International Journal of Computer Science and Information Technology Research (IJCSITR)
Vol. 1, Issue 1, pp: (16-24), Month: October-December 2013, Available at: www.researchpublish.com

Comparison of Edge Detectors

Ayaz Akram, Asad Ismail
Department of Electrical Engineering
University of Engineering and Technology
Lahore, Pakistan

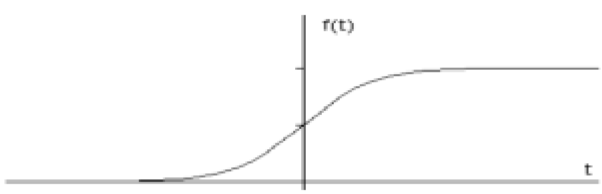
Abstract: **Edge detection is one of the most significant tasks in image processing systems. Edge map of an image contains vital information about objects present in an image and is used to recognize certain objects in an image. Information contained in edge map will only be useful if edge map contains accurate edges. Process of edge detection is an extremely difficult task. For the last few decades a lot of research has been done in this field. This paper tries to provide a comparison of different edge detection schemes that fall in three main categories of edge detectors: Gradient based edge detectors, Laplacian based edge detectors and Non-derivative based edge detectors. Pratts figure of merit is used to compare quantitatively results of edge maps for a synthetic image at various levels of noise. Results of real life image are analyzed qualitatively. Non-derivative based edge detector SUSAN gives the best results even in presence of noise.**

I. INTRODUCTION

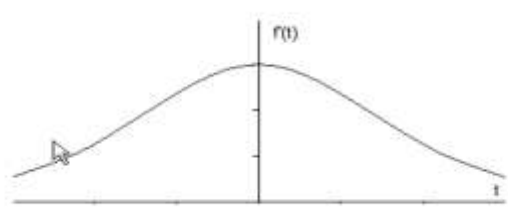
Edge can be defined as a sharp discontinuity or geometrical change in an image. The edges carry significant information regarding the objects present in an image. Edge detection, the process of determining edge pixels within an image, is a task of huge importance in feature-based image processing. Accurately detected edges separate objects from the background and help in calculation of different features of objects like area, perimeter and shape. There is a large number of image processing and computer vision applications that rely on correctly detected edges within an image. For example, military applications involving tasks such as object recognition and motion analysis, security applications including data coding, data hiding, and watermarking also benefit from improved edge detection capabilities. There has been a lot of research in this field for the last few decades. The performance measure for the edge detection is how well edge detector markings match with the visual perception of object boundaries [6]. The detection process is carried out by the examination of local intensity changes at each pixel element of an image. This paper is further organized as follows. Section 2 describes different methodologies for edge detection. Section 3 describes working of some edge detection algorithms. Section 4 deals with quantitative comparison of those algorithms. In section 5 a comparison is made between results of those algorithms after applying them to real life images. Section 6 provides with conclusions.

II. METHODOLOGIES

There are many ways to perform edge detection. However, the majority of different methods may be grouped into three categories:
Gradient Based Edge Detection: In this category of edge detectors derivative of image is taken and edges are detected by looking for maximum and minimum in that derivative.



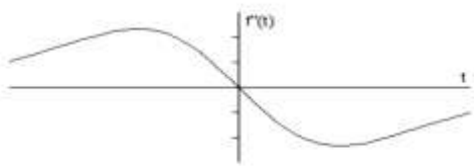
(a) figure 1



(b) figure2

Consider a one dimensional ramp edge as shown in figure. 1.Taking its gradient with respect to t gives signal as shown in figure2.Clearly, the derivative shows a maximum located at the center of the edge in the original signal.This method of locating an edge is characteristic of the gradient filter family of edge detection filters A pixel location is declared an edge location if the value of the gradient exceeds some threshold. As edges have higher pixel intensity values as compared to neighboring pixels. So once a threshold is set, gradient value can be compared to the threshold value and an edge is detected whenever the threshold is exceeded [3].

Laplacian Based Edge Detection: The Laplacian method searches for zero crossings in the second derivative of the image to find edges. Furthermore, when the first derivative is at a maximum, the second derivative is zero. As a result, another alternative to finding the location of an edge is to locate the zeros in the second derivative. This method is known as the Laplacian and the second derivative of the signal of figure1 is shown in figure below.



(c) figure 3

Non-derivative Based Edge Detection: This category of edge detectors do not require image derivatives at all.

There are many problems associated with edge detection such as false edge detection, missing true edges, edge localization, high computational time and problems due to noise. Past research and experience with numerous edge detectors indicates that the problem of locating edges in real images is extremely difficult. The performance of an edge detector depends on how well localized its response to real and synthetic images is. All real life images contain noise. Usually, to minimize the effect of this noise low pass filtering (using Gaussian kernels) is performed prior to edge detection. But, this smoothing also reduces the effect of sharp discontinuities due to edges [7]. Smoothing performed by filter can be controlled by varying parameters of filter. Increasing strength of filter too much would result in effective removal of noise but detected edges will have large localization errors and many edges would not be picked. On the other hand decreasing strength of filter would result in ineffective removal of noise but fine details would be preserved [1].Keeping in view the problems of Gaussian kernels and gradient based edge detectors SUSAN Edge detector [12] was presented in 1995 and the fact that it uses no image derivatives makes its performance good in presence of noise. Marr and Hildreth [9] in 1980 argued that an edge detecting operator should be a scalable differential operator, which can compute the first or second derivatives at different scales. They achieved these goals using a Laplacian of Gaussian (LoG) operator,
which was as:

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

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

figure 4

The magnitude and directions of the gradient can be given

$$|G| = sqrt(Gx^2 + Gy^2) \qquad (1)$$

$$a(x, y) = arctan(Gy/Gx) \qquad (2)$$

In above equations Gx and Gy are the two images of the further approximated by the Difference of Gaussian (DoG). Zero-crossings are needed to detect edges in images which are filtered using these filters. This operator opened new horizons in the field of edge detection.

Zero-crossings from derivatives of the Gaussian are only reliable if edges are well separated and the signal-to-noise ratio in the image is high. A problem with the Gaussian differential algorithm is that it produces false edges i.e. those which do not result from major intensity changes in the image. [5] contains a detailed analysis of such phantom edges.

Canny [4] presented edge detection as an optimization problem with constraints. His optimization objectives were high signal to noise ratio, well localized edge points, and single edge response. He formulated a mathematical expression for these objectives and then showed that a successful use of the first derivative of a Gaussian approximation achieved optimal results. However, Cannys algorithm is more sensitive to weak edges, making it declare fake and unstable boundaries as edges, resulting in a corrupted edge map [2].

In short, most Gaussian based edge detectors have problems like false contours, localization errors and missing infor- mation. Much work has been done to overcome the issues related to these detectors but most of the techniques are computationally expensive. This paper is further organized as follows. Section 2 describes working of some edge detection algorithms. Section 3 deals with quantitative comparison of those algorithms. In section 4 a comparison is made between results of those algorithms after applying them to real life images. Section 5 provides with conclusions.

III. EDGE DETECTION OPERATORS

In this section a brief description of some famous edge detection algorithms is provided. Comparison of these detectors will be presented in next sections.
Prewitt: The Prewitt operator [11] is a discrete differentiation operator used to compute the gradient of image intensity function. The Prewitt masks are simple to implement but are very sensitive to noise [8]. The operator uses two 3x3 size masks which gives more information regarding the direction of the edges as they consider the nature of data on the opposite sides of the center point of the mask. These masks are then convolved with the original image to obtain the approximations of derivatives for the horizontal and vertical edge changes, separately. The mask used to calculate the gradients are shown in figure 4.Same size as the original image and these show horizontal and vertical gradient at each point.

Sobel: The Sobel operator is a discrete differentiation operator which computes the gradient for the intensity changes at each point in an image just like Prewitt operator. This operator is better for noise suppression as compared to Prewitt operator [7]. Masks used are shown in figure 5. Magnitude and direction of gradient are calculated using equation (1) and (2).

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

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

Figure 5

LoG: This operator belongs to Laplacian based edge detectors class. Laplacian operator highlights the regions of rapid intensity changes in an image. As the Laplacian of an image detects the noise along with the edges in an image, the image is smoothened first by convolving by a 2-D Gaussian kernel of standard deviation σ.

$$G(x, y) = e^{(x^2+y^2)/2\sigma^2} \qquad (3)$$

The expression for LOG is given as

$$\nabla^2 G(x,y) = [\frac{x^2+y^2-2\sigma^2}{\sigma^4}]e^{-(x^2+y^2)/2\sigma^2} \qquad (4)$$

LoG is then convolved with input image I(x,y) giving resultant edge map.

$$g(x,y) = I(x,y) * \nabla^2 G(x,y) \qquad (5)$$

A 5*5 mask used for this operator is shown in figure 6

0	0	1	0	0
0	1	2	1	0
1	2	-16	2	1
0	1	2	1	0
0	0	1	0	0

Figure 6

The kernels of any size can be approximated by using the above expression for LoG. The edge detection in an image using LoG operator can thus be obtained by the following steps:
1.Apply Log to the input image.
2.Detect the zero-crossings of the image.
3.Apply threshold to minimize the weak zero-crossings caused due to noise.

Canny: The Canny edge detection algorithm constitutes the following basic steps [7]
1. Noise is filtered and image is smoothed using Gaussian filter.
2. Edge strength is found by computing the gradient magnitude and angle of gradient vector for edge direction.
3. Non-maxima suppression is applied to the gradient magni- tude to trace move along the edge direction and suppress those pixel values that are not considered edge and thus resulting in thinning of edge.
4. Final step is to use hysteresis and connectivity analysis to detect and connect edges.
If threshold value for edge detection is kept too low or too high there can be problem of either false positive or false negative edges. Canny algorithm solves this problem by using two thresholds: A low threshold and a high threshold.

Susan: SUSAN Edge detector [12] was presented in 1995 and the fact that it uses no image derivatives makes its performance good in presence of noise. SUSAN stands for smallest Uni value Segment Assimilating Nucleus. The idea behind this detector is to use a pixel's similarity to its neighbors gray values as the classification criteria (a non linear filter). Figure 7 shows that the area of the USAN carries information about the image structure around a given point. The area of the USAN is at a maximum in a flat region, becomes half when USAN is near a straight edge and becomes further low when mask is used near a corner. Circular masks placed at different locations of an image containing a rectangle can be seen in figure. USAN is marked in dark color for each circular mask.

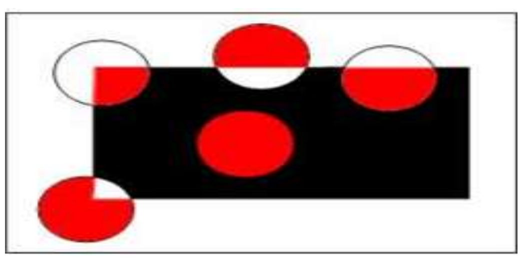


Figure 7

The steps of the edge detection are as follows:
Circular mask is placed at each pixel and weight of the circular mask is calculated. The weight of the USAN is

$$n(r_0) = \sum_r compare(r, r_0) \qquad (6)$$

Where compare(r; r0) is defined as:

$$compare(r, r_0) = \begin{cases} 1 \; if \; |I(r) - I(r_0)| \le t \\ 0 \; if \; |I(r) - I(r_0)| > t \end{cases} \qquad (7)$$

Here 't' is a threshold defining pixel gray level similarity. Edge strength at each pixel is calculated using the formula:

$$response(r_0) = \begin{Bmatrix} g - n(r_0) \ if \ n(r_0) < g \\ 0 \ otherwise \end{Bmatrix} \qquad (8)$$

Here g is a geometric threshold which is set to 3/4. After computing the edge response image non maxima suppression is performed for which direction perpendicular to edge is required. The direction depends on the edge type which is being examined either inter-pixel (edge is between pixels) or intra-pixel (pixel itself is part of the edge).For inter pixel case, if the USAN area is greater than the mask diameter and the center of gravity of the USAN lies more than one pixel from the nucleus. The center of gravity of the USAN is defined as:

$$CG(r_0) = \sum_r r \ compare(r, r_0) / \sum_r compare(r, r_0) \qquad (9)$$

Required direction is given by $r_0 - CG(r_0)$. For intra pixel case, if the USAN area is smaller than the mask diameter or the USAN center of gravity lies less than one pixel from the nucleus. Compute the second order moments of the USAN about the nucleus $r_0 = (x_0, y_0)$:

$$\overline{(x - x_0)^2} = \sum_r (x - x_0)^2 \ compare(r, r_0) \qquad (10)$$

$$\overline{(y - y_0)^2} = \sum_r (y - y_0)^2 \ compare(r, r_0) \qquad (11)$$

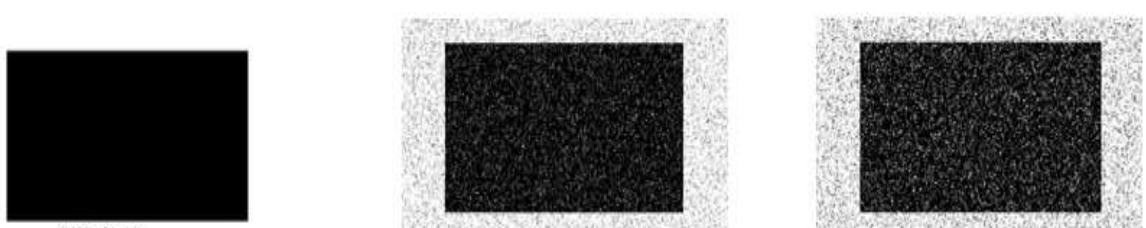
Edge orientation is given by ratio of equation 10 and equation 11.

IV. QUANTITATIVE COMPARISON

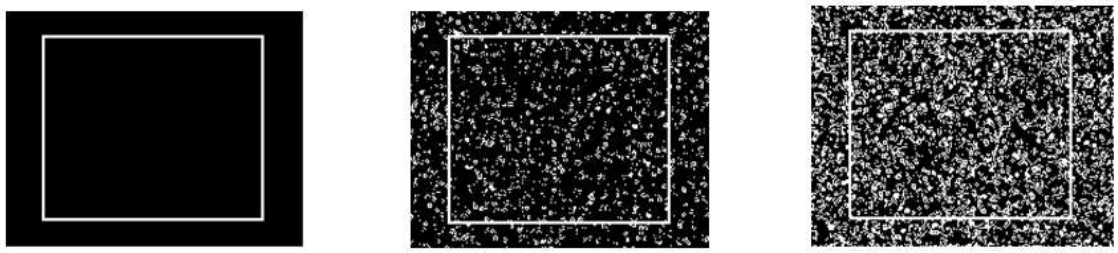
In this section we have tried to compare edge detectors described in the previous section. There are three common errors associated with edge detectors: (1)missing valid edge points,(2) failure to localize edge points and (3) classification of noise fluctuations as edge points. Pratt has introduced a figure of merit that balances these three types of error [10]. Pratt's Figure of Merit is chosen to quantify the results of edge detectors. This quantitative measure is determined as follows.

$$F = \frac{1}{\max\{N_I, N_A\}} \sum_{k=1}^{N_A} \frac{1}{1 + \alpha d^2(k)}$$

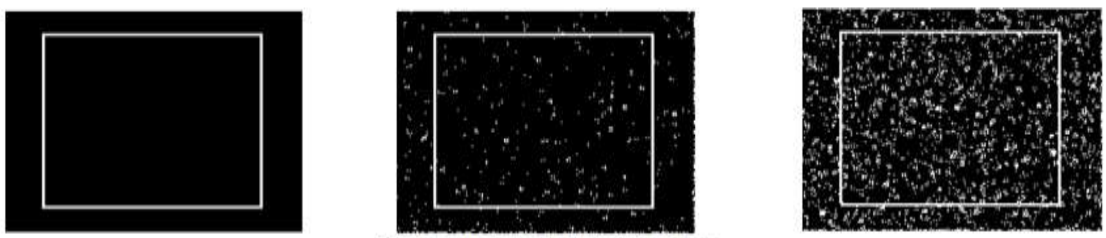
where, NI is number of actual edge pixels, NA is the number of detected edge pixels, and d(k) is the distance from the kth actual edge to the corresponding detected edge. α is a scaling constant, which is set to 1/9 as is often done in the literature. We have taken a synthetic image (box shape) as an input, and find out its edge map at different levels of independent Gaussian noise. Threshold parameters of every edge detector are chosen to maximize Pratt's FOM. Outputs of all detectors are shown in figure 8 and resulting values of Pratts FOM are given in table 1.



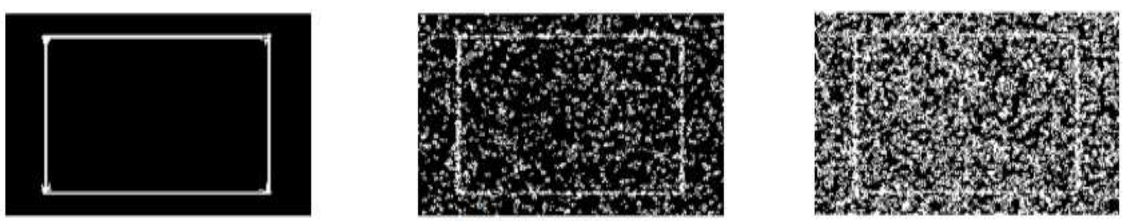
(a)No Noise (b)Gaussian noise of 0.05 var (c)Gaussian noise of 0.1 var



(d) Edge maps obtained with Prewitt Edge detector

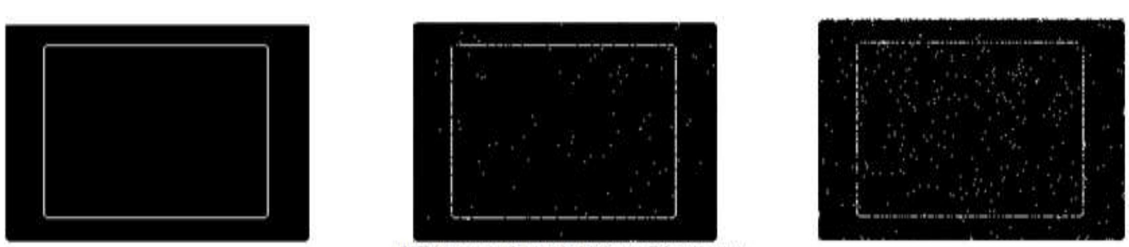


(e) Edge maps obtained with Sobel Edge detector



(f) Edge maps obtained with LoG

(g) Edge maps obtained with Canny's algorithm



(h)Edge maps obtained with Susan Edge detector

Figure 8. Edge maps of different edge detectors at different levels of noise.

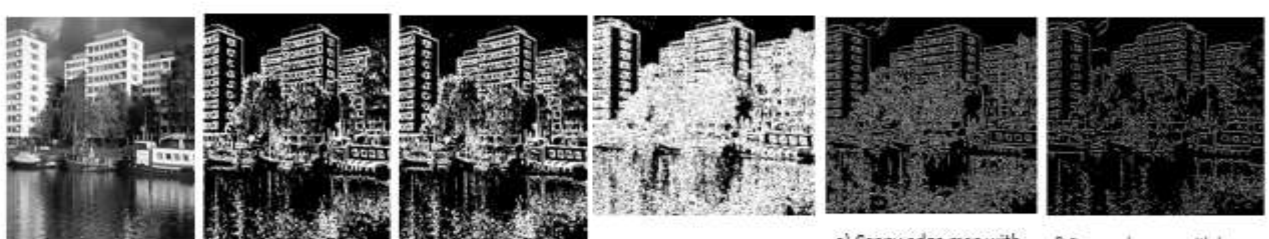
PRATT'S FOM FOR VARIOUS EDGE DETECTORS			
Operator	Image without noise	Image with Gaussian noise of 0.05 variance	Image with Gaussian noise of 0.1 variance
Prewitt	0.8743	0.6757	0.3708
Sobel	0.8743	0.3601	0.2528
LoG	0.9095	0.378	0.254
Canny	0.8740	0.8580	0.343
Susan	0.9611	0.86	0.6378

Table 1

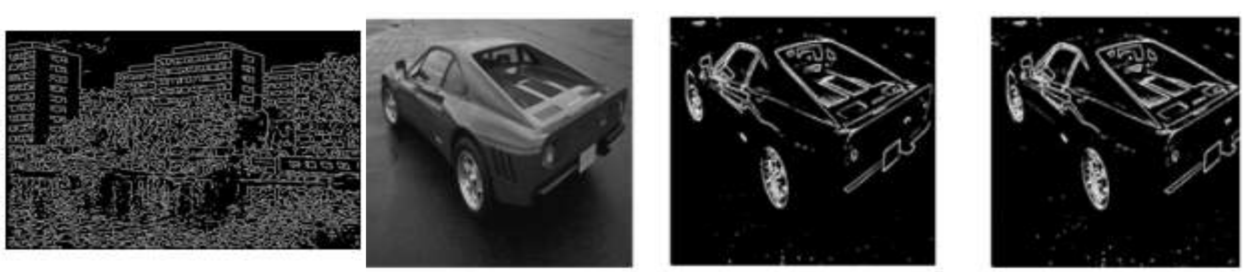
Also, it is observed from figure 8 that the visual appearance of the output isn't always as good as the numerical. This is due to the limitations of the figure of merit measure (for which the output edge maps were optimized).

V. COMPARISON FOR REAL IMAGES

As explained in previous section quantitative comparison of detected edge maps require ground truth images. However, manually constructing ground truth for real intensity images is problematic. Even the definition of an intensity edge is debatable. The difficulties involved in obtaining ground truth for real images are so great that, researchers simply do not conduct quantitative evaluations of edge detectors using real images. In this section we have applied edge detection algorithms to three real life images and tried to analyze algorithms qualitatively. Images are taken from Berkeley Segmentation Data set [10]. It has been tried to make sure that images contain necessary features to test abilities of edge detection techniques. Images taken contain areas of fine detail as well as areas of consistent colors. Three images and their results can be seen in figure 9.Results of Sobel and Prewitt are much similar but their edge maps miss many edges which can be observed in results. LoG produces edges that are much thicker. Canny with low Gaussian smoothing give many wrong edges but miss many if Gaussian smoothing is increased therefore a tradeoff between the two is required to produce better results.Susan give much better results which are obvious from figures. Note that parameters of all detectors are selected to give best possible results.



(a) First image (b) Prewitt edge map (c) Sobel edge map d) LoG edge map e) Canny edge map with high Gaussian smoothing f) Canny edge map with low value of Gaussian smoothing



(g) Susan edge map (h) Second image (i) Prewitt edge map (j) Sobel edge map

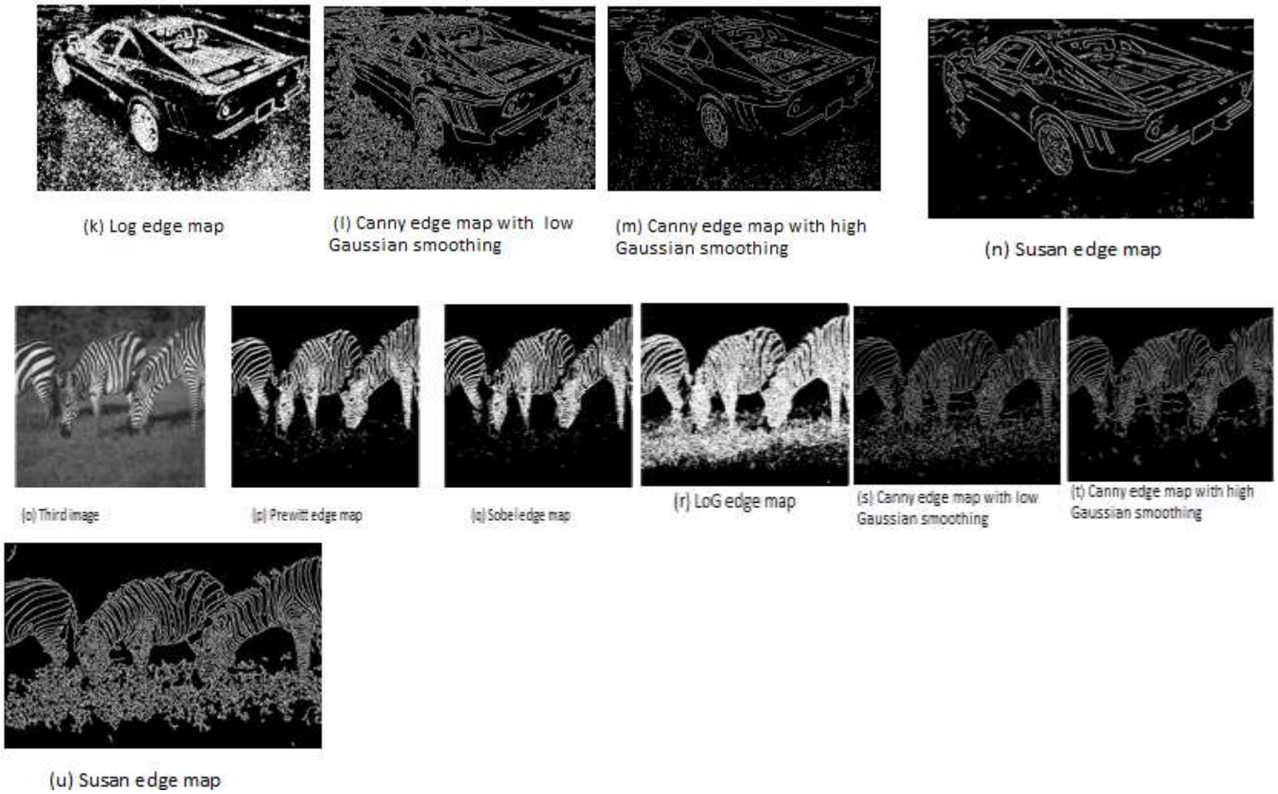


Figure 9. Edge maps for real images

VI. CONCLUSIONS

Edge detection is a key tool for image segmentation used for object detection and many other applications. Therefore, it is necessary to use a robust edge detector which gives the best results at all conditions. In this paper we have tried to explain the differences between some famous edge detection algorithms and evaluate them on the basis of their results to different images. Gradient based edge detectors like Prewitt and Sobel are relatively simple and easy to implement, but are very sensitive to noise. LoG tests wider area around the pixel and find the edges correctly, but malfunctions at corners and curves. It also does not find edge orientation because of using Laplacian filter. Cannys algorithm is an optimal solution to problem of edge detection which gives better detection specially in presence of noise, but it is time consuming and require a lot of parameter setting. SUSAN edge detector uses no image derivatives which explains why the performance in the presence of noise is good. The integrating effect of the principle, together with its non-linear response, give strong noise rejection. This can be understood simply if an input sig nal with identically independently distributed Gaussian noise is considered. As long as the noise is small enough for the USAN function to contain each “similar” value, the noise is ignored. The integration of individual values in the calculation of areas further reduces the effect of noise. Another strength of the SUSAN edge detector is that the use of controlling parameters is much simpler and less arbitrary (and therefore easier to automate) than with most other edge detection algorithms [12]. Numerical analysis of these algorithms is done for synthetic image (with known edges) at various noise levels using Pratts figure of merit. For natural image results are analyzed visually.

REFERENCES

[1] Volker Aurich and Jörg Weule. Non-linear gaussian filters performing edge preserving diffusion. In Mustererkennung 1995, pages 538–545. Springer, 1995.

[2] Mitra Basu. Gaussian-based edge-detection methods-a survey. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Trans- actions on, 32(3):252–260, 2002.

[3] F. Bergholm. Edge focusing. Proc. 8th Int. Conf. Pattern Recogni- tion,Paris,France, 3(1):597–600, 1986.

[4] John Canny. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, (6):679–698, 1986.

[5] James J. Clark. Authenticating edges produced by zero-crossing algo- rithms. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 11(1):43–57, 1989.

[6] Werner Frei and Chung-Ching Chen. Fast boundary detection: A generalization and a new algorithm. Computers, IEEE Transactions on, 100(10):988–998, 1977.

[7] Rafael C Gonzalez and RE Woods. Digital image processing (interna- tional ed.), 2008.

[8] Raman Maini and Himanshu Aggarwal. Study and comparison of various image edge detection techniques. International Journal of Image Processing (IJIP), 3(1):1–11, 2009.

[9] David Marr, Tomaso Poggio, Ellen C Hildreth, and W Eric L Grimson. A computational theory of human stereo vision. Springer, 1991.

[10] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, volume 2, pages 416–423. IEEE, 2001.

[11] Judith MS Prewitt. Object enhancement and extraction, volume 75. Academic Press, New York, 1970.

[12] Stephen M Smith and J Michael Brady. Susana new approach to low level image processing. International journal of computer vision, 23(1):45–78, 1997.