Edge Detection Algorithms

M. K., Abdulmajeed
kbmajeed1@gmail.com,

Abstract—**In this work, we implement different edge detection algorithms on three unique images. Due to the importance of edge detection in higher level image processing and computer vision tasks, it is important to understand performance characteristics of edge detection algorithms. Edge detection using Sobel, Roberts, Prewitt and the Laplacian method is presented in this report.**

Keywords— Edge detection; Gradient; Sobel; Prewitt.

1. INTRODUCTION

Edge detection is a classical image processing phenomena and a primary requirement for many higher-level image processing tasks such as feature extraction and segmentation. It is frequently used as the first step in extracting information from an image. The understanding of an edge is based on visual perception of regions of steep changes in intensity. The process of edge detection can be observed as the process of finding discontinuities in the image which can thereafter be used for segmentation and image understanding. Among the common edge detectors include: Sobel, Roberts, Prewitt and Canny.

The Fundamental steps of Edge detection can be summarized in 3 steps [3], [2]: (a) Image smoothing for noise reduction, (b) Detection of edge points, and (c) Edge localization.

2. EDGE DETECTION

2.1 Introduction to Edge Detection

As previously mentioned, edge detection algorithms are crucial and are continuously the subject of research. A good edge detector can be evaluated by the correct edges detected, missing edges, and false edges. To proceed, we present the following definitions as obtained from [1].

Definition 1: An edge point is a point in an image $(f(x,y))$ with coordinates $[i,j]$ at the location of a significant local intensity change in the image.

Definition 2: An edge detector is an algorithm that produces a set of edges from an image.

Definition 3: Gradient of an image (∇f) is a measure of the change in pixel intensities in 2-dimensions; similarly referred to as the derivative of the image and has magnitude $|\nabla f|$ and direction $\boldsymbol{\theta}$.

$$\nabla f = \mathrm{G}[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}.$$

$$|\nabla f| = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y| \text{ and } \boldsymbol{\theta} = \tan^{-1}(G_y/G_x)$$

Similarly,

$$\frac{df}{dx}(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$

While:

$$G_x = \frac{\partial f}{\partial x} \cong f[i, j+1] - f[i,j]$$

$$G_y = \frac{\partial f}{\partial y} \cong f[i,j] - f[i+1, j]$$

2.2 Edge Detection Algorithms

The edge detection schemes utilized in this work, both the gradient and Laplacian operators are used. Spatial filters (mask or kernel) are used to compute the derivatives at every pixel location in an image. The resulting response of using a $\mathrm{w} \times \mathrm{w}$ mask at a pixel point is given as the sum of products of the mask coefficients (ω) with the intensity values (z) in the region encompassed by the mask [3].

$$R = \sum_{k=1}^{\mathrm{w}\times\mathrm{w}} \omega_k z_k$$

2.2.1 First-Order Edge Detectors

These methods are defined using the first derivative of the image to obtain a kernel which is convolved with the image. The three first-order kernels utilized in this work are briefly highlighted below:

Sobel Operator (kernel):

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \text{ and } G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Prewitt Operator (kernel):

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \text{ and } G_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Robert Operator (kernel):

$$G_x = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \text{ and } G_y = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$

First order derivatives generally produce thicker edges in an image, second order derivatives have a stronger response to fine detail and produce double-edge response at ramp and step transitions in intensity. The sign of the second derivative can be used to determine whether the transition into an edge is from light to dark or dark to light.

2.2.2 Second-Order Edge Detectors

These methods are defined using the second derivative (Laplacian) of the image to obtain a kernel operator which is convolved with the image. Where the Laplacian operator ($\boldsymbol{\nabla}^2$) is defined:

$$\boldsymbol{\nabla}^2 f(x,y) = \frac{\partial^2}{\partial x^2} f(x,y) + \frac{\partial^2}{\partial y^2} f(x,y)$$

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

$$\boldsymbol{\nabla}^2 f(x,y) = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

Laplacian Operator (kernel):

$$\mathcal{L}[f(x,y)] = \boldsymbol{\nabla}^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

The Laplacian results in the filter kernel defined as:

$$G_{xy} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

3. ALGORITHMIC IMPLEMENTATION

(a) **Image Acquisition**: Here the image is read into Matlab as an $m \times m$ array. The two images used in this work are shown in Figure 1 below:

Figure 1: Image Acquisition

(b) **Image Smoothing**: This is done for noise reduction and basically helps to attenuate the problems associated with finding the derivative of noisy signals. This step basically makes edge detection more effective. It however could lead to the loss of edge information if smoothing is excessive. In this work, a Gaussian blurring kernel is implemented on the image with a kernel size of $n \times n$ and variance σ^2.

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

Figure 2 shows the 3D surface plot of the Gaussian function.

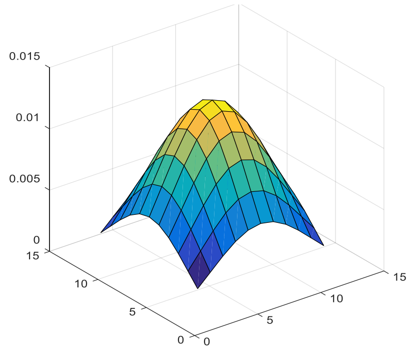


Figure 2: Gaussian Blur 3D Surface

(c) **Edge Detection and Localization**: this operation extracts possible edge points from the image using the gradient operators.

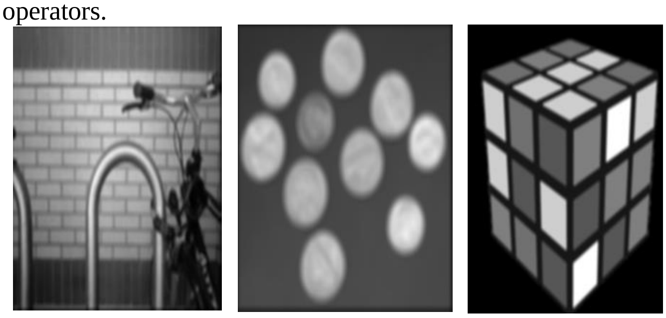


Figure 3: Blurred Images ($\sigma = 4$)

4. RESULTS

(a) Gradient based Edge detection with Sobel kernel

The following figures represent the results from the application of first order Sobel gradient operator on two images. The gradient operator using Sobel kernel is applied to obtain the vertical and horizontal edges of the image. The gradient magnitude and direction is then obtained using the previously defined equations.

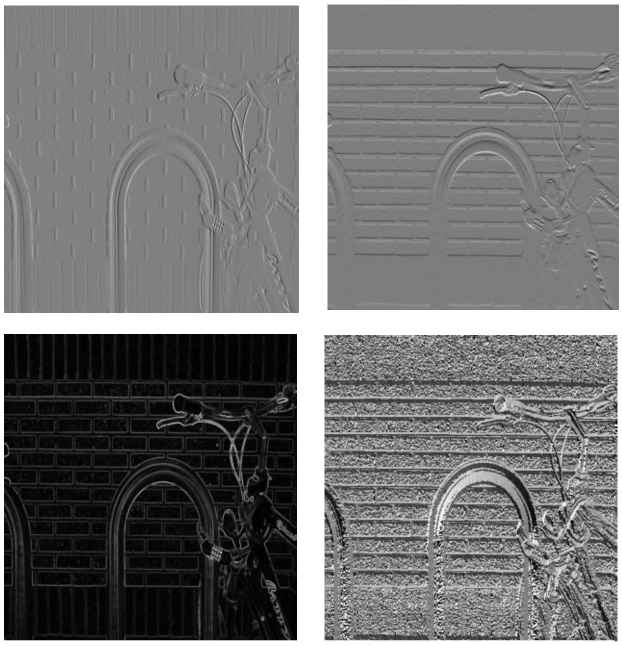


Figure 4(left to right): Vertical gradient, Horizontal gradient, Gradient (Edge) magnitude, Gradient direction.

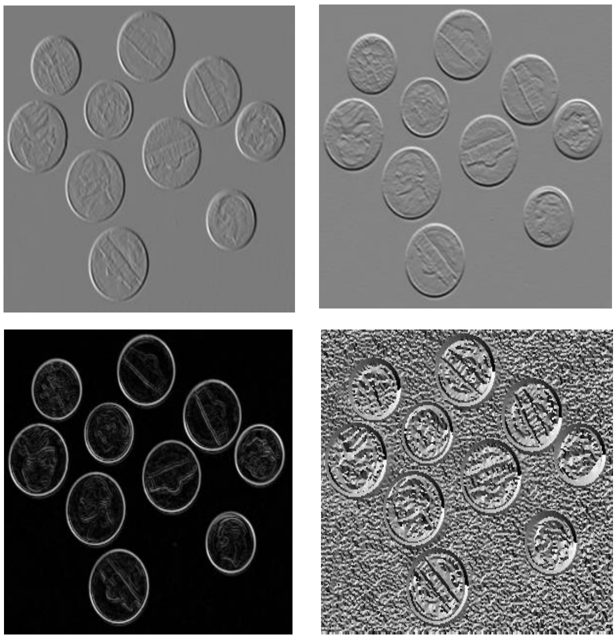


Figure 5(left to right): Vertical gradient, Horizontal gradient, Gradient (Edge) magnitude, Gradient direction.

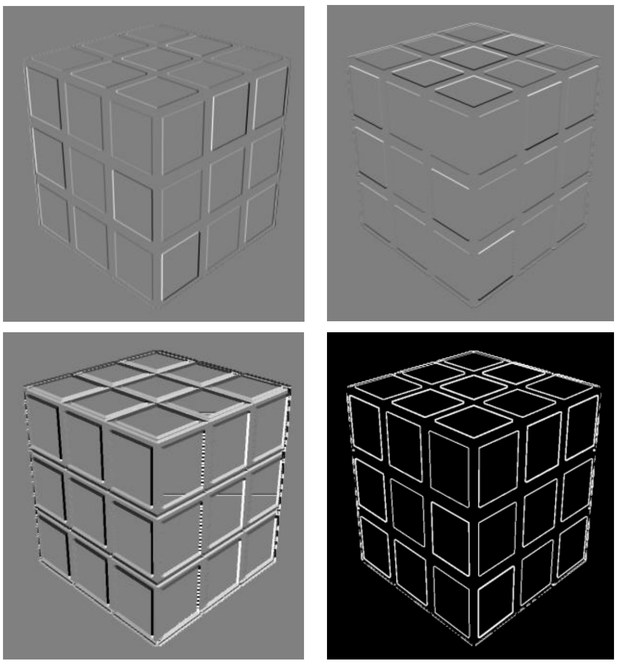


Figure 6(left to right): Vertical gradient, Horizontal gradient, Gradient (Edge) magnitude, Gradient direction.

(b) **Gradient based Edge detection with Prewitt kernel**

The following figures represent the results from the application of first order Prewitt gradient operator on two images. The gradient operator using Prewitt kernel is applied to obtain the vertical and horizontal edges of the image. Gradient magnitude and direction are then obtained using the previously defined equations.

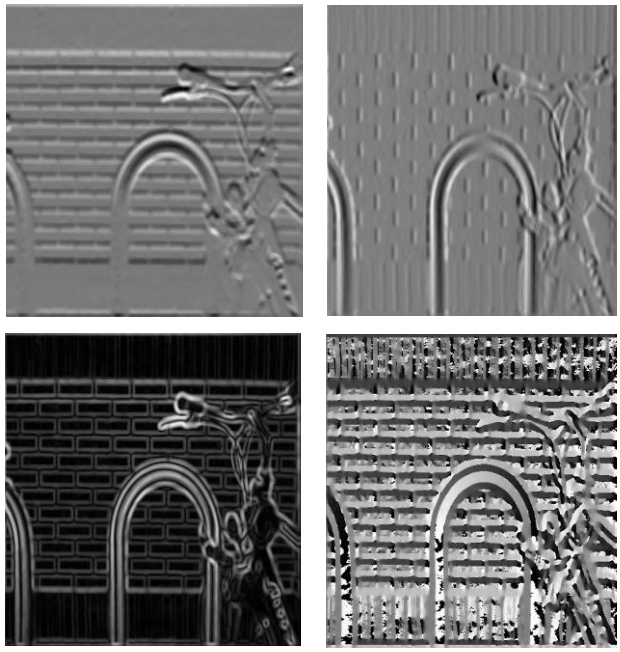


Figure 7(left to right): Vertical gradient, Horizontal gradient, Gradient (Edge) magnitude, Gradient direction

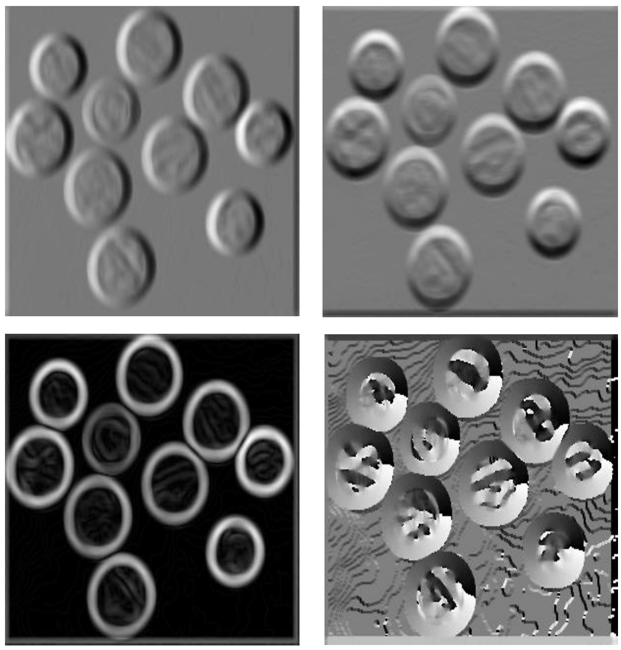


Figure 8(left to right): Vertical gradient, Horizontal gradient, Gradient (Edge) magnitude, Gradient direction

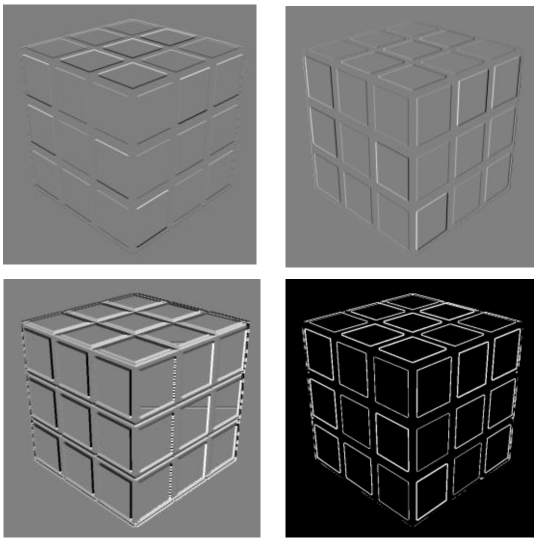


Figure 9(left to right): Vertical gradient, Horizontal gradient, Gradient (Edge) magnitude, Gradient direction

(c) Gradient based Edge detection with Roberts kernel

The following figures represent the results from the application of first order Roberts gradient operator on two images. The gradient operator using Robert kernel is applied to obtain the gradient magnitudes.

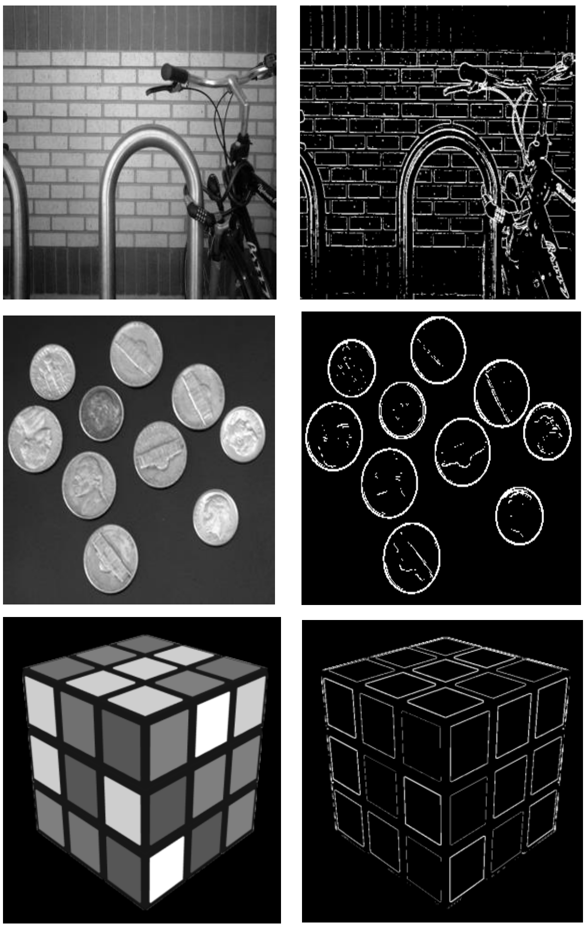


Figure 10(left to right): Original images versus Gradient (Edge) magnitude

(d) Second-Order Edge Detection (Laplacian Kernel)

Since second derivatives are more sensitive to noise, it is necessary to pre-process the images by blurring using Gaussian kernels. For the sake of brevity, results for the clearest blurring using the third kernel ($17 \times 17, \sigma = 3$) is presented.

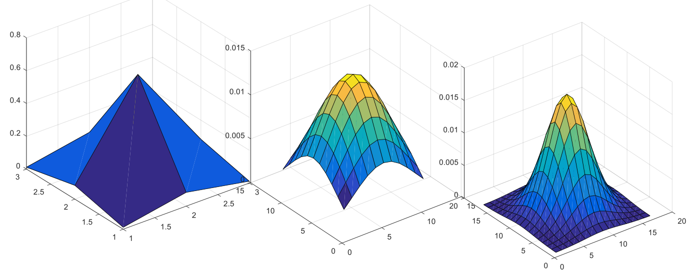


Figure 11(Left to Right): Three different Gaussian blurring functions used for smoothing; (a) $3 \times 3, \sigma = 0.5$, (b) $11 \times 11, \sigma = 4$, (c) $17 \times 17, \sigma = 3$.

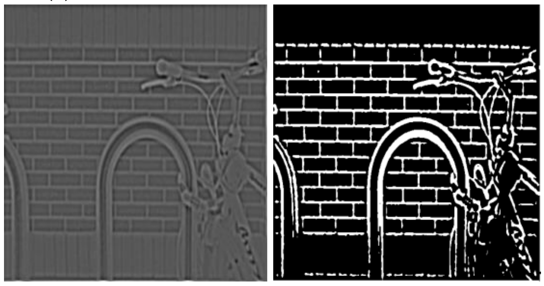


Figure 12(left to right): Laplacian image, Edges after binarization

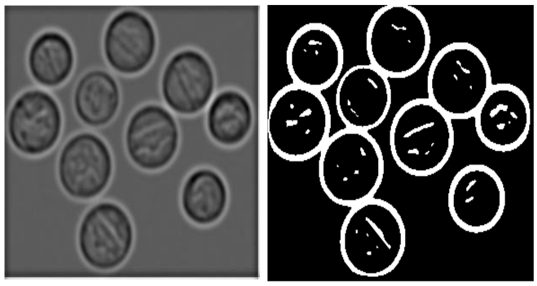
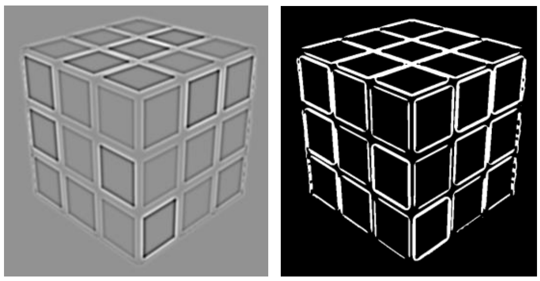


Figure 13(left to right): Laplacian image, Edges after binarization



REFERENCES

[1] J. Ramesh, K Rangachar, B. G. Schunck, *Machine Vision*. p. 140, 1995

[2] M. Abo-Zahhad et. al, “Edge detection with a Preprocessing Approach.” *Journal of Signal and Information Processing*, vol. 5, p. 123-134, 2014

[3] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing. p. 715, 2008