# QUALITATIVE AND QUANTITATIVE STUDY OF EDGE DETECTION ALGORITHMS

**Mr. Hasamukh Patel***

**Dr. PriyaSwaminarayan****

**ABSTRACT**

Edge detection is one of the most commonly used operations in image analysis.An edge is the border between an object and the background, and indicates the boundary between overlapping objects. Basic properties of the objects such as area, perimeter, and shape can be measured if the objects in the image can be identified accurately. In this paper, we have compared techniques for edge detection in image processing. We consider various well-known measuring metrics used in image processing applied to standard images in this comparison.

* *Rai University, Gujarat, India*

** *Professor, Department of CS, ISTAR,GTU, V.V.Nagar, Gujarat, India*

## Introduction

Virtually every branch of science has sub-disciplines that use recording devices or sensors to collect image data from the universe around us. This data is often multidimensional and can be arranged in a format that is suitable for human viewing. Viewable datasets like this can be regarded as images and processed using established techniques for image processing; even if the information has not been derived from visible light sources. One aspect of image processing that makes it such an interesting topic of study is the amazing diversity of applications that make use of image processing or analysis techniques.

Edge detection is one of the most commonly used operations in image analysis, and there are probably more algorithms in the literature for enhancing and detecting edges than any other single subject. Edge detection is a low level operation used in image processing and computer vision applications. The main goal of edge detection is to locate and identify sharp discontinuities from an image. These discontinuities are due to abrupt changes in pixel intensity which characterizes boundaries of objects in a scene.

The reason for this is that edges form the outline of an object, in the generic sense. Objects are subjects of interest in image analysis and vision systems. An edge is the boundary between an object and the background, and indicates the boundary between overlapping objects. These boundaries are used to identify objects for segmentation and matching purpose (Bhardwaj & Mittal, 2012).This means that if the edges in an image can be identified accurately, all the objects can be located, and basic properties such as area, perimeter, and shape can be measured. Since computer vision involves the identification and classification of objects in an image, edge detection is an essential tool.

Edges are places in the image with strong intensity contrast. Since edges often occur at image locations representing object boundaries, edge detection is extensively used in image segmentation when we want to divide the image into areas corresponding to different objects. Representing an image by its edges has the further advantage that the amount of data is reduced significantly while retaining most of the image information (Blackledge, 2005; Marr & Hildreth, 1980).

Since edges consist of mainly high frequencies, we can, in theory, detect edges by applying a high pass frequency filter in the Fourier domain or by convolving the image with an appropriate kernel in the spatial domain. In practice, edge detection is performed in the spatial domain, because it is computationally less expensive and often yields better results.

Since edges correspond to strong illumination gradients, we can highlight them by calculating the derivatives of the image. This is illustrated for the one-dimensional case in Figure 1.

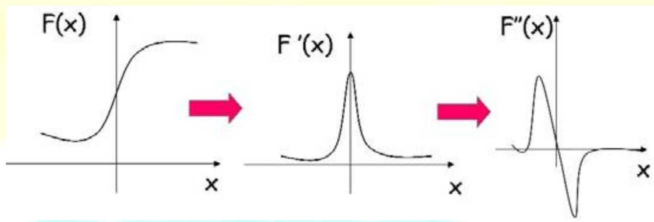


**Figure 1:** F(x): The one dimensional illumination Gradient as an edge; F'(x): first order derivative of F(x) and F''(x) the second order derivative

The position of the edge can be estimated with the maximum of the 1st derivative or with the zero-crossing of the 2nd derivative.

Apart from standard methods based on illumination gradient at gray level few standard operators are available for edge detection. They can be further classified in to two main categories,

In First order derivative (Gonzalez & Woods, 2002) the input image is convolved by an adapted mask to generate a gradient image in which edges are detected by thresholding. Most classical operators like sobel, prewitt, robert(Canny, 1986) are the first order derivative operators. These operators are also said as gradient operators. These gradient operators detect edges by looking for maximum and minimum intensity values. These operators examine the distribution of intensity values in the neighborhood of a given pixel and determine if the pixel is to be classified as an edge. These operators have more computational time and can't be used in real-time application.

In second order derivative (Gonzalez & Woods, 2002), these are based on the extraction of zero crossing points which indicates the presence of maxima in the image. In this, image is first smoothed by adaptive filters (Geng, Chen, & Hu, 2012).Since the second order derivative is very sensible to noise, and the filtering function is very important. These operators are derived from the Laplacian of a Gaussian (LOG), and proposed by Marr and Hildreth(Geng et al., 2012; Marr & Hildreth, 1980), in this, the image is smoothed by a Gaussian filter. For this operator we have to fix some parameters such as the variance of the Gaussian filter and thresholds. Some methods are available for their automatic computation (Marr & Hildreth, 1980; Yuesong & Jianqiao, 2010), but in most cases their values have to be fixed by the user. A significant problem of LOG is that the localization of edges with an asymmetric profile by zero-crossing points introduces a bias which increases with the smoothing effect of filtering (Xiaobing, Baokui, & Qingbo, 2012).An interesting solution to this problem was proposed by Canny (Canny, 1986), which says in an optimal operator for step edge detection includes three criteria: good detection, good localization, and only one response to a single edge.

**Methodology:**

**First order edge detection:**

It is based on the use of a first order derivative, or can say gradient based. If I (i, j) be the input image, then image gradient is given by following formula

$$\nabla I(i,j) = i\frac{\partial I(i,j)}{\partial i} + j\frac{\partial I(i,j)}{\partial j}$$

Where: $\frac{\partial I(i,j)}{\partial i}$ is the gradient in the i direction.

$\frac{\partial I(i,j)}{\partial j}$is the gradient in the j direction.

The gradient magnitude can be computed by the formula:

$$|G| = \sqrt{\left(\frac{\partial I}{\partial i}\right)^2 + \left(\frac{\partial I}{\partial i}\right)^2} \; or \; |G| = \sqrt{G_i^2 + G_j^2}$$

The gradient magnitude can be computed by the formula:

$$\theta = \arctan\left(\frac{G_i}{G_j}\right).$$

The magnitude of gradient computed above gives edge strength and the gradient direction is always perpendicular to the direction of edge.

**Classical operators:**

Robert, Sobel, Prewitt are classified as classical operators which are easy to operate but highly sensitive to noise.

**Robert operator (Ahammer & DeVaney, 2004; Gonzalez & Woods, 2002)**

It is gradient based operator. It firstly computes the sum of the squares of the difference between diagonally adjacent pixels through discrete differentiation and then calculate approximate gradient of the image. The input image is convolved with the default kernels of operator and gradient magnitude and directions are computed. It uses following 2 x2 two kernels:

$$D_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \; and \; D_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

The plus factor of this operator is its simplicity but having small kernel it is highly sensitive to noise not and not much compatible with today's technology.

**Sobel operator (Ahammer & DeVaney, 2004; Gonzalez & Woods, 2002)**

Sobel operator is a discrete differentiation operator used to compute an approximation of the gradient of image intensity function for edge detection. At each pixel of an image, sobel operator gives either the corresponding gradient vector or normal to the vector. It convolves the input image with kernel and computes the gradient magnitude and direction. It uses following 3x3 two kernels:

$$D_i = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \; and \; D_j = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

As compared to Robert operator have slow computation ability but as it has large kernel so it is less sensitive to noise as compared to Robert operator. As having larger mask, errors due to effects of noise are reduced by local averaging within the neighborhood of the mask.

**Prewitt operator (Gonzalez & Woods, 2002)**

The function of Prewitt edge detector is almost same as of Sobel detector but have different kernels:

$$D_i = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \; and \; D_j = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Prewitt edge operator gives better performance than that of Sobel operator.

**Canny edge detector (Canny, 1986)**

Canny edge detector have advanced algorithm derived from the previous work of Marr and Hildreth. It is an optimal edge detection technique as provide good detection, clear response and good localization. It is widely used in current image processing techniques with further improvements.

**Canny edge detection algorithm (Li, Song, & Cao, 2014)**

**STEP I: Noise reduction by smoothing**

Noise contained in image is smoothed by convolving the input image I (i, j) with Gaussian filter G. Mathematically, the smooth resultant image is given by Prewitt operators are simpler to operator as compared to sobel operator but more sensitive to noise in comparison with sobel operator.

$$F(i,j) = G * I(i,j)$$

**STEP II: Finding gradients**

In this step we detect the edges where the change in grayscale intensity is maximum. Required areas are determined with the help of gradient of images. Sobel operator is used to determine the gradient at each pixel of smoothened image. Sobel operators in i and j directions are given as

$$D_i = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \; and \; D_j = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

These Sobel masks are convolved with smoothed image and giving gradients in i and j directions.

$$G_i = D_i * F(i,j) \; and \; G_j = D_j * F(i,j)$$

Therefore edge strength or magnitude of gradient of a pixel is given by

$$G = \sqrt{G_i^2 + G_j^2}$$

The direction of gradient is given by,

$\theta = \arctan\left(\frac{G_i}{G_j}\right)$ $G_i$ and $G_j$ are the gradients in i- and j- directions respectively.

**STEP III: Non maximum suppressions:**

Non maximum suppression is carried out to preserves all local maxima in the gradient image, and deleting everything else this results in thin edges. For a pixel M (i, j):

• Firstly round the gradient direction θ nearest45°,then compare the gradient magnitude of the pixels in positive and negative gradient directions i.e. If gradient direction is east then compare with gradient of the pixels in east and west directions say E (i, j) and W (i, j)respectively.

• If the edge strength of pixel M (i, j) is largest than that of E (i, j) and W (i, j), then preserve the value of gradient and mark M (i, j) as edge pixel, if not then suppress or remove.

**STEP IV: Hysteresis thresholding**:

The output of non-maxima suppression still contains the local maxima created by noise. Instead choosing a single threshold, for avoiding the problem of streaking two thresholds $t_{high}$ and $t_{low}$ are used.

For pixel M(i, j) having gradient magnitude G following conditions exists to detect pixel as edge: i) If G $<t_{low}$ than discard the edge. ii) If G > $t_{high}$ keep the edge. iii) If $t_{low}<$ G < $t_{high}$ and any of its neighbors in 3x3 region around it have gradient magnitudes greater than $t_{high}$, keep the edge. iv) If none of pixel (x, y)'s neighbors have high gradient magnitudes but at least one falls between $t_{low}$ and $t_{high}$ search the 5x5 region to the see if any of these pixels have a magnitude greater than $t_{high}$. If so, keep the edge else discard the edge.

**SECOND ORDER EDGE DETECTOR (Blackledge, 2005; Ganesan & Bhattacharyya, 1997; Gonzalez & Woods, 2002)**

It is based on second order derivative, in particular, the Laplacian$\nabla^2$.In this operator a pixel is marked as an edge at a position where second derivative of an image becomes zero. The laplacian operator $\nabla^2$ for a 2D image I (i, j) is defined by following formula:

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

**Laplacian of Gaussian or Marr Hildrith operator (Ahammer & DeVaney, 2004; Woolford, Hankamer, & Ericksson, 2007)**

The Marr-Hildreth edge detector was a very popular edge operator before Canny proposed his algorithm. It is a gradient based operator which uses the Laplacian to take the second derivative of an image. It works on zero crossing method. It uses both Gaussian and laplacian operator so that Gaussian operator reduces the noise and laplacian operator detects the sharp edges.

This paper focuses on the evaluation of the performance of edge detector algorithms considering synthetic images and images of real scenes. Several measures are obtained by comparison to manually specified ideal image (ground truth).

Comparison of an edge map, obtained by a detector of edges, with its ground truth can be achieved through a set of direct measurements, such as the number of correctly detected edge pixels, called true positive (TP), the number of pixels erroneously classified as edge pixels, called false positive (FP), the amount of edge pixels that were not classified as edge pixel, called false negative (FN). From these measures, the following statistical indices have been proposed:

The Gaussian function is defined by the formula:

$$G(i,j) = \frac{1}{\sqrt{2\pi\sigma^2}} exp - \left(\frac{i^2 + j^2}{2\sigma^2}\right)$$

Where, σ is standard deviation and LoG operator is computed from

$$LoG = \frac{\partial^2}{\partial i^2} G(i,j) + \frac{\partial^2}{\partial j^2} G(i,j) = \frac{i^2 + j^2 - 2\sigma^2}{\sigma^4} exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right)$$

The Marr–Hildreth operator, however, suffers from two main limitations. It generates responses that do not correspond to edges, so-called "false edges", and the localization error may be severe at curved edges.

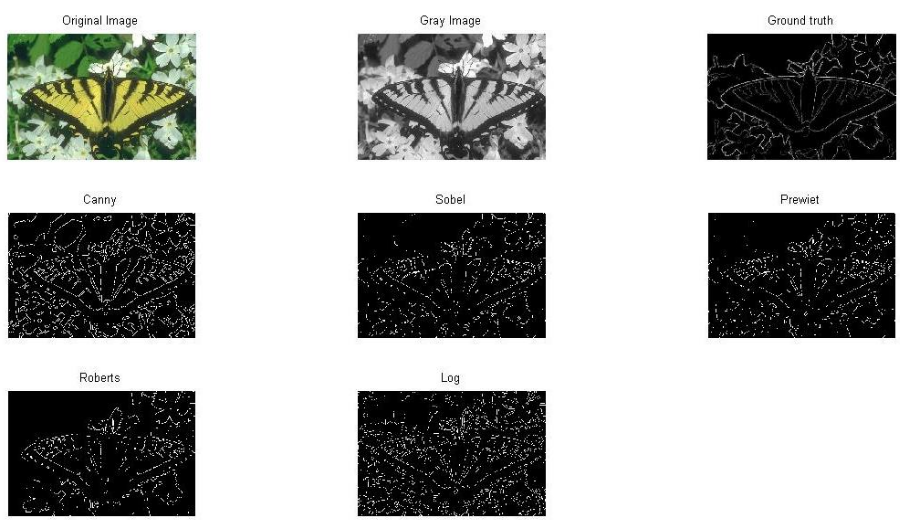


**Figure 2:** Output of various operators on Gray Scale Original Image 1.

**Statistical Analysis**

The experiments are performed on 6 images (taken from the database of images *Berkeley Segmentation Dataset*, *Matlab test images* and this database brings a set of 200 images of natural scenes and their ground truth produced manually). Their performance evaluation was carried qualitatively as well as by means of performance ratio and PSNR.

The statistical significance of *Peak Signal to Noise Ratio*

Peak signal-to-noise ratio, is ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The PSNR usually expressed in terms of the decibel (dB) scale. PSNR is a rough estimation to human perception of reconstruction quality.

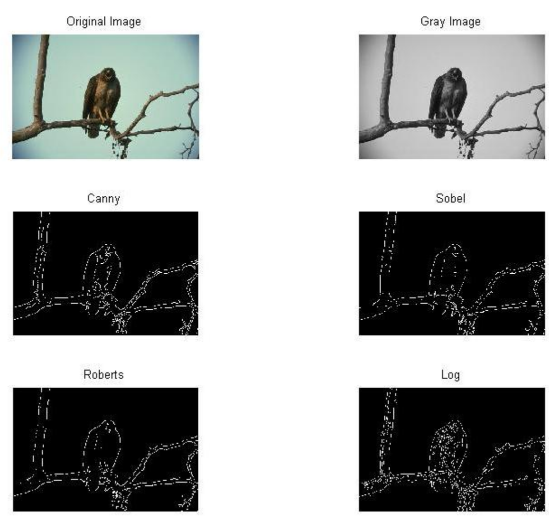


**Figure 3:**Output of various operators on Gray Scale Original Image 2.

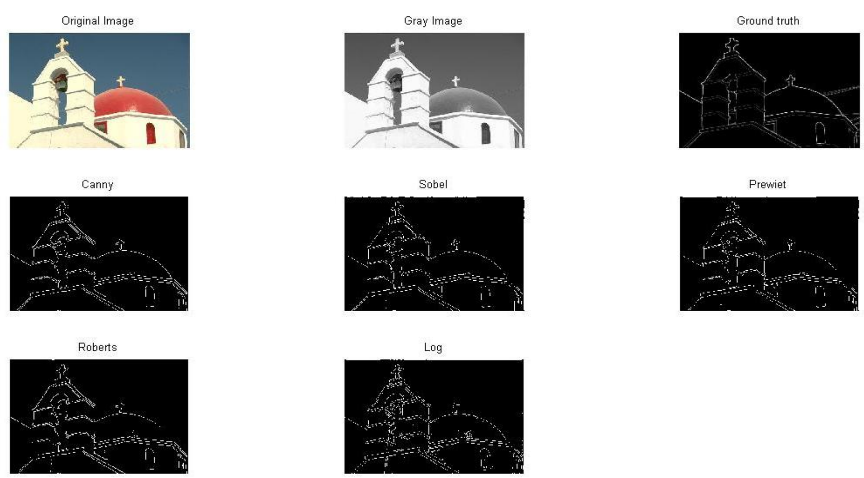


**Figure 4:**Output of various operators on Gray Scale Original Image 3.

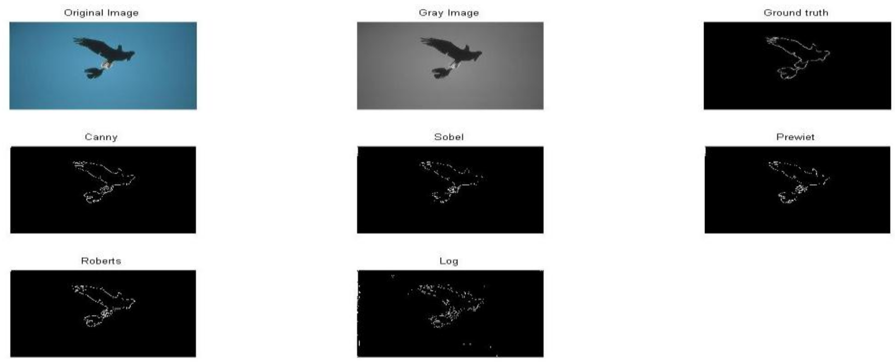


**Figure 5:**Output of various operators on Gray Scale Original Image 4

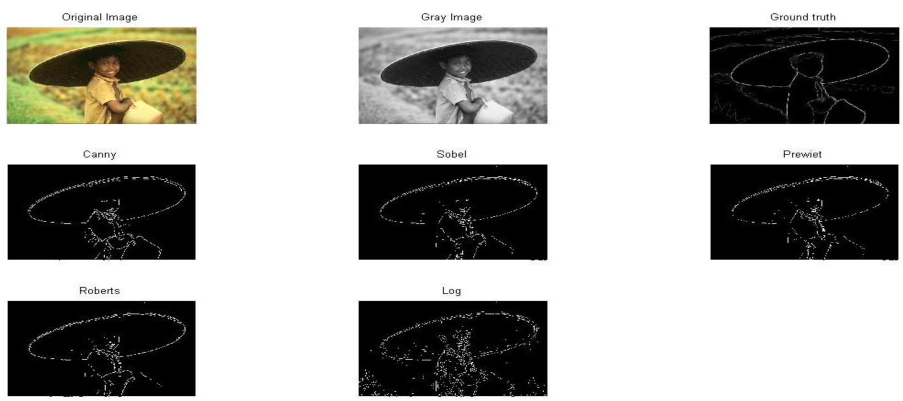


**Figure 6:**Output of various operators on Gray Scale Original Image 5.

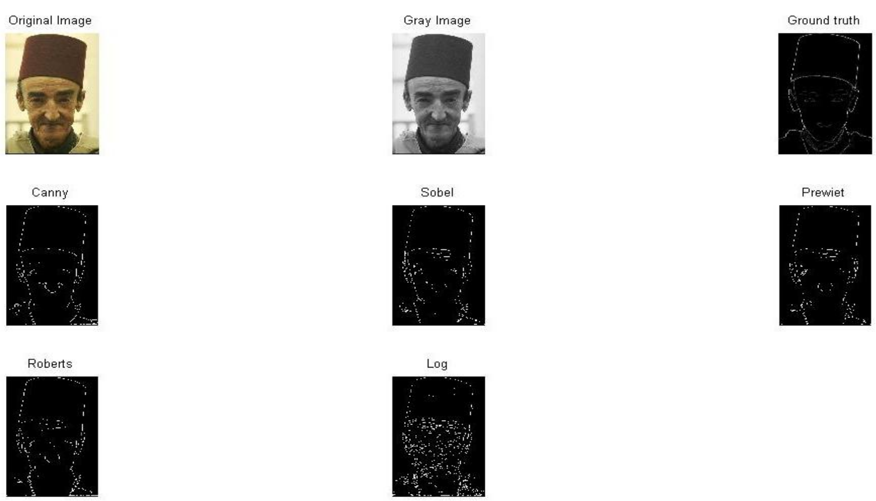


**Figure 7:**Output of various operators on Gray Scale Original Image 6.

**The statistical significance of *Peak Signal to Noise Ratio***

Peak signal-to-noise ratio, is ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The PSNR usually expressed in terms of the decibel (dB) scale. PSNR is a rough estimation to human perception of reconstruction quality. Although a higher PSNR generally indicates that the reconstruction is of higher quality in image compression. But in some cases like edge detection PSNR should lesser to achieve proper results. The PSNR calculated based on the MSE by

$$PSNR = 10log_{10}\left(\frac{R^2}{MSE}\right)$$

*R* is the maximal variation in the input image data. If it has an 8-bit unsigned integer data type, *R* is 255.

Performance ratio is the ratio of True edge pixels identified as Edges (TP) to non-edge pixels identified as edges (FN) and edge pixels identified as non-edge pixels (TN).

$$PR = \frac{TP}{FN + TN}$$

Table 1 represents the PR and PSNR values for each operator

| Image | | Canny | Sobel | Prewiet | Roberts | LoG |
|---|---|---|---|---|---|---|
| 1 | PR | 17.1933 | 9.6267 | 9.5487 | 9.8715 | 11.6908 |
| | PSNR (db) | 17.2235 | 17.2105 | 17.2104 | 17.2154 | 17.2118 |
| 2 | PR | 20.3959 | 15.4506 | 15.4654 | 19.83 | 15.6133 |
| | PSNR (db) | 17.9855 | 17.974 | 17.974 | 17.9862 | 17.9726 |
| 3 | PR | 15.2432 | 11.1709 | 11.113 | 14.7262 | 10.808 |
| | PSNR (db) | 19.9279 | 19.9181 | 19.9179 | 19.9284 | 19.9168 |
| 4 | PR | 15.2432 | 11.1709 | 11.113 | 14.7262 | 10.808 |
| | PSNR (db) | 19.9279 | 19.9181 | 19.9179 | 19.9284 | 19.9168 |
| 5 | PR | 7.9504 | 6.1243 | 6.145 | 7.4097 | 7.4878 |
| | PSNR (db) | 20.6949 | 20.6859 | 20.686 | 20.6925 | 20.6874 |
| 6 | PR | 11.5043 | 8.9997 | 8.9302 | 10.4054 | 10.1631 |
| | PSNR (db) | 20.9899 | 20.9797 | 20.9796 | 20.9901 | 20.9811 |

It is clearly observed from the results (Table 1) that, Canny edge detector (14.58838) yields the highest average performance in terms of both PR. While the gradient based operator Roberts using 2x2 kernels have fairly good average performance ratio (12.82817) followed by second order derivative Marr-Hildreth edge detector (LoG) (11.09517) then other 3x3 kernels i.e. Sobel (10.58838) and Prewiet (10.38588). While, the Figures 2-7 demonstrate the qualitative assessment for stated operators.

## Conclusion

When the object dimensions are important their boundary must be precise and accurate i.e. in the field of industrial automations the perfect edges are required. It can be concluded that 'Canny' operator is the best in comparison to other standard operators which can be used further in image segmentation and object tracing as well as object detection algorithms.

## References

[1] Ahammer, H., & DeVaney, T. T. (2004). The influence of edge detection algorithms on the estimation of the fractal dimension of binary digital images. *Chaos, 14*(1), 183-188. doi: 10.1063/1.1638947

[2] Bhardwaj, S., & Mittal, A. (2012). A Survey on Various Edge Detector Techniques. *Procedia Technology, 4*(0), 220-226. doi: http://dx.doi.org/10.1016/j.protcy.2012.05.033

[3] Blackledge, J. M. (2005). Chapter 16 - Segmentation and Edge Detection. In J. M. Blackledge (Ed.), *Digital Image Processing* (pp. 487-511): Woodhead Publishing.

[4] Canny, J. (1986). A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell, 8*(6), 679-698

[5] Ganesan, L., & Bhattacharyya, P. (1997). Edge detection in untextured and textured images-a common computational framework. *IEEE Trans Syst Man Cybern B Cybern, 27*(5), 823-834. doi: 10.1109/3477.623235

[6] Geng, X., Chen, K., & Hu, X. (2012). An improved Canny edge detection algorithm for color image *Industrial Informatics (INDIN), 2012 10th IEEE International Conference on* (pp. 113-117).

[7] Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing* (2nd ed.). Upper Saddle River, N.J.: Prentice Hall.

[8] Li, W., Song, W., & Cao, J. (2014). An Image Sub-pixel Edge Detection Algorithm of Plant Roots Based on Non-linear Fourth-order Interpolation Method. *International Journal of Hybrid Information Technology, 7*(3), 275-283

[9] Marr, D., & Hildreth, E. (1980). Theory of edge detection. *Proc R Soc Lond B Biol Sci, 207*(1167), 187-217

[10] Woolford, D., Hankamer, B., & Ericksson, G. (2007). The Laplacian of Gaussian and arbitrary z-crossings approach applied to automated single particle reconstruction. *J Struct Biol, 159*(1), 122-134. doi: S1047-8477(07)00076-7 [pii]; 10.1016/j.jsb.2007.03.003

[11] Xiaobing, W., Baokui, L., & Qingbo, G. (2012). Runway Detection and Tracking for Unmanned Aerial Vehicle Based on an Improved Canny Edge Detection Algorithm *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on* (Vol. 2, pp. 149-152).

[12] Yuesong, M., & Jianqiao, Y. (2010). An Algorithm for Automatic Extraction of Moving Object in the Image Guidance *Intelligent System Design and Engineering Application (ISDEA), 2010 International Conference on* (Vol. 1, pp. 226-230).