International Journal of Computational and Applied Mathematics
ISSN 1819-4966 Volume 5 Number 1 (2010), pp. 53–65

http://www.ripublication.com/ijcam.htm

Better Edgegap in Grayscale Image Using Gaussian Method

[1]D.S. Pathania, [2]S.K. Singla, [3]Pardeep Kumar and H.K.Grewal

[1] Department of Applied Mathematics, Guru Nanak Dev Engg. College, Ludhiana (Punjab) INDIA, E-mail: despathania@yahoo.com
[2,3]Department of Information Technology, Guru Nanak Dev Engg. College, Ludhiana.
Deptt of Elecronics and Electrical Engg, SSIET, Dera Bassi

Abstract

In this paper, different first and second derivative gradient operators are investigated with experiments to find edge map in a gray scale image. Comparative analysis of various edge detection gradient operators of first derivative like Sobel, Prewitt, Roberts and second derivative like Laplacian and Laplacian of Gaussian are presented. In this paper, we have proposed a new derivative filter of first derivative gradient type and a novel approach with an aim to find better edge map in a grayscale image. The proposed derivative operator is convolved to the input image for all coordinate points to obtain both the horizontal and vertical gradient components. The magnitude of the gradient component is calculated. The gradient component is normalized and threshold to a level to find the edge map information. Subjective and objective methods are used to evaluate the performance of the proposed operator with other existing edge detection operators. The subjective method is used by visually comparing the different edge detected output images with characteristics like contrast value in the image, strength of edge map, and noise content obtained by different derivative filters. The objective method like root-mean-square error is used to find the performance of different derivative filters. Image sharpening is a common technique in the edge detection process with the objective of enhancing edges in an image. We have proposed one such technique called ‘Fuzzy Gaussian Filtering’. Finally, to validate the efficiency of the edge detection and edge sharpening scheme, different algorithms are proposed and simulation study has been carried out using MATLAB 5.0.

Keywords: edge detection, gradient operator, image processing, root-mean-square error.

1. Introduction

The edge of an image describes the boundary between an object and the background. It represents a sudden change in the value of the image intensity function. So an edge separates two regions of different intensities. However all the edges in an image are not due to the change in intensity values, where parameters like poor focus or refraction can result in edge in an image [1]. The shape of edges in an image depends on different attributes like, lighting conditions, the noise level, type of material and the geometrical and optical properties of the object [2]. Gradient operators of first derivative like Sobel, Prewitt, Roberts and second derivative like Laplacian are used to find the edge in an image [3, 4, 6, 22, 25]. The efficient edge detection operator is evaluated subjectively by visually comparing the output images obtained with certain characteristics [5].

The rest of the paper is presented as follows: section (2) describes the different first and second derivative filters for finding edge in a gray scale image. Section (3) describes the Fuzzy Gaussian filtering to enhance the edge in an image. Section (4) presents algorithms required and section (5) shows the experimental results of different edge detection images with the subjective and objective evaluation methods and finally the discussion of the results and conclusion is presented in section (6).

2. First & Second Derivative Filters

The gradient of an image f(x, y) at the location (x, y) is given by the two dimensional column vector [11, 22].

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{1}$$

The magnitude of the first derivative is used to detect the presence of an edge in the image. The magnitude of this vector is given by [18]:

$$mag(\nabla f) = [G^2_x + G_y{}^2]^{1/2} \tag{2}$$

Here $\partial f / \partial x..and...\partial f / \partial y$ are the rates of change of two dimensional function f(x, y) along x and y axis respectively. A pixel position is declared as an edge position if the value of the gradient exceeds some threshold value, because edge points will have higher pixel intensity values than those surrounding it [10, 11].

We have used a 3X3 region to denote image points of an input image [8, 9] as follows:

Figure 1: A 3X3 region of an image.

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

$$W_1 = f(x-1, y-1), W_2 = f(x-1, y), W_3 = f(x-1, y+1)$$
$$W_4 = f(x, y-1), W_5 = f(x, y), W_6 = f(x, y+1)$$
$$W_7 = f(x+1, y-1), W_8 = f(x+1, y), W_9 = f(x+1, y+1)$$

2.1. Sobel Operator

The Sobel operator is given by the equations [8, 12, 22]:

$$G_x = (W_7 + 2W_8 + W_9) - (W_1 + 2W_2 + W_3)$$
$$G_y = (W_3 + 2W_6 + W_9) - (W_1 + 2W_4 + W_7) \quad (3)$$

Where, W_1 to W_9 are pixels values in a sub image as shown in Fig.1.

2.2. Roberts Operator

The Roberts operator is given by the equations [8, 13]:

$$G_x = W_9 - W_5$$
$$G_y = W_8 - W_6 \quad (4)$$

2.3. Prewitt Operator

The Prewitt's operator is given by the equations [8, 14]:

$$G_x = (W_7 + W_8 + W_9) - (W_1 + W_2 + W_3)$$
$$G_y = (W_3 + W_6 + W_9) - (W_1 + W_4 + W_7) \quad (5)$$

2.4. Proposed Operator

Our proposed operator is given by equations:

$$G_x = (W_7 + 3W_8 + W_9) - (W_1 + 3W_2 + W_3)$$
$$G_y = (W_3 + 3W_6 + W_9) - (W_1 + 3W_4 + W_7) \quad (6)$$

2.5. Laplace Operator

The sign of the second derivative is used to decide whether the edge pixel lies on the dark side or light side of an edge [15, 20]. The second derivative at any point in an image is obtained by using the laplacian operator [22]. The Laplacian for an image function f(x, y) of two variables is defined as [11, 16]:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (7)$$

The Laplacian operator is given by the equation:

$$\nabla^2 f = (W_2 + W_4 + W_6 + W_8) - 4W_5 \quad (8)$$

2.6. Laplacian of Gaussian Operator

The Laplace operator is very much sensitive to noise. Hence it is clear that some sort of noise cleaning procedures must be preceded before the Laplacian. For noise smoothing the Gaussian filter can be applied. Then the resultant procedure is called Laplacian of Gaussian (LOG). It can be defined as:

$$g''(r,c) = \nabla^2 \{ g(r,c) * G(r,c) \} \quad (9)$$

The root-mean-square error [8] between the input and output image is defined as:

$$e_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]^2 \right]^{1/2} \quad (10)$$

Where, $f(x, y)$ denotes the original input image and $\hat{f}(x, y)$ denotes the output image. Both the images have M rows and N columns and e_{rms} is the root-mean-square error.

3. Fuzzy Gaussian Filter

Image sharpening is a common technique in the edge detection process with the objective of enhancing edges. Typical algorithms adjust the value of each pixel according to a weighted average of the pixel in a surrounding neighbourhood. The Fuzzy Gaussian filtering algorithm is very immune to particular information content in an image. The general Gaussian filtering is expressed as the convolution of the original image ‘I” with a Gaussian shaped mask ‘W’, then $I' = I * W$

where W has the discrete form:

$$W(i, j) = \frac{1}{2 * pi * sigma^2} \exp\left[\frac{I^2 + J^2}{2 * sigma^2} \right] \quad (11)$$

The value of ‘sigma’ effectively determines the breadth of W and the value of W is constant for a given radius from the centre of the function. This filter is controllable because the size of filter is detected by sigma, and the shape of the filter is to be determined dynamically according to local data characteristics. The controllable value of sigma determines the degree of smoothing of the image. The smoothing is done to clean the distortions caused by noise but preserves dominant shape. The filter shape squashing is done in both horizontal and vertical directions and if necessary along the direction perpendicular to the gradient. The Gaussian mask can be represented in a matrix form:

$$\begin{bmatrix} -0.3033 & -0.2186 & -0.1479 & -0.2186 & -0.3033 \\ -0.2186 & 0.1608 & 0.4776 & 0.1608 & -0.2186 \\ -0.1479 & 0.4776 & 1.0000 & 0.4776 & -0.1479 \\ -0.2186 & 0.1608 & 0.4776 & 0.1608 & -0.2186 \\ -0.3033 & -0.2186 & -0.1479 & -0.2186 & -0.3033 \end{bmatrix}$$

Figure 2: A 5X5 Gaussian mask.

The following figure shows the graphical view of the Gaussian filter obtained with mask size 5 X 5 and sigma value 1.0. The first row of graphs shows the variation of pixel values along different rows and the succeeding row of graphs are variation of pixel values along the columns.

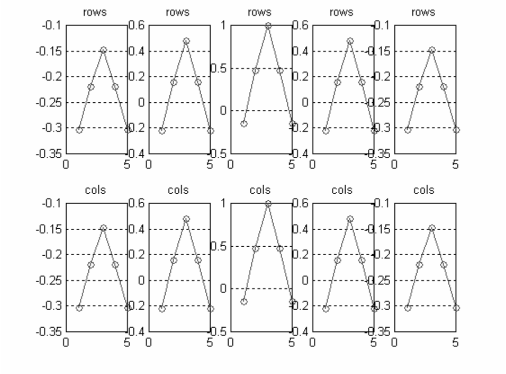

Figure 3: Gaussian filter with mask of 5X5 and sigma = 1.0.

3.1. Fuzzyfication and Defuzzyfication process

The fuzzy logic is different from the conventional binary logic. In binary logic the variable can be 0 and 1, but the fuzzy variable can be any values in between o and 1. In digital images conformation of existence of an edge can not have binary answer. The images have a continuous change in brightness level as well as in shades. This information will be lost if we take binary answer to this problem. All the edge points constitute as set which is called an edge map. It is not wise to tell whether a certain pixel belongs to the edge map or not, rather each of the pixels should be assigned some membership towards the set. This process is called 'Fuzzyfication'. In this process, we perform Gaussian filtering with different values of sigma. Pixels with membership value one definitely belong to the edge map where as pixels with membership value equal to zero do not belong to the set. However pixels with intermediate membership values may or may not belong to the edge map depending upon a prescribed threshold value. Thresholding operation is called a 'Defuzzyfication' process. After thresholding we obtain a binary image which is the edge map representation of the image. Though fuzzy sets have a realistic approach towards the real world problems, they are seldom implemented directly, because for computer representation the data has to be in binary value. Hence the fuzzy set is now transformed into binary set. Around 5 to 10% of the total pixels generally belong to the edge map. Pixels having membership less than threshold value are assigned value 0 and those having greater membership are assigned value of 1.

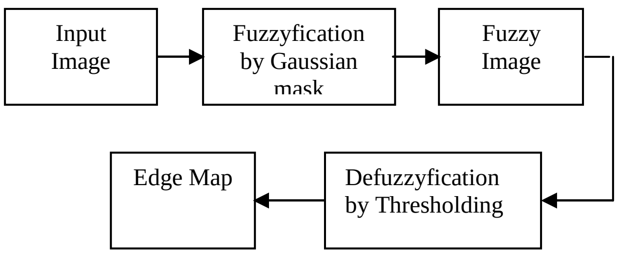

Figure 4: Block Diagram of Fuzzy Gaussian Filter.

4. Methods

We propose the following algorithms to get the edge map of the input gray scale image. The first algorithm is used to convolve the mask to the input image for all coordinate points of the image. The second algorithm describes the process of normalization and the third algorithm describes the process used to threshold the image at a particular level to find the edge map of a gray scale image. The fourth algorithm describes the general procedure used to find the edge map for all gradient operators.

ALGORITHM 4.1. Convolving an image with odd mask.

Begin

(1) Read all the pixel vales of input image with M rows and N columns where, f(x, y) represents the pixel value at x and y co-ordinate.
(2) Store all the pixel vales in an integer matrix of dimension M X N.
(3) Select the mask w, which is an array with dimension m X n, indexed from 0 to m -1 horizontally and 0 to n -1 vertically for m rows and n columns.
(4) Fill the mask w with mask coefficients
(5) The sum of all the coefficients of each mask must be zero.
(6) Compute Mask half-width, a = (m – 1)/2 // for a mask of odd size
(7) Compute Mask half-height, b= (n – 1)/2 // for a mask of odd size
(8) Create an M X N output image, g with M rows and N columns
(9) for all pixel coordinates, x and y, do
(10) g(x, y) = 0
(11) end for
(12) for y = b to N – b - 1 do // column pixels in the image excluding Border pixels
(13) for x = a to M – a – 1 do //row pixels in the image excluding Border
(14) sum = 0
(15) for k = -b to b do // For a single pixel in the image
(16) for j = -a to a do
(17) sum = sum + w(k, j) f(x + k, y + j)
(18) end for
(19) end for
(20) end for
(21) end for
(22) g(x, y) = sum

End

ALGORITHM 4.2. Normalization of an image for display.

Begin

(1) Read all the pixel vales of input image with M rows and N columns where, f(x, y) represents the pixel value at x and y co-ordinate.
(2) Store all the pixel vales in an integer matrix of dimension M X N which is to be normalized for M rows and N columns.
(3) The output image is g(x, y) with M rows and N columns

(4) Calculate the minimum value for each column of the input matrix.
(5) Calculate the smallest value among all the minimum column values.
(6) Calculate the maximum value for each column of the input matrix.
(7) Calculate the largest value among all the maximum column values.
(8) Calculate range = largest value – smallest value
(9) for x = 1 to N do
(10) for y = 1 to M do
(11) g(y, x) = (f(y, x) – smallest pixel value) * 255 / range.
(12) end for
(13) end for
End

ALGORITHM 4.3. Thresholding an image
Begin
(1) Select a gray scale image with M rows and N columns where f(x, y) represents the pixel value at x and y coordinates.
(2) Store all the pixel values of the image in matrix form.
(3) Choose a value for the label.
(4) for x = 1 to N do
(5) for y = 1 to M do
(6) if f(y, x) is greater than level then
(7) f(y, x) = 255, it sets the point to white
(8) else
(9) f(y, x) = 0, it sets the point to black
(10) end
(11) end for
(12) end for
End

ALGORITHM 4.4. Edge detection by derivative operators.
Begin
(1) Select a gray scale input image.
(2) Store all the pixel values of the image along x and y coordinates in matrix form.
(3) Generate the convolution mask for different gradient operators and store it in different matrices.
(4) The sum of all the coefficients of each mask must be zero.
(5) Each mask along the horizontal and vertical direction is convolved with the input image.
(6) The magnitude of the gradient vector is obtained.
(7) Finally, the gradient vector is normalized and threshold to a particular level for display of edge map information.
End

Initially, we started with a 8-bit grayscale image of size 256 X 256 and the input image is processed using the different gradient first and second derivative operators like Sobel, Robert, Prewitt and Laplacian to find edge map [5, 7, 23, 24]. The proposed mask for horizontal and vertical direction is convolved to the input image and then the magnitude of the gradient vector is obtained [11, 17]. Finally it is normalized and threshold to find the edge map information. For execution of all the algorithms used in this paper, we have used MATLAB [19, 21].

5. Results

5.1. Output from Sobel Operator

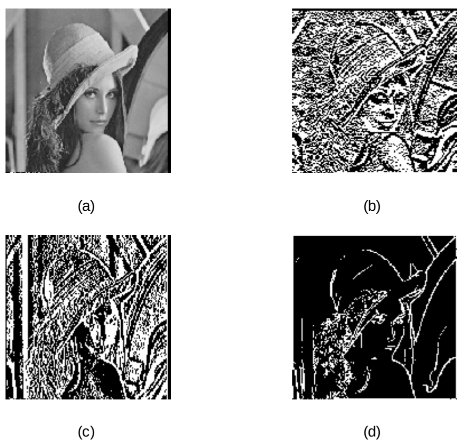

Figure 5: Results of Sobel Operator, (a) Original ‘Lena’ image (b) Horizontal component, (c)Vertical component (d) output image with edge map, (‘Lena’ image source by MathWorks Inc., USA (MATHLab))

5.2. Output from Robert Operator

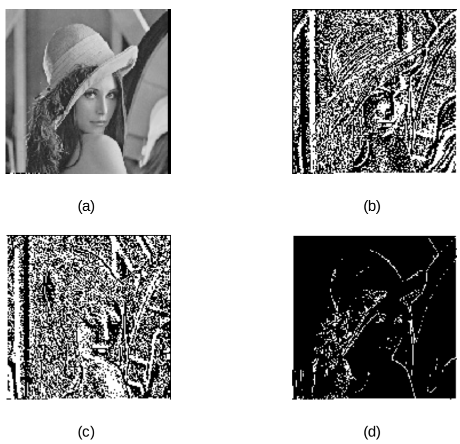

Figure 6: Results of Roberts Operator, Original ‘Lena’ image (b) Horizontal component, (c)Vertical component (d) output image with edge map.

5.3. Output from Prewitt Operator

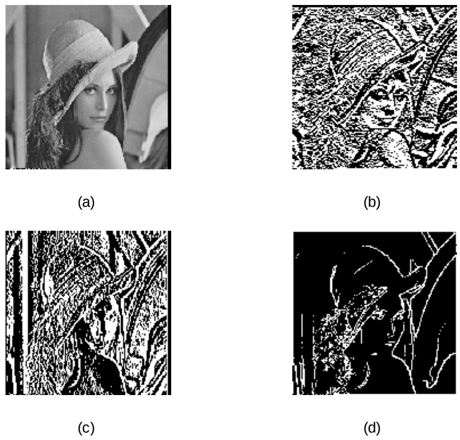

Figure 7: Results of Prewitt Operator, (a) Original 'Lena' image (b) Horizontal component, (c)Vertical component (d) output image with edge map.

5.4. Output from Laplacian Operator

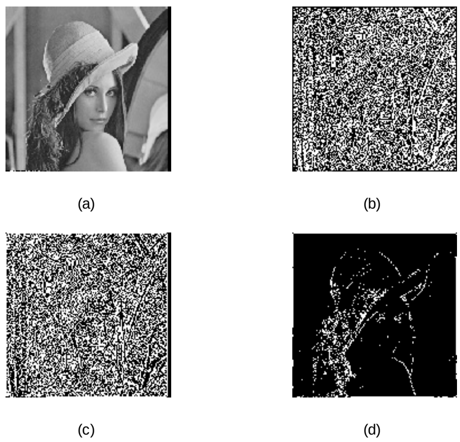

Figure 8: Results of Laplacian Operator, Original 'Lena' image (b) Horizontal component, (c)Vertical component (d) output image with edge map

5.5. Output from Laplacian of Gaussian Operator

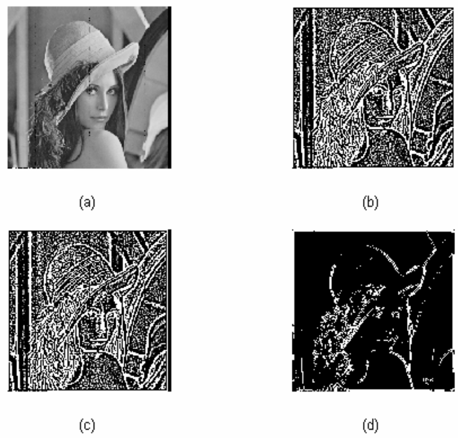

Figure 9: Results of Laplacian of Gaussian Operator, Original 'Lena' image (b) Horizontal component, (c)Vertical component (d) output image with edge map

5.6. Output from proposed Operator

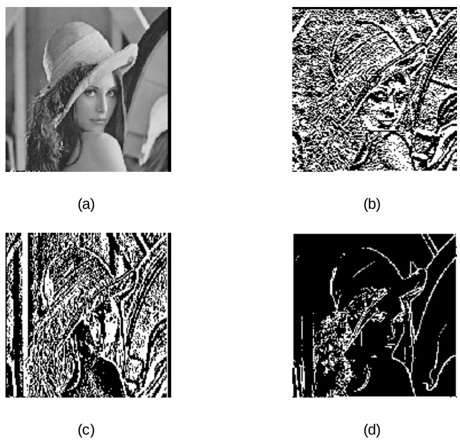

Figure 10: Results of Proposed Operator, Original 'Lena' image (b) Horizontal component, (c)Vertical component (d) output image with edge map.

5.7. Output from Fuzzy Gaussian Filter:

Figure 11: Edge Sharpening by Fuzzy Gaussian filter.

Table 1: Subjective Fidelity Scoring Scales

Quality	Comparison
A – Very good	+ 3 Very high
B - Good	+2 High
C - Fair	+1 Medium
D - Poor	- 1 Less
E - Bad	- 2 Much less

Table 2: Comparison of Different Gradient Operator.

Operators\Factors for Comparison	Contrast	Edge Map	Noise Content
Sobel Operator	B	B	+1
Roberts Operator	C	C	+1
Prewitt Operator	B	B	+1
Laplacian Operator	D	D	+2
Laplacian of Gaussian Operator	C	B	-1
Fuzzy Gaussian Filter	C	B	-1
Proposed Operator	B	B	+1

Table 3: Comparision of Root-Mean-Square Error.

Operators	e_{rms} after applying the operator
Sobel Operator	141.5792
Roberts Operator	134.6837
Prewitt Operator	140.2256
Laplacian Operator	138.2917
Proposed Operator	142.8385

6. Conclusion

In this paper, the proposed operator's performance for edge detection in a noisy image is evaluated both subjectively and objectively against the first and second order derivative filters and the results are shown in Fig. 5 to Fig. 10 and from Table 2 and Table 3 respectively.

The subjective evaluation of edge detected images show that proposed operator, Sobel and Prewitt operator exhibit better performances respectively. Table2 shows that Robert and Laplacian have poor performance in terms of contrast, edge map strength and noise content. Prewitt, Sobel and proposed operator have good contrast, edge map strength and low noise content then Laplacian of Gaussian. Prewitt is more acceptable than Roberts, Laplacian and Laplacian of Gaussian, while Sobel and proposed are more acceptable than Prewitt. It also shows that Laplacian is very much sensitive to noise. The objective evaluation of edge detection results as in Table 3 agree the subjective evaluation as in Table 2 that proposed, Sobel and Prewitt operators are better than Laplacian, Laplacian of Gaussian and Robert operator. The root mean square error of Laplacian and Robert is less than Prewitt, which is less than Sobel and proposed operator. The fuzzy Gaussian filter has a good contrast and sharpness which is required to sharpen an image. This paper concludes that the subjective and objective evaluation of edge map shows that proposed, Sobel, Prewitt, Laplacian of Gaussian, Roberts and Laplacian exhibit better performance for edge detection respectively and the results of the subjective evaluation matches with the results of the objective evaluation.

References

[1] E. Argyle. “Techniques for edge detection,” Proc. IEEE, vol. 59, pp. 285-286, 1971.

[2] H.Chidiac, D.Ziou, “Classification of Image Edges”,Vision Interface’99, Troise-Rivieres, Canada, 1999.pp. 17-24.

[3] Hueckel.,M., “ A local visual operator which recognizes edges and line”. J. ACM, vol. 20, no. 4, pp. 634-647, Oct. 1973.

[4] M.Heath, S. Sarkar, T. Sanocki, and K.W. Bowyer. “A Robust Visual Method for Assessing the Relative. Performance of Edge Detection Algorithms”. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19(12), pp. 1338-1359, Dec. 1997.

[5] M. Heath, S. Sarkar, T. Sanocki, and K.W. Bowyer. “Comparison of Edge Detectors: A Methodology and Initial Study “Computer Vision and Image Understanding, vol. 69, no. 1, pp. 38-54 Jan. 1998.

[6] M.C. Shin, D. Goldgof, and K.W. Bowyer. “Comparison of Edge Detector Performance through Use in an Object Recognition Task”. Computer Vision and Image Understanding, vol. 84, no. 1, pp. 160-178, Oct. 2001.

[7] T. Peli and D. Malah. “A Study of Edge Detection Algorithms”. Computer Graphics and Image Processing, vol. 20, pp. 1-21, 1982.

[8] R. C. Gonzalez, R. E. Woods, Digital Image Processing, 2nd ed., Upper Saddle River, New Jersey, Prentice-Hall, Inc., 2002.

[9] Pratt, W.K., Digital Image Processing, 4 th ed., Hoboken, New Jersey, John Wiley & Sons, Inc, 2007.

[10] F. Bergholm. “Edge focusing,” in Proc. 8th Int. Conf. Pattern Recognition, Paris, France, pp. 597- 600, 1986.

[11] Chanda, B., and Dutta Majumder, D. Digital Image Processing and Analysis, India, Prentice Hall of India, 2008.

[12] Sobel, I.E., “Camera Models and Machine Perception,” Ph.D. dissertation, Stanford University, Palo Alto, Calif, 1970.

[13] Roberts, L.G., Tippet, J.T., Machine Perception of Three-Dimensional Solids, Cambridge, Mass, MIT Press, 1965.
[14] Prewitt, J.M.S., Lipkin, B.S., and Rosenfeld, A. Object Enhancement and Extraction., New York, Academic Press, 1970.
[15] Chanda, B., Chaudhuri, B.B. and Dutta majumder, D., “A differentiation/ enhancement edge detector and its properties”, IEEE Trans. On System, Man and Cybern., SMC- 15:pp. 162-168, 1985.
[16] Marr, D.C. and Hildreth, E., “Theory of edge detection”, Proc. Royal Soc. Lond., vol. B, pp. 187-217, 1980.
[17] Hueckel, M., “An operator which locates edges in digitized pictures”, J. Assoc. Comput., vol. 18, pp. 113-125, 1971.
[18] Forsyth, D.A., and Ponce, J., A Modern Approach, India, Prentice Hall of India, 2003.
[19] Gilat, A., Matlab An Introduction with Applications, New York, John Wiley & Sons, Inc, 2004.
[20] Duda, R.O, Hart, P.E., Pattern Classification and Scene Analysis, New York, Wiley-Interscience, 2001.
[21] Image Processing Toolbox, User guide for use with Matlab, the MathWorks Inc., 2001.
[22] Cyganek, C., and Siebert, J.P., An Introduction to 3D Computer Vision Techniques and Algorithms, New York, John Wiley & Sons, Ltd, 2009.
[23] Ziou, D. and S. Tabbone, “Edge detection techniques an overview”. Int. J. Patt. Recog. Image Anal., vol. 8, pp. 537-559, 1998.
[24] Yakimovsky Y., "Boundary and object detection in real world image", Journal ACM, vol. 23, pp. 599-618, 1976.
[25] Davis, L. S., "Edge detection techniques", Computer Graphics Image Process., vol. 4, pp. 248-270, 1995.