Improved Simplified Novel Method for Edge Detection in Grayscale Images Using Adaptive Thresholding

Tirath P. Sahu and Yogendra K. Jain

***Abstract*—Edges play a crucial role in digital image processing. In this paper, an improved simplified novel method is presented to improve the quality of edge detection in grayscale images. In this method, a 3x3 binary matrix is employed to process each and every pixel by means of vicinity. Each pixel is considered an input and by examining binary matrix, the edge pixel is specified and by utilizing conditions and adaptive thresholding of the algorithm, edges are displayed more clearly. This method not only detects edge in images but removes noise as well. A comparative study of different edge detection techniques like Sobel, Prewitt, Roberts, LoG, zero-cross, Canny, fuzzy techniques and SNM with the proposed technique is presented. The proposed technique gives better results in terms of smoothness, sharpness, continuity, true edges and execution time than the conventional edge detection techniques.**

***Index Terms*—Edge detection, digital image processing, additive noise, image segmentation, adaptive thresholding, morphological processing.**

I. Introduction

Edge detection is a fundamental tool used in most image processing applications to obtain information from the frames as a precursor step to feature extraction and object segmentation. An edge is defined by a discontinuity in gray level values. This process detects outlines of an object and boundaries between objects and the background in the image [1]. An edge detection filter [1]-[3] can also be used to improve the appearance of blurred image. An overview of some of the studies done in this subject is given below.

One of the most important applications of edge detection is image segmentation. The process of partitioning a digital image into multiple regions or sets of pixels is called image segmentation [1]. Edge is a boundary between two homogeneous regions. Edge detection refers to the process of identifying and locating sharp discontinuities in an image. The Canny algorithm [4] uses an optimal edge detector based on a set of criteria which include finding the most edges by minimizing the error rate, marking edges as closely as possible to the actual edges to maximize localization, and marking edges only once when a single edge exists for minimal response. The non-maximal suppression stage identifies pixels that are local maxima in the direction of the gradient using the magnitude and orientation of the pixels. The major orientation of the gradient, either horizontal or vertical, is obtained by comparing the individual components, G_x and G_y, which are the result of convolving the smoothed image with the derivative of the Gaussian. Since most edges are at an angle, it is possible to obtain further granularity in the orientation of the gradient by comparing the sign bit of the gradient [2], [3].

An Edge in an image is a significant local change [5] in the image intensity, usually associated with a discontinuity in either the image intensity or the first derivative of the image intensity. Discontinuities in the image intensity can be either Step edge, 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 Edges, where the image intensity abruptly changes value but then returns to the starting value within some short distance [6], [7]. Many points in an image have a nonzero value for the gradient, and not all of these points are edges for a particular application. Therefore, some method should be used to determine which points are edge points. Frequently, threshold provides the criterion used for detection [8]. The Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image [9]. It thus highlights regions of high spatial frequency which often correspond to edges. In its most common usage, the input to the operator is a grayscale image, as is the output. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point. The Prewitt edge detector is an appropriate way to estimate the magnitude and orientation of an edge [10]. Although differential gradient edge detection needs a rather time consuming calculation to estimate the orientation from the magnitudes in the x and y-directions, the compass edge detection obtains the orientation directly from the kernel with the maximum response. The Prewitt operator is limited to 8 possible orientations however experience shows that most direct orientation estimates are not much more accurate. This gradient based edge detector is estimated in the 3×3 neighborhood for eight directions. All the eight convolution masks are calculated. One convolution mask is then selected, namely that with the largest module. In all cases these operators are not suitable for sharp detection of different geometrical shapes in images. Moreover, the visible clarity of hidden alphabets in images is not well detected. Some advanced techniques based on soft computing like fuzzy logic [11], [12] give better results than the conventional techniques but take longer execution time. The simplified novel method [8] reduces the execution time but results with blunt edges. Our method gives better results as compared to conventional techniques and SNM with less computational time.

The proposed algorithm is based on adaptive thresholding approach of edge detection for grayscale images and is

Manuscript received September 10, 2014; revised April 13, 2015.

Tirath P. Sahu is with the Department of IT, NIT Raipur, India (e-mail: tirasahu.it@nitrr.ac.in).

Yogendra K. Jain is with Department of CSE, SATI Vidisha, India (e-mail: ykjain_p@yahoo.co.in).

DOI: 10.7763/JACN.2015.V3.159

implemented in Matlab 10.0. The 3×3 binary matrix shown in Fig. 1 considered defining center pixel as an edge pixel.

[*i*-1,*j*-1]	[*i*-1,*j*]	[*i*-1,*j*+1]
[*i*,*j*-1]	[*i*,*j*]	[*i*,*j*+1]
[*i*+1,*j*-1]	[*i*+1,*j*]	[*i*+1,*j*+1]

Fig. 1. A 3×3 binary matrix.

II. Image Segmentation

The first step in image analysis is to segment the image. Segmentation subdivides an image into its constituent parts or objects. The level to which this subdivision is carried depends on the problem being solved. Image segmentation is typically used to locate objects and boundaries (lines, curves etc.) in images [13]. Sometime we need to segment the object from the background to read the image correctly and identify the content of the image, for this reason there are two techniques of segmentation: discontinuity detection technique and similarity detection technique. In the first technique the image is partitioned based on abrupt changes in gray-levels. The second technique is based on the regions and threshold. These paper discuses the second technique for Edge Detection based on thresholding.

III. Similarity Detecyion

Similarity detection partitions an image based on some similarity like gray-level values in image pixels. The edge pixels are determined using the gradient operator, but the resulting edges have small openings or gaps which are not desirable. Thus the boundary is not completely defined by the edge pixels. In order to fill openings or gaps at each pixel, we have to consider the characteristics of pixels in a small neighborhood (3×3 or 5×5). All the neighborhood pixels which are similar to this boundary pixel are linked. Two important properties used for checking the similarity of the neighborhood pixels with respect to the edge pixels are:

1) The strength of the gradient operator response to produce the edge pixel.
2) The direction of the gradient.

Now let us consider an edge pixel with coordinates (x, y) and a neighborhood pixel at the coordinate (x', y'). Then the neighborhood pixel (x', y') is similar to the edge pixel (x, y) if

$$|\boldsymbol{\nabla} f(x,y) - \boldsymbol{\nabla} f(x',y')| \tag{1}$$

where T is a non-negative threshold value. Then neighborhood pixel (x', y') with respect to edge pixel at (x, y) has an angle similar to the edge pixel if

$$|\propto f(x,y) - \propto f(x',y')| < A \tag{2}$$

where

$$\propto (x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{3}$$

and A is an angle of threshold.

A neighborhood pixel *(x',y')* is linked to the edge pixel *(x,y)* if both magnitude and direction given in above equations are satisfied [1].

Thresholding based image segmentation aims to partition an input image into pixels of two or more values through comparison of pixel values with the predefined threshold value T individually. This is the simplest way of segmentation. Using thresholding technique regions can be classified on the basis of range values, which is applied to the intensity values of the image pixels. The thresholding technique can be put into three different types based on function T and its associated parameters as given by:

$$T = T[(x,y), p(x,y), f(x,y)] \tag{4}$$

where $f(x, y)$ is the gray level at the point (x, y) and $p(x, y)$ denotes some local property at that point (e.g. the average gray level of neighborhood center on (x, y)). Thresholding is the transformation of an input image $f(x, y)$ to an output (segmented) binary image $g(x, y)$ as follows [1]:

$$g(x,y) = \begin{cases} 1, & f(x,y) \geq T \\ 0, & f(x,y) < T \end{cases} \tag{5}$$

where T is the threshold, $g(x, y)$ =1 for image elements of objects and $g(x, y)$=0 for image elements of the background(or vice versa). When the threshold value T depends only on $f(x, y)$ the threshold technique is called global. If the T depends on both $f(x, y)$ and $p(x, y)$, the threshold is called local. If T depends on all the three parameters, that is, the coordinates (x, y), local property $p(x, y)$, and $f(x, y)$ then the threshold is called dynamic. Thresholding is computationally inexpensive and fast, it is the oldest segmentation method and is still widely used in simple applications. Using range values or threshold values, pixels are classified using either of the thresholding techniques like global and local thresholding. Global thresholding method selects only one threshold value for the entire image. Local thresholding selects different threshold values for different regions. To segment complex images multilevel thresholding is required. Fig. 2 shows how thresholding works on gray-scale images to find the pixels i.e. candidate of edge.

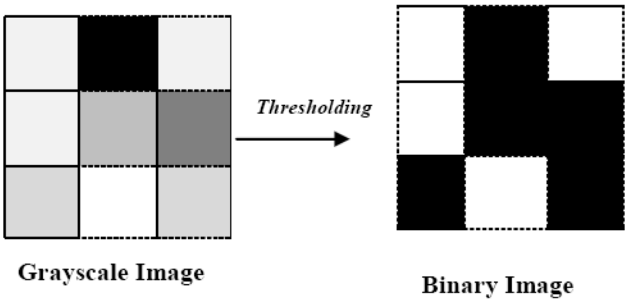


Fig. 2. Finding edge pixels in grayscale image.

IV. Proposed Technique for Edge Detection

As some of the conventional techniques the proposed technique is also based on the first order derivative of the gradient operator [1]. In this technique image is first converted into grayscale image. There is no need to convert image into a binary image as in *Simplified Novel Method* [8]. This is the key factor of our algorithm. Many techniques are there for image segmentation based on discontinuity and similarity detection [13]. Here we use adaptive thresholding based on similarity detection for image segmentation. The thresholding technique can be put into three different types based on function T and its associated parameters as given by:

$$T = T[(x, y), p(x, y), f(x, y)] \tag{6}$$

where $f(x, y)$ is the gray level at the point (x, y) and $p(x, y)$ denotes some local property at that point (e.g. the average gray level of neighborhood center on (x, y)). Thresholding is the transformation of an input image $f(x, y)$ to an output (segmented) binary image $g(x, y)$ as follows: [1]

$$g(x, y) = \begin{cases} 255, & f(x, y) \geq T \\ 0, & f(x, y) < T \end{cases} \tag{7}$$

where T is the adaptive threshold value, $g(x, y)$=255 for image elements of objects and $g(x, y)$=0 for image elements of the background(or vice versa) [1]. Consequently a 3×3 widow is employed to process each and every pixel with central pixel as a query pixel. The method is used to decide whether the query pixel is a part of edge or not, based on gray values of the pixels. In general, it is found that the best results are achieved if gray values from 0 to 80 are set as black and that from 81 to 255 are set as white. We have added the border pixels throughout the image to decide the edginess of border pixels comfortably. The 3×3 window spans across the image from one pixel to another and so on. The condition of 3×3 window is given in the flowchart. The conditions given in the flowchart decide the central pixel as an edge pixel. Thickness of obtained edges is not desirable; so the post-processing morphological [14] thin function is applied to thin the obtained edges. Flowchart of this technique shown in Fig. 3 explains step by step procedure to be followed.

V. Experimental Results

The proposed system is simulated in MATLAB 10.0 and tested with different images without noise and with additive noise. The results are then compared to that of the conventional technique like Sobel, Prewitt, Roberts, LoG, zero-cross, Canny and fuzzy techniques.

From Fig. 4(a), it is clear that the proposed system gives the best result in terms of smoothness; sharpness and continuity of outer boundary which the conventional technique fails to detect it. The edges of hole are continuous relative to other techniques. This image is tested without the addition of noise, and the corner of the image get sharper than the conventional techniques.

Fig. 5(a) shows that the image containing Gaussian noise is tested with all techniques. The proposed technique yields smoother boundary relative to other conventional techniques and removes noise as well. Fuzzy technique, using 2×2 mask also removes noise but boundary is blunt and using 3×3 mask, the bottom edge formed by boundary pixel of image is not detected.

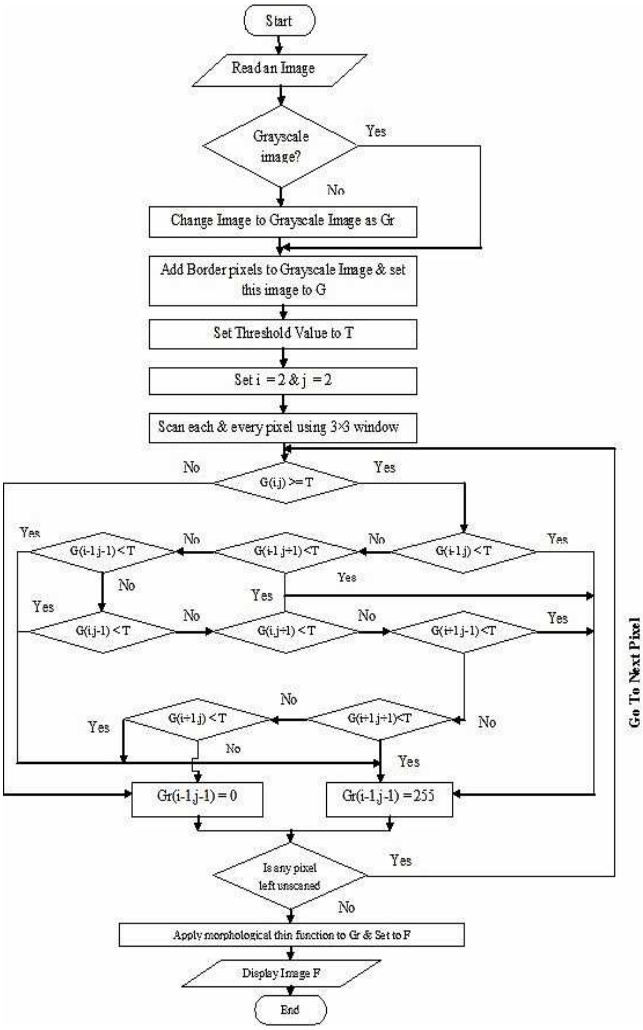


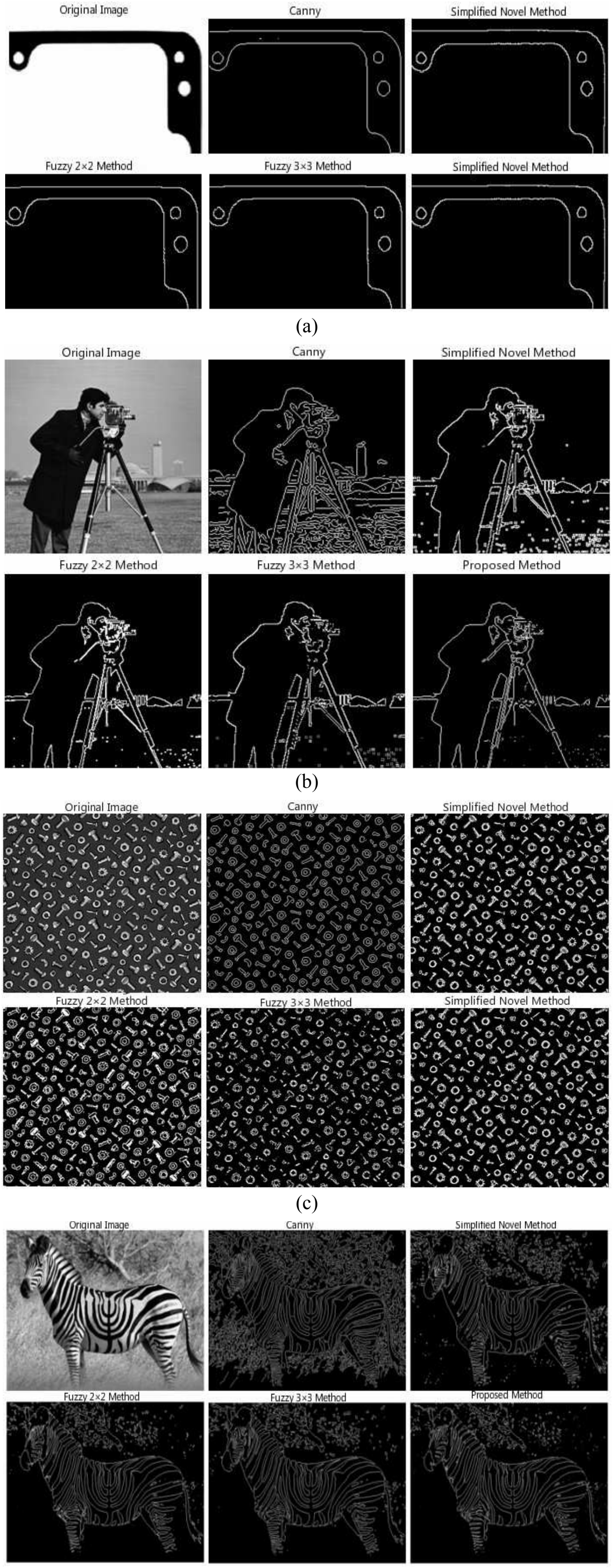
Fig. 3. Flowchart of proposed technique.

In Fig. 4(b) and Fig. 5(b) it is seen that the image of cameraman with the proposed technique detects the object clearly. When conventional techniques are applied to same image the generated output has lot of false edges generated from image background. The proposed method results with thin edges and less noisy flat areas. It also gives sharper result in the edgy regions than the conventional techniques.

Fig. 4(c) and Fig. 5(c) shows that the image of nuts-bolts with Canny operator gives smoother result but it detects the edge of nuts as circle shape. The proposed technique detects the edge of nuts precisely. The edge detected by the fuzzy techniques contains more pixels resulting thick edges. The edges of nuts-bolts image with additive noise are detected precisely by proposed method whereas fuzzy techniques, SNM fails to detect the edges of the object clearly. The edges of the image get smoother with the proposed method.

In Fig. 4(d) and Fig. 5(d) the edge detection techniques is applied to zebra image and by visualization our proposed method gives best results in terms of object. Canny operator detects edges of object as well as false edges from background. The outputs of fuzzy techniques also have similar issues.

For quantitative measurement of the performance of proposed method with classical edge detectors and fuzzy techniques we have used two parameters, namely PSNR and Execution Time. The simulation results are tabulated below. Simplified Novel Method [8] gives best results in terms of execution time. But in terms of the quality of the detected edge, Canny edge detector is the best among the conventional techniques. So we are comparing the result of proposed technique relative to Canny Operator. As indicated in the tables, the output of proposed method is best on all the three parameters.



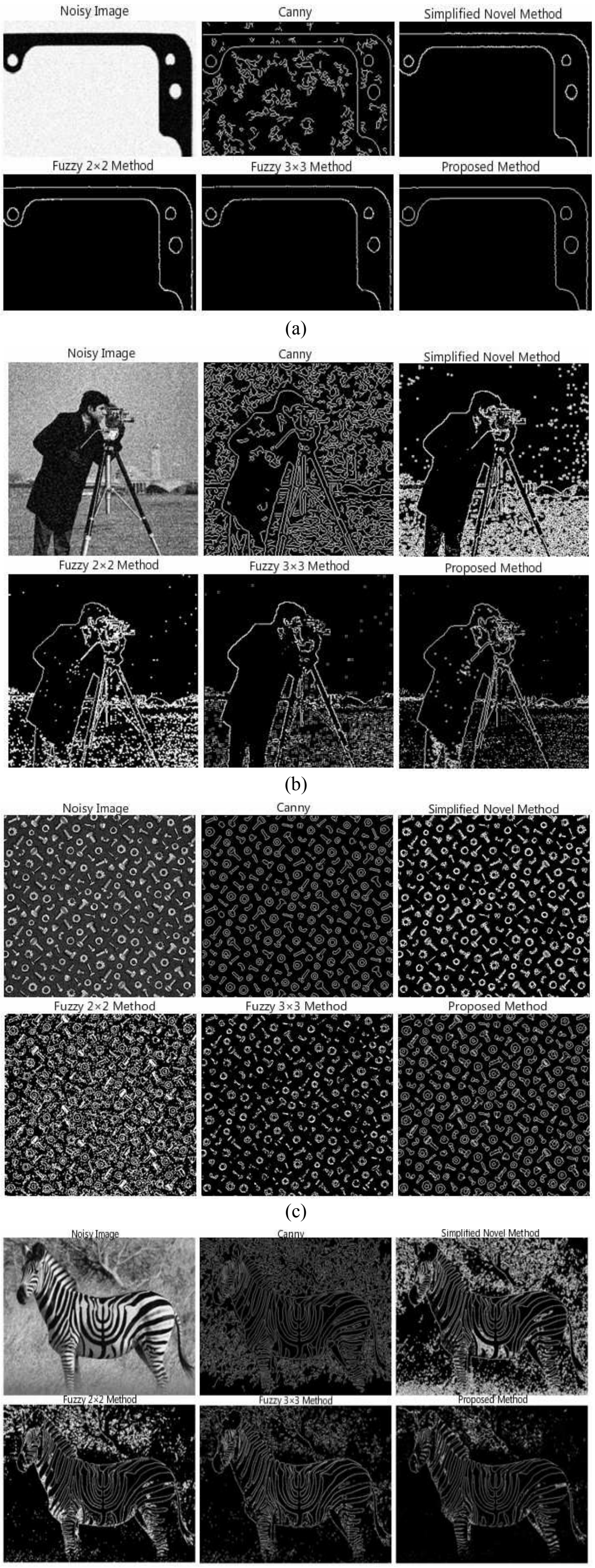
(a)

(b)

(c)

(d)

Fig. 4. Edge detection without noise on (a) angle (b) camera man (c) nut-bolts (d) zebra images.



(a)

(b)

(c)

(d)

Fig. 5. Edge detection with additive noise on (a) Angle (b) camera man (c) nut-bolts (d) zebra images.

Table I shows the simulation results without additive noise and Table II shows the same for images with additive noise.

TABLE I: PSNR AND EXECUTION TIME WITHOUT ADDITIVE NOISE

Image	Evaluation Parameters	Edge Detection Techniques			
		Fuzzy (2×2)	Fuzzy (3×3)	SNM	**ISNM**
angle	PSNR(dB)	+16.49	+16.66	+62.57	**+64.03**
	Execution Time(Sec)	2.0163	3.3469	0.5051	**0.4274**
Camera -man	PSNR(dB)	+12.46	+13.61	+57.58	**+58.04**
	Execution Time(Sec)	2.8153	5.3394	0.4135	**0.3949**
Nuts-bolts	PSNR(dB)	+ 7.86	+ 9.96	+56.13	**+56.24**
	Execution Time(Sec)	4.7894	7.6493	0.2585	**0.2227**
Zebra	PSNR(dB)	+10.97	+11.07	+56.16	**+56.29**
	Execution Time(Sec)	9.5771	29.772	0.5644	**0.4906**

TABLE II: PSNR AND EXECUTION TIME WITH ADDITIVE NOISE

Image	Evaluation Parameters	Edge Detection Techniques			
		Fuzzy (2×2)	Fuzzy (3×3)	SNM	**ISNM**
angle	PSNR(dB)	+16.17	+16.73	+57.96	**+58.40**
	Execution Time(Sec)	1.8080	4.3230	0.4088	**0.3725**
Camera -man	PSNR(dB)	+ 8.90	+11.31	+53.83	**+54.88**
	Execution Time(Sec)	2.8889	5.1032	0.3213	**0.2761**
Nuts-bolts	PSNR(dB)	+ 6.02	+10.06	+56.05	**+57.22**
	Execution Time(Sec)	3.6563	11.0372	0.6301	**0.4380**
Zebra	PSNR(dB)	+ 8.75	+10.64	+53.84	**+55.44**
	Execution Time(Sec)	10.290	29.311	0.5369	**0.3999**

VI. CONCLUCIONS AND FUTURE WORK

The proposed method gives the best result in terms of noise removal, smoothness, sharpness and continuous edge boundary. The classical edge detectors doesn't remove noise that causes blunt edges which are minimized using the proposed method that offers good result with less computational time and greater accuracy. Hence edge detection is improved to a great extent. Adding border pixels to image and applying morphological thin function makes this method versatile and robust. Though simple the method offers best result among the classical operators. The method can be useful in a wide variety of applications where digital image processing plays an important role as in medical backgrounds, robotics where fast processing of image is required. The proposed technique is tested and compared in MATLAB 10.0 environment. A drawback of the proposed technique is that it fails to detect edges in images with lot of details. The enhancement of this technique can be devised by using other intelligent techniques for image segmentation which would clearly identify the objects from the background in detailed images.

REFERENCES

[1] R. Gonzalez and R. Woods, *Digital Image Processing*, 2nd ed. Prentice Hall Publisher, 2002.

[2] M. Sharifi, M. Fathy, and M. T. Mahmoudi, "A classified and comparative study of edge detection algorithm," in *Proc. the International Conference on Information Technology: Coding and Computing, IEEE*, 2002.

[3] M. Roushdy, "Comparative study of edge detection algorithms applying on the grayscale noisy image morphological filter," *GVIP Journal*, vol. 6, issue 4, December 2006.

[4] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679-714, November 1986.

[5] C. Li, K. He, and J. Zhou, "Edge detection of image on the local feature," in *Proc. the Second International Symposium on Intelligent Information Technology Application, IEEE*, 2008.

[6] I. E. Sobel, "Camera models and machine perception," Ph.D. dissertation, Stanford University, Palo Alto, Calif, 1970.

[7] M. Abdulghafour, "Image segmentation using fuzzy logic and genetic algorithms," *Journal of WSCG*, vol. 11, no. 1, 2003.

[8] M. K. Ray, D. Mitra, and S. Saha, "Simplified novel method for edge detection in digital images," in *Proc. the 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies IEEE*, 2001, pp. 197-202.

[9] L. G. Roberts, "Machine perception of three-dimensional solids," in *Optical and Electro-optical Information Processing*, J. T. Tipped, Ed. Cambridge, Mass: MIT Press, 1965.

[10] J. M. S. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictorics*, B. S. Lipkin and A. Rosenfeld, Eds. New York: Academic Press, 1970.

[11] S. Mathur and A. Ahlawat, "Application of fuzzy logic on image edge detection," *Intelligent Information and Engineering Systems*, INFOS 2008.

[12] A. A. Alshennawy and A. A. Aly, "Edge detection in digital images using fuzzy logic technique," *World Academy of Science, Engineering and Technology*, 2009.

[13] N. Senthilkumaran and R. Rajesh, "Edge detection techniques for image segmentation — A survey," in *Proc. the International Conference on Managing Next Generation Software Applications*, 2008, pp. 749-760.

[14] L. Rey-Sern, *Edge Detection by Morphological Operations and Fuzzy Reasoning*, 2008.

Tirath P. Sahu is with the Dept. of IT, NIT Raipur. He was born in Baikunthpur, India in 1987. He received the B.Tech. degree in information technology from NIT Raipur, India, in 2009 and the M.Tech. degree in computer science & engineering from SATI, Vidisha, India in 2012. He is currently pursuing his Ph.D. degree with the Department of Computer Science & Engineering, NIT Raipur. His research interests include image processing, data mining and text analytics.

Yogendra K. Jain presently working as the head of the Department of Computer Science & Engineering at Samrat Ashok Technological Institute Vidisha M.P India. He received the B.E. (Hons) degree secured in E&I from SATI Vidisha in 1991, the M.E. (Hons) degree in digital tech. & instrumentation from SGSITS, DAVV Indore (M.P), India in 1999. He received the Ph.D. degree from Rajiv Gandhi Technical University, Bhopal (M.P.) India in 2010. His research interests include image processing, image compression, network security, and data mining. He published more than 40 research papers in various journals/conferences.