ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 7, July 2013

Image Edge Detection: A Review

Sunanda Gupta **Charu Gupta** **S.K. Chakarvarti**

Abstract— **Edge detection is important part of image processing for object detection. So it becomes extremely important to have a good understanding of edge detection algorithms. An edge is the real or imagined line that marks the limit and divides of plane, object or appearance from other places or things. This means that if the edges in an image can be identified accurately, all of the objects can be located and basic properties can be measured. This paper introduces a classification of most important and commonly used edge detection algorithms, namely Sobel, Robert, Prewitt, Laplacian of Gaussian, Canny, Ant colony Optimization**.

Keywords—**Ant colony optimization, Canny, Edge Detection, Image Processing, Laplacian of Gaussian, Prewitt, Robert, Sobel.**

I. Introduction

Digital image is composed of a finite number of components, each of which has a special place or position and value. These components are cited to as picture elements, image elements, and pixels [1].

Image processing is any form of signal processing for which image is the input, such as a photograph and the image processing output may be whether an image or, a set of characteristics or parameters associated to the image [1]

Edge can also be defined as in binary images as the black pixels with one nearest white neighbour [2]. Edges include large amount of important information about the image. The changes in pixel intensity describe the boundaries of objects in a picture [2].

Feature detection and Feature extraction are the main areas of image processing, where Edge detection is used as a basic and important tool [2].

Image edge detection trades with drawing out of edges in an image by recognizing high intensity variations in the pixels. This action discovers outlines of an object and boundaries between objects and the back part of the image [2].

Detection of edges for an image may help in image segmentation, data compression, and also for image reconstruction and so on [2, 3]. Variables involved in selection of an edge detection operator include edge orientation, noise environment and edge structure [4, 5]. Edge detection is difficult in noisy images, since both the noise and the edges include high-frequency essence. Attempts to reduce the noise consequence are blurred and distorted edges [6].

Edge detection is used mainly to extract the information about the image e.g. image sharpening and enhancement, location of object present in the image, their shape, size. Depending upon variation of intensity / grey level various types of edges are shown in in Figure 1.

Charu Gupta, *ECE Department, Manav Rachna International University, , India.*

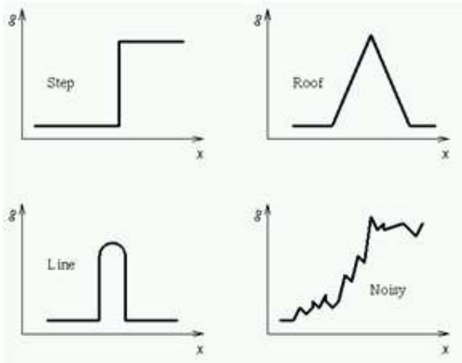

Fig.1 Typical edge profiles

Traditional methods of edge detection involves convolving the image with an operator, which is made to be sensitive to large gradients in the image while returning values of zero in uniform regions [7,8].

II. Image Edge Detection

We can define image edge detection as the discovery of the real or imagined line that marks the limit and divides of image appearance from other places or things in a digital image [2].

It is a series of actions whose purpose is to recognize points in an image where clear and defined changes occur in the intensity. This series of action is necessary to realize the meaning of the content of an image and has its applications in the evaluation of image and machine vision. The end usage of discovering clear and defined changes in image intensity is to represent the bottom line events and changes in the material properties of the world.

Causes of Intensity alteration normally represent two types of events: one is geometric events and other is non-geometric events.

A. Geometric Events

- Surface (boundary) discontinuities
- Discontinuities in depth
- Colour and Texture discontinuities

B. Non-Geometric Events

- Illumination changes
- Specularities
- Shadows
- Inter-Reflections[9]

II. Flow Chart for Edge Detection

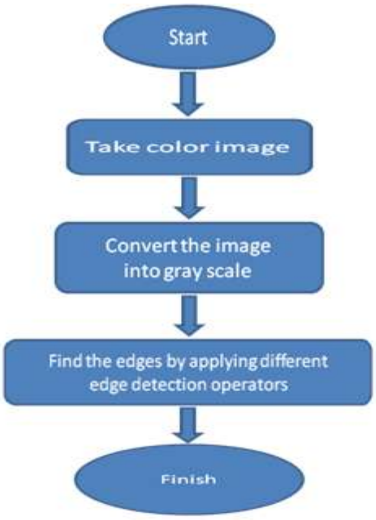

Fig. 2 Flow chart for edge detection

A. Algorithm for edge detection

Step 1-Take a color image.

Step 2-**Smoothing:** Annihilate as adequate noise as accessible, without wrecking genuine edges.

Step 3- **Enhancement:** the quality of edges is enhanced by applying differentiation.

Step 4- **Threshold**: Apply edge magnitude threshold to determine which edge pixels should be retained and which should be discarded as noise.

Step 5- **Localization**: Ascertain the postulate edge bearings.

Step 6- Evaluation with the algorithms.

Step 7- Get the image after edge disclosure.

III. Approaches Of Edge Detection

The course for edge detection is classified into two categories; first method is gradient based and second is laplacian based [10, 11].

In gradient based edges are detected by taking first order derivative. It calculates a measure of edge strength by computing the gradient magnitude, and then looking for local directional maxima of the gradient magnitude using a computed estimate of the local orientation of the edge, normally the gradient direction [10, 11].

In laplacian based methods, edges are found by searching for zero crossings in a second-order derivative expression computed from the image, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression. A smoothing stage, typically Gaussian smoothing, is applied as a pre-processing step to edge detection [10, 11].

A. Edge Detection Based on Gradient Operator

The gradient operators are also called as masks in digital images which calculate finite differential approximations of either horizontal or vertical directions [2].

The edge is the place where image grey value changes briskly, so to find out for the maximum and minimum values in the first derivative of the image [7, 10] of the gradient operator is most widely used [11]. First-order derivatives in image processing are implemented using the magnitude of the gradient. For a function f(x, y), the differential of 'f' at coordinates (x, y) is denoted [12] as the two dimensional column vector

$$\nabla f = G[f(x,y)] = \begin{bmatrix} Gx \\ Gy \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1)$$

The quantity ∇f is known as the gradient of a vector.

With the help of vector evaluation it can be observed that the gradient vector is directing in the direction of maximum rate of change at (x, y) coordinates. The vector sum of these two gradients is assumed to be taken as the magnitude of the gradient and the angle represents the gradient angle.

Magnitude of vector ∇f, denoted as M(x, y):

$$M(x,y) = magnitude(\nabla f) = |G| = \sqrt{Gx^2 + Gy^2} \quad (2)$$

To simplify computation, this quantity is approximated sometimes by omitting the square root operation

$$M(x,y) = Gx^2 + Gy^2 \quad (3)$$

Or by using absolute values,

$$M(x,y) \approx |Gx| + |Gy| \quad (4)$$

The direction of the gradient is given as:

$$\theta = \tan^{-1}\left(\frac{Gy}{Gx}\right) \quad (5)$$

Here the angle is measured with reference to x-axis. The direction of the edge at any point is perpendicular to the direction of the gradient at that point.

In a 2D image the [13] gradient is given as:

$$Gx = f(x+1, y) - f(x, y) \quad (6)$$

And

$$Gy = f(x, y+1) - f(x, y). \quad (7)$$

In this edge detection method the edges are assumed high gradient pixels. A derivative of intensity at some direction given by the angle of the gradient vector is beheld at edge pixels. Let Figure 3, denotes the intensities of image points in a 3x3 region. The center point z5 denotes f(x, y) at arbitrary location (x, y) [1].

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Fig. 3 Intensities of image points in a 3x3 region [1]

Here intensities z_1 denotes f(x-1, y-1), z_2 denotes (x-1, y), z_3 denotes (x-1, y+1), z_4 denotes (x, y-1), z_6 denotes (x, y+1), z_7 denotes (x+1, y-1), z_8 denotes (x+1, y), z_9 denotes (x+1, y+1) [1].

An edge pixel is determined using two crucial features [10, 14].

- The magnitude of the gradient which is equal to edge strength.
- The angle of the gradient which is equal to edge direction.

In the process step, we will learn gradient based Roberts edge detector, Sobel edge detector and Prewitt edge detector, Laplacian of Gaussian detector.

1)Robert Detector: The Roberts cross operator provides a simple proximity of 2 × 2 mask

$G_x =$

-1	0
0	1

$G_y =$

0	-1
1	0

Fig. 4 Convolution masks for Roberts operator [1]

The two masks can be applied separately on the image, results in separate analysis of the two gradient components Gx and Gy in the directions, perpendicular and parallel, is determined respectively [1]:

$$Gx = (z_9 - z_5)$$
$$Gy = (z_8 - z_6) \qquad (8)$$

Masks of even sizes are awkward to implement because they do not have a center of symmetry [1]. Further equation above can be written as given below

$$G[f(x,y)] = [f(x,y) - f(x+1,y+1)] + [f(x+1,y) - f(x,y+1)] \qquad (9)$$

The gradient magnitude is given by:

$$G[f(x,y)] = \sqrt{Gx^2 + Gy^2} \qquad (10)$$

The approximate magnitude is given by:

$$G[f(x,y)] = |Gx| + |Gy| \qquad (11)$$

Here Gx and Gy are calculated using the masks shown in Figure 4. The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \arctan(Gy / Gx) - 3\pi / 4 \qquad (12)$$

The differences are to be intended at the interpolated point [i + 1/2, j + 1/2]. The Roberts operator is a proximity to the ceaseless gradient at this interpolated point and not at the point [i, j] as might be apprehend [14, 15]. The smallest filter mask in which we are interested are of size 3x3.

2) Sobel Detector: The Sobel detector is one of the most frequently used in edge detection [16].

Sobel edge detection can be implemented by filtering an image with left mask or kernel. Filter the image again with the other mask. After this square of the pixels values of each filtered image. Now add the two results and compute their root. The 3 × 3convolution masks for the Sobel based operator as shown in Figure 5 [1]:

$Gx=$

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

$Gy=$

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

Fig. 5 convolution masks for the Sobel operator [1, 7]

The difference is taken between the 3rd and 1st rows of 3x3 image region is implemented by the left mask of figure 6 approximates the partial derivative in x-direction. The difference between the 3rd and 1st columns in the other mask approximates the derivative in y-direction [1].

Here the partial derivatives are to be intended by:

$$Gx = (z_7 + cz_8 + z_9) - (z_1 + cz_2 + z_3)$$
$$Gy = (z_3 + cz_6 + z_9) - (z_1 + cz_4 + z_7) \qquad (13)$$

With the constant c = 2. Further above equation can be written as given below

$$Gx = [f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)]$$
$$-[f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1)] \qquad (14)$$

And

$$Gy = [f(x-1, y+1)] + 2f(x, y+1) + f(x+1, y+1)]$$
$$-[f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1)] \qquad (15)$$

The magnitude of the gradient computed by:

$$G[f(x,y)] = (Gx^2 + Gy^2)^{1/2} \qquad (16)$$

The angle of orientation of the edge giving boost to the spatial gradient is given by:

$$\theta = \arctan(Gy / Gx) \qquad (17)$$

3) Prewitt Detector: The prewitt operator uses the same equations as the Sobel operator, other than the constant c =1. Therefore the convolution masks for the Prewitt operator shown in Figure 6:

0	1	1
-1	0	1
-1	-1	0

-1	-1	0
-1	0	1
0	1	1

1	1	1
0	0	0
-1	-1	-1

-1	0	1
-1	0	1
-1	-1	0

Fig. 6 Mask for Prewitt operators [7]

The Prewitt filter [17] is analogous to Sobel filter. Note that, contrary the Sobel operator, this operator does not place any emphasis on pixels that are closer to the centre of the masks [14].Classical operators are simple in which detection of edges & their orientation are possible but classical operators [18, 19] are sensitive to noise, and are inaccurate [20-26].

IV. Edge Detection Based On Laplacian Detection

To find edges the Laplacian method searches for zero crossings in the second derivative of the image.

The gradient operator as presented earlier is anisotropic, i.e., they are rotation invariant [13]. An isotropic operator is one which before and after the resultant image is having no effect on the image.

However, calculating 2nd derivative is very sensitive to noise. Before edge detection, this noise should be filtered out. To accomplish this, "Laplacian of Gaussian" is used [14].

A. Laplacian Of Gaussian

Laplacian of gaussian is also known as Marr-Hildreth Edge Detector. Laplacian of Gaussian function is referred to as LoG [27].

In this approach, firstly noise is reduced by convoluting the image with a Gaussian filter. After that isolated noise points and small structures are filtered out with smoothing. Those pixels, that have locally maximum gradient, are contemplated as edges by the edge detector in which zero crossings of the second derivative are used. Only the zero crossings, whose corresponding first derivative is above some threshold, are selected as edge point in order to avoid detection of insignificant edges. By using the direction in which zero crossing occurs we can obtain the edge direction [14, 28, 29].The LoG (`Laplacian of Gaussian')[30] kernel can be pre-calculated in advance so only one convolution needs to be performed at run-time on the image. The 2-D LoG function [31, 32] centred on zero and with Gaussian standard deviation σ has the form:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (18)$$

The three dimensional plot given by the LOG function [33] looks like Mexican hat as shown in Figure 7.

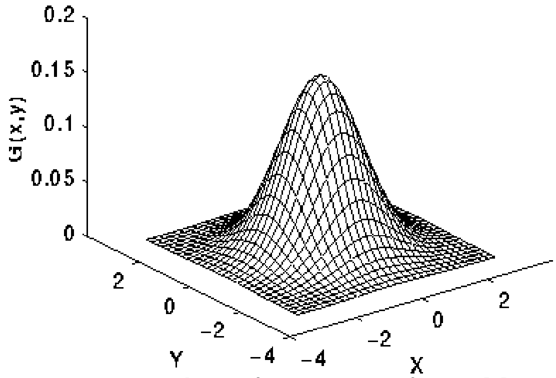

Fig. 7 2-D Laplacian of Gaussian (LoG) function [7]

Unlike the Sobel edge detector, the Laplacian edge detector uses only one mask. It can compute second order derivatives in a one pass. The mask used for it is shown in Figure 8.

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

Fig. 8 Three commonly used discrete approximations to the Laplacian filter [7]

You can use either one of these. Or if you want an improve proximation, you can construct a 5×5 mask (it has a 24 at the middle and everything else is -1).The LoG [27] kernel can be pre-calculated in advance so only one convolution needs to be performed at run-time on the image.

In LoG Detection of edges and their orientations are possible and we can test the wider area around the pixel which has fixed characteristics [20-26], in all directions so that it can find the correct places of edges but there is malfunctioning at corners, curves and where the grey level intensity function varies and because it uses the Laplacian filter, LoG is unable to find the orientation of edge[3,6] and is also sensitivity to noise because of which it does respond to some of the existing edges.

V. Edge Detection Based On The Optimum Detection

The maximum value of image is the gradient of the image edge. It means the inflection point of the gray image is the edge. From the mathematical frame of reference, inflection point of the second derivative of the function is equal to 0. After detecting this point, whose second derivative is 0 is a way of edge detection. Canny operator is the example of edge detection based on the optimum operator [34].

A. Edge Detection Using Canny Operator

The Canny edge detection algorithm is also known as the optimal edge detector [18, 35]. Canny's intentions were to appreciate the many edge detectors already out at the time he started his work [36]. Canny approach is based on three basic archetypes. First objective is low error rate. It is essential that edges appearing in images should not be missed and there should be no response to non-edges. The second archetype is that the edge points be well confined. In other words, the distance among the edge pixels as found by the detector and the actual edge is to be at a minimum. A third archetype is to have only one reaction to a single edge [18] [35].

Based on these archetype, the canny edge detector first smoothes image in order to eliminate noise. It then acquires the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these arena and suppresses any pixel that is not at the maximum. The gradient array is now reduced by hysteresis. Hysteresis is used to mark forward the remaining pixels that have not been restrained. Hysteresis uses two verges and if the magnitude is below the first verge, it is set to zero (made a non-edge). If the Beehive is above the high verge, it is made an edge. And if the beehive is between the 2 verges, then it is set to zero lest there is a path from this pixel to a pixel with a gradient above 2nd verge [18, 35].

The Canny Edge Detection Algorithm has the following steps:

Step1 Smooth the image with a two dimensional Gaussian filters. In most cases the computation of a two dimensional Gaussian is costly, so it is approximated by two one dimensional Gaussians, one in the x direction and the other in the y direction [33]. Filter out any noise in the original image before trying to locate and detect any edges.

Step2 Take the gradient of the image. This shows changes in intensity, which signifies the appearance of edges. This in fact gives two outcomes, the gradient in the x direction and the gradient in the y direction [33].

Step3: Direction of the edge is computed using the gradient in the x and y directions. Still, an error will be produced when sum X is set to zero. So in the code there has to be a limitation set when so ever this happens. The edge direction is set to 90 degree or 0 degree whenever the gradient in the x direction is set to 0, rely on what the value of the gradient in the y-direction is equal to. The edge direction will be equal to 0 degrees, if Q[i, j] has a value of zero. Else it is set to 90 degrees [35].

$$\theta = \tan^{-1}\left(\frac{Q[i,j]}{P[i,j]}\right) \qquad (19)$$

Step4: Once the edge direction is recognized, the coming step is to agnate the edge direction to a direction that can be marked in an image. So if the pixels of a 5x5 image are arranged as follows

x x x x x
x x x x x
x x a x x
x x x x x
x x x x x

It can be seen by looking at pixel "a", there are only four possible directions when describing the surrounding pixels - 0 degrees (in the horizontal direction), 45 degrees (adjacent the actual diagonal), 90 degrees (in the vertical direction), or 135 degrees (adjacent the negative diagonal). So now the edge position has to be adjudicate into one of these four directions depending on which direction it is closest to (e.g.

if the orientation angle is found to be 3 degrees, adjust it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions [35].

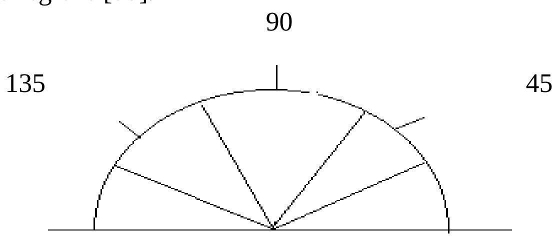

Figure 9 Edge directions [35]

Therefore, any edge direction falling within the range (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction lies in the range (22.5 to 67.5 degrees) is equal to 45 degrees. Any edge direction lies in the range (67.5 to 112.5 degrees) is equal to 90 degrees. And lastly, any edge direction lies within the range (112.5 to 157.5 degrees) is set to 135 degrees [35].

Step5:Non-Maxima Suppression

Non-maxima suppression traces along the edge direction and suppresses any pixel value not considered to be an edge and give a thin line for the edge. This will give a thin line in the output image [35].

Step6: Thresholding

The method of thresholding used by the Canny Edge Detector is referred to as "hysteresis" [18, 37]. Hysteresis thresholding is used to eliminate streaking. It makes use of both a high threshold and a low verge. If a pixel has a value above the high verge, it is equal to an edge pixel. If a pixel has a value above the low threshold and is the neighbour of an edge pixel, it is equal to an edge pixel as well. If a pixel has a value above the low verge but is not the neighbour of an edge pixel, it is not equal to an edge pixel. If a pixel has a value below the low verge, it is never equal to an edge pixel [33, 38].

The Canny operator [20-26] uses probability to find error rate so that it can provide localization and response to improve signal to noise ratio which gives better detection especially in noise conditions[18, 19] but Canny is having complex computations which shows false zero crossing and is time consuming.

VI. Edge Detection Based On Ant Colony Optimization

Ant colony optimization is a nature-influenced optimization algorithm, which is motivated by the natural phenomenon that ants deposit pheromone on the ground in order to mark or find some favorable path [11].

Ant algorithms are inspired by observing the real ant colonies. Ants are the amusing insects i.e. insects that live in colonies and whose behavior is to the survival of the colony as a whole than to that of directed more a single individual component of the colony. These social insects have captured the attention of many scientists because of their colonies can achieve the high structuring level, especially when they are compared to the relative simplicity of the colony's individuals. Foraging behaviour of ants is an important and interesting behaviour of ant colonies, and, in particular, how ants can find the shortest paths between food sources and their nest.

A substance known as pheromone is deposited by ants on the ground while walking from food sources to the nest and vice versa. In this way ants form a pheromone trail. Ants can odour pheromone and, when exercising their way, they tend to exercise, in anticipation, aisles marked by strong concentrations of pheromone. The pheromone trail allows the ants to find their way back to the food source or to the nest. It can also be used by the other ants to find the location of the food sources found by their nest mates [39].

Acknowledgments

I would like to express my sincere gratitude to my project guide "Sunanda Gupta" for giving me the opportunity to work on this topic.

I would like to give deference to Prof S.K. Chakarvarti of Manav Rachna International University for their generosity towards me and helpful to me in every possible way.

I am especially grateful to my parents and my brother for their constant support, love and keeping their faith in me, helped me to grow as a personality and making me believe that I can do achieve anything in life with hard work and strong determination.

References

[1] Rafael C. Gonzalez, R.E. Woods, *Digital Image Processing*, third edition, pp. 700-702, 2008.

[2] A. K. Jain, *Fundamentals of digital image processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[3] H. Voorhees and T. Poggio," Detecting textons and texture boundries in natural images" ICCV 87:250-25, 1987.

[4] S. Ullman, "The Interpretation of Visual Motion". Cambridge, MA: M.I.T. Press, 1979.

[5] A. Huertas and G. Medioni, "Detection of intensity changes with sub-pixel accuracy using Laplacian-Gaussian masks" *IEEE Trans. on Pattern Analysis and Machine Intelligence*. PAMI-8(5):651-664, 1986.

[6] S. Selvarajan and W. C. Tat, "Extraction of man-made features from remote sensing imageries by data fusion techniques" *22nd Asian conference on Remote Sensing*, 5-9 Nov. 2001, Singapore.

[7] Raman Maini & Dr. Himanshu Aggarwal. "Study and Comparison of Various Image Edge Detection Techniques". *International Journal of Image Processing (IJIP),* Volume (3): Issue (1).

[8] E. Argyle. "Techniques for edge detection,*" Proc. IEEE,* vol. 59, pp. 285-286, 1971.

[9] Anna Veronica. Baterina, Carlos M. Oppus,"Image Edge Detection Using Ant Colony Optimization" Department Of Electronics And Communications Engineering.

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

[11] Shweta Agarwal M.Tech Scholar. "A Review Paper of Edge Detection Using Ant Colony Optimization Techniques." *International Journal of Latest Research in Science and Technology,* Vol.1, Issue 2: Page No.120-123 July ñ August (2012).

[12] R. C. Gonzalez and R. E. Woods, Digital Image Processing. Reading, MA: Addison-Wesley, 1992.

[13] T. Acharya and A. K. Ray, "IMAGE PROCESSING Principles and Applications", Wiley Interscience, 2005.

[14] Gradient Based Edge Detection,[Online] Available: euclid.ii.metu.edu.tr/~ion528/demo/lectures/6/2/index.html.

[15] L. G. Roberts, " Machine perception of 3-D solids" ser. *Optical and Electo-Optical Information Processing,* MIT Press, 1965.

[16] J. Matthews, "An introduction to edge detection: The sobel edge detector,"[Online]Available:www.generation5.org/content/2002/im01.asp.

[17] R. C. Gonzalez and R. E. Woods. "*Digital Image Processing*". 2nd ed. Prentice Hall, 2002.

[18] Rajwinder Kaur, Monika Verma, Kalpna, Harish Kundra, "Classification of various edge detectors", Department of Computer Science, RIEIT, Railmajra, 2008.

[19] D. Marr and E. Hildreth. "Theory of Edge Detection ", *Proceedings of the Royal Society of London.* Series B, Biological Sciences., vol. 207, no. 1167.(29 Feb. 1980), pp. 187-217.
[20] T. Peli and D. Malah. " A Study of Edge Detection Algorithms", Computer Graphics and Image Processing, vol. 20, pp. 1-21, 1982.
[21] M. C. Shin, D. Goldgof, and K. W. Bowyer "Comparison of Edge Detectors Performance through Use in an Object Recognition Task". Computer Vision and Image Understanding, vol. 84, no. 1, pp. 160-178 Oct. 2001.
[22] 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.
[23] 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, pp. 12, pp. 1338-1359, Dec. 1997.
[24] A. Yuille and T. A. Poggio. "Scaling theorems for zero crossings". *IEEE Trans. Pattern Anal. Machine Intelligence,* vol. PAMI-8, no. 1, pp. 187-163, Jan. 1986.
[25] Y. Yakimovsky, "Boundary and object detection in real world images". JACM, vol. 23, no. 4, pp. 598-619, Oct. 1976.
[26] M. H. Hueckel. " A local visual operator which recognizes edges and line", JACM, vol. 20, no. 4, pp. 634-647, Oct. 1973.
[27] V. Torre and T. A. Poggio,"On edge detection", *IEEE Trans. Pattern Analysis Machine Intell.,* vol. PAMI-8, no. 2, pp. 187-163, Mar. 1986.
[28] W. Frei and C.C. Chen. "Fast boundary detection: A generalization and a new algorithm "*IEEE Trans. Pattern Comput.,* vol. C-26, no. 10, pp. 988-998, 1977.
[29] R. M. Haralick. "Digital step edges from zero crossing of the second directional derivatives, "*IEEE Trans. Pattern Anal. Machine Intelligence*, vol. PAMI-6, no. 1, pp. 58-68, Jan. 1984.
[30] D. C. Marr and T. Poggio, "A theory of human stereo vision," *Proc.Roy. Soc. London.,* vol. B 204, pp. 301-328, 1979.
[31] K. A. Stevens, "Surface perception from local analysis of texture and contour," M.I.T. Artificial Intell. Lab., Cambridge, MA, Rep. Al- TR-512, 1980.
[32] E. R. Davies, "Constraints on the design of template masks for edge detection", *Pattern Recognition Lett.,* vol. 4, pp. 11 1-120, Apr. 1986.
[33] Ehsan Nadernejad, Sara Sharifzadeh, Hamid Hassanpour. "Edge Detection Techniques: Evaluations and Comparisons", Applied Mathematical Sciences, Vol. 2, 2008, no. 31, 1507 – 1520, 2008.
[34] X. Zhuang, "Edge Feature Extraction in Digital Images with the Ant Colony System", *in proc. of the IEEE international Conference an computational intelligence for Measurement Systems and Applications*, pp. 133-136,2004.
[35] Amit Verma, Iqbaldeep Kaur, Parvinder S. Sandhu, Amandeep Kaur and Gagandeep Jindal, "Edge Detection and Optimization Algorithm for 2-D Images", *International Journal of Research in Engineering and Technology (IJRET)* Vol. 1 No. 1.
[36] J. F. Canny, "A computational approach to edge detection". *IEEE Trans. Pattern Anal. Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-697, 1986.
[37] J. Canny, "Finding edges and lines in image". Master's thesis, MIT, 1983.
[38] S. Price, "Edges: The Canny Edge Detector", July 4, 1996. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MARBLE/low/edges/canny.htm.
[39] Marco Dorigo and Gianni Di Caro and Luca M. Gambardella. "Ant Algorithms for Discrete Optimization", MIT Press, 1999.

Author Profile

I, Charu Gupta is pursuing M. Tech. from Manav Rachna International University (MIRU), Faridabad. I have received my B. Tech. degree in ECE from Al-Falah School of Engineering and Technology, Faridabad (Affiliated to MDU, Rohtak) in 2011. I have published my paper on "Edge Detection of an Image based on Ant Colony Optimization Technique" in International Journal of Science and Research (IJSR), Volume 2 Issue 6, June 2013.

Sunanda Gupta is pursuing her Ph. D from MRIU, Faridabad. She received M.Tech. Degree in ICE from Apeejay College of Engineering Sohna Gurgaon (Affiliated to M.D.U Rohtak) in 2000. AMIE Degree in ECE from The Institution of Engineers (India) in 2000. She is a life member of ISTE & member of IEI (India).

Prof SK Chakarvarti is Director Research & Development at Manav Rachna International University, Faridabad.
Dean Academic, controller of Examination, Professor and Chairman Department of Physics at National Institute of Technology, Kurukshetra.