ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor: 5.2
IJAR 2016; 2(11): 257-262
www.allresearchjournal.com
Received: 09-09-2016
Accepted: 10-10-2016

Bhakti Batra
B.Tech Maharaja Surajmal Institute of Technology, Janakpuri, Delhi (GGSIPU), Delhi, India

Saurav Singh
B.Tech Maharaja Surajmal Institute of Technology, Janakpuri, Delhi (GGSIPU), Delhi, India

Jyotirmay Sharma
B.Tech Maharaja Surajmal Institute of Technology, Janakpuri, Delhi (GGSIPU), Delhi, India

Shaifali M Arora
Assistant Professor Maharaja Surajmal Institute of Technology, Janakpuri, Delhi (GGSIPU), Delhi, India

Correspondence
Bhakti Batra
B.Tech Maharaja Surajmal Institute of Technology, Janakpuri, Delhi (GGSIPU), Delhi, India

Computational analysis of edge detection operators

Bhakti Batra, Saurav Singh, Jyotirmay Sharma and Shaifali M Arora

Abstract
Edge detection is a fundamental tool in Digital Image Processing. It is widely used for image segmentation in the areas such as image processing, computer vision, and machine vision, particularly for feature detection and feature extraction. Edge detectors are used as pre-processing tools in computer vision that makes image segregation and pattern recognition more comfortable. Many edge detectors are available for pre-processing in computer vision but Canny, Sobel, Prewitt, Roberts and Laplacian of Gaussian (LoG) are among the most applied algorithms. In this paper, each of these algorithms is compared by the manner of comparing the Peak Signal to Noise Ratio (PSNR) and Mean Squared Error (MSE) of the resulting images. Together with PSNR and MSE, time required by each algorithm is also computed to compare these algorithms. The experiment is performed and evaluated on MATLAB. The performance of canny operator is found to be best amongst all the edge detection operators compared in this paper.

Keywords: Canny, Laplacian of Gaussian, Sobel, Prewitt, Robert, edge detection, MSE, PSNR

1. Introduction
While most signals are a measure of parameter over time, Images are a measure of parameter over space. Edge detection is a process of locating boundaries of objects present in an image. Edge detection is one of the important steps towards understanding features of an image because they consist of meaningful features and contain significant information. It consequently reduces the image size by filtering out irrelevant information, thus preserving the important structural properties of an image. Redundant information of an image can sometimes be removed by detecting edges and replacing them during reconstruction. Therefore, edge detection algorithms may notably minimise the amount of data to be processed by preserving the salient structural properties of an image. This in turn reduces the space occupied by an image and the processing time [1]. Since edges often occur at image locations representing object boundaries, edge detection is broadly used in image segmentation when images are divided into areas corresponding to different objects. An edge in an image is an abrupt change in the image intensity associated with a discontinuity in the image intensity. The function of an edge detector is to eliminate independent pixels present in the input image. Discontinuities in the image intensity can be either step discontinuities, where the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side, or line discontinuities, where the image intensity abruptly changes value but then returns to the starting value within some short distance [2]. Because of the smoothing introduced by most sensing devices, sharp discontinuities rarely exist in real signals. Consequently, resulting in ramp edges and roof edges from step and line edges respectively, where intensity changes are not instantaneous but occur over a finite distance. By definition, an edge is a significant local change in image intensity. The changes due to noise are not edges even though the changes are local, because the changes are not significant. The changes due to shading, which gives characteristics of the ramp, are not edges even though the changes are significant, because the changes are not local. Usually, real images are very noisy. It is difficult to develop an edge detection operator that reliably finds step edges and is immune to noise [2]
The edge detectors are responsible for determining and effectively acquiring the lines which are in the exact locations of the image edges. Each algorithm has its own gradient operator for detection, so results may vary when the lines are plot. The notable edge detectors are compared on the basis of accuracy in results, edge continuity, noise level, processing time

etc. In order to measure noise level, two standards are universally followed, Mean Squared Error (MSE), and Peak Signal to Noise Ratio (PSNR).

The Mean Squared Error (MSE) measures the average of the squares of the errors or deviations—that is, the difference between the resultant images produced by the edge detector operator and ground truth image. PSNR represents a measure of the peak error [9]. While comparing edge detection operators with common image, if an operator gives resultant image with low PSNR, high MSE and less processing time then the operator is said to be of high edge detection quality [9].

A productive comparison of renowned edge detection operators is proposed in this paper based on their abilities to produce unambiguous edges to determine object boundaries.

2. Edge Detection Techniques

The motivation behind identifying notable local changes in image brightness is to encapsulate important events and changes in properties of the world [3]. It can be shown that under general assumptions for an image formation model, discontinuities in image brightness are likely to correspond to:

- discontinuities in depth,
- discontinuities in surface orientation,
- variations material properties and
- Discrepancy in scene illumination.

The outcome of implementing an edge detector to an image gives rise to a set of connected curves that denote the borderlines of objects, the boundaries of surface contours as well as curves that dovetail with discontinuities in surface orientation [3].

There are mainly two types of edge detection techniques, search-based and zero-crossing based.

The search-based techniques identify edges by first finding a first-order derivative expression like the gradient magnitude which gives a measure of edge strength, and then by using a computed estimate of the local orientation of the edge, usually the gradient direction, it searches for directional maxima which are local of the gradient magnitude [4]. Gradient vector and magnitude for an image f are given as:

$$\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1)$$

The following expression (angle) gives the direction of the gradient vector:

$$\propto (x, y) = \tan^{-1} \begin{bmatrix} g_y \\ g_x \end{bmatrix} \quad (2)$$

In zero crossing based methods firstly the second order derivative expression, represented by equation (3), is computed from the image and edges are detected by searching zero crossings. Mostly the zero-crossings of the non-linear differential expression or the zero-crossings of a Laplacian differential expression are searched.

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \quad (3)$$

In this paper, the five renowned edge detection operators are compared. These are explained as follows:

i. Roberts operator

The following masks are used by Roberts's edge detector to approximate digitally the first derivatives as differences between adjacent pixels [2]:

$$\underset{g_x}{\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}} \quad \underset{g_y}{\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}}$$

It is one of the oldest edge detectors in DIP and the above figure shows that it is also the simplest. It is non symmetric and cannot be used to detect edges that are the multiple of 45 degree. However, it is still used widely in hardware applications. As its kernel is a smaller one, so it is quite sensitive to noise.

ii. Sobel operator

The Sobel edge detector computes the gradient by using discrete differences between rows and columns of a 3×3 neighbourhood where the centre pixel in each row or column is weighted by 2 to provide smoothening. The output image is a logical image 1s at locations where edges were detected and 0s elsewhere. This also increases the edge intensity and it becomes enhanced comparatively to the original image.

$$\underset{g_x}{\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}} \quad \underset{g_y}{\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}}$$

The difference between the pixel intensities of the particular edge can be calculated by the above-mentioned mask. As the centre row of mask consist of zeros so it does not have the real (native or first) values of edge in the image but it rather calculates the difference of above and below pixel intensities of the particular (given) edge. Thus increasing the sudden change of intensities and making the edge more visible.

iii. Prewitt operator

The prewitt edge detector is slightly simpler to implement than sobel detector but produce somewhat noisier results. The horizontal and vertical derivative approximation at each point is represented by two images G_x and G_y respectively [5].

$$\underset{g_x}{\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}} \quad \underset{g_y}{\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}}$$

Computational analysis of edge detection operators will be highlighted by the mask, G_x. It operates on the principle of above-mentioned mask and calculates the difference among the pixel intensities of a particular edge. Since the centre row of mask contains zero hence, it does not incorporate the original values of edge in the image, instead it calculates the difference between above and below pixel intensities of the particular edge, thus elevating the sudden difference in intensities and making the edge more discernible. The above mentioned masks follow the principle of derivate mask.

Masks have opposite sign in them and sum of both masks equals to zero. The gradient magnitude at every point in the image can be combined with the resulting gradient approximations, using:

$$|G| = \sqrt{{G_x}^2 + {G_y}^2} \quad (4)$$

iv. Laplacian of Gaussian operator

The second spatial derivative of an image is calculated by the LoG operator. This means that in areas where the intensity gradient is zero i.e., image has a constant intensity, the LoG response will be zero. In the zone of a change in the intensity, however, on the darker side the LoG response will be positive and on the lighter side it will be negative. This implies that at a respectably sharp edge between two regions of uniform but different intensities, the LoG response will be zero(at a long distance from the edge), positive just to one side of the edge, negative just to the other side of the edge, at some point in between it will be 0, on the edge itself [7]. This can be visualised with the help of figure2 which shows the second derivation of the image.

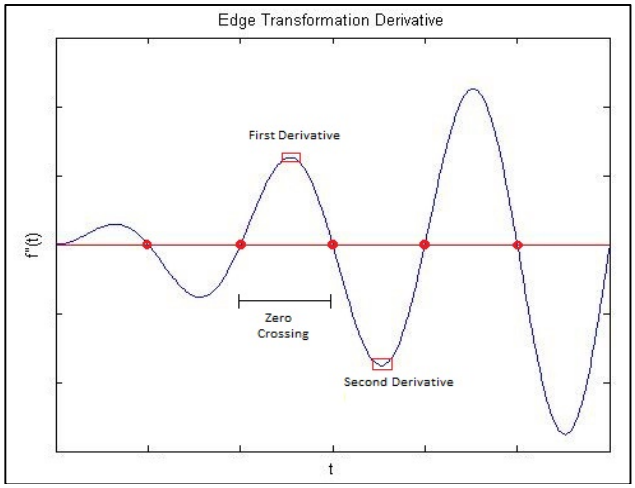


Fig 1: Edge transformative derivation

The laplacian of Gaussian uses the following Gaussian function for smoothening:

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5)$$

The degree of blurring is determined by the value of σ. The laplacian of this function is given by:

$$\nabla^2 G(x,y) = \frac{\partial^2 G(x,y)}{\partial x^2} + \frac{\partial^2 G(x,y)}{\partial y^2} \quad (6)$$
$$\left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Hence, it is named as laplacian of Gaussian. The image is convolved with $\nabla^2 G(x,y)$because it has 2 effects: it smoothes the image, thus reducing noise and the laplacian is also computed by it which produces a double edge image. Locating edges then consists of finding zero crossings.

v. Canny operator

The Process of Canny edge detection algorithm can be divided in 5 different steps. In the first step, Gaussian filter, with a specified standard deviation is applied to smooth the image in order to reduce the noise. In the second step, we find the intensity gradients and edge direction at each point of the image using any of the 3 mentioned search based algorithms. In the third step, non-maximum suppression is applied to get rid of spurious response to edge detection. In the fourth step, double threshold is applied to determine potential edges. Edges are tracked by hysteresis thresholding, which is based on using 2 thresholds, T1and T_2, with $T_1<T_2$. Ride pixels with value > T_2 are known as "strong" edge pixels. Ridge pixels with value between T_1 andT_2 are said to be "weak" edge pixels. In the end, the weak pixels are incorporated to the strong pixels by the algorithm hence performing edge linking [6].

3. Methodology

The performance of edge detections algorithms are evaluated by detection of true edges, processing time, error ratio, and noise level etc. In this paper, comparison between five famous edge detectors is accomplished with the Mean Squared Error and Peak Signal to Noise Ratio.

Edge detection of test images is performed by applying all selected algorithms using MATLAB software and then calculating PSNR and MSE between each resultant edge detected image and ground truth image. Also, the execution time of each algorithm is also measured [8].

A. Mean Squared Error

The average of the squares of the "errors" is measured by the mean squared error of an estimator, i.e. the difference between the estimator and what is estimated. The contrast occurs because of randomness or because the estimator does not account for information that could produce a more accurate estimate. The MSE varies inversely with the PSNR. The M.S.E can be calculated from the following equation:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \quad (7)$$

Where I (i, j) and K (i, j) are the edge detected image and ground truth image respectively and M, N are the dimensions of the image.

B. Peak signal to noise ratio

PSNR is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Here, PSNR cites the ratio between the edge detected images i.e. the estimator output and the ground truth image which is also said to be the estimated image. PSNR can be evaluated by using the following equation:

$$PSNR = 20.\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \quad (8)$$

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

4. Experimentation

Five different edge detection algorithms, i.e. Robert, Prewitt, Sobel, Canny and Laplacian of Gaussian have been compared in this experiment. The comparison is done by computing MSE and PSNR of each image with respective to their corresponding ground truth images. The images taken for analysis are face, butterfly, child and church. For a better understanding, the time taken by each algorithm is also computed.

This experiment has been conducted using MATLAB which is a high level language that allows matrix manipulations, plotting of functions and data, implementation of algorithms. The table I shows the test image and the output images generated by each algorithm. Table II shows the computed MSE, PSNR and Time Elapsed for execution of the five algorithms on the image Butterfly. Similarly, table III, IV, V, VI and VII shows results for Child, face and Church respectively.

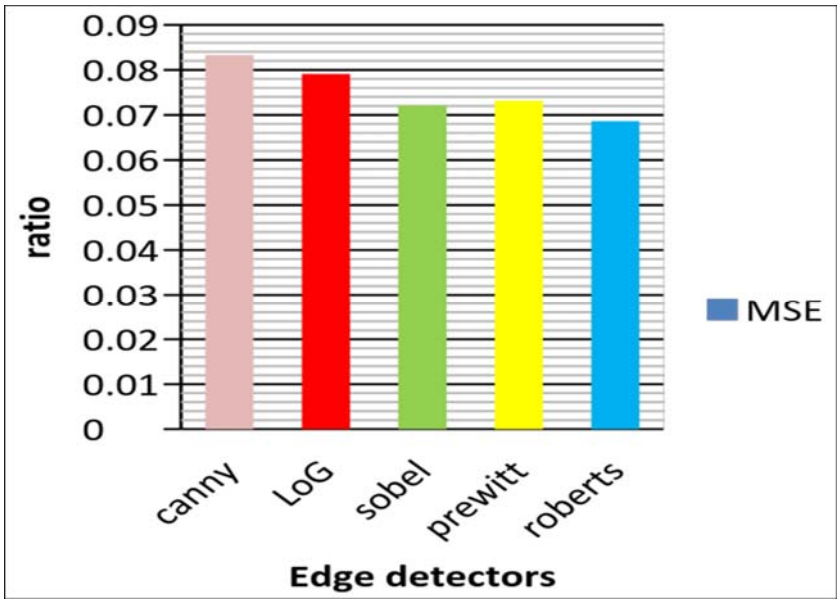


Fig 2: Average MSE of resultant images produced by edge detectors on 4 test images.

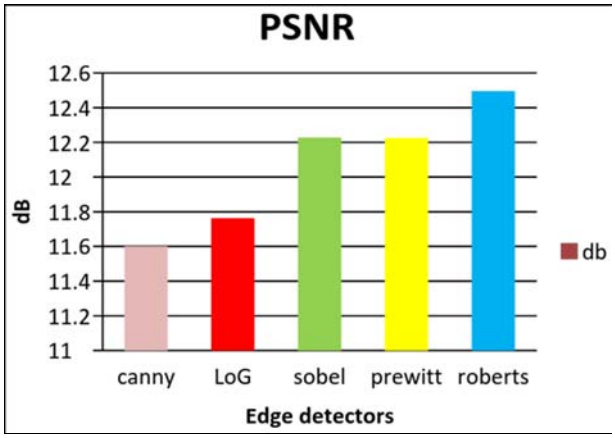


Fig 3: Average PSNR of the resultant images produced by edge detectors on 4 test images

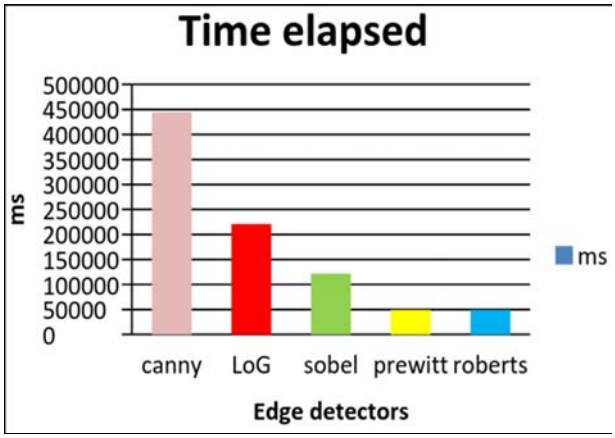
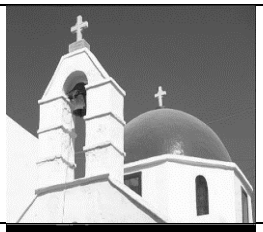
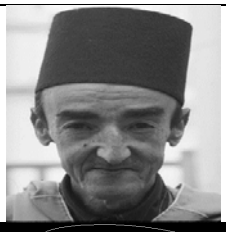
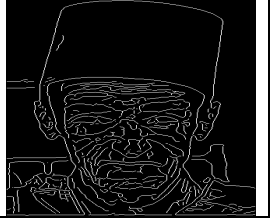


Fig 4: Average time taken by each edge detector to detect the edges.

Table 1: original gray scale images and the results produced by each edge detection operator

	BUTTERFLY	CHILD	CHURCH	FACE
ORIGINAL IMAGE				
CANNY				

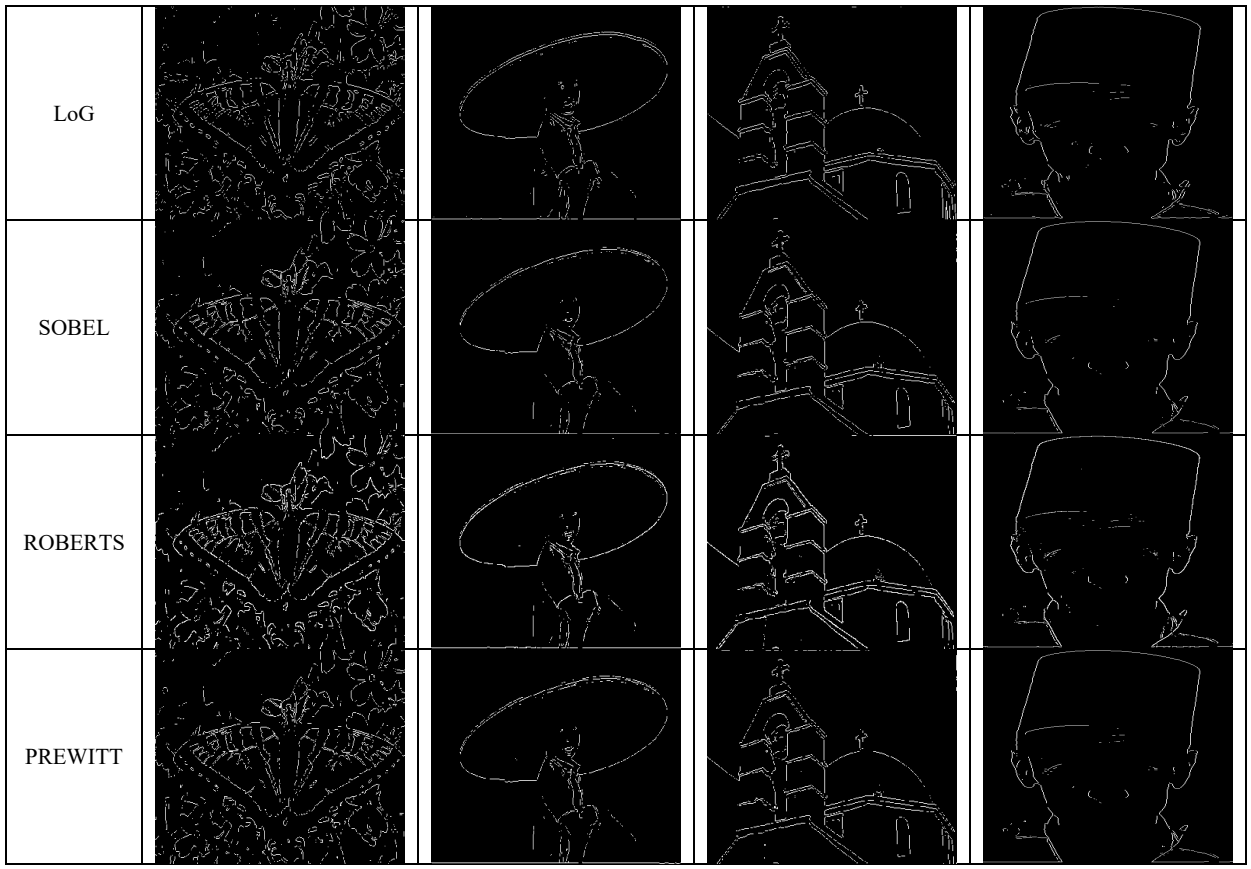


Table 2: Results produced by the operators on the image of a face

FACE	LOG	Canny	Sobel	Prewitt	Roberts
MSE	0.0606	0.0866	0.0568	0.0566	0.0538
PSNR(dB)	12.3065	10.6584	12.4594	12.4743	12.6904
Time (s) elapsed	0.2509	0.3957	0.1067	0.0499	0.0567

Table 3: Results produced by the operators on the image of a church

CHURCH	LOG	Canny	Sobel	Prewitt	Roberts
MSE	0.0825	0.0888	0.0743	0.0743	0.0705
PSNR(dB)	10.8354	10.5161	11.2911	11.2877	11.5170
Time (s) elapsed	0.2404	0.4831	0.1224	0.0464	0.0468

Table 4: Results produced by the operators on the image of a child

CHILD	LOG	Canny	Sobel	Prewitt	Roberts
MSE	0.0858	0.0962	0.0815	0.0813	0.0789
PSNR(dB)	10.6649	10.1676	10.8861	10.9004	11.0307
Time (s) elapsed	0.2248	0.4586	0.1158	0.0398	0.0481

Table 5: Results produced by the operators on the image of a butterfly

Butterfly	LOG	Canny	Sobel	Prewitt	Roberts
MSE	0.1544	0.1386	0.1427	0.1445	0.1394
PSNR(dB)	8.1145	8.5825	8.4544	8.4021	8.5587
Time (s) elapsed	0.1622	0.5091	0.1529	0.0611	0.0614

5. Results and discussion

The tables II to VII shows the numerical parameters as a measure of efficiency of each edge detection operator, i.e. MSE and PSNR. The Mean Squared Error of different algorithms on various test images is found to be varying in a range from 0.0130 to 0.1600 and the PSNR (dB) ranges from 8.1100 to 18.6100 in this experiment.

It can be observed from the numerical results that the Roberts operator shows the highest average PSNR which is approximately12.496 dB. On the other hand, canny operator shows the least average PSNR among the five tested algorithms, approximately 11.601 dB. It is important to note that an edge detector operator with the least PSNR have the highest edge detection capabilities and Canny appears to have the least maximum average MSE and least average PSNR. It can be noticed that in less complex images like Eagle, the canny algorithm shows high PSNR. This is because canny operator is capable of detecting weak edges and thus, it has a higher chance of detecting false edges.

When talking about time complexity, looking at average processing time, Prewitt operator gives much better performance than others. On the other hand, canny operator takes consumes maximum time among other four operators.

Fig 2&3 shows a bar graph for MSE and PSNR respectively for the five algorithms under test.

6. Conclusion

Five prominent edge detectors, i.e. Laplacian of Gaussian, Canny, Sobel, Prewitt, and Roberts have been discussed in this paper. The performance on different test images have been evaluated based on Mean Squared Error, Peak Signal to Noise Ratio, and Processing Time needed for each image to detect edges. Six test images are used for experiment. MATLAB is used for implementation of edge detection algorithms and computation of PSNR and MSE. The experiments lead to the conclusion that canny edge detection

algorithm provides better performance than others. Further, LoG, Prewitt, Sobel and Robert's are ranked so based on their relative performance. It should also be noticed that canny takes comparatively more time for processing while prewitt requires least amount of time to compute edges of an image.

7. References

1. Debosmit Ray. Edge Detection in Digital Image Processing. 2013.
2. Ramesh Jain, Rangachar Kasturi, Brian G Schunck. Machine Vision, Published by McGraw-Hill, Inc., ISBN 0-07-032018-7, 1995.
3. Lindeberg, Tony. Edge detection", in Hazewinkel, Michiel, Encyclopedia of Mathematics, Springer, ISBN 978-1-55608-010-4. 2001.
4. Ziou D, Tabbone S. Edge detection techniques: An overview, International Journal of Pattern Recognition and Image Analysis, 1998; 8(4):537-559.
5. Prewitt JMS. Object Enhancement and Extraction in "Picture processing and Psychopictorics, Academic Press, 1970.
6. Moeslund T. Canny Edge Detection. Retrieved. 2014, 2009.
7. Horn B. *Robot Vision*, MIT Press, 1986, 8.
8. Poobathy D, Dr. Manicka Chezian R. Edge Detection Operators: Peak Signal to Noise Ratio Based Comparison, International Journal of Image, Graphics and Signal Processing. 2014; 10:55-61.
9. Lehmann EL, Casella George. Theory of Point Estimation (2nd ed.). New York: Springer. ISBN0-387-98502-6.MR 1639875. 1998.