Variable Edge Detection based on Improved Canny Filter

Shashi Mishra, Prof. Bharti Chourasia
Department of Electronics and Communication Engineering,
Scope College of Engineering, Bhopal MP India
Shashi_mishra01@rediffmail.com, bhartichourasia@gmail.com

Abstract- **Edge detection is main significant part of any image. By accurate edges we are able to predict the property of any image. For the detection of the object, it is important to find exact boundary of the image. So it's very important to have a well knowledge of edge detection. For this purpose different edge detection methods are available but canny edge detection is more appropriate than others. In canny edge detection, different kind of operator and filter are used. This operator improves the detection level of edge. OTSU thresholding is used for the automatic dual level of threshold. This makes canny edge detection method more usable from astronomy to photography, medicine to war. In this paper we are using canny edge detection for finding the edges, and thresholding is based on OTSU multi-level.**

Keywords- **Edge Detection, Image processing, OTSU, Canny edge detector.**

I. INTRODUCTION

The Edge detection is a set of mathematical methods which aim identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities [3]. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. Means of edges detection extraction of image features, this is a fundamental tools of image processing, machine vision and computer vision. Localization of exact point from real boundary is main part of detection. Canny edge detection gives the best result. OTSU method is used which is for the dual level thresholding. We improved this work by multi-level threshold.

II.THE PREVIOUS METHOD

i) Gaussian Filter to Reduce Noise

A Gaussian filter (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. The equation of a Gaussian function in one dimension is-

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

In two dimensions, it is the product of two such Gaussians, one in each dimension:

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

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. When applied in two dimensions, this formula produces a surface whose contours are concentric circles with a Gaussian distribution from the center point. Values from this distribution are used to build a convolution matrix which is applied to the original image. Each pixel's new value is set to a weighted average of that pixel's neighborhood. The original pixel's value receives the heaviest weight (having the highest Gaussian value) and neighboring pixels receive smaller weights as their distance to the original pixel increases. This results in a blur that preserves boundaries and edges better than other, more uniform blurring filters;

ii) Get High And Low with OTSU Algorithm

In computer vision and image processing OTSU's method is used to automatically perform clustering based image thresholding or the reduction of a gray level image to a binary image the algorithm assumes that the image to be threshold contains two classes of pixels or bi-model histogram (foreground and background) then calculates the optimum threshold separating those two classes so that their combined spread (intra class variance) is minimal. The extension of the original method or multi-level thresholding is referred to as the multi OTSU method.

Method:-

In OTSU method we exhaustively search for the threshold that minimizes the intra class variance (the variance with in the class), defined as a weighted sum of variances of the two classes.

$$\sigma_w^2(t) = w_1(t)\,\sigma_1^2(t) + w_2(t)\sigma_2^2(t)$$

Weights w_i are the probabilities of two classes separated by a threshold t and $\sigma_i^2(t)$ variances of these classes.

$$\mu 1(t) = [\textstyle\sum_0^t p(i)x(i)]/w_1$$

where x(i) is the value at the center of the ith histogram bin. Similarly, we computed $w_2(t)$ and μ_2 on the right-hand side of the histogram for bins greater than t.

Algorithm

1. Compute histogram and probability of each intensity level,
2. Set up initial $w_i(0)$ and $\mu_i(0)$
3. Step through all possible thresholds t= 1 . . maximum intensity
4. Update w_i and μ_i
5. Compute $\sigma_b^2(t)$
6. Desired threshold corresponds to maximum $\sigma_b^2(t)$
7. We computed two maxima. $\sigma_{b1}^2(t)$ is the greater max and $\sigma_{b2}^2(t)$ is the greater or equal maximum.
8. Desired threshold = ($threshold_1$+$threshold_2$)/2

Flow chart of this work is shown below -

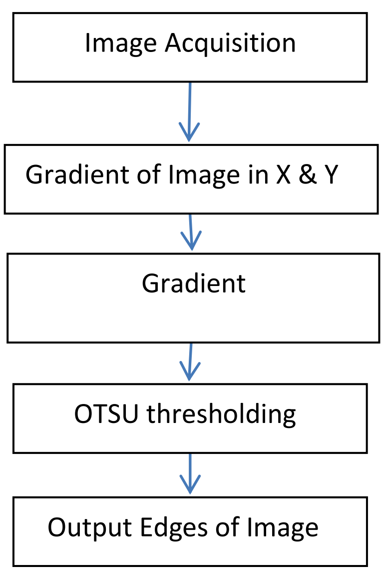

Fig.1 Previous Method with simple OTSU's thresholding Method

III. PROPOSED METHOD

i) OTSU: - OTSU is a linearization algorithm named after its inventor Nobuyuki OTSU. OTSU's method selects the threshold by minimizing the within-class variance of the two groups of pixels or in other words optimal threshold is selected by maximizing the between class variance. The basic idea is to find the threshold value where the sum of foreground and background spreads is at its minimum [2]. An image is a 2 D gray scale intensity function, and contains N pixels with gray levels from 1 to L. The number of pixels with gray level I is denoted fi, giving a probability of gray level I in an image of

$$P_i = fi/N$$

Let M*N is an image histogram with L intensity level, i.e. [0…..L-1]. The number of pixels with gray level i is denoted $n_{i,}$ such that

$$MN = \sum ni$$

Lower limit is I = 0, higher limit L-1 so probability of gray level I in an image is:

$$Pi = ni/MN$$

$$\textstyle\sum_{i=0}^{L-1} Pi = 1$$

$$Pi \geq 0$$

Using a threshold say k pixels of image are partitioned into two classes say C1 and C2 (e.g., object and background) such that 0<k<L-1, as threshold, T=k . the gray level probability distributions for two classes C1 (pixels in [0,k]) and C2 (pixels in [k+1, L-1]) are:

$$P1 = P\,(C1) = \textstyle\sum_{i=0}^{k} Pi$$

Probability of class C1

$$P2 = P(C2) = (x+a)^n = \textstyle\sum_{i=k+1}^{L-1} pi = 1 - P1$$

Probability of class C2

Means m1 and m2 for classes c1 and C2 respectively are given by following equations:

$$m1 = \sum_{i=0}^{k} i.\,P(i/C1)$$

$$= 1/P1 \textstyle\sum_{i=0}^{k} i.\,Pi$$

Where P(C1/i)=1, P(i) = Pi e P(C1) = P1

$$m2 == 1/P2 \textstyle\sum_{i=k+1}^{L-1} i.\,pi$$

Mean global intensity, m_G :

$$m_G = \textstyle\sum_{i=0}^{L-1} i.\,Pi$$

While the mean imtensity up to the k level, m:

$$m = \sum_{i=0}^{k} i.\,Pi$$

hence

$$P_1 m_1 + P_2 m_2 = m_G$$

$$P_1 + P_2 = 1$$

The global variance, σ^2 is given by :

$$\sigma^2 = \textstyle\sum_{i=0}^{L-1} (i - mg)^2 .\,Pi$$

Hence between class variance of the threshold image is defined by OTSU as :

$$\sigma^2 = (m_G p_1 - m)^2 / p_1(1 - p_1)$$

ii) Linear filtering

This block implements two convolutions of the input image (intensity distribution function) with the x- and y-derivatives of a Gaussian function. (This is equivalent to first smoothing the input image with a Gaussian function and then computing the x- and y-derivatives.) The two convolution results present the x- and y-components of the scale dependent gradient of the image intensity distribution function.

iii) Sigma (σ) –

The standard deviation of the Gaussian function - is the only parameter of this step. Valid values are real positive numbers. Choose a small value (e.g. 1, 2 or 3) to detect sharp intensity transitions and a large value (e.g. 10 or 20) to detect gradual transitions. The filter kernel will be calculated for the specified value of the parameter sigma (σ) and will be displayed as an intensity map image in the output window. It is also possible to display the power spectrum of the filter kernel. (Light and dark gray colors correspond to positive and negative function values, respectively.)

iv) Surround Suppression Type

Default is "no surround suppression". This corresponds to the original Canny algorithm. If "isotropic surround suppression" is selected, all edges in the surroundings of a given edge have a suppression effect on that edge. The relative orientation of these edges has no influence on the suppression effect.

v) Superposition for isotropic suppression

If "isotropic suppression" is selected, a superposition of the convolution results for the x- and y-orientations is computed and deployed for surround suppression. Different types of superposition can be used: L1, L2 and L-infinity norms.

vi) Alpha (α)

This parameter controls the strength of surround suppression the higher the value of Alpha (α), the more the strength (gradient magnitude) of an edge surrounded by other edges will be reduced. Default is 1 but one may need larger values in order to completely suppress texture edges.

vii) K_1 and K_2

The surround that has a suppression effect on an edge in a given point has annular form with inner radius controlled by the combination of values of the parameters K_1 and K_2. The contribution of points in the annular surround is defined by a weighting function which is a half-wave rectified difference of Gaussian functions with standard deviations of $K_1\sigma$ and $K_2\sigma$ where σ is the standard deviation of the Gaussian function deployed in the linear filtering step. One can visualize the weighting function by selecting option "Inhibition kernel" under parameter "Output image". The inner radius of the annular surround increases with K_1. The size of the annual surround which has substantial contribution to the suppression increases with K_2. Default values are $K_1 = 1$ and $K_2 = 4$.

Flow chart of our proposed work is shown below-

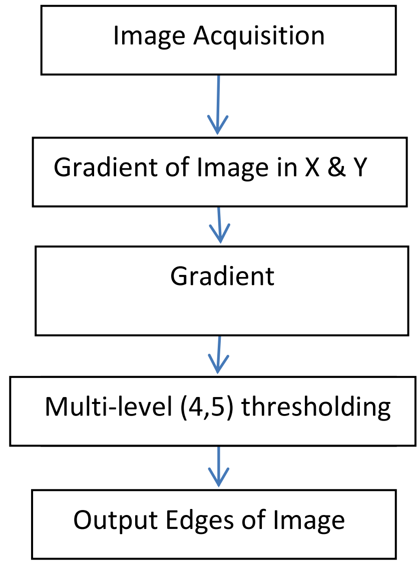

Fig.2 proposed Method of multi-level thresholding method

IV. RESULT

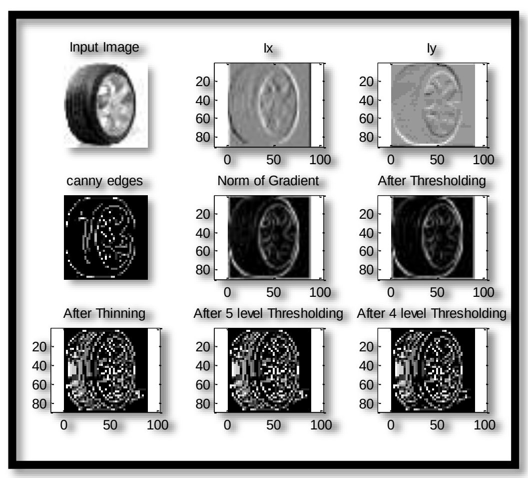

Fig.3 Edges of wheel

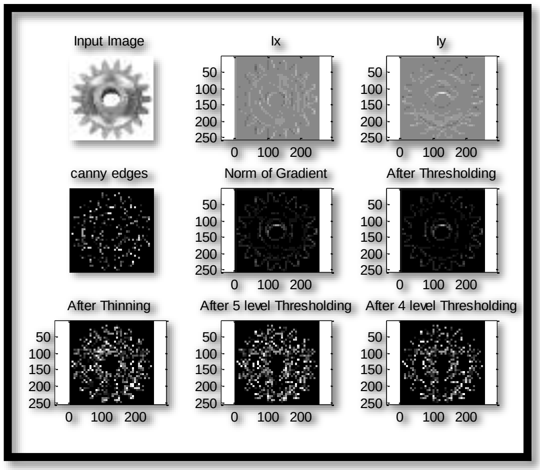

Fig.4 Edges of Metal

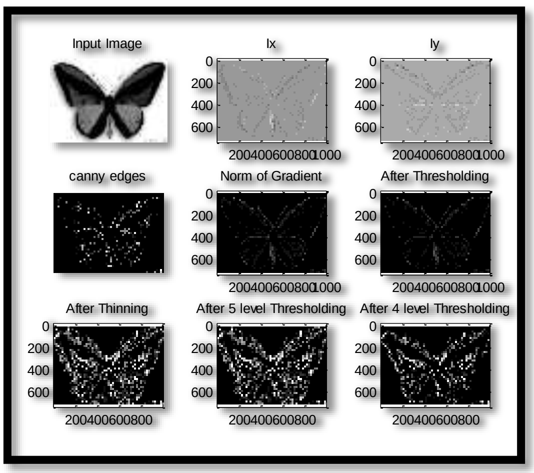

Fig.5 Edges of Butterfly colorful

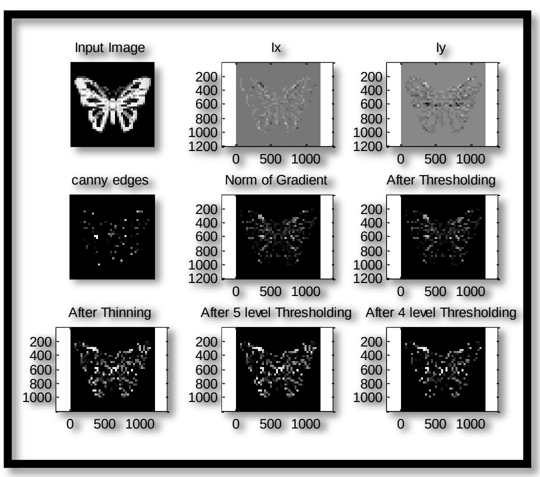

Fig.6 Edges of Butterfly Silver

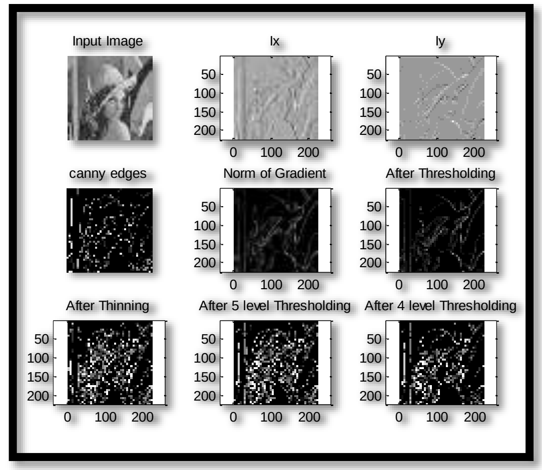

Fig.7 Edges of Lena

Performance Parameter table for different images

SN	Name of Image	Method	Variance	Mean	SNR	MSE	UIQI
1	Wheel	Base Paper	0.0803	0.0880	0.4002	.0099	.9338 $*e^{-6}$
		Proposed	1.7086	0.3979	0.9213	1.3717 $*e^{-14}$	0.0022
2		Base Paper	0.0477	0.0502	0.2238	.0020	2.7822 e^{-6}
	Metal	Proposed	0.9214	0.1911	0.4182	1.372 $*e^{-14}$	0.0020
3	Lena	Base Paper	0.0731	0.0794	0.3592	.0035	4.7400 e^{-6}
		Proposed	0.7309	0.1852	0.4909	.0016	0.0024
4	Butter fly (Silver	Base Paper	0.0178	0.0181	0.0794	1.51* e^{-4}	4.1582 e^{-6}

	)						
		Proposed	0.3146	0.0722	0.1796	0	0.0028
5	Butter fly (colorful)	Base Paper	.0304	.0314	.1387	.06836	.006198
		Proposed	.7572	.1548	.3369	2.7711*10^{-7}	.0020

On the basis of the above table, graph of different parameter is given below, between the base paper method and our proposed method.

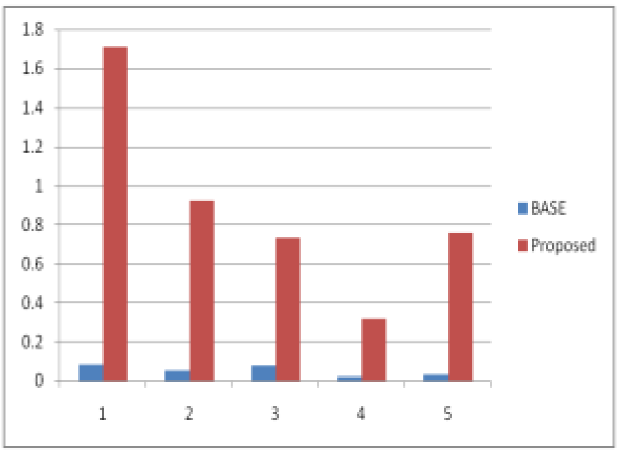

Fig. 8 Variance for different images

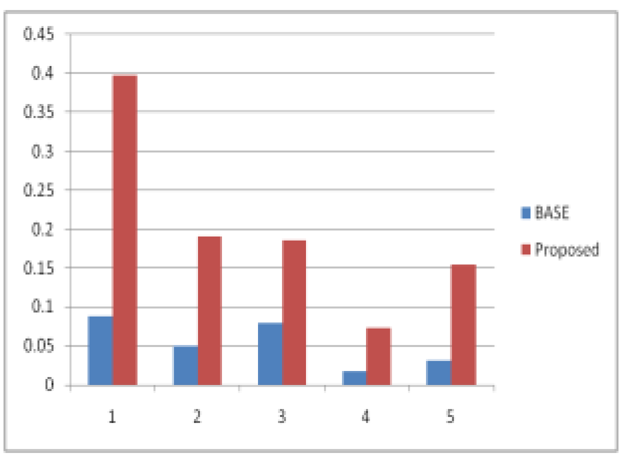

Fig. 9 Mean for different images

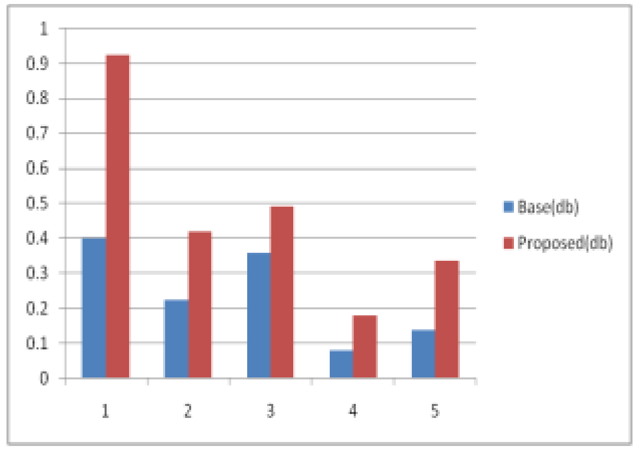

Fig. 10 SNR for different images

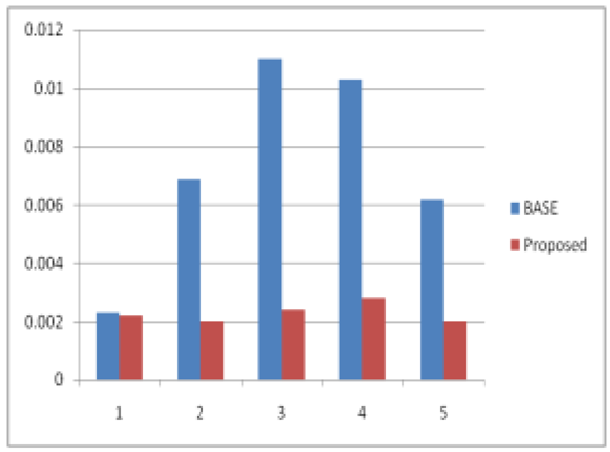

Fig. 11 UIQI for different images

III. CONCLUSIONS

The purpose of this paper is to present a result of various approaches for canny edge detection techniques. The study of different canny edge detection technique and their experimental results shows that OTSU multi-level thresholding gives the better result than simple OTSU thresholding. OTSU based canny, works on the dual threshold level which is automatic decided by the canny operator. Canny edge detection is more reliable [19]. Canny improves signal to noise ratio, and gives the better result in noisy conditions [7]. Performance parameters and graphs which are shown above prove that OTSU based on multilevel thresholding canny edge detection method improves the quality of detection.

REFERENCES:-

[1] G. T. Shrivakshan, Dr.C. Chadraseker IJCI Internetional Journal of Computer Science Issues, Vol. 9, issue 5, No 1, September 2012 ISSn (online): 1694-0814.

[2] Raman Maini & Dr. Himanshu Aggarwal International Journal of Image Processing (IJIP), Volume (3): Issue (1).

[3] Shazia Akram, Dr. mehraj-Ud-Din Dar, Aasia Quyoum International Journal of Computer application (0975-8887) Volume 10- No.5 november 2010.

[4] V.Torre and T.A. Poggio," On edge detection" IEEE Transactions on Patern Analysis , Vol. 8, pp. 147-163, march,1986.

[5] Erdon, A., 1992. Edge detection methods in digital images. Msc. Thesis, ITU.,Istanbul,Turkey.

[6] C.Tomasi, R. Manduchi. Bilateral Filtering for Gray and Color Images[C]. Proceedings of the 1998 IEEE International Conferences on Computer Vsion.Bombay,India,,pp:839-846,1998.

[7] J.Canny-"A Computational Approach to Edge Detection" IEEE Trans. Pattern Analysis and Machine Intelligence. Vol.8,no.6,1986,pp.679-698.

[8] J.J. Clark,"Authenticating Edges Produced by zero Crossing Algorithm" IEEE Trans. Pattern Analysis and Machine Intelligence.vol.8, no.1, 1989,pp.43-57.

[9] CAI-XIA DENG, GUI-BIN WANG, XIN-RUI YANG, International conference on Wavelet analysis and Pattern Recognition, Tianjin, 14-17 july, 2013, 978-1-4799-0417-4/13/2013 IEEE.

[10] Geng Hao, Luo Min, Hu Feng 2013 Fifth International Conference on Intelligent Human-Machine Systems and Cybernetics 978-0-7695-5011-4/13 2013 IEEE.

[11] Yanpeng sun, yongtao Wei, Jun Xu, Simin Xu, 2012 5th International Conference on Image and Signal Processing (CISP 2012) 978-1-4673-0964-6/12, 2012 IEEE.

[12] F.Bergholm, "Edge Focusing" IEEE Trans. Pattern Analysis and Machine Intelligence. Vol.9, 1987, pp. 726-741.

[13] Er. Snigdha Mohanty and Er. Mahesh Prasad Sahoo Bhubaneswer Engineering college, KGI Campus-4, Bhubneswer.

[14] V.Torre and T.A. Poggio," On edge detection" IEEE Transactions on Patern Analysis , Vol. 8, pp. 147-163, march,1986.

[15] John Canny. "A computational approach to edge detection. Pattern Analysis and Machine Intelligence,"IEEE Transactions on, PAMI- 8(6):679–698, Nov. 1986. (Book style).

[16] Biman Debbarma and Dibyendu Ghoshal International Journal of Emerging Trends in Electrical and Electronics (IJETEE) Vol. 1, Issue. 2, March-2013.

[17] www.imageprocessingplace.com.

[18] Sunanda Gupta, Charu Gupta, S.K. Chakarvarti International journal of Advanced Research in Computer engineering & technology (IJARCET) ISSN: 2278-1323, Volume 2, Issue 7,July 2013.

[19] Poonam Dhankhar, Neha Sahu International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology ISSN 2320–088X IJCSMC, Vol. 2, Issue. 7, July 2013, pg.86 – 92.

