

Implementation of Various Edge Detection Techniques for Gray Scale Images in VC++

Er. Harsimran Singh¹, Er. Tajinder Kaur²

¹M.Tech. Scholar, CSE/IT, S.B.B.S.I.E.T Padhiana, Jalandhar, Punjab, India

²Assistant Professor, CSE/IT, S.B.B.S.I.E.T Padhiana, Jalandhar, Punjab, India

Abstract: - Edge Detection is an important task in Digital Image Processing. It is commonly used for feature extraction from the images. Various techniques are used to detect edges in the images. Every technique has its own convolution mask which is applied on the image to detect true edges in a gray scale image. In this paper implementation of various edge detection techniques is done in Visual C++. In future after working on the result of these existing operators a novel algorithm will be developed. We will compare the optimal existing technique with the proposed technique and show the better results.

Keywords: Edge Detection; Sobel; Canny; Threshold; Mask; Orientation; Suppression

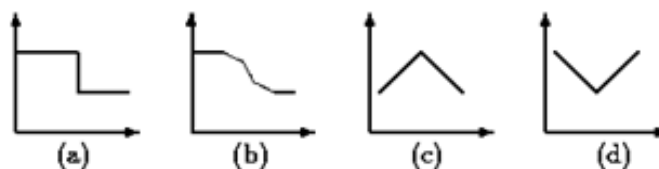
I. Introduction

Edge detection is the most common approach for detecting discontinuities in gray scale images. An edge is defined as a set of connected pixels that lie on a particular boundary between two regions. Edges are used to define the ability to measure the transitions in a meaningful way. When edges points are located, boundaries can be formed by linking the points. Edges are formed at the place where change occurs. By using edge detection important features can be extracted from an image in the form of edges. Therefore edges are used for boundary estimation and segmentation in the scene. Since computer vision involves the identification and classification of objects in an image, edge detection is an essential tool.

A. Types of Edges

All edges are locally directional. Edges can be modeled according to their intensity profiles. A Sharp Step, as shown in Fig. 2(a), is an idealization of an edge. Since an image is always band limited, this type of graph cannot ever occur. A Gradual Step, as shown in Fig. 2(b), is very similar to a Sharp Step, but it has been smoothed out. The change in intensity is not as quick or sharp. A Roof, as shown in Fig. 2(c), is a ridge edge where the intensity change is not instantaneous but occurs over a finite distance. The Trough, also shown in Fig. 2(d), is the inverse of a Roof. Edge detection is very useful in a number of contexts. Edges characterize object boundaries and are, therefore, useful for segmentation, registration, and identification of objects in scenes.

Figure 1 Types of Edges (a) Sharp step (b) Gradual step (c) Roof (d) Trough



II. TECHNIQUES FOR EDGE DETECTION

Edge detectors can be classified into two broad categories [4]:

A. First Derivative operators [4]

1) *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°.

Figure 2 Convolution masks of sobel operator

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

Gx

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

Gy

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:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y|$$

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:

$$\theta = \arctan (G_y/G_x)$$

2) *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°.

Figure 3 Convolution masks of Robert cross operator

+1	0
0	-1

Gx

+1	0
0	-1

Gy

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:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

although typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y|$$

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:

$$\theta = \arctan(G_y/G_x) - 3\pi/4$$

3) *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.

Figure 4 Convolution masks of Prewitt operator

-1	0	1
-1	0	1
-1	0	1

Gx

1	1	1
0	0	0
-1	-1	-1

Gy

B. Second Derivative Operator's [4]

1) *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.

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

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

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. Three commonly used small kernels are shown in Fig. 6.

Figure 5 Three commonly used discrete approximations to the Laplacian filter.

0	1	0
1	-4	1
0	1	0

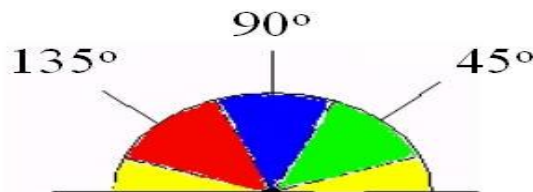
1	1	1
1	-8	1
1	1	1

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

2) *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 6 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. Non maximum suppression 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.

III. Implementation

Steps to be followed:

Step1: Take a gray scale image $f(x,y)$ that is free from noise. If there is noise in the image then remove it first.

Step2: Apply the convolution mask G_x and G_y on the input image.

Step3: Determine the gradient magnitude in the form of $|G| = \sqrt{G_x^2 + G_y^2}$

Step4: Compare the Gradient Magnitude with threshold value and find true edges.

Step5: Compare the results on the basis of MSE and PSNR.

IV. Parameters for Comparison

A. 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]

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

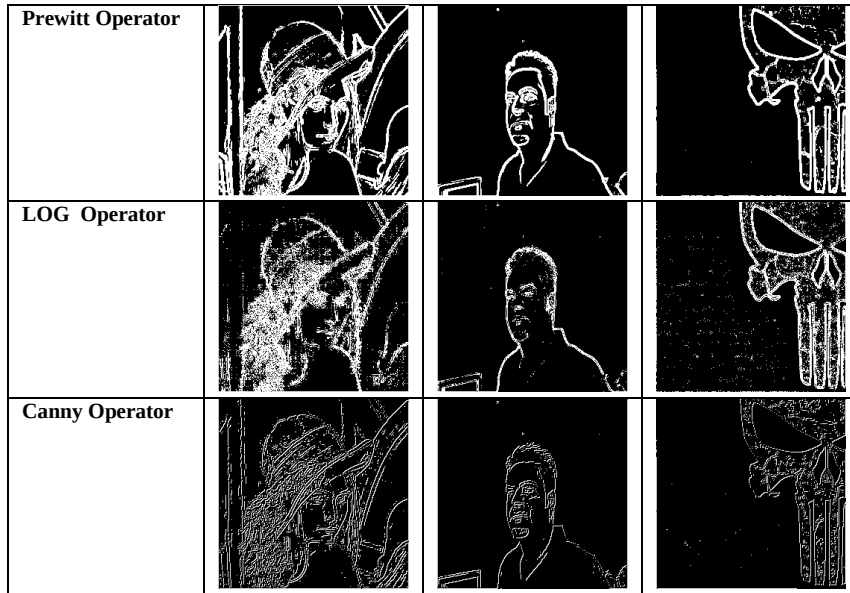
B. Mean square error (MSE) of an estimator is to quantify the difference between an estimator and the true value of the quantity being estimated. [6]

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

IV. Experimental results

Table 1 Shows the experimental results of various operators

	Image (a)	Image (b)	Image (c)
Original Image			
Sobel Operator			
Robert cross Operator			



V. Comparison

Operator	Image (a)		Image (b)		Image (c)	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
Sobel	20009.0419	5.1185	19216.5023	5.2940	8070.9617	9.0615
Robert Cross	19607.0285	5.2066	17782.3607	5.6309	6857.3654	9.7692
Prewitt	19797.5665	5.1646	18711.3543	5.4097	7726.6245	9.2509
LOG	19845.1533	5.1542	17731.7856	5.6432	7610.5951	9.3166
Canny	19044.7414	5.3330	17541.1504	5.6902	5944.7840	10.3894

VI. Conclusion

In this paper we have evaluated various Edge Detection Operators that are Sobel, Robert Cross, Prewitt, LOG and Canny. From the above results, it is concluded that Canny operator is the best method to detect edges. Comparison is done on the basis of two parameters PSNR and MSE. For an Edge detector to be effective PSNR value should be high and MSE should be low. Hence both are inversely proportional to each other. From the above comparison it has been observed that Canny operator has higher PSNR and lower MSE in all the images, So it is the best method to detect edges. In the future work we will propose a novel algorithm that will yield better results as compared to canny operator.

References

- [1] Rafael c. Gonzalez and richard e. Woods "Digital Image Processing", Third Edition, Pearson Prentice Hall.
- [2] Ireya. 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] Wenshuo Gao, Lei Yang, Xiaoguang Zhang, Huizhong Liu (2010) , "An Improved Sobel Edge Detection", IEEE, 978-1-4244-5540.
- [9] Bhadauria H S, Dewal M L (2010), "Comparison of Edge Detection Techniques on Noisy Abnormal Lung CT Image before and after Using Morphological Filter", Published in International Journal of Advanced Engineering & Application.
- [10] 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.
- [11] Mohamed Roushdy (2006), "Comparative Study of Edge Detection Algorithms Applying on the Grayscale Noisy Image Using Morphological Filter", GVIP Journal, Volume 6, Issue 4.
- [12] 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.
- [13] 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.