Performance Evaluation of Image Edge Detection Techniques

Maher I. Rajab *mirajab@uqu.edu.sa*
College of Computer & Information Systems
Computer Engineering Department
Umm Al-Qura University
Makkah 21955, Saudi Arabia

Abstract

The success of an image recognition procedure is related to the quality of the edges marked. The aim of this research is to investigate and evaluate edge detection techniques when applied to noisy images at different scales. Sobel, Prewitt, and Canny edge detection algorithms are evaluated using artificially generated images and comparison criteria: edge quality (EQ) and map quality (MQ). The results demonstrated that the use of these criteria can be utilized as an aid for further analysis and arbitration to find the best edge detector for a given image.

Keywords: Image Edge Detection, Gaussian Noise, Gaussian Smoothing.

1. INTRODUCTION

Edge extraction is usually the starting step in segmentation because it effectively detects the limits of the objects. These limits are commonly called "edges" [1]. Edge extraction is used mostly in image recognition, classification or interpretation procedures because it provides a compressed amount of information for processing [2]. The evidence for the best detector type is judged by studying the edge maps relative to each other through statistical evaluation [3]. Singh et al. [4] concluded that the Sobel, Prewitt, Roberts, Canny provide low quality edge maps as compared to Laplician of Gaussian (LoG). Comparison is done on the basis of two parameters PSNR and MSE.

This paper is organized into five sections as follows:
Section one introduces the topics of edge detection and scale space analysis. Section two reviews the theory and the mathematical background that is used to assess the analysis of the techniques and procedures followed throughout this work. Section three starts with the images that are used, and also a specific generated reference image are described along with other related methods which are adopted to achieve the aims of this research. Section four summarizes the results found when evaluating the success rate of the investigated edge detectors. Finally, in section five conclusions and suggestions for future work are considered.

2. THEORY

In this section, the essential derivative concept for edge detectors (sec 2.2) In 2-D images is introduced, along with a suggested approach which has been used in edge detector implementations.

2.1 Edge Detection

Firstly an "edge point" represents an abrupt change in an image brightness function which occurs at boundaries; each edge point has a magnitude and direction, i.e. it is a vector variable [5]. The edge magnitude represents edge intensity or the strength of the edge point and edge direction depicts the tangent of the direction at that point.

Since the image varies in two dimensions, accordingly edge variations are also calculated in two directions, i.e. horizontal direction i and vertical direction j. Therefore, two partial derivative components are needed to describe the edge gradient in both i and j directions:

$$I_i = \partial I / \partial i \quad (1)$$

$$I_j = \partial I / \partial j \quad (2)$$

also the angle between Ii and Ij can also be specified as:

$$\theta = arctan \frac{I_i}{I_j} \quad (3)$$

The edge strength magnitude at each pixel (i,j) can be simply expressed as:

$$E(i, j) = \sqrt{I_i^2 + I_j^2} \quad (4)$$

or

$$E(i, j) = |I_i| + |I_j| \quad (5)$$

To simplify image calculations and transformations, i.e. square and square root transformations, in this work, instead of using the definition in equation (4), it is suggested that E(i,j) be expressed as the maximum of the two directional gradients [5] and :

$$E(i, j) = \max\left(I_i , I_j\right) \quad (6)$$

The definition of in equation (3) is the same as before. Equation (6) has been used extensively throughout the implementation procedure in various edge detectors. It is noticed that equation (6) calibrates the edge strength E within its maximum allowable range, i.e. E has a value not exceeding 255 when representing an image in 8-bit or 256 gray levels. In contrast, the standard magnitude notation in (4) increases the value of E by $\sqrt{2}$ at equal gradients, i.e. I_i equals I_j and the simple addition in (5) doubles E.

2.2 Image Edge Detectors

Various edge detectors will be evaluated according to the maximum edge map [1] that they produce as discussed later. The robustness of the edge detectors to different levels of noise will be investigated as well. Common edge detectors are considered in this section. The first and second derivative approaches are discussed below.

2.2.1 First Derivative Edge Detectors

Some of the edge operators utilize pixel differences to approximate derivatives for example, the Roberts operator [5]. In contrast other edge detectors apply several rotational kernels to approximate first derivatives such as Sobel, Prewitt, Robinson, and Kirsch [5].

Since digital images are discrete in nature, so derivatives can be approximated as differences between pixel values. Therefore, the partial derivatives in (1) and (2) can be approximated by one pixel difference as in (7) and (8), respectively:

$$I_i(i\ ,\ j) = \left| I(i+1\ ,\ j) - I(i\ ,\ j) \right| \qquad (7)$$

$$I_j(i\ ,\ j) = \left| I(i\ ,\ j+1) - I(i\ ,\ j) \right| \qquad (8)$$

The Roberts operator involves two kernels (templates) to measure the rate of change of image intensity in two orthogonal directions as shown in the figure below.

1	0
0	-1

(a)

0	1
-1	0

(b)

FIGURE 1: The Roberts edge operator kernels.

This will yield two images Ia and Ib from the above kernel in Figures 1(a) and 1(b), respectively. The absolute pixel values of $I_a(i\ ,\ j)$ and $I_b(i\ ,\ j)$ are:

$$I_a(i\ ,\ j) = \left| I(i\ ,\ j) - I(i+1\ ,\ j+1) \right| \qquad (9)$$

$$I_b(i\ ,\ j) = \left| I(i\ ,\ j+1) - I(i+1\ ,\ j) \right| \qquad (10)$$

Consequently, the Roberts edge magnitude would be calculated by applying equation (6), using the results from (9) and (10), as

$$E(i,j) = max\left(I_i\ ,\ I_j\right) \qquad (11)$$

It is noticed that the Roberts operator uses a total of 4-neighbourhoods, or two pixels in each orthogonal direction, to approximate the derivative which results in high sensitivity to noise [5]. In contrast, other first derivative edge detectors utilize up to 8-neighbourhood pixels in each 3x3 rotational kernel, for example Robinson and Kirsch [5]. For example, the first three possible successive rotational kernels in the Robinson edge detector are :

$k_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{pmatrix}$ $k_2 = \begin{pmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{pmatrix}$ $k_3 = \begin{pmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{pmatrix}$

The eight rotational kernels (k1 to k8) for Sobel and Prewitt are generated and listed in Appendix A. The simple rotation [5] is applied onto the 8-neighbourhoods pixels by rotating them one consistent circular step, in a clockwise or anticlockwise direction, to yield a kernel k_i , as shown between k_1 to k_2 or k_2 to k_3.

Edge operator is a good high-pass filter for an image, i.e. it can detect sharp intensity changes, when the image is free of noise. However, this operator is not considered as a good edge detector because it also detects the noise pixels as edges and increases their intensity [2,7].

A practical approach in attenuating or reducing the amount of noise is called noise smoothing or simply smoothing [2,8]. This can be tackled with the Gaussian smoothing operator $G(i,j)$, or Gaussian for short, and is given by [2]

$$G(i,j;\sigma) = e^{-\frac{i^2+j^2}{2\sigma^2}} \quad (12)$$

where σ is the scale parameter, the standard deviation, which defines the bandwidth of the filter and therefore the level of smoothing [2]; the larger the parameter σ , the less the presence of higher frequencies, and the more the blurring of the image [2]. Noise smoothing using the Gaussian kernel is a linear filtering operator, since the image is convolved with a constant kernel or kernel model (G) as [8]

$$I_G = I * G \quad (13)$$

The integer Gaussian filter, refer to Appendix B, is adopted in this work because it rounds to integers the real numbers contained in the kernel. The smallest integer used in the kernel is 1. Appendix B lists ten integer Gaussian kernels at different scales σ ; which have been used throughout this study.

3. METHODLOGY

In the previous section a literature review on the topics relating to this work was presented. In this section the generated synthetic images used in evaluating various edge detection algorithms is introduced. Within this section some examples of edge detection techniques are implemented and the generated and used images are also considered. Finally, the necessary comparison criteria, which are used as an aid for arbitration, are investigated.

3.1 Gaussian Noise Deviates

Noise, in general, is defined differently according to the case it is introduced in [8]. In image and signal processing:

- It might be considered as false contours when trying to group meaningful objects or lines.
- In edge or line detection, noise might be the spurious fluctuations of pixel values which may be introduced by the image acquisitions system [8].

In our case, the noise is chosen to be an additive amount [8] to an artificially generated [1] noise-free image (discussed in the next section). Therefore, a generated noise image $n(i , j)$ added to the true image, free-of-noise, $I(i , j)$ will yield $In(i , j)$ as

$$I_n(i,j) = I(i,j) + n(i,j) \quad (14)$$

The Gaussian noise deviates model is chosen in this study because:
1. Define the Gaussian noise values which are normally distributed deviates with zero mean and at fixed standard deviations, σ_n as $n(i , j; \sigma_n)$. These noise values are completely independent of each other and also, of the true image e.g. $I(i , j)$. Noise is considered to be additive.

2. The symmetrical nature of this model is a practical approximation. In contrast impulsive noise, otherwise known as spots or peak noise, randomly alters pixels so that their values deviate considerably from the true values $I(i , j)$.

Two Gaussian deviates are generated by the algorithm introduced by Press et al. [9]. This algorithm has been developed to generate these normal deviates at the desired σ_n and in two dimensions i and j to yield finally a noise image, file, of size mxn .

3.2 Generated Images

Ramalho's artificially generated images [1,10] Band A and Band B are shown in Figures 3.1 and 3.2; these are mainly used in this study. The Ramelho's Band B reference edge image, as shown in figure 3.3, is also used as a reference for correct and wrong marked edge pixels; it is a binary image i.e. one level represents the correct edges while the other refers to wrong marked edges. Therefore, edge images which result from the set of edge detection algorithms (discussed in section 3.4) are compared with the aid of this reference image to compute the percentage of missed edges. Comparison criteria are investigated in the next section.

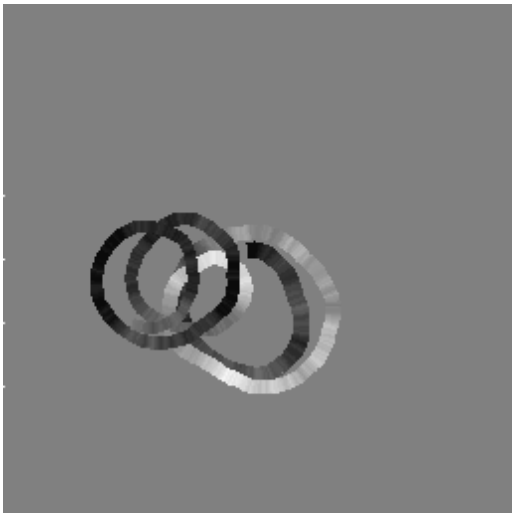

FIGURE 3.1: Ramalho's [1] Band A original image.

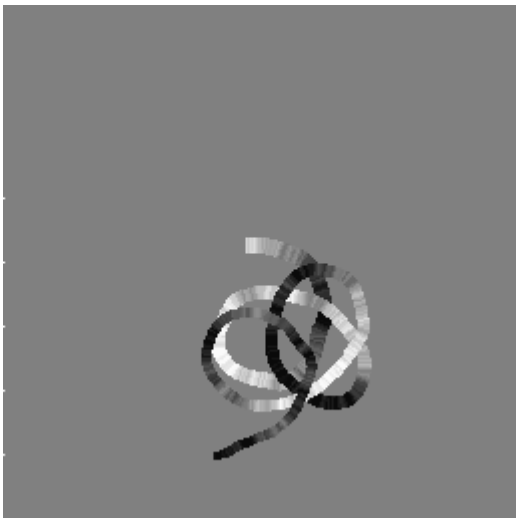

FIGURE 3.2: Ramalho's [1] Band B original image.

Enhancement image operations are performed on the Band B reference image to yield closed contours of one pixel wide edges coinciding with the centers of corresponding pixels of the original image [11] (Band B original image shown in figure 3.2). The enhancement process has included the following three major steps:

1. Simple segmentation operation to segment the original image into a one closed contour object and background.
2. Logically based routine to mark outside boundaries of the main object.
3. Logical and mathematical operations between Ramelho's original reference and the image result from step 2.

Figure 3.4 shows the output images resulting from the steps followed to prepare this enhanced reference.

FIGURE 3.3: Ramalho's[1] Band B Reference image (256x256) pixels.

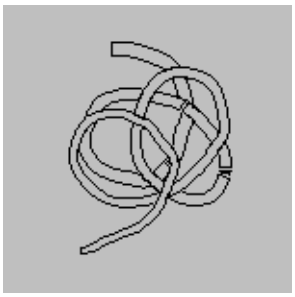

FIGURE 3.4: Enhanced Band B Reference image (150x150) pixels.

3.3 Comparison Criteria

According to Pratt [12], the only performance measure of ultimate importance is the completeness and quality of and edge map. A feasible approach introduced by Ramalho [1] shows significant advantages of using a neural network in the comparison of edge detection techniques, in contrast with the conventional edge detection techniques. Ramalho's [1] has developed two ratios to compare and evaluate this quality of edges which are produced by a selected set of edge detection algorithms. These ratios are designated as edge quality (EQ) and map quality (MQ) as follows

$$EQ = \frac{C}{W + C} \tag{15}$$

$$MQ = \frac{C}{W + C + M} \tag{16}$$

where C refers to the correctly marked edges, W to the wrongly marked edges, and M to missed edges. The map quality ratio is chosen in this study as a base measure for the quality of edges produced by a set of edge detection algorithms (as discussed in the next section) because (i) edge detectors which mark a small number of correct edges are penalized [1], (ii) it is more accurate since the missed points are also included; in contrast EQ gives in such case a misleading result. For example the case when an edge operator marks a very small number of both wrong and correct points, In this case if W goes to zero then EQ will measure a false ratio of one.

3.4 Edge Detection Algorithms

A diversity of edge detection algorithms and their implementations are considered in this section. First derivative edge detectors which are based on the convolution operations, are introduced in section two. Due to the similarities between these filters, one algorithm is introduced ; the Sobel edge detector algorithm is selected as an example from this category.

The thresholding step is commonly used in edge detection algorithms [1,8] to determine a threshold value (τ) above which the image points are considered as edges [8]. Different thresholding techniques are reviewed by Shaoo et al. [13]. The problem with this type of traditional edge detection approach is that a low threshold produces false edges, but a high threshold misses important edges [14]. Panda et al. [15] investigated different first and second derivative filters to find edge map after denoising a corrupted gray scale image. Subjective method has been used by visually comparing the performance of the proposed derivative filter with other existing first and second order derivative filters. For objective evaluation of the derivative filters, the root mean square error and root mean square of signal to noise ratio are used.

In this research the suggested method for noisy images, requires a strategy to take into consideration the overall probability of these images. The standard deviation of the Gaussian noise should be taken into account when calculating the threshold.

The discussion above can be summarized in the following proposed edge detection algorithm:

Algorithm: *Proposed Sobel edge detection based on eight Sobel rotational kernels.*

{Detection}

For i=1 to 8 do {eight rotational kernels k_1 to k_8 }

$\boldsymbol{I_k} = \boldsymbol{I} * \boldsymbol{k_i}$ {convolve image I with kernel $\boldsymbol{k_i}$ }

end for

$E(i,j) \Leftarrow \max\left(I_{k_1} \; to \; I_{k_8} \right)$ {maximum gradients}

{Thresholding}

$\boldsymbol{\tau = \mu + \alpha \sigma}$ { computes threshold τ. μ is the mean of $E(i,j)$, σ is its standard deviation, and α is an adjustable parameter }

$E(i,j) > \tau$ { mark edges which have edge strength greater than τ }

The above algorithm consists mainly of two major parts: *Detection* and *Thresholding*.

The *Detection* step performs eight convolution operations on the image *I*, by operating eight rotational kernels k_1 to k_8 , which model eight orientation images (at eight orientations i.e. $0, \pi/4, \pi/2, \ldots 7\pi/4$). Each orientation image represents a directional gradient and the greatest gradient will constitute the edge strength $E(i,j)$: this was discussed in section two.

In the *Thresholding* operation, the selection of τ is designed to be such that all pixels points of the edge strength $E(i,j)$ are greater than τ. However it is important to determine a method to calculate τ. A method is suggested for choosing this threshold according to a combination of the mean and the standard deviation σ of the edge detector output image, $E(i,j)$, as discussed in the next section.

Extensive experiments have been carried out on noisy images (e.g. Band A and Band B shown in figures 3.1 and 3.2, respectively). It is found that a common approximation that is made to the threshold value τ is a percentage of the highest value of $E(i,j)$ as follows:

$$\tau = \alpha \cdot max(E(i,j)) \qquad (17)$$

For example $\alpha = 0.5$ will mark all pixels of $E(i,j)$ which are above 50% of its maximum. This relation is misleading, especially when an image is corrupted by a large amount of noise, for example when noise $n(i,j)$ with large variance, is added to the true image $I(i,j)$. For example, in this study, five levels of noise are chosen at $\sigma_n = 1,10,30,50, and\ 75$.

The suggested method to compute τ will takes into consideration the amount of additive noise. The derived relation given as

$$\boldsymbol{\tau = \mu + \alpha \sigma} \qquad (18)$$

where μ is the mean value of the strength image $E(i, j)$, σ is its standard deviation, and α is an adjustable parameter which may give a flexible range to control the output of an edge detector more. Here τ is taken to be the sum of μ and σ i.e. $\alpha = 1$.

The relation in (18) is successfully experimented on different images and yields a qualitatively good edge output. Therefore, the *{Thresholding}* section in the above algorithm is developed as

{Thresholding}

$\mu = Mean(E(i, j))$ {Get the mean value of image $E(i, j)$ }

$\sigma = StdDev(E(i, j))$ {Get the standard deviation of image $E(i, j)$}

$\tau = \mu + \sigma$

In the next section the performance of different edge detectors are evaluated according to the comparison criteria which were mentioned earlier in this section.

4. RESULTS AND DISCUSSION

In this section the output of three edge detectors are evaluated using the map quality ratio (MQ), section 3.3. Firstly, five levels of noise are added to the Band B image (Figure 3.2), then each level is smoothed at gradual scales; ten gradual smoothing steps are used. Finally, for each edge detector there are five curves showing the fluctuation of edge success rate; the percentage of map quality MQ represents this rate.

4.1 Effect of Gaussian Noise Deviates on Edge Detectors

To achieve the foregoing objectives, the performance of each edge detector is analyzed for five levels of Gaussian noise deviates; $\sigma_n = 1,10,30,50,$ and 75. Figure 4.1 shows the Band B image exposed to these five levels of noise:

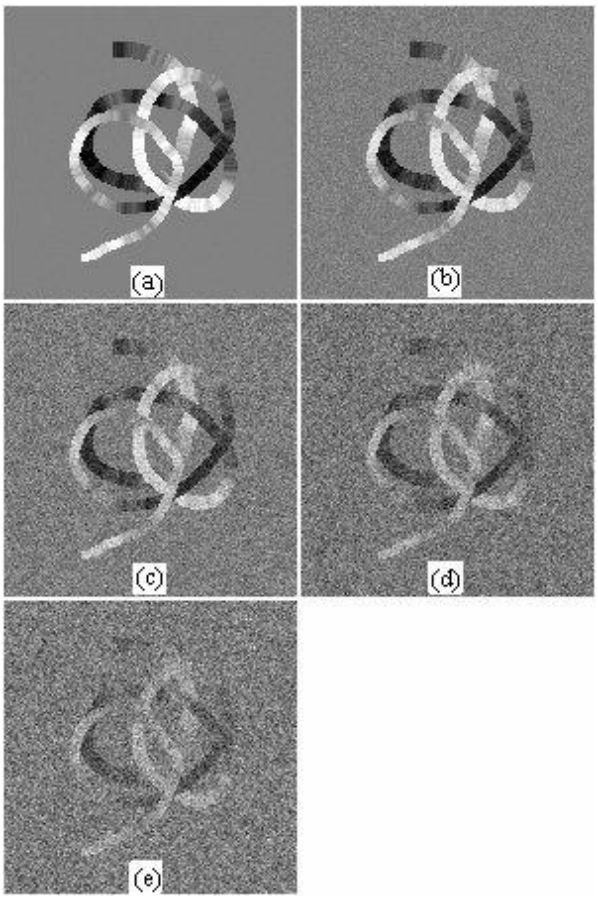

FIGURE 4.1: Five noise levels added to Band B image.
Images (a) at $\sigma_n = 1$, (b) $\sigma_n = 10$, (c) $\sigma_n = 30$, (d) $\sigma_n = 50$, (e) $\sigma_n = 75$.

At each noise level there are ten gradual steps of smoothing according to convolution kernel width given as

$$W = 5\sigma_g \qquad (19)$$

This is an adequate width of the Gaussian filter because it subtends 98.76% of the area under the curve [8]. The degree of smoothing is determined by the standard deviation of the Gaussian σ_g; larger standard deviation Gaussians require larger convolution kernels. To clear ambiguities, Appendix C lists in tabular form the detailed data summarizing the values of σ_g, kernel sizes, percentage of correct, wrong, and missed edges, and the overall map quality MQ. Figures 4.2 to 4.4 illustrate the performance of the three edge detectors: Sobel, Prewitt, and Canny. The edge quality decreases with the increase of scale σ_g of the Gaussians.

In this research the suggested method for noisy images, requires a strategy to take into consideration the overall probability of these images. The improved selection of thresholding in Sobel and Prewitt algorithms using the sum of mean and standard deviation of the Gaussian noise. Then the image is run through the proposed Sobel or Prewitt algorithms, this process is hardly affected by noise. The MQ results indicated that the Proposed Sobel and Prewitt edge

detection algorithms outperform Canny algorithm. The enhanced detection step in Sobel and Prewitt performs eight convolution operations and this will enrich the detection of edge pixels.

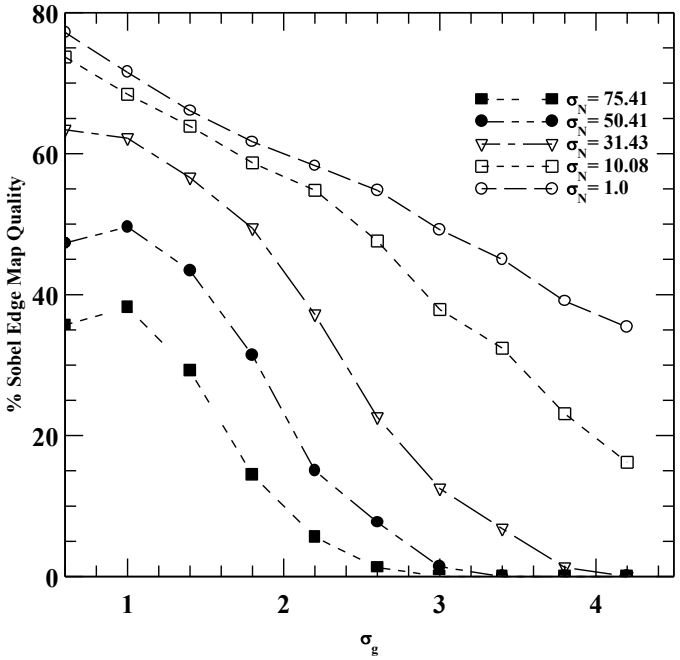

FIGURE 4.2: Edge map quality of smoothed Sobel edge detector corrupted by Gaussian noise.

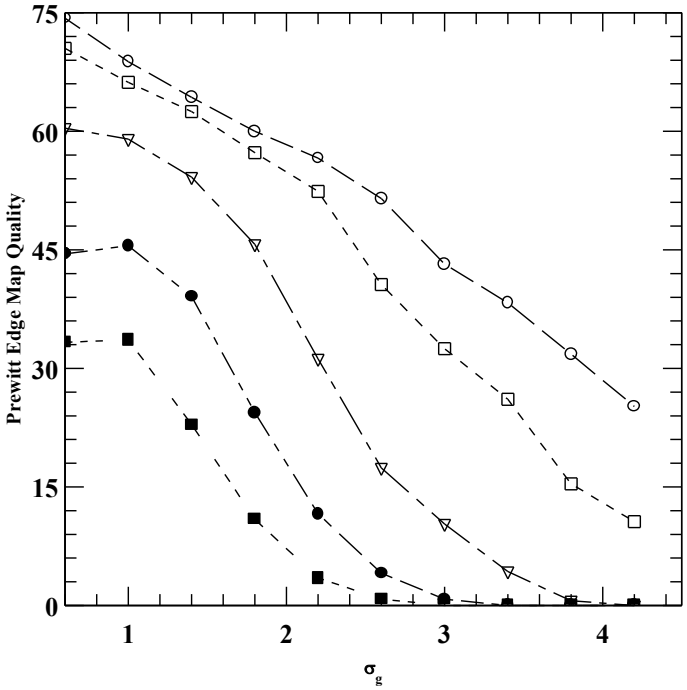

FIGURE 4.3: Edge map quality of smoothed Prewitt edge detector corrupted by Gaussian noise.

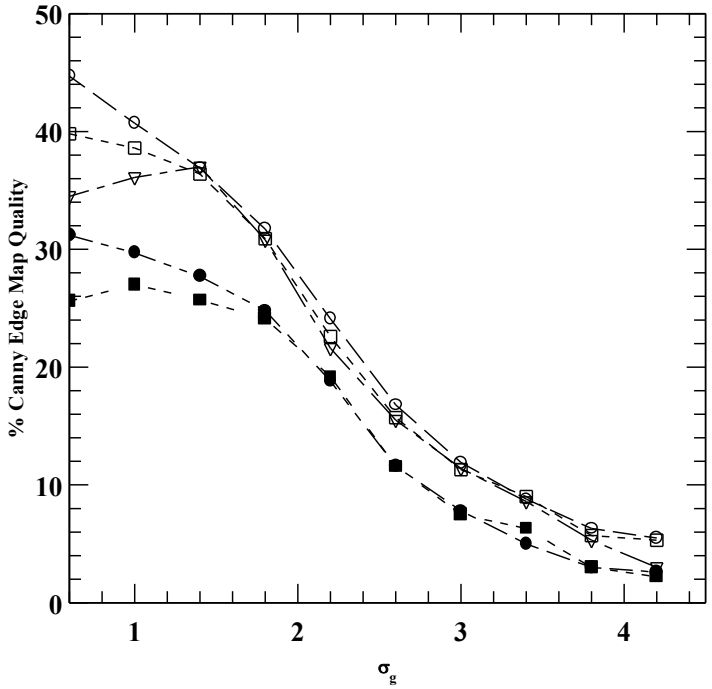

FIGURE 4.4: Edge map quality of smoothed Canny edge detector corrupted by Gaussian noise.

5. CONCLUSION AND FUTURE WORK

The aims of this research have been stated and methods are also suggested to aid in achieving the following objectives:

- The necessary comparison criteria that assist in the evaluation of the performance of edge detector are investigated.
- Edge detectors outputs 'edge marked' are evaluated by using the map quality MQ ratio.

The performance of three edge detectors are analyzed for five levels of Gaussian noise deviates; $\sigma_n = 1,10,30,50,$ and $75.$ At each noise level there are ten gradual steps of smoothing according to convolution kernel width given as $W = 5\sigma$.

Future work suggests that the learning set of artificially generated images used by Ramalho can be used to train the neural network which will (i) be utilized as an aid to the arbitration strategy, and (ii) used to define the ideal sigma σ_g that corresponds to the ideal noisy image within every level of noise. The neural network will also be used to determine the overall performance of the edge detector itself.

6. REFERENCES

[1] M. Ramalho. "Edge detection using neural network arbitration." Ph.D. Thesis, University of Nottingham, UK, 1996.

[2] M. Garcý´a-Silvente, J.A. Garcý´a, J. Fdez-Valdivia, A. Garrido. "A new edge detector integrating scale-spectrum information." Image and Vision Computing, vol. 15, pp. 913–923, 1997.

[3] M. Juneja, P. Sandhu. "Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain." International Journal of Computer Theory and Engineering, vol. 1, no. 5, 2009.

[4] I. Singh, A. Oberoi, M. Oberoi. "Performance Evaluation of Edge Detection Techniques for Square, Hexagon and Enhanced Hexagonal Pixel Images, International Journal of Computer Applications, vol. 121, no.12, 2015.

[5] M. Sonka, V. Hlavac, R. Boyle. Image Processing, Analysis, and Machine Vision. Toronto, CA: Thomson, 2008, pp. 132-146,

[6] D. Marr, E. Hildreth. "Theory of Edge Detection." Royal Society of London B, vol. 207, no. 1167, pp. 187-217, Feb. 29, 1980.

[7] R.C. Gonzalez, P. Wintz. Digital Signal Processing. Addison-Wesley Publishing Company, 1993.

[8] E. Trucco, A. Verri. Introductory Techniques for 3-D Computer Vision. Prentice Hall, 1998.

[9] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling. Numerical Recipes in Pascal, Cambridge University Press, 1989.

[10] M. Ramalho, and K.M. Curtis. "Neural Network Arbitration of Edge Maps," in Transputer Applications and Systems '94; A. Gloria., M. Jane., D. Marini. (Eds), IOS Press, Netherlands, 1994.

[11] J. Canny. "A computational approach to edge detection." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679-698, 1986.

[12] W.K. Pratt. Digital Image Processing, New York: Wiley InterSciencce, 1991.

[13] P.K. Shaoo, S. Soltani, A.K.C. Wong. "A survey of thresholding techniques." Computer Vision, Graphics, and Image Processing, vol. 41, pp. 233-260, 1998.

[14] S. Lakshmi, V. Sankaranarayanan (2010), "A study of Edge Detection Techniques for Segmentation Computing Approaches.", IJCA Special Issue on Computer Aided Soft Computing Techniques for Imaging and Biomedical Applications CASCT. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.206.3030&rep=rep1&type=pdf

[15] C.S. Panda, S. Patnaik (2009), "Filtering Corrupted Image and Edge Detection in Restored Grayscale Image Using Derivative Filters.", International Journal of Image Processing (IJIP). Vol. 3 (3), 105-119. Available: http://www.cscjournals.org/manuscript/Journals/IJIP/Volume3/Issue3/IJIP-28.pdf

APPENDIX A. Convolution Kernels:

$k_1 =$ 1 0 -1 2 0 -2 1 0 -1	$k_2 =$ 2 1 0 1 0 -1 0 -1 -2	$k_3 =$ 1 2 1 0 0 0 -1 -2 -1	$k_4 =$ 0 1 2 -1 0 1 -2 -1 0
$k_5 =$ -1 0 1 -2 0 2 -1 0 1	$k_6 =$ -2 -1 0 -1 0 1 0 1 2	$k_7 =$ -1 -2 -1 0 0 0 1 2 1	$k_8 =$ 0 -1 -2 1 0 -1 2 1 0

TABLE A.1: Sobel eight rotational kernels.

$k_1 =$ 1 0 -1 1 0 -1 1 0 -1	$k_2 =$ 1 1 0 1 0 -1 0 -1 -1	$k_3 =$ 1 1 1 0 0 0 -1 -1 -1	$k_4 =$ 0 1 1 -1 0 1 -1 -1 0
$k_5 =$ -1 0 1 -1 0 1 -1 0 1	$k_6 =$ -1 -1 0 -1 0 1 0 1 1	$k_7 =$ -1 -1 -1 0 0 0 1 1 1	$k_8 =$ 0 -1 -1 1 0 -1 1 1 0

TABLE A.2: Prewitt eight rotational kernels.

$k_1 =$ 3 3 3 3 0 3 -5 -5 -5	$k_2 =$ 3 3 3 -5 0 3 -5 -5 3	$k_3 =$ -5 3 3 -5 0 3 -5 3 3	$k_4 =$ -5 -5 3 -5 0 3 3 3 3
$k_5 =$ -5 -5 -5 3 0 3 3 3 3	$k_6 =$ 3 -5 -5 3 0 -5 3 3 3	$k_7 =$ 3 3 -5 3 0 -5 3 3 -5	$k_8 =$ 3 3 3 3 0 -5 3 -5 -5

TABLE A.3: Kirsch eight rotational kernels.

APPENDIX B. Gaussian Smoothing Kernels:

Gaussian Integer Kernels G_i:

$$G\left(x,y,\sigma\right)=e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The normalization factor f can be determined by

$$f = 1/g_{\min}$$, $g_{\min}$ is the minimum floating point value of kernel G

The entries of the non-normalized filter are

$$G_{int}\left(i,j,\sigma_g\right)=int\left[f\cdot G\left(i,j,\sigma_g\right)\right]$$, *int* indicates the closest integer

Ten integer Gaussian kernels at different scales σ :

3x3 Kernel:	**5x5 Kernel:**	**7x7 Kernel:**	**9x9 Kernel:**
$\sigma_g = 0.6$	$\sigma_g = 1.0$	$\sigma_g = 1.4$	$\sigma_g = 1.8$
1 2 1	0 1 1 1 0	0 0 1 1 1 0 0	0 0 0 1 1 1 0 0 0
2 10 2	1 4 6 4 1	0 1 3 4 3 1 0	0 1 1 2 2 2 1 1 0
1 2 1	1 6 10 6 1	1 3 6 8 6 3 1	0 1 3 5 5 5 3 1 0
	1 4 6 4 1	1 4 8 10 8 4 1	1 2 5 6 8 6 5 2 1
	0 1 1 1 0	1 3 6 8 6 3 1	1 2 5 8 10 8 5 2 1
		0 1 3 4 3 1 0	1 2 5 6 8 6 5 2 1
		0 0 1 1 1 0 0	0 1 3 5 5 5 3 1 0
			0 1 1 2 2 2 1 1 0
			0 0 0 1 1 1 0 0 0

11x11 Kernel:	**13x13 Kernel:**	**15x15 Kernel:**
$\sigma_g = 2.2$	$\sigma_g = 2.6$	$\sigma_g = 3.0$
0 0 0 0 1 1 1 0 0 0 0	0 0 0 0 1 1 1 1 1 0 0 0 0	0 0 0 0 0 1 1 1 1 1 0 0 0 0 0
0 0 1 1 2 2 2 1 1 0 0	0 0 0 1 1 1 2 1 1 1 0 0 0	0 0 0 1 1 1 1 1 1 1 1 1 0 0 0
0 1 2 3 4 4 4 3 2 1 0	0 0 1 2 2 3 3 3 2 2 1 0 0	0 0 1 1 2 2 2 2 2 2 2 1 1 0 0
0 1 3 4 6 6 6 4 3 1 0	0 1 2 3 4 5 5 5 4 3 2 1 0	0 1 1 2 2 3 4 4 4 3 2 2 1 1 0
1 2 4 6 8 8 8 6 4 2 1	1 1 2 4 6 6 6 6 6 4 2 1 1	0 1 2 2 4 5 6 6 6 5 4 2 2 1 0
1 2 4 6 8 10 8 6 4 2 1	1 1 3 5 6 8 8 8 6 5 3 1 1	1 1 2 3 5 6 8 8 8 6 5 3 2 1 1
1 2 4 6 8 8 8 6 4 2 1	1 2 3 5 6 8 10 8 6 5 3 2 1	1 1 2 4 6 8 8 8 8 8 6 4 2 1 1
0 1 3 4 6 6 6 4 3 1 0	1 1 3 5 6 8 8 8 6 5 3 1 1	1 1 2 4 6 8 8 10 8 8 6 4 2 1 1
0 1 2 3 4 4 4 3 2 1 0	1 1 2 4 6 6 6 6 6 4 2 1 1	1 1 2 4 6 8 8 8 8 8 6 4 2 1 1
0 0 1 1 2 2 2 1 1 0 0	0 1 2 3 4 5 5 5 4 3 2 1 0	1 1 2 3 5 6 8 8 8 6 5 3 2 1 1
0 0 0 0 1 1 1 0 0 0 0	0 0 1 2 2 3 3 3 2 2 1 0 0	0 1 2 2 4 5 6 6 6 5 4 2 2 1 0
	0 0 0 1 1 1 2 1 1 1 0 0 0	0 1 1 2 2 3 4 4 4 3 2 2 1 1 0
	0 0 0 0 1 1 1 1 1 0 0 0 0	0 0 1 1 2 2 2 2 2 2 2 1 1 0 0
		0 0 0 1 1 1 1 1 1 1 1 1 0 0 0
		0 0 0 0 0 1 1 1 1 1 0 0 0 0 0

17x17 Kernel:

$\sigma_g = 3.4$

0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 1 1 1 2 2 2 2 2 1 1 1 0 0 0
0 0 1 1 2 2 3 3 3 3 3 2 2 1 1 0 0
0 1 1 2 3 3 4 5 5 5 4 3 3 2 1 1 0
0 1 1 2 3 5 6 6 6 6 6 5 3 2 1 1 0
1 1 2 3 4 6 6 8 8 8 6 6 4 3 2 1 1
1 1 2 3 5 6 8 8 10 8 8 6 5 3 2 1 1
1 1 2 3 5 6 8 10 10 10 8 6 5 3 2 1 1
1 1 2 3 5 6 8 8 10 8 8 6 5 3 2 1 1
1 1 2 3 4 6 6 8 8 8 6 6 4 3 2 1 1
0 1 1 2 3 5 6 6 6 6 6 5 3 2 1 1 0
0 1 1 2 3 3 4 5 5 5 4 3 3 2 1 1 0
0 0 1 1 2 2 3 3 3 3 3 2 2 1 1 0 0
0 0 0 1 1 1 2 2 2 2 2 1 1 1 0 0 0
0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0

19x19 Kernel:

$\sigma_g = 3.8$

0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 1 1 1 2 2 2 2 2 1 1 1 1 0 0 0 0
0 0 1 1 1 2 2 3 3 3 3 3 2 2 1 1 1 0 0
0 0 1 1 2 2 3 4 4 4 4 4 3 2 2 1 1 0 0
0 1 1 2 2 3 4 5 6 6 6 5 4 3 2 2 1 1 0
0 1 1 2 3 4 5 6 6 6 6 6 5 4 3 2 1 1 0
1 1 2 3 4 5 6 8 8 8 8 8 6 5 4 3 2 1 1
1 1 2 3 4 6 6 8 8 10 8 8 6 6 4 3 2 1 1
1 1 2 3 4 6 6 8 10 10 10 8 6 6 4 3 2 1 1
1 1 2 3 4 6 6 8 8 10 8 8 6 6 4 3 2 1 1
1 1 2 3 4 5 6 8 8 8 8 8 6 5 4 3 2 1 1
0 1 1 2 3 4 5 6 6 6 6 6 5 4 3 2 1 1 0
0 1 1 2 2 3 4 5 6 6 6 5 4 3 2 2 1 1 0
0 0 1 1 2 2 3 4 4 4 4 4 3 2 2 1 1 0 0
0 0 1 1 1 2 2 3 3 3 3 3 2 2 1 1 1 0 0
0 0 0 1 1 1 2 2 2 2 2 1 1 1 1 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0

21x21 Kernel:

$\sigma_g = 4.2$

0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 1 1 1 1 2 2 2 2 1 1 1 1 1 0 0 0 0
0 0 0 1 1 1 2 2 2 2 2 2 2 2 1 1 1 0 0 0 0
0 0 1 1 1 2 2 3 3 4 4 4 3 3 2 2 1 1 1 0 0
0 0 1 1 2 2 3 4 4 5 5 5 4 4 3 2 2 1 1 0 0
0 1 1 2 2 3 4 5 6 6 6 6 6 5 4 3 2 2 1 1 0
0 1 1 2 3 4 5 6 6 8 8 8 6 6 5 4 3 2 1 1 0
1 1 1 2 3 4 6 6 8 8 8 8 8 6 6 4 3 2 1 1 1
1 1 2 2 4 5 6 8 8 8 10 8 8 8 6 5 4 2 2 1 1
1 1 2 2 4 5 6 8 8 10 10 10 8 8 6 5 4 2 2 1 1
1 1 2 2 4 5 6 8 8 8 10 8 8 8 6 5 4 2 2 1 1
1 1 1 2 3 4 6 6 8 8 8 8 8 6 6 4 3 2 1 1 1
0 1 1 2 3 4 5 6 6 8 8 8 6 6 5 4 3 2 1 1 0
0 1 1 2 2 3 4 5 6 6 6 6 6 5 4 3 2 2 1 1 0
0 0 1 1 2 2 3 4 4 5 5 5 4 4 3 2 2 1 1 0 0
0 0 1 1 1 2 2 3 3 4 4 4 3 3 2 2 1 1 1 0 0
0 0 0 1 1 1 2 2 2 2 2 2 2 2 1 1 1 0 0 0 0
0 0 0 0 1 1 1 1 2 2 2 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0

APPENDIX C. Tabular Form of Results:

σ_G	Kernel	Threshold	%Correct	%Missed	%Wrong	%EQ	%MQ
0.6	3x3	107	85.5	14.5	10.7	88.9	77.2
1.0	5x5	104	80.2	19.8	12.1	86.9	71.5
1.4	7x7	98	74.6	25.4	12.8	85.3	66.1
1.8	9x9	88	69.9	30.1	13.3	84.0	61.7
2.2	11x11	78	66.2	33.8	13.5	83.1	58.3
2.6	13x13	68	62.2	37.8	13.5	82.2	54.8
3.0	15x15	61	55.8	44.2	13.4	80.6	49.2
3.4	17x17	55	50.8	49.2	12.9	79.7	45.0
3.8	19x19	50	43.8	56.2	12.0	78.5	39.1
4.2	21x21	46	39.5	60.5	11.6	77.3	35.4

TABLE C.1: Sobel Edge Detector: Image title "Band B". Noise standard deviation σ_N=1.0. Threshold= (mean+ standard deviation).

σ_G	σ_N=1.0	σ_N=10.08	σ_N=31.43	σ_N=50.41	σ_N=75.41
0.6	77.2	73.7	63.4	47.3	35.7
1.0	71.5	68.4	62.2	49.6	38.2
1.4	66.1	63.9	56.6	43.4	29.2
1.8	61.7	58.7	49.4	31.4	14.4
2.2	58.3	54.8	37.2	15.0	5.6
2.6	54.8	47.6	22.6	7.7	1.3
3.0	49.2	37.9	12.5	1.4	0.0
3.4	45.0	32.4	6.8	0.0	0.0
3.8	39.1	23.1	1.3	0.0	0.0
4.2	35.4	16.2	0.0	0.0	0.0

TABLE C.2: Sobel %Map Quality at 5 Noise levels. Threshold = (mean + standard deviation).

σ_G	σ_N=1.0	σ_N=10.08	σ_N=31.43	σ_N=50.41	σ_N=75.41
0.6	74.3	70.5	60.4	44.5	33.3
1.0	68.8	66.2	59.0	45.5	33.6
1.4	64.3	62.5	54.2	39.1	22.9
1.8	60.0	57.3	45.7	24.4	10.9
2.2	56.6	52.4	31.2	11.6	3.5
2.6	51.5	40.6	17.4	4.1	0.8
3.0	43.2	32.5	10.3	0.8	0.0
3.4	38.3	26.1	4.3	0.0	0.0
3.8	31.8	15.4	0.6	0.0	0.0
4.2	25.2	10.6	0.0	0.0	0.0

TABLE C.3: Prewitt Edge Detector: Prewitt %Map Quality at 5 Noise levels. Threshold = (mean + standard deviation).

σ_G	σ_N=1.0	σ_N=10.08	σ_N=31.43	σ_N=50.41	σ_N=75.41
0.6	44.7	39.8	34.5	31.2	25.6
1.0	40.7	38.6	36.1	29.7	27.0
1.4	36.9	36.4	37.0	27.7	25.7
1.8	31.7	30.9	30.8	24.8	24.1
2.2	24.1	22.6	21.6	18.8	19.2
2.6	16.8	15.7	15.5	11.6	11.6
3.0	11.9	11.3	11.4	7.8	7.5
3.4	8.8	9.0	8.6	5.0	6.3
3.8	6.3	5.7	5.3	3.0	3.0
4.2	5.5	5.3	3.0	2.6	2.2

TABLE C.4: Canny Edge Detector: Canny %Map Quality at 5 Noise levels.