# *Result Analysis Of Different Image Edges By Applying Existing And New Techniques*

Kalyan Kumar Jena

Department Of Computer Science Engineering & Application
Indira Gandhi Institute Of Technology
Sarang,Odisha,India

***Abstract*— Edges of an image are considered a type of crucial information that can be extracted by applying detectors with different methodology. Edge detection is a basic and important subject in computer vision and image processing. Here a new approach to edge detection using fuzzy method is presented. The proposed method works better for low contrast images. Fuzzy Image processing is a widely used technique used in image processing and often gives better result in many such occasions as compared to the conventional image processing. Gray-scale images can be processed based on fuzzy topology and color images can be processed by regarding the color image as multi-dimensional gray-scale images. In this paper existing and new construction methods for interval-valued fuzzy relations from fuzzy relation is presented. This construction method is based on the concepts of triangular norm . We analyze the effect of using different *triangular*-norms . Again we examine the result of some traditional edge detection methods. Finally, We apply the construction method to image processing and compare the result of our approach with those obtained by means of other.**

***Keywords:Concept of image processing, Fuzzy method, Sobel Operator, Prewitt Operator, canny operator, Robert operator and lower constructor with laplacian operator.***

## I. INTRODUCTION

Contours of image of object or in other words , edges in context of image processing and computer vision provide valuable information towards human image understanding .It probably most important processing step in human picture recognition system consist of edge detection process naturally , edge detection has became serious challenge to image processing scientist .How does edge comes in an image? The answer to this question probably , provide the early clue for locating edge in an image .the variation of image feature ,usually brightness give rise to edges. objectively the edges are representation of the discontinuities of image intensity function .There could be various reason ,such as lighting conditions ,object geometry , type of material surface texture etc , as well as there mutual interaction , for discontinuities.

Therefore, edge detection algorithm is essentially a process of detection of this discontinues in an image. Since the abrupt change in brightness level indicates edge, its detection in binary or segmented image is quite straightforward. However the process of edge localization is quite complex in case of gray level or intensity image .the transition in intensity in gray scale image is relatively smooth in nature rather than abrupt in case of segmented image or binary image .The nature of intensity variation points to application of derivative operators for detecting edges , Application of derivative operator on intensity image produce another image ,usually called gradient image as it reveals the rate of intensity variation .The image is then made to undergo thresholding and /or edge linking in order to yield contours .

There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories, Gradient and Laplacian. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. The Laplacian method searches for zero crossings in the second derivative of the image to find edges. An edge has the one-dimensional shape of a ramp and calculating the derivative of the image can highlight its location.

## II. THE PRINCIPLE OF EDGE DETECTION

The goal of edge detection is to mark the points in a digital image at which the luminous intensity changes sharply. Sharp changes in image properties usually reflect important events and changes in properties of the world. These include discontinuities in depth, discontinuities in surface orientation, changes in material properties and Variations in scene illumination. Edge detection is a research field within image processing and computer vision, in particular within the area of feature extraction.

Edge detection of an image reduces significantly the amount of data and filters out information that may be regarded as less relevant, preserving the important structural properties of an image. There are many methods for edge detection, but most of them can be grouped into two categories:

- Derivative Approach – Edge pixel or edge elements are detected by taking derivative followed by thresholding (e.g. Robert operator and 4 neighbor operator) they occasionally incorporate noise

cleaning scheme(e.g. Prewit operator Sobel operator ) two dimensional derivative are computed by means of what we call is edge mask.

- Pattern Fitting Approach - A series of edge approximating functions in the form of edge template over a small neighborhood are analyzed. Parameters along with there properties corresponding to best fitting function are determined. Based on this information whether or not edge is present, is decided. We also call them edge filter.

Both the approaches have advantage and disadvantage. However it is our common experience that the second approach gives better result compared to derivative approach the reason could be followed from the primary aim of latter which is

(i) To detect derivative of image intensity and at the same time
(ii) To make the process robust to noise.

### A. *Edge Properties*

Edges may be viewpoint dependent - these are edges that may change as the viewpoint changes, and typically reflect the geometry of the scene, objects occluding one another and so on, or may be viewpoint independent - these generally reflect properties of the viewed objects such as surface markings and surface shape. In two dimensions, and higher, the concept of perspective projection has to be considered.

A typical edge might be (for instance) the border between a block of red color and a block of yellow; in contrast a line can be a small number of pixels of a different color on an otherwise unchanging background. There will be one edge on each side of the line. Edges play quite an important role in many applications of image processing. During recent years, however, substantial (and successful) research has also been made on computer vision methods that do not explicitly rely on edge detection as a pre-processing.

### B. *Detecting An Edge*

Taking an edge to be a change in intensity taking place over a number of pixels, edge detection algorithms generally compute a derivative of this intensity change. To simplify matters, we can consider the detection of an edge in one dimension. In this instance, our data can be a single line of pixel intensities. For instance, we can intuitively say that there should be an edge between the 4th and 5th pixels in the following 1-dimensional data:

| 5 | 7 | 6 | 4 | 152 | 148 | 149 |
|---|---|---|---|---|---|---|

To firmly state a specific threshold on how large the intensity change between two neighboring pixels must be for us to say that there should be an edge between these pixels is, however, not always an easy problem. Indeed, this is one of the reasons why edge detection may be a non-trivial problem unless the objects in the scene are particularly simple and the illumination conditions can be well controlled.

### C. *Approaches To Edge Detection*

There are many methods for edge detection[10], but most of them can be grouped into two categories, search-based and zero-crossing based. The search-based methods detect edges by first computing a measure of edge strength, usually a first-order derivative expression such as the gradient magnitude, and then searching for local directional maxima of the gradient magnitude using a computed estimate of the local orientation of the edge, usually the gradient direction. The zero-crossing based methods search for zero crossings in a second-order derivative expression computed from the image in order to find edges, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression, as will be described in the section on differential edge detection following below. As a pre-processing step to edge detection, a smoothing stage, typically Gaussian smoothing[7], is almost always applied.

The edge detection methods that have been published mainly differ in the types of smoothing filters that are applied and the way the measures of edge strength are computed[3]. As many edge detection methods rely on the computation of image gradients, they also differ in the types of filters used for computing gradient estimates in the x- and y-directions.

## III. SOBEL EDGE DETECTOR

The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image.

In theory at least, the operator consists of a pair of 3×3 convolution kernels as shown in Figure 1. One kernel is simply the other rotated by 90°. This is very similar to the Roberts Cross operator.

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

Gx

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

Gy

Figure 1.Sobel Operator Mask

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in

each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$\mathbf{G} = \sqrt{G_x^{\,2} + G_y^{\,2}}$$

Typically, an approximate magnitude is computed using:

$$\mathbf{G} = |\mathbf{G}_x| + |\mathbf{G}_y|$$

This is much faster to compute.

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

In this case, orientation $\theta$ is taken to mean that the direction of maximum contrast from black to white runs from left to right on the image, and other angles are measured anti-clockwise from this.

Often, this absolute magnitude is the only output the user sees --- the two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator shown in Figure 2.

| | | |
|---|---|---|
| P1 | P2 | P3 |
| P4 | P5 | P6 |
| P7 | P8 | P9 |

Figure 2. Pseudo-convolution kernels used to quickly compute approximate gradient magnitude

## IV. PREWITT EDGE DETECTOR

The Prewitt operator is used in image processing, particularly within edge detection algorithms. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Prewitt operator is either the corresponding gradient vector or the norm of this vector. The Prewitt operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation which it produces is relatively crude, in particular for high frequency variations in the image. The Prewitt operator is named for Judith Prewitt.

Mathematically, the operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical. If we define $\mathbf{A}$ as the source image, and **Gx** and **Gy** are two images which at each point contain the horizontal and vertical derivative approximations, the latter are computed as:

$$\mathbf{G}_X = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G}_Y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * \mathbf{A}$$

.

Since the Prewitt kernels can be decomposed as the products of an averaging and a differentiation kernel, they compute the gradient with smoothing. For example, **Gx** can be written as

$$\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} [-1 \quad 0 \quad 1]$$

The *x*-coordinate is defined here as increasing in the "right"-direction, and the *y*-coordinate is defined as increasing in the "down"-direction. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using:

$$\sqrt{Gx^2 + Gx^2}$$

Using this information, we can also calculate the gradient's direction:

$$\mathbf{\Theta} = \operatorname{atan2}(\mathbf{G}_y, \mathbf{G}_x)$$

where, for example, **Θ** is 0 for a vertical edge which is darker on the right side.

## V. ROBERT EDGE DETECTOR

It can also detect the edges by using the following methodology:

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

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

Figure 3. Robert Operator Mask

$d_1 = g_o - g_1$ $d_2 = g_o - g_3$

The differential along diagonal of a 2 X 2 mask is used and edge value after the convolution correspond to central point $(\frac{r-1}{2}, \frac{c-1}{2})$.

## VI. CANNY EDGE DETECTOR

The Canny operator was designed to be an optimal edge detector according to particular criteria.The operator works in a multi-stage process. First of all the image is smoothed by Gaussian convolution.Then a simple 2-D first derivative operator isapplied to the smoothed image to highlightregions of the image with high first spatial derivatives.Edges give rise to ridges in the gradient magnitude image.

Step 1

- Noise is filtered out – usually a Gaussian filter is used
- Width is chosen carefully

Step 2

- Edge strength is found out by taking the gradient of the image
- A Roberts mask or a Sobel mask can be used

Step 3

- Find the edge direction

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

Step 4

- Resolve edge direction

Step 5

- Non-maxima suppression – trace along the edge direction and suppress any pixel value not considered to be an edge. Gives a thin line of edge

Step 6

- Use double / hysterisis thresholding to eliminate streaking

## VII. MIN CONSTRUCTOR WITH LAPLACIAN EDGE DETECTOR

### A. Min Construction Method

The *lower constructor* is a generalization of the tn-processing

A t-norm T: $[0, 1]^2 \rightarrow [0, 1]$ is an associative, commutative, increasing function, such that, T(1, x) = x for all $x \in [0, 1]$. At-norm T is called idempotent if T(x, x) = x for all $x \in [0, 1]$.

The four basic *t*-norms are as follows

1. The minimum $T_M(x,y)=\min(x,y)$.
2. The product $T_P(x, y) = x \cdot y$.
3. The Łukasiewicz t-norm
   $T_L(x, y) = \max(x + y - 1, 0)$.
4. The nilpotent minimum t-norm
5. $T_{nM}(x, y) = \min(x, y)$, *if* $x + y > 1$
   $0$, otherwise.

- Let $R \in F(X \times Y)$ be an FR. Consider two t-norms T1 and T2 and two values $n,m \in N$ so that $n \leq P - 1/2$ , and $m \leq Q - 1/2$ .We define the lower constructor associated with T1 , T2 , n, and m in the following way:
- $L^{n,m}T1 ,T2 : F(X \times Y) \rightarrow$ given by

$$L^{n,m}T1 ,T2 [R](x, y) = \underset{i=-n}{\overset{}{T_1^{m,n}}}\underset{J=-m}{\overset{}{(T_2 (R(x - i, y - j),R(x,y)))}}$$

for all $(x,y) \in (X,Y)$

The Algorithm begins with reading an M x N image. The first set of nine pixels of a 3x3 window are chosen with central pixel having values (2,2) i.e for each pizel(i,j) we are taking the 8 neighbourhood of (i,j).After the initialization, the pixel values are initially marked as edge pixel after an observation to the 8 neighbourhood. After the subjection of the pixel values the algorithm generates an intermediate image using a construction method stated below. It is checked whether all pixels have been checked or now, if not then first the horizontal coordinate pixels are checked. If all horizontal pixels have been checked the vertical pixels are checked else the horizontal pixel is incremented to retrieve the next set of pixels of a window. In this manner the window shifts and checks all the pixels in one horizontal line then increments to check the next vertical location.

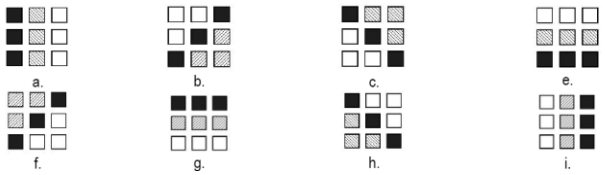



Figure 4. Initial conditions to check 8 neighborhood

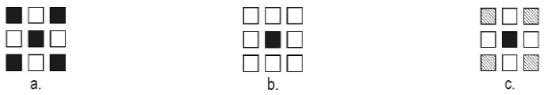



Figure 5. (a,b)Type of unwanted edge pixels (c) condition for removal of unwanted edge pixels.

Pixel checked

Edge pixel

Unchecked pixel

After edge highlighting image is subjected to another set of condition with the help of which the unwanted parts of the output image of type shown in Fig.(a-b) are removed to generate an image which contains only the edges associated with the input image. Let us now consider the case of the

fuzzy condition displayed in Fig. (g). For an input image A and an output image B of size M x N pixels respectively we have the following set of conditions that are implemented to detect the edges pixel values.

Input*:* An image A of M x N pixels (Phase 1)
Output*:* An image B of M x N pixels
Initial Edge Detection (A, B)using Min Construction
For I←2 to M-1
  For J←2to N-1
    If A (I-1, J)>A (I-1, J+1)
      Then If A (I-1, J-1)>A (I, J)
        Then If A (I, J-1)>A (I+1, J-1)
          Then
            B (I-1, J+1) ←0
            B (I, J) ←0
            B (I+1, J-1) ←0
  End For
End For
For I←2 to M-1
  For J←2to N-1
    If B(I-1,J)=255& B(I,J)=0& B(I+1,J)=255& B(I,J-1)=255
     Then B (I, J) is minimum and highlighted as edge initially.
  End For
End For

In the above algorithm Min construction[1] is used but not after fuzzification as after fuzzification the membership values would become fractions that can't be stored in unsigned char. Hence the same technique of min construction is used but on true picture and taking into consideration 8-nbd of a pixel (i,j). We can observe in the above algorithm written for a particular fuzzy condition that the nesting of statements is done in a manner that only the edge associated pixels are granted black pixel values and initially min valued edge pixels are given white value. These pixels are initially marked as edge.

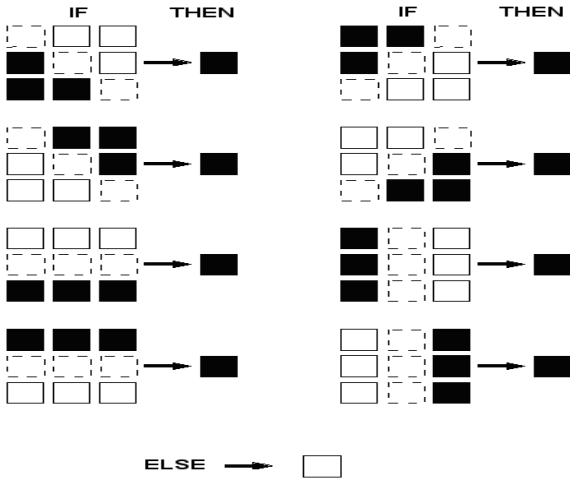
Phase 2. Input: An image B (256 color true bmp image) of size MxN
Output: Edge image of size MxN
We now use Laplacian of Gaussian (log) operator[7] on the intermediate image to get the edge image. And In this way whatever image is being constructed is compared with edges found on same image by other existing techniques.

LOG operator is

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

which is stored in a 5x5 array.

The phase 1 actually performs a check like




Black pixels are having min values.

## VIII. EXPEIMENTAL RESULTS

### *A. Output Based On Sobel Operator*

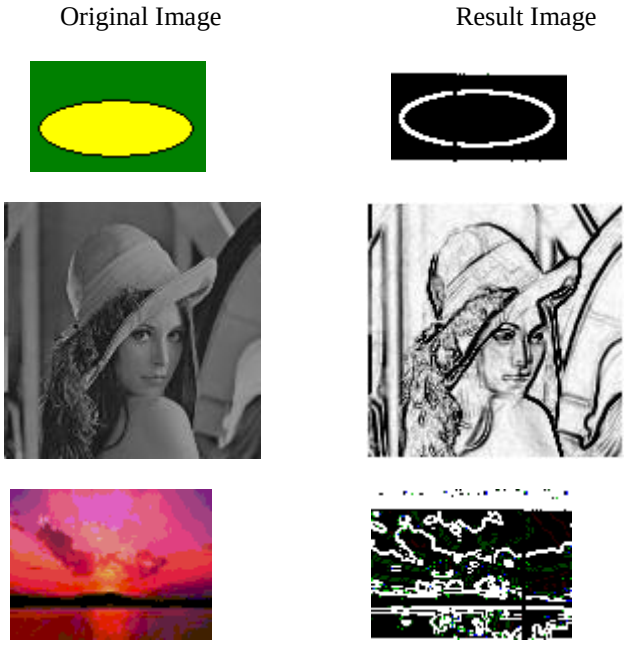



Figure 6. Result Using Sobel Operator

### B. Output Based On Prewitt Operator

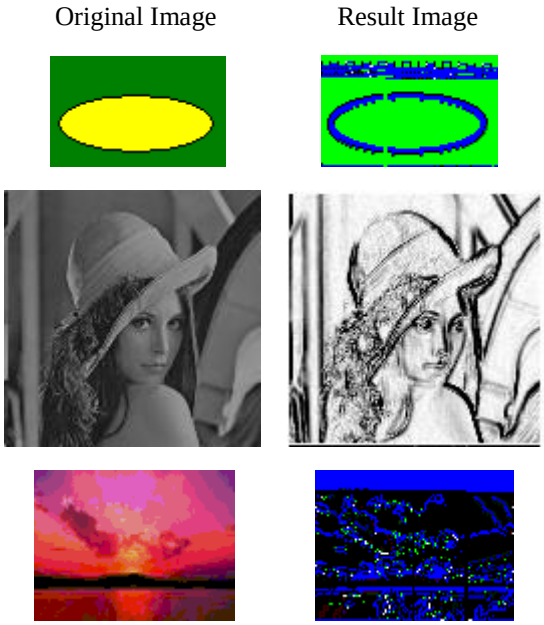



Figure 7. Result Using Prewitt Operator

### C. Output Based On Canny Operator

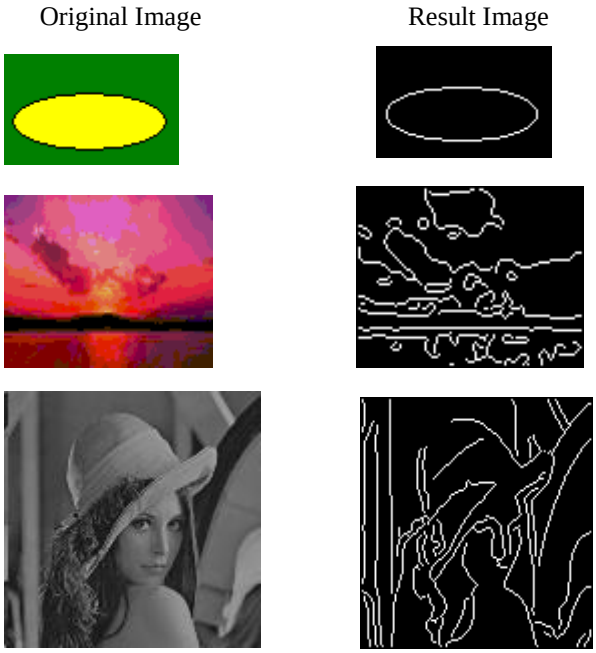



Figure 8. Result Using Canny Operator

### D. Output Based On Robert Operator


Figure 9. Result Using Robet Operator

### E. Output Based On Min Laplacian Edge ConstructorMethod

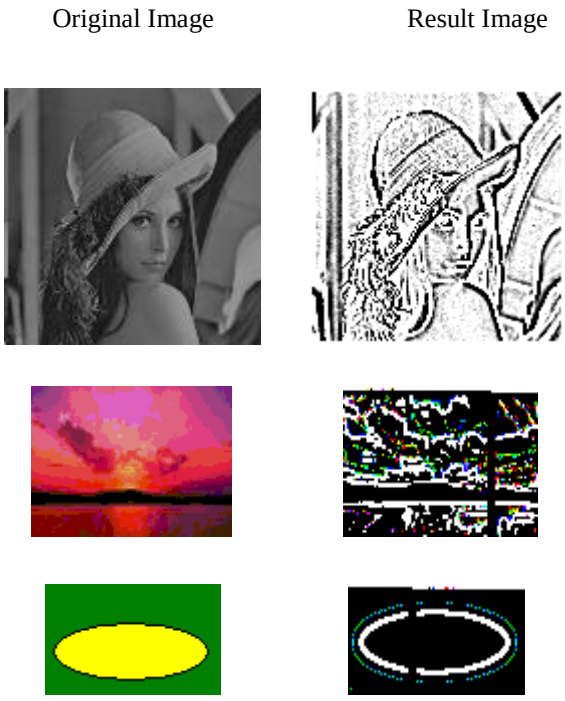



Figure 10. Result Using Min Laplacian Edge Constructor Method

## IX. CONCLUSION

In this paper, the algorithm to find the edges associated with an image had been implemented.Comparison were made amongst the various edge detection algorithms. Sobel operator works better for brighter images .It can be affected by noise and it is slower than Prewitt operator.Prewitt operator works better for brighter images.It is faster than Sobel operator and it is also affected by noise.Canny operator works better for brighter Gray images.It requires better localization of images.It cannot provide better result in the width of the image increases.Robert operator works better for brighter images.It provides better result for small images.These four operator provides better result for Brighter images or high constract images and these operators can be affected by noise.But these operators can not provide better result for darker images. The new concept lower constructor with laplacian operator provides better result for darker images compared to the previous operators and this operator also less affected by noise.

## REFERENCES

[1] Edurne Barrenechea, Humberto Bustince, Member, IEEE, Bernard De Baets,And Carlos Lopez-Molina, Student Member, IEEE, "Construction Of Interval-Valued Fuzzy Relations With Application To The Generation Of Fuzzy Edge Images", IEEE Transactions On Fuzzy Systems, Vol. 19, No. 5, October 2011, pp. 819-830.

[2] Debashis Sen and Sankar K. Pal, "Gradient Histogram: Thresholding In A Region Of Interest For Edge Detection", ELSEVIER Image And Vision Computing 28 ,2010, pp.677–695.

[3] Jinbo Wu, Zhouping Yin and Youlun Xion, "The Fast Multilevel Fuzzy Edge Detection Of Blurry Images", IEEE Signal Processing Letters, Vol. 14, No. 5, May 2008, pp.344-347.

[4] Bo Lia,, Aleksandar Jevtic and Ulrik Söderström, " Fast Edge Detection By Center Of Mass**"**, IEEE Intelligent Systems And Image Processing 2013.

[5] Mitra Basu, Senior Member, IEEE, "Gaussian-Based Edge-Detection Methods", IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 32, No. 3, August 2008.

[6] Z.J. Hou and G.W. Wei, "A New Approach To Edge Detection", ELSIVIER Pattern Recognition 35 ,2009.

[7] Abdallah A. Alshennawy, and Ayman A. Aly, "Edge Detection Indigital Images Using Fuzzy Logic Technique", World Academy Of Science Engineering And Technology 51, 2009. Pp. 178-186.

[8] Begol, Moslem and Maghooli,Keivan, "Improving Digital Image Edge Detection By Fuzzy Systems", World Academy Of Science, Engineering And Technology 81 , 2011.

[9] Wafa Barkhoda, Fardin Akhlaqian Tab, and Om-Kolsoom Shahryari , "Fuzzy Edge Detection Based On Pixel's Gradient And Standard Deviation Values", Computer Science And Information Technology, 2009. IMCSIT09.

[10] Abdallah A. Alshennawy and Ayman A. Aly, "Edge Detection In Digital Images Using Fuzzy Logic Technique", World Academy Of Science, Engineering And Technology Vol:27 2009-03-24.

[11] M. A. Nikouei Mahani, M. Koohi Moghadam and H. Nezamabadi-Pour, "A Fuzzy Difference Based Edge Detector", Iranian Journal Of Fuzzy Systems Vol. 9, No. 6, 2012, pp. 69-85.

[12] Somya Saxena,Sunil Kumar and Vijay Kumar Sharma, " Edge Detection Using Soft Computing In Matlab", IJARCSSE,Volume 3, Issue 6, June 2013.

[13] Mamta Juneja and Parvinder Singh, "Performance Evaluation Of Edge Detection Techniques For Images In Spatial Domain", International Journal Of Computer Theory And Engineering, Vol. 1, No. 5, December, 2009 1793-8201.

[14] Ritaban Das and Pinaki Pratim Acharjya, "Edge Detection Using The Magnitude Of The Gradient", International Journal Of Innovative Research In Computer And Communication Engineering Vol. 1, Issue 2, April 2013.

[15] Adrian-Viorel Diaconu, "Simple, Xor Based, Image Edge Detection", ECAI 29 June, 2013.

[16] Er. Navneet Kaur. and Er. Rishm, "A Survey Of The Algorithms For Image Noise Removal And Edge Detection", International Journal Of Advanced Research In Computer Engineering & Technology (IJARCET) Volume 2, Issue 11, November 2013.

[17] Bernd Jähne, "Digital Image Processing", 5th Revised And Extended Edition SPRINGER Engineering Online Library,2009.

[18] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", 2nd Edition.

## AUTHOR PROFILE

Kalyan Kumar Jena

B.Tech(CSE),M.Tech(CSE.),Ph.D.(Continuing)
Asst. Prof., Dept. Of Computer Science Engg. & App.
Indira Gandhi Institute Of Technology,Sarang,Odisha,India
Research Area: Image Processing (Edge Detection)
kalyankumarjena@igitsarang.ac.in