A project report on

EDGE DETECTION ON GRAYSCALE IMAGES USING THE PYTHAGOREAN THEOREM

Submitted in partial fulfillment of the requirement for the award of degree of

Master of Technology
In
Information Systems

Submitted By:

Monika Rabha
(2K14/ISY/08)

Under the Guidance of:

Mr. Anil Singh Parihar

(Assistant Professor, Department of Computer Science and Engineering, DTU)

2014-2016

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
Bawana Road, Delhi – 110042

CERTIFICATE

This is to certify that **Monika Rabha (2K14/ISY/08)** has carried out the major project entitled "**Edge Detection On Grayscale Images Using The Pythagorean Theorem**" in partial fulfillment of the requirements for the award of Master of Technology Degree in Information Systems during session 2014-2016 at Delhi Technological University.

The major project is bonafide piece of work carried out and completed under my supervision and guidance. To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any degree or diploma.

Mr. Anil Singh Parihar
Assistant Professor
Department of Computer Science and Engineering
Delhi Technological University
Delhi-110042

ACKNOWLEDGEMENT

I express my sincere gratitude to my project mentor Mr. Anil Singh Parihar, Assistant Professor, Department of Computer Science and Engineering, Delhi Technological University, Delhi, for providing valuable guidance and constant encouragement throughout the project. It is my pleasure to record my sincere thanks to him for his constructive criticism and insight without which the project would not have shaped as it has.

I extend my gratitude to Dr. O.P. Verma, Head of Computer Science and Engineering Department for permitting me access to the department facilities and giving me opportunity to work on this project.

I thank God for making all this possible, my parent and friends for their constant support and encouragement throughout the project work.

Monika Rabha

Roll No. 2K14/ISY/08

M.Tech (Information Systems)

Department of Computer Science and Engineering

Delhi Technological University

ABSTRACT

Edge detection is one of the important task performed in computer vision and image processing as edges that are present in an image in some ways makes it possible to form a very sparse representation of what is actually in the image. In this work, a new method is proposed for edge detection using the universal concept behind the Pythagorean Theorem and the geometric surface two-sheeted hyperboloid. A triangular kernel is used which computes values for every possible direction first, followed by mutually subtracting each resulting image. After this, thresholding is applied in resultant images to generate binary image and to discard any false edges. Morphological operation is also applied as post processing step for refining the edges furthermore. Final results shows that the proposed method has comparatively consistent performance on various test images, which is competitive with traditional edge detecting algorithm like Sobel, Prewitt, Canny operator etc. and also proves how edge detection method strongly depends upon the application domain as well. Evaluation of results for quantitative comparisons was done with two assessment techniques Cohen's Kappa and Shannon's Entropy and results were shown for a number of test images.

Table of Contents

Figures and Tables

Chapter 1

INTRODUCTION

A digital image is made up of pixels. More accurately a pixel (a word invented from "picture element") is the smallest element, individual locations of colors or certain level of grayscale intensity. There are basically four kinds of images which are grayscale images, binary images, color (RGB) images and indexed color images.

An edge in a image is represented as some sort of physical phenomenon, an occlusion between objects in an image. Different images formed by different sensing modalities have different properties and what really constitutes an edge can be domain-dependent. So, edges in an image are basically the locations where there is a sharp change in intensity which contains important visual and semantic information and the change is from high intensity value to low intensity value or vice-versa. A high intensity value indicates steep change and a low intensity value indicates shallow change. Therefore, these edges are boundaries including boundaries of objects and to find those boundaries we make use of these discontinuous levels of intensity. To be specific, an edge is a concise or sparse descriptor of what is in the image. Some of them corresponds to important boundaries and some of them do not. But the important thing is that the edges that are present in an image in some ways makes it possible to form a very sparse representation of what is actually in the image. For example we can take all the pixels of an image, then reduce them down to basic black and white values and followed by again reducing them to very sparse number of pixels that actually have any information in them and we still get the key idea of what is in the image. So, in images especially black and white images, edges can be used to reduce the amount of data tremendously without losing much about what is in the image itself i.e. extracting meaningful edges from images, amounts to a dramatic reduction in the amount of data. Color discontinuity, depth discontinuity, surface normal discontinuity surface, illumination discontinuity are some of the causes which forms edges.

Edge detection is fundamentally related to the gradient of the image and is the process of identifying the pixels which falls along an edge. Edge detection operators use methods to detect these locations. One thing all edge operators deal with is noise, which can be defined as fluctuations in image data and can carry a intensity value as high as an edge pixel can. Even a small amount of noise greatly affects the output. So to overcome this problem, one solution is to smooth the image. Smoothening is done on the image so as to get rid of false edges, to get rid of noise. But the question is now, how much to smooth the image? It depends on the application. Well in some images smoothening is done to a greater extent comparatively, when big things are of importance in an image and in other images to a little extent small

or fine details are needed. Then any sharpening filter is applied in order to enhance the quality of the edges in the image. This is followed by a very useful feature while performing edge detection that is carrying out the operations using the selected local gradient and enhance the edge pixels even more, suppressing the non-edge pixels. To keep only two types of data in an image i.e. edge or non-edge, first a binary image is produced by using thresholding where black regions with lowest intensity value and white regions highest intensity value are obtained. As edges have high gradient magnitude whereas places like background have much weaker i.e. low gradient magnitude, a image results in only edges of important features in the image. As described earlier, highly intensified pixels gives the position of edge pixels, so correct labeling of edge and non-edge pixels often requires subjective interpretation. Deciding the threshold is also one of the main task as with low threshold value, lots of edge points can be visible and similarly with high threshold value, a few edge points. Plus even with thresholding using a high threshold, it is noticed that edge points that are more than one pixel wide along a contour come into picture. What would be feasible is that there should be only one response to an edge. For a line to be seen as a one pixel edge line, there are different operations which can be carried out. As for example considering that point as an edge pixel which is local maximum in the direction of the gradient i.e. edge thinning etc.

1.1 MOTIVATION

Edge detecting operators are discrete operators used within edge detection algorithms which approximates gradient of the image intensity function. These are actually the spatial filter which enhances the discontinuities or the shift of the intensities.

For example: First derivative [1], an edge operator which enhances high spatial frequencies.

$$\frac{\partial f\,(x,y)}{\partial x} \approx f(x+1,y) - f(x,y) \tag{1}$$

Equivalent filter in frequency domain-

$$\frac{\partial^2 f(x)}{\partial x^n} \approx (2\pi ju)^n - F(u) \tag{2}$$

For first derivative ($n=1$) this acts like a high pass filter, attenuating frequencies near 0.

One drawback with this edge operator is that it also enhances noise. Thus, practical edge detectors first go for image smoothing for noise reduction. For example there is Sobel filter [1] which effectively does a smoothing followed by a differencing. This filter compute for edges in both vertical and horizontal directions then combines the information into a single metric.

The masks are defined as follows:

$$x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Edge Magnitude = $\sqrt{x^2 + y^2}$ Edge Direction = $\tan^{-1}\left[\frac{y}{x}\right]$

There is also Prewitt operator [3] which is very similar to the Sobel, with mask coefficients:

$$x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Therefore all edge detectors should follow the steps mentioned below-

Step1: Smoothing for noise detection

Step2: Detection of candidate edge points

Step3: Edge localization i.e. keeping only the points that are closed to the true edge.

It is useful to be able to detect edges at different scales, depending on the features of interest. Intensity changes occur at different scales in an image, so operators of different sizes is needed to detect them and that is where Scale-space edge operators have important applications, where a space of images is formed by applying a series of operators at different scales. Gaussian of different sizes can be used to filter the image at different scales. Gaussian is a natural choice for smoothing operator because it has this parameter σ. Here σ, which is the standard deviation of the Gaussian can be adjusted. Adjusting the σ of a Gaussian low pass filter is a convenient way to select the scale. By making the Gaussian wider or increasing the σ more smoothing can be done and less of less structure can be seen whereas with a very less σ only very fine details is noticed and then one can also find the scale σ at which a given edge is most significant.

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)} \tag{3}$$

There is also Laplacian of Gaussian (or "Marr-Hildreth") [4] and Canny operator[5]. Laplacian of Gaussian operator involves convolving with a Gaussian and then computing the laplacian or vice-versa as,

$$\nabla^2(g * I) = (\nabla^2 g) * I \tag{4}$$

As explained earlier the operators's scale is determined by σ and when σ is changed the scale will change.

$$\nabla^2 g(x, y) = -\frac{1}{\pi\,\sigma^4}\left(1 - \frac{r^2}{2\sigma^2}\right) e^{-r^2/(2\sigma^2)} \tag{5}$$

where $r^2 = x^2 + y^2$ (it is circularly symmetric).

Canny edge Operator [5] is an optimal edge operator to find step edges in the presence of white noise, where "optimal" means good detection, localization and single response.

Step1: The image is first convolved with derivative of Gaussian operator (dG/dx, dG/dy).

Step2: The gradient direction at each pixel is then calculated and is quantized into one of four directions (east-west, northeast-southwest, north-south, and northwest-southeast).

Step3: Here an edge is a candidate edge point only if the magnitude of gradient is larger than the two neighbors along the direction.

Step4: And in the final step, maximum of convolution results approximately in true edge location followed by edge linking. Here, in edge linking, edge points are to be joined into connected curves or lines which facilitate object recognition. As previously stated, some edge points along the curve may be weak, causing a miss, which then results in broken curve, so first a high threshold is used to make sure only true edge points are considered into account and then linking the additional edge points using a comparatively lower threshold (which is called hysteresis) i.e. from each strong edge point, following the chains of connected edge points in both directions perpendicular to the edge normal.

Chapter 2

LITERATURE REVIEW

In 1986 John Canny proposed a computational approach to edge detection in which detection and localization criteria are defined for a class of edges, and mathematical forms are presented for these criteria as functions on the operator impulse response [5]. An additional criterion to ensure that the detector has only one response to a single edge is also added. This can also be used in numerical optimization. In 1994 Sugata Ghosal and Rajiv Mehrotra presented a new parametric model-based approach using orthogonal Zernike moment-based operators to high-precision composite edge detection, where composite edges are combinations of steps, and roof edges [7]. Their method involves parameterization of the image intensity function followed by computation of a set of projection coefficients onto a set of orthogonal moment-generating polynomials. In 1995 Russell C. Hardie and Charles G. Boncelet have examined the use of non-linear edge enhancers as pre-filters for edge detectors which are able to suppress noise simultaneously leading to minimization of false alarms due to noise and, large and localized edge gradient estimates resulting in significantly improved edge maps [8]. Later in 1999 Alberto De Santis and Carmela Sinisgalli proposed an adaptive method for edge detection based on a linear stochastic signal model in monochromatic unblurred noisy images which does not require overall thresholding[9]. For all pixels, the statistical parameters are estimated by using a Bayesian procedure, followed by an hypothesis test which is adopted to mark a pixel as an edge point.

In 2001 C. Grecos, J. Jiang and E. A. Edirisinghe have presented algorithms for the detection and prediction of edge pixels in JPEGLS (Lossless Joint Photographic Experts Group), where their proposed schemes leads to the reduction of predictive Mean Squared Error[10]. In 2003 Scott Konishi, Alan L. Yuille, Song Chun Zhu and James M. Coughlan, presented edge detection process as statistical inference [11]. This statistical edge detection is data driven and they have even shown that one can learn these conditional distributions on one set of data and adapt them to the other with only a slight degradation of performance without knowing the second data set. In 2005 Renyan Zhang, Guoliang Zhao and Li Su proposed a edge detection method which is based on the integer logarithm ratio of gray levels between two image points where sensitivity of this edge detection methods can be easily adjusted by a single parameter serves as an advantage [12]. In 2007, J.-A. Jiang, C.-L. Chuang, Y.-L. Lu and C.-S. Fahn proposed a new edge detector based on mathematical morphology to preserve thin edges in low-contrast regions [13]. The input video frame is first processed by a contrast enhancement operator and a dilation residue edge detection operator. The morphological dilation residue edge detector is then used in their algorithm for detecting the edges. In 2010 Olivier Laligant and Frédéric Truchetet, proposed a non-linear scheme which is a combination of two polarized derivatives allowing the detection and univocal edge localization [14]. Later in 2010 Sos S. Agaian, Karen A. Panetta, Shahan C. Nercessian,and Ethan E. Danahy, introduced a new concept of Boolean derivatives as a combination of partial derivatives of Boolean functions and proved that Boolean derivatives functions can be used for the application of identifying the edge pixels in binary images and the same can be extended grayscale images [15]. Again later in 2010 Yang Chen proposed a method using the logarithm function for detection of edges where firstly normalization of pixel values is performed within a mask so to be considered as probabilities followed by addition with a positive to avoid zero values [16]. After this, the sum of the logarithms of all the adjusted probabilities is calculated. A threshold is chosen for generating the binary edge image followed by morphological operation for post processing.

Chapter 3

THE PYTHAGOREAN THEOREM AND TWO-SHEETED HYPERBOLOID

3.1 CONCEPT OF THE PYTHAGOREAN THEOREM AND TWO-SHEETED HYPERBOLOID

This section describes the universal concept behind the Pythagorean Theorem and the geometric surface two-sheeted hyperboloid stating how these two important mathematical figures can be used in detecting edges.

3.1.1 PYTHAGOREAN THEOREM

In algebraic terms, Pythagorean Theorem is defined as,

$$a^2 + b^2 = c^2 \tag{6}$$

where, a and b are the legs of a triangle and c is the hypotenuse [17].

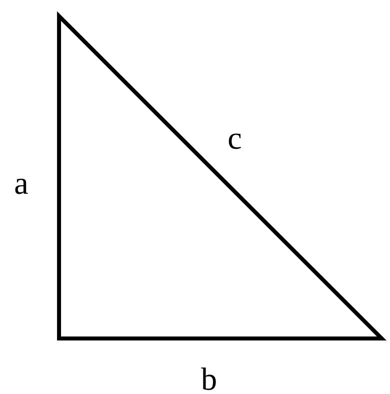

Figure 1. A Right Angled Triangle

Traditionally it is always known that triangles are used in Pythagorean Theorem, which is the simplest 2-D shape. In reality this is not only confined to line segments but can belong to any shape [18]. For example, taking circles into account, by applying Pythagorean Theorem, we have,

$$Circle\ of\ radius\ 50 = Circle\ of\ radius\ 40 + Circle\ of\ radius 30 \quad (7)$$

Therefore, the Pythagorean Theorem helps to relate the areas of any similar shapes or property i.e. the said "length" of a side can be energy, distance, time, work or even individuals in a social network as proved below:

- Social Networks:

As known already, Metcalfe's Law [19] says that the value of a network is approximately n^2 (i.e. the number of relationships). Therefore in terms of value,

$$Network\ of\ radius\ 500M = Network\ \ of\ radius\ 400M + Network\ \ of\ radius 300M \quad (8)$$

Here, the 2nd and 3rd networks have a total of 700 million people, but they aren't a coherent whole, instead it is a fact that a 500 million people network is as valuable as the others combined.

- Computer Science:

Since, some programs having n inputs take n^2 time to run (e.g. bubble sort). Therefore, in terms of processing time:

$$5inputs = 4inputs + 3inputs \quad (9)$$

Again here, 7 elements among two groups can be sorted as 5 items in one. Thus, here the Pythagorean theorem proved that sorting 5 elements combined can be slow as sorting 3 and 4 separately.

- Physics:

Since it is known that the kinetic energy is defined as: $\frac{1}{2}mv^2$

Where, mass ‘m’ and velocity ‘v’ of an object.

Therefore, in terms of energy,

$$Energy\ at\ 50mph = Energy\ at\ 40mph + Energy\ at\ 30mph \quad (10)$$

Here it can be clearly proved that with the energy used to accelerate one bullet to 50 mph, two others can be to accelerated to 40 and 30 mph can be accelerated.

3.1.2 TWO-SHEETED HYPERBOLOID

Quadrics of a two sheeted hyperboloid:

$$x^2 + y^2 - z^2 = -1 \quad (11)$$

where $\{(x, y, z) = f(x, y)\}$ is graph of function$f(x, y)$and which defines a surface.

From an image processing point of view, by applying Laplace-Beltrami operator on the hyperboloid, it is noticed that the gradient on the hyperboloid is proportional to the gradient of the Euclidean plane i.e its gradient is a scaled gradient of the plane [21]. Hence, the derived gradient operator mentioned below can be used to localize contours.

$$\nabla_{D_+} f = \frac{(1-x^2-y^2)^2}{4} \nabla_{R^2} f \quad (12)$$

Chapter 4

PROPOSED METHODOLOGY

4.1 PROPOSED APPROACH

Firstly smoothing is done on the image, suppressing noise as much as possible, without destroying the true details followed by sharpening thereby enhancing the quality of the edges [1]. Next, as explained previously, the Pythagorean Theorem explains the value of each element on both sides of the equation. Therefore, if a equation looks as is mentioned in equation (6), shifting the elements from one side of the equation to the other, we get,

$$x(i,j) = C^2 - (A^2 + B^2) \tag{13}$$

where, if ‘x’ is a digitized image having L gray levels then $x(i,j)$ is the pixel luminance at every location [i,j] $(0 \leq x(i,j) \leq L-1)$.

It can be clearly seen that equation(13) is similar to the equation of a two-sheeted hyperboloid as mentioned earlier in equation (11) which when used on a image can be used to localize contours. 3D Surface plot for the equation (13) is shown in Figure 2:

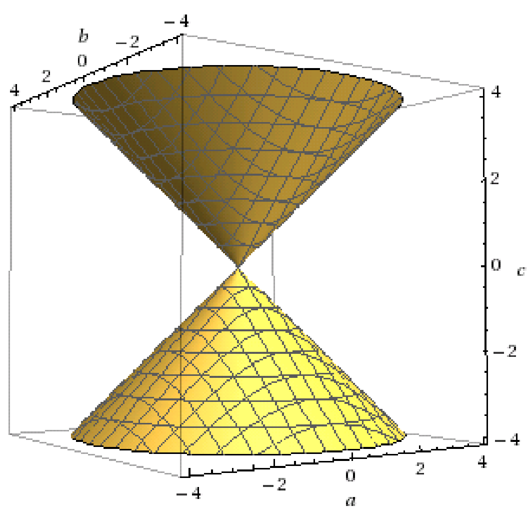

Figure 2. 3-dimentional Surface Plot for $x = c^2 - a^2 - b^2$

Next as in general properties and applications of logarithms, these are used in various types of circumstances for example to calculate alterations in atmospheric CO_2, population growth, when estimating radioactive decay-dating, sedimentology, etc.[22]. Because of such properties, using natural logarithms in the derived expression, so as to make the result better, we get,

$$x(i,j) = |[C^2 - \{A^2 + B^2\}]|/ln(10) \quad (14)$$

where, ln(10) is proved to perform better than any other values applied.

Therefore to calculate for $x(i,j) = |[C^2 - \{A^2 + B^2\}]|/ln(10)$, A, B, and C for a pixel $x(i,j)$ would look like as shown in Figure 4 showing for all possible 'triangles'.

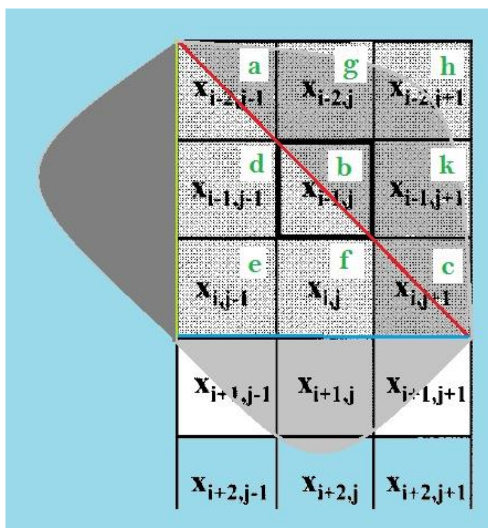

Figure 3. Pixel map

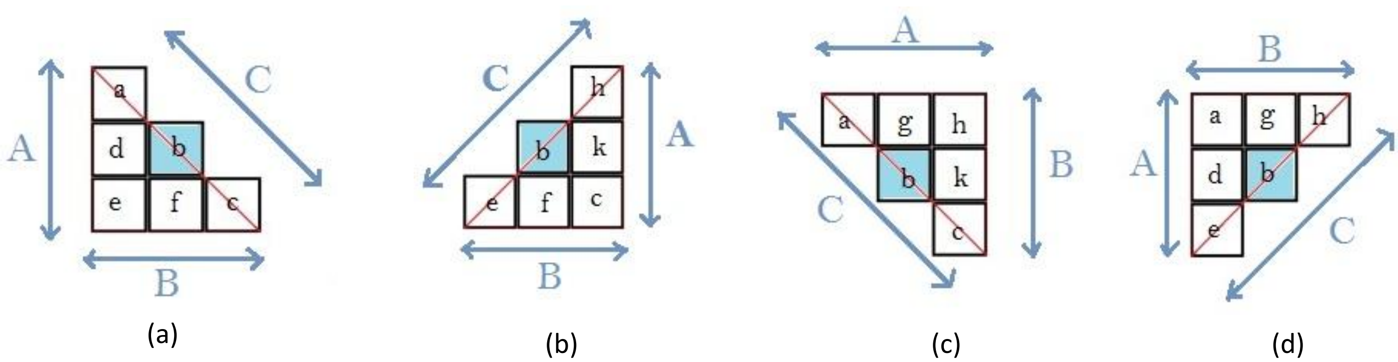

Figure 4. Pixel Nomenclature

i.e. for figure 4(a),

$$U = x(i,j) = |[(a + b + c)^2 - \{(a + d + e)^2 + (e + f + c)^2\}]|/ln(10) \quad (15)$$

Similarly for Figure 4(b), Figure 4(b), and Figure 4(c), respectively:

$$V = x(i,j) = |[(e + b + h)^2 - \{(h + k + c)^2 + (e + f + c)^2\}]|/ln(10) \quad (16)$$

$$W = x(i,j) = |[(a + b + c)^2 - \{(a + g + h)^2 + (h + k + c)^2\}]|/ln(10) \quad (17)$$

$$Y = x(i,j) = |[(e + b + h)^2 - \{(a + g + h)^2 + (a + d + e)^2\}]|/ln(10) \quad (18)$$

NOTE: As letter ‘X’ is already used for nomenclature of original image, the sequence here is U,V,W and Y for nomenclature of resulting images for equation (15), (16), (17), and (18) respectively.

Taking grayscale images as shown in Figure 5 and Figure 6 into account, and carrying out equations (15), (16), (17) and (18) on these test images, output will be as shown in Figure 7 and Figure 8 respectively.

Figure 5. Original Grayscale Duckling Image

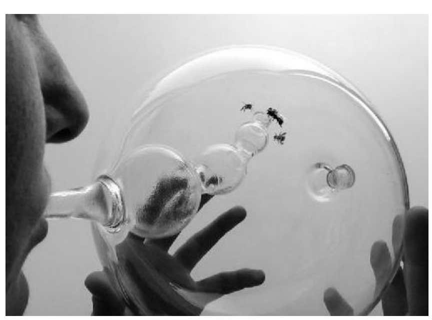

Figure 6. Original Grayscale Glass Image

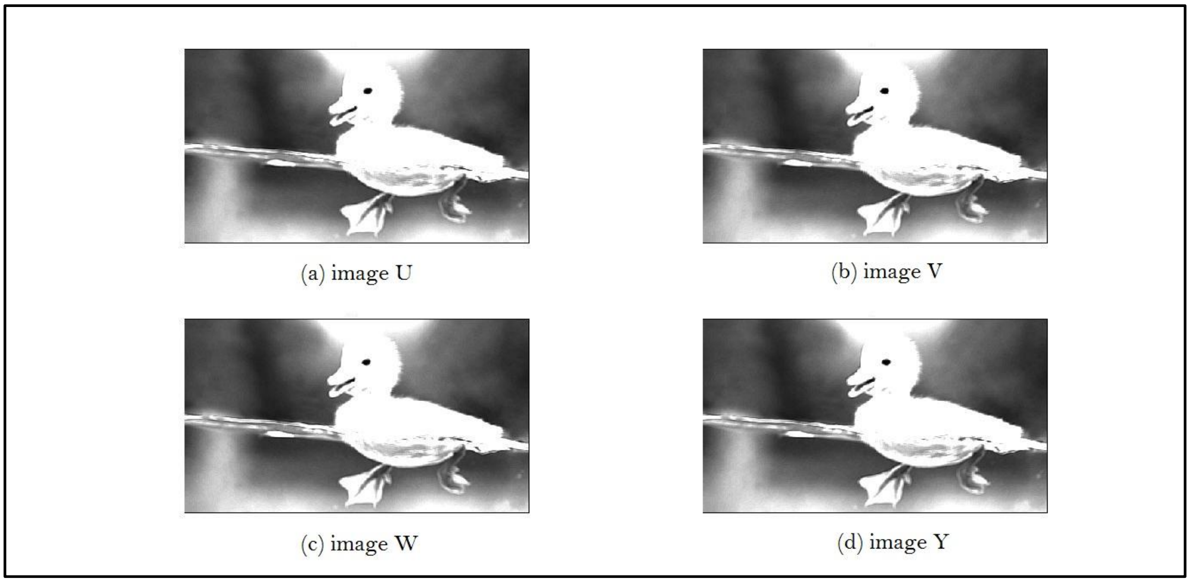

Figure 7. Output for Duckling image after applying equations (15),(16),(17)&(18) respectively

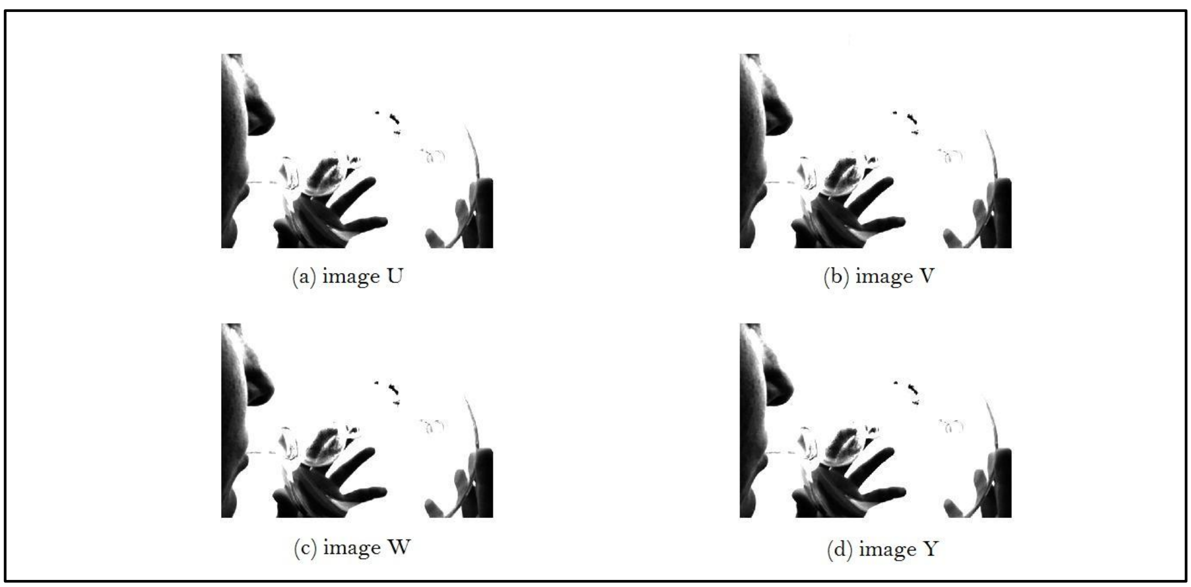

Figure 8. Output for Glass image after applying equations (15),(16),(17)&(18) respectively

Furthermore, subtraction is carried out between every two output combination possible for the present set of four outputs, i.e.:

$$M1 = U - V \tag{19}$$

$$M2 = U - W \tag{20}$$

$$M3 = U - Y \tag{21}$$

$$M4 = V - W \tag{22}$$

$$M5 = V - Y \tag{23}$$

$$M6 = W - Y \tag{24}$$

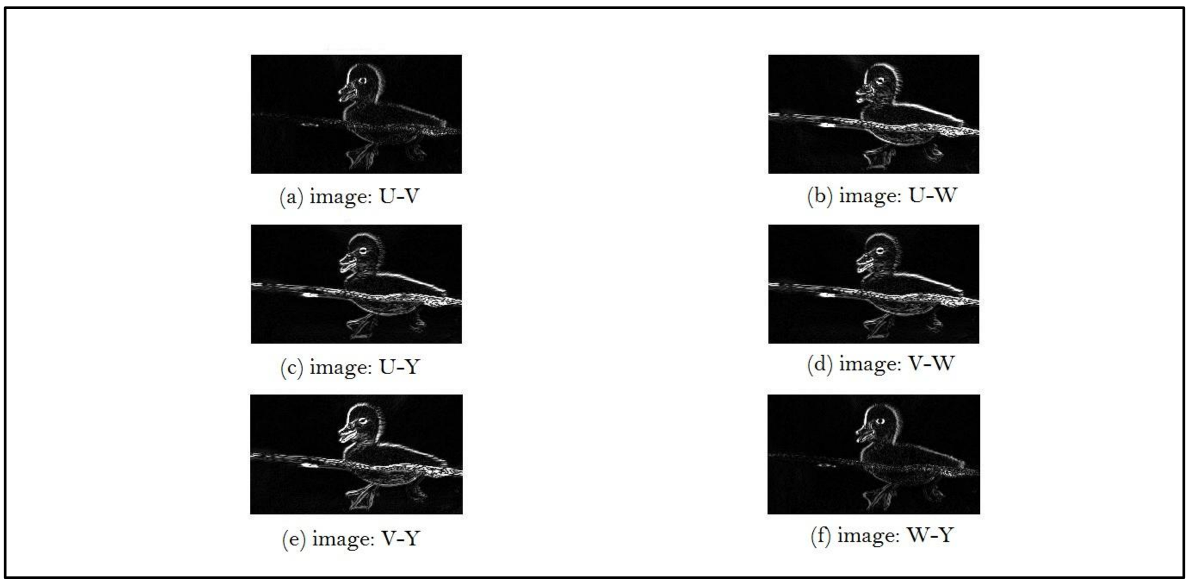

Figure 9. Resulting Images after mutual subtraction in Duckling image

Figure 10. Resulting Images after mutual subtraction in Glass image

Thresholding technique is then performed on every image of the previous step, result of which is shown in Figure 11 and Figure 12 for both the test images, followed by combining all the thresholded images produced, as shown in Figure 13 and Figure 14.

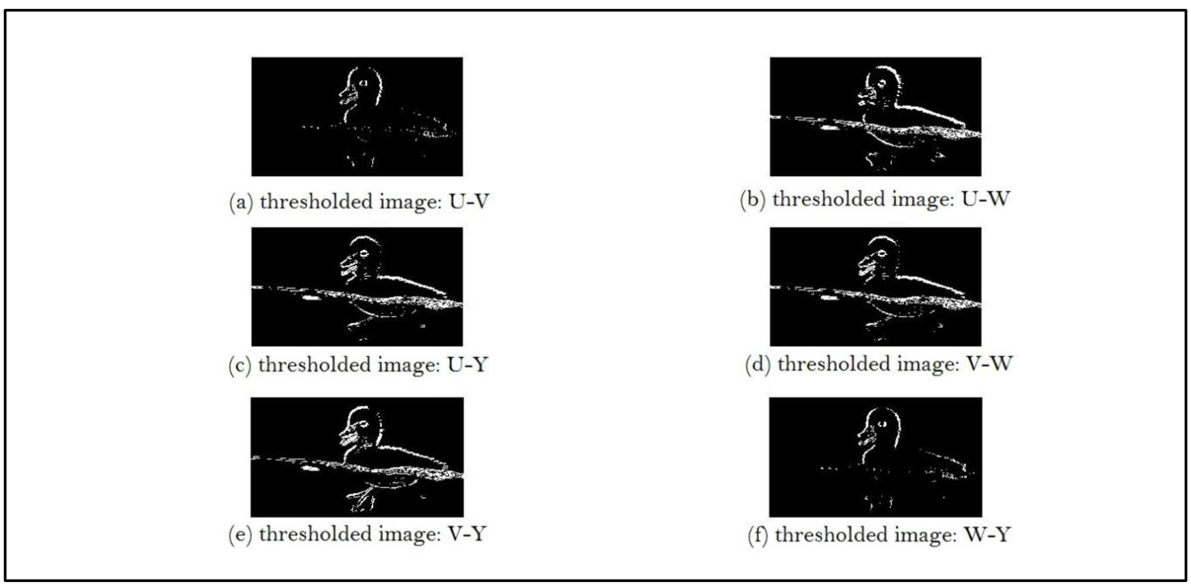

Figure 11. Resulting Images for Duckling image after applying threshold

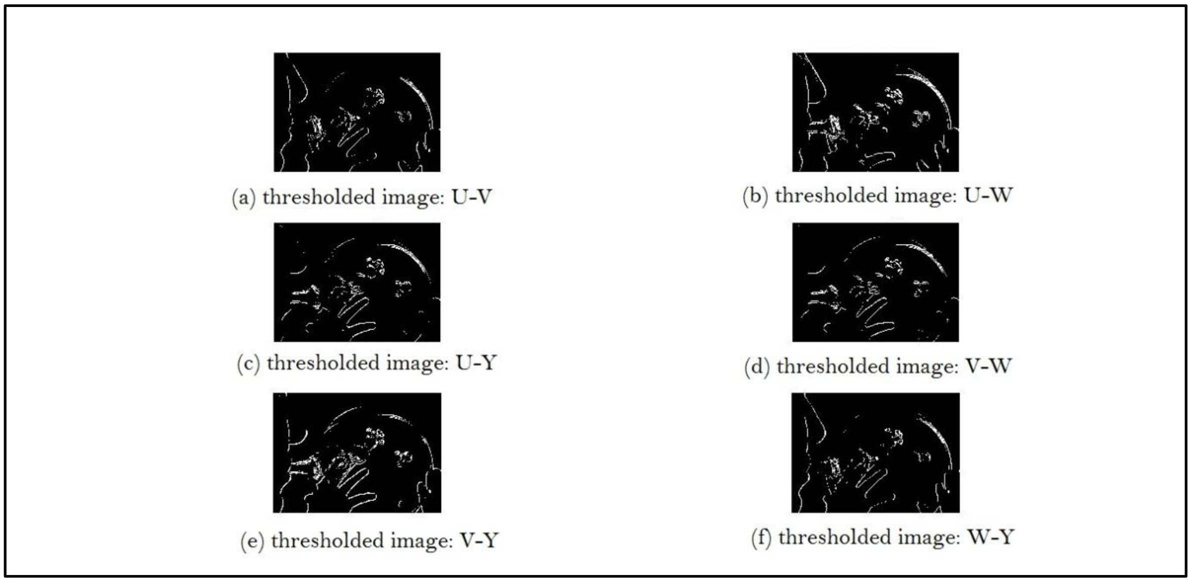

Figure 12. Resulting Images for Glass image after applying threshold

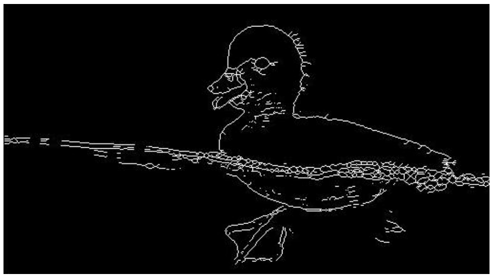

Figure 13. Final output of the proposed approach for Duckling image

Figure 14. Final output of the proposed approach for Glass image

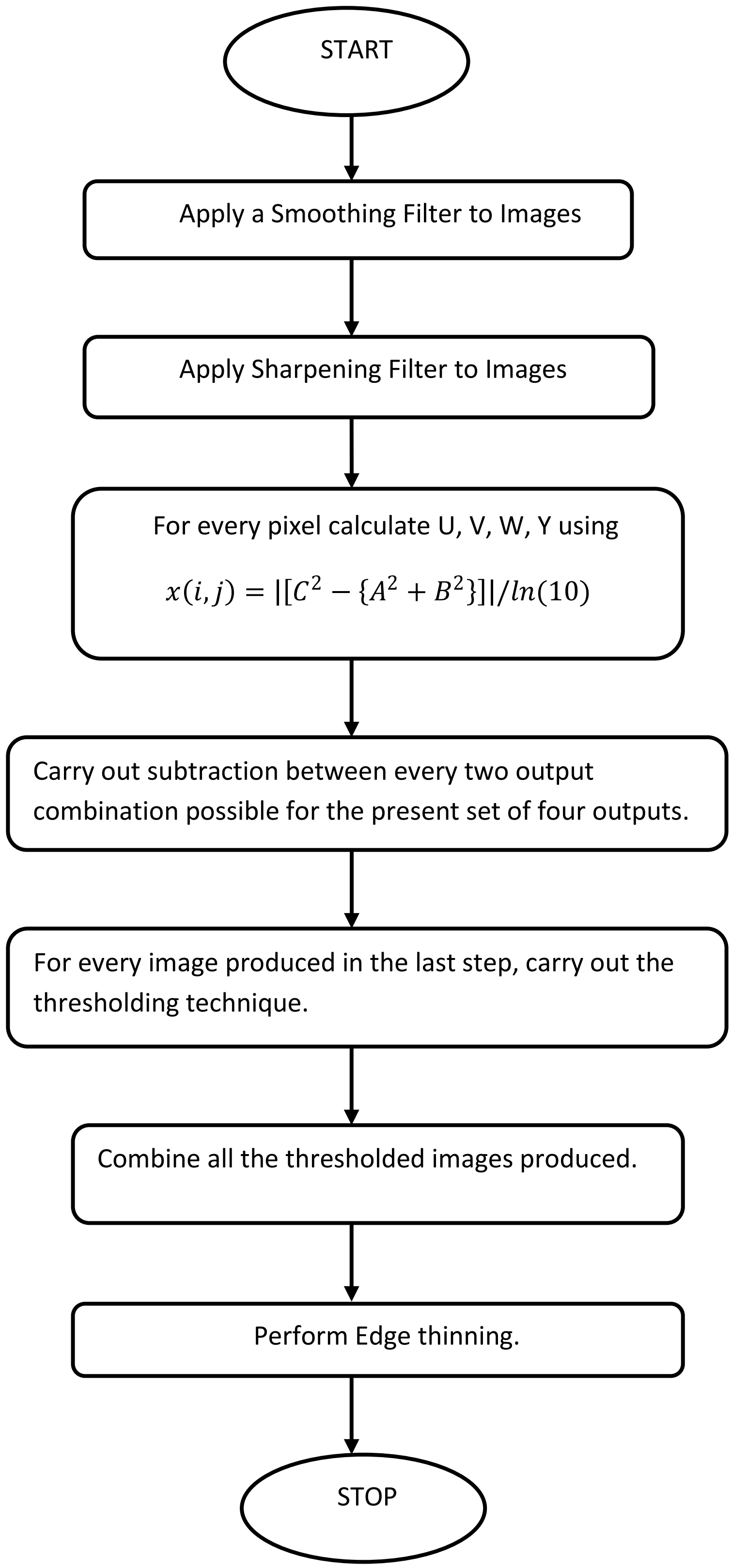

Figure 15. Flow Chart for Proposed Approach

4.2 ALGORITHM

Step1: Firstly smoothing the image, hence suppressing noise as much as possible, without destroying the true details followed by sharpening thereby enhancing the quality of the edges.

Step2: For every pixel calculate, U, V, W, Y using $x(i,j) = |[C^2 - \{A^2 + B^2\}]|/ln(10)$

Step3: Carry out subtraction between every two output combination possible for the present set of four outputs, i.e.:

$$M1 = U - V$$

$$M2 = U - W$$

$$M3 = U - Y$$

$$M4 = V - W$$

$$M5 = V - Y$$

$$M6 = W - Y$$

Step4: For every image produced in the last step, carry out the thresholding technique.

Step5: Combine all the thresholded images produced.

Step6: Perform Edge thinning which determines the exact location of an edge.

Chapter 5

EXPERIMENTAL RESULTS

Following is the system configuration used:

Processor: Intel® Core™ i5

Clock Speed: 2.50 GHz

Main Memory: 4GB

Hard Disk Capacity: 1TB

Software Used: MATLAB R2010a

5.1 QUANLITATIVE ASSESSMENT

A number of images of different sizes and formats have been used as test images to evaluate the performance of the proposed approach and some of the test images and their details are given in Figure 16. All the images used are grayscale images. The output of the proposed edge detection method is given in Figure 17. The results are then compared to the most commonly used edge-detectors such as Roberts [23], Sobel [1], Prewitt [3], Laplacian of Gaussian [4] and, Canny [5] as shown in Figure13.

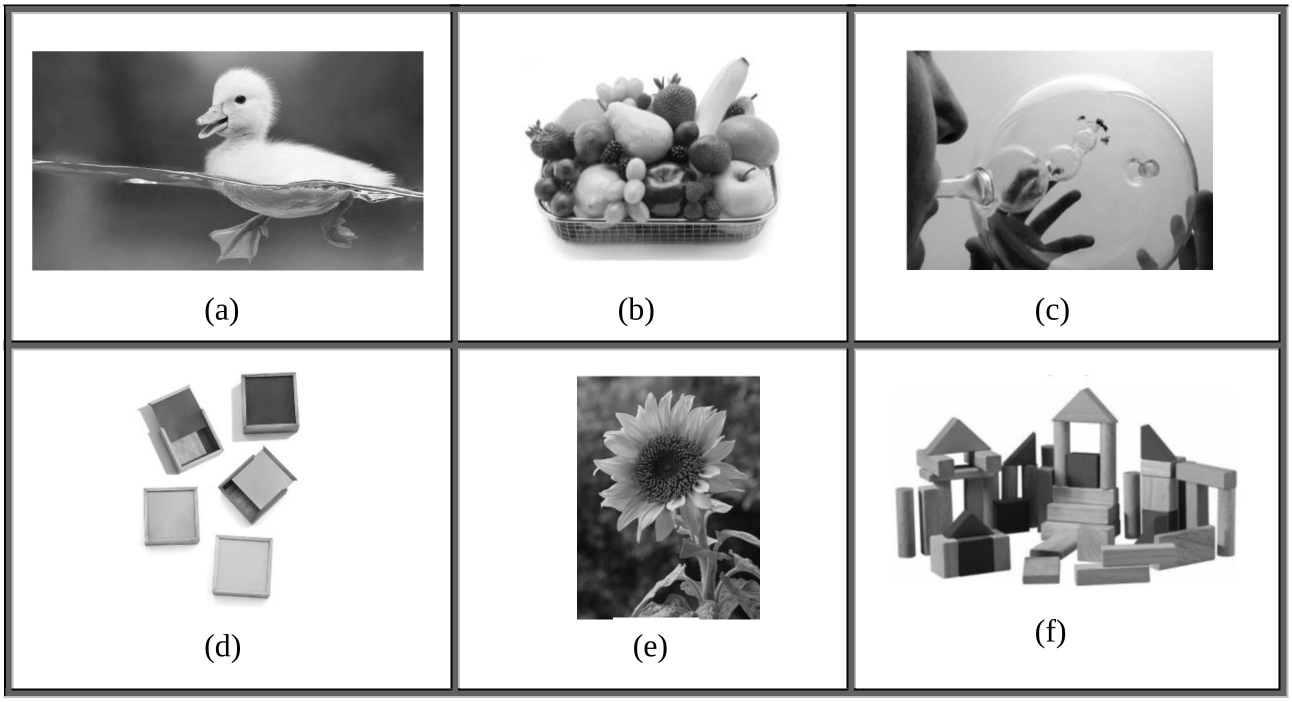

Figure 16. Test images (a) 512 x 288 Duckling image (b) 415 x 312 Fruits image (c) 770 x 550 Glass image (d) 650 x 774 Boxes image (e) 768 x 1024 Sunflower image (f) 512 x 303 Lego

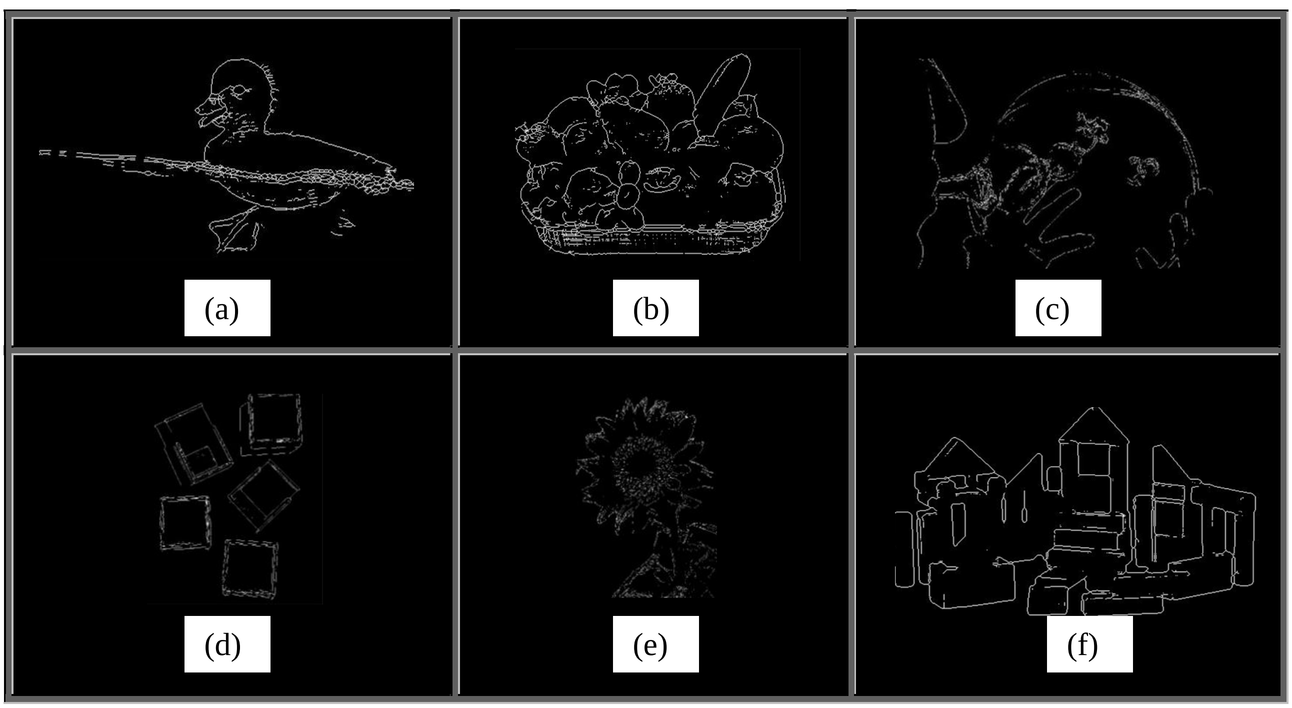

Figure 17. The resultant images using the proposed approach for test images:
(a) 512 x 288 Duckling image (b) 415 x 312 Fruits image (c) 770 x 550 Glass image (d) 650 x 774 Boxes image (e) 768 x 1024 Sunflower image (f) 512 x 303 Lego image

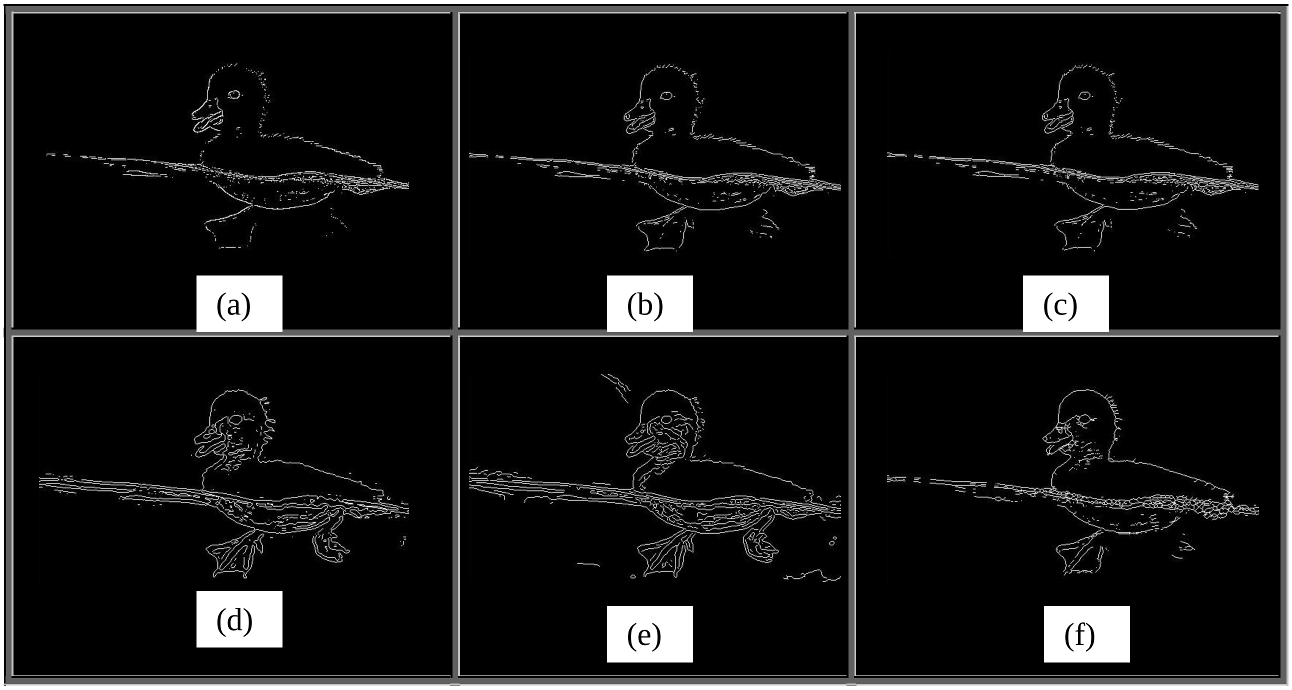

Figure 18. The results of different edge detectors on Duckling image

(a) Roberts (b) Sobel (c) Prewitt (d) Laplacian of Gaussian (e) Canny (f) Proposed approach.

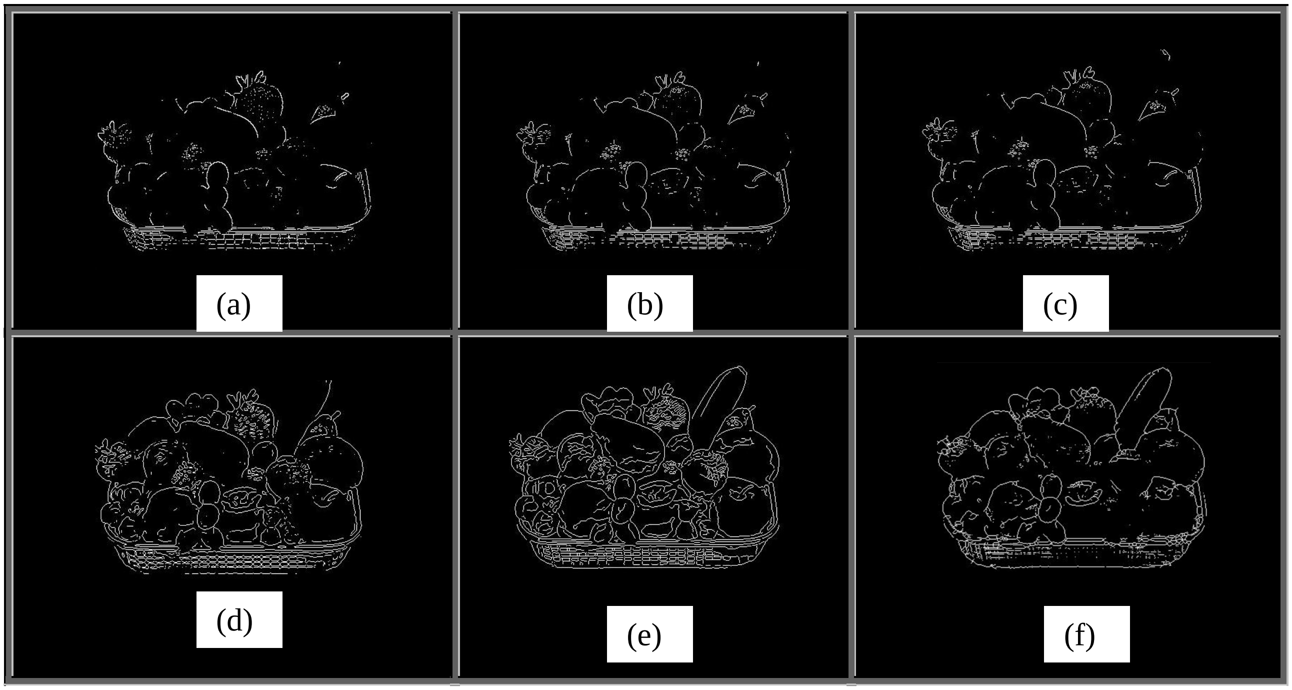

Figure 19. The results of different edge detectors on Fruits image

(a) Roberts (b) Sobel (c) Prewitt (d) Laplacian of Gaussian (e) Canny (f) Proposed approach.

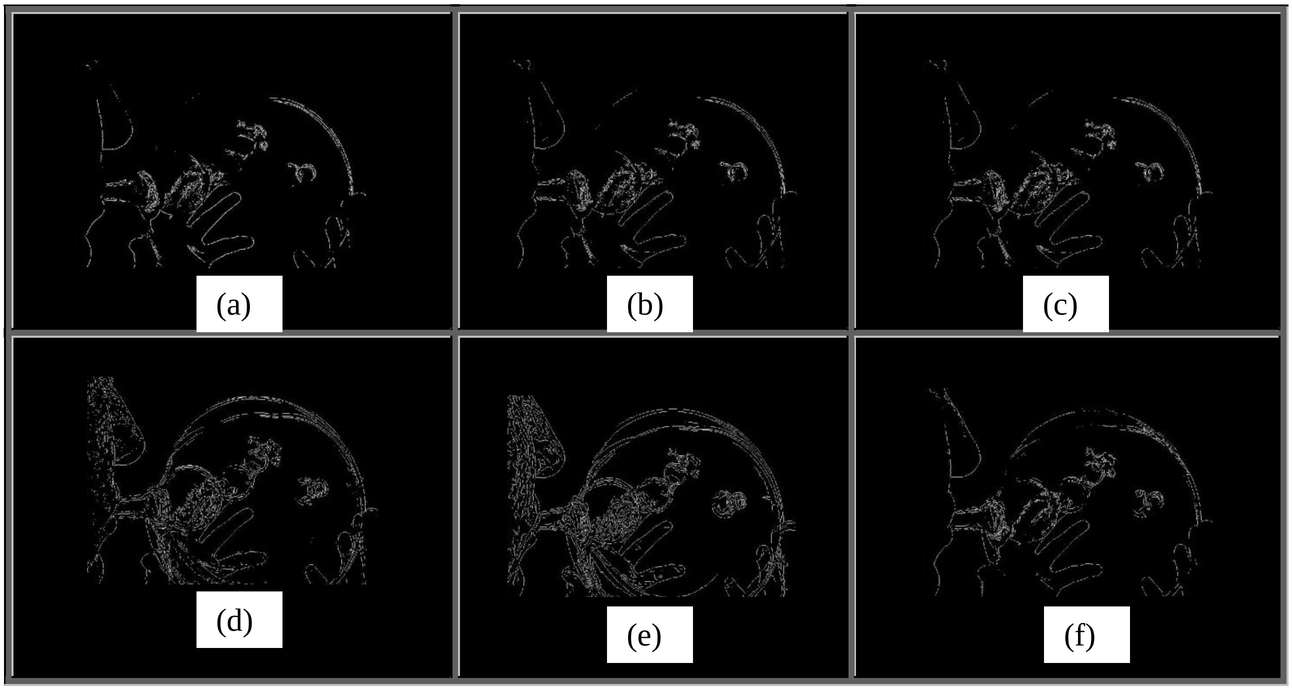

Figure 20. The results of different edge detectors on Glass image

(a) Roberts (b) Sobel (c) Prewitt (d) Laplacian of Gaussian (e) Canny (f) Proposed approach.

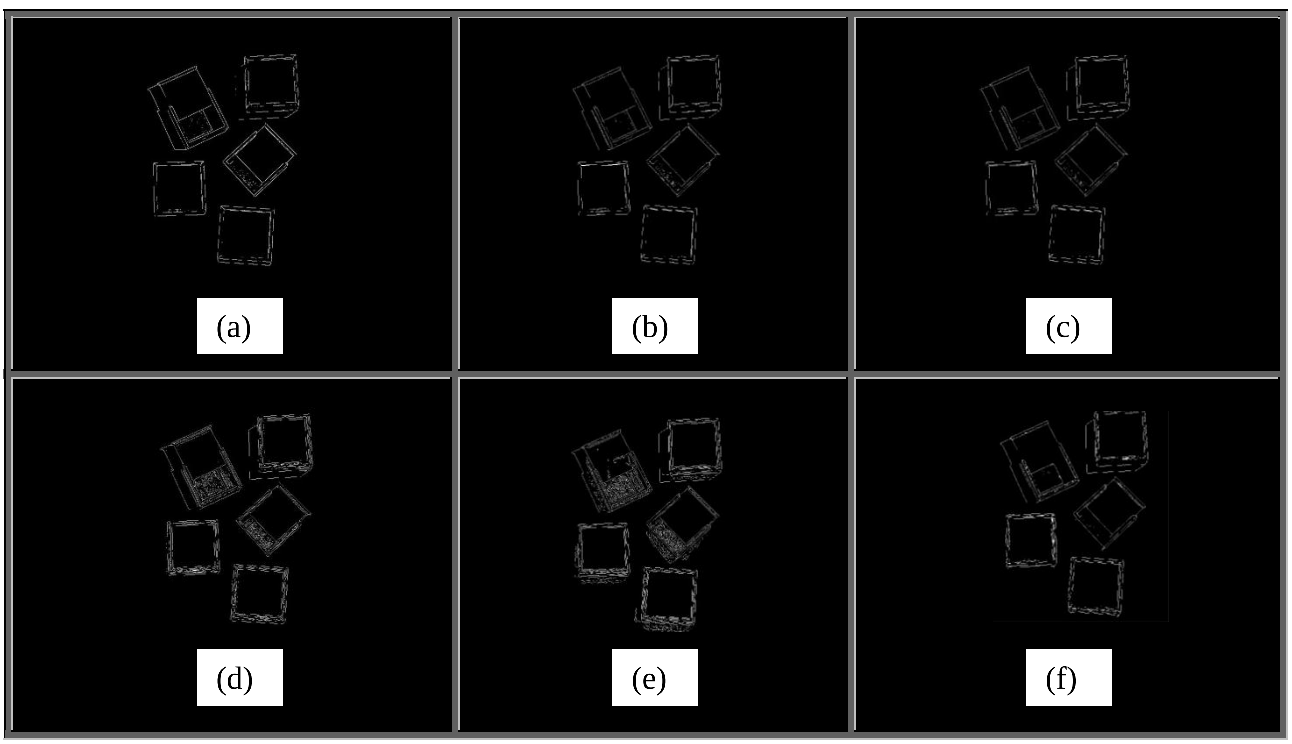

Figure 21. The results of different edge detectors on Boxes image

(a) Roberts (b) Sobel (c) Prewitt (d) Laplacian of Gaussian (e) Canny (f) Proposed

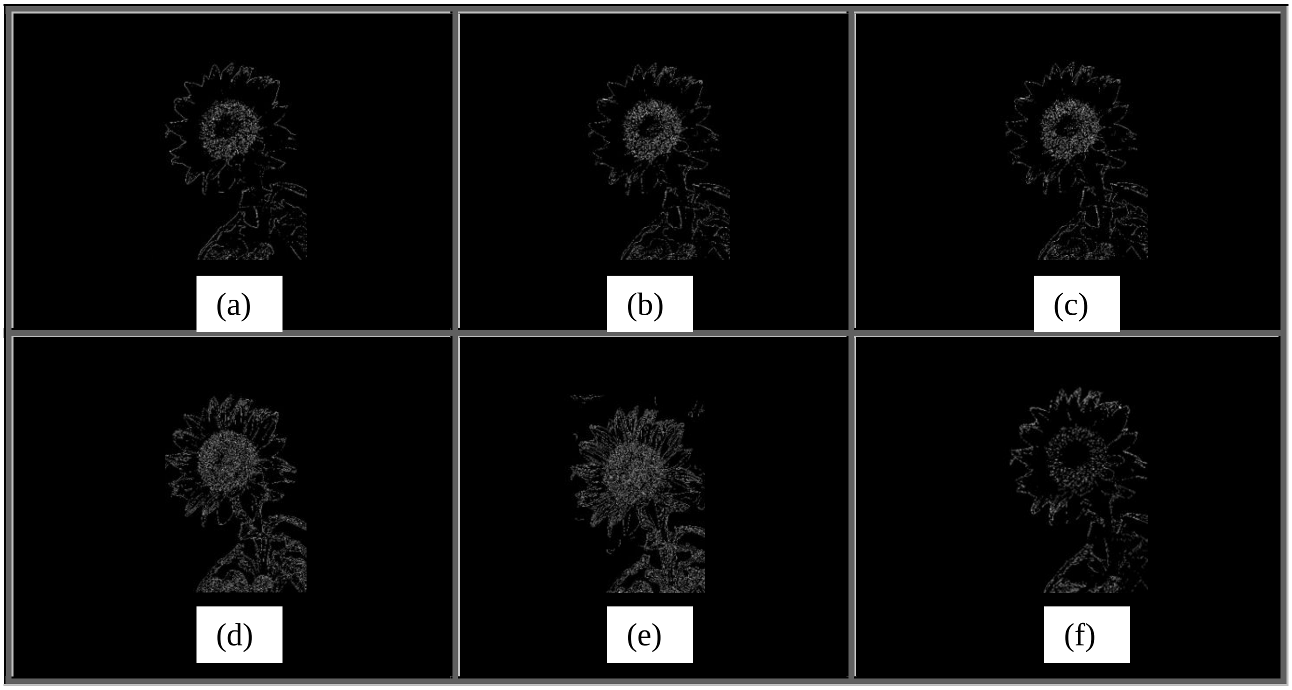

Figure 22. The results of different edge detectors on Sunflower image

(a) Roberts (b) Sobel (c) Prewitt (d) Laplacian of Gaussian (e) Canny (f) Proposed

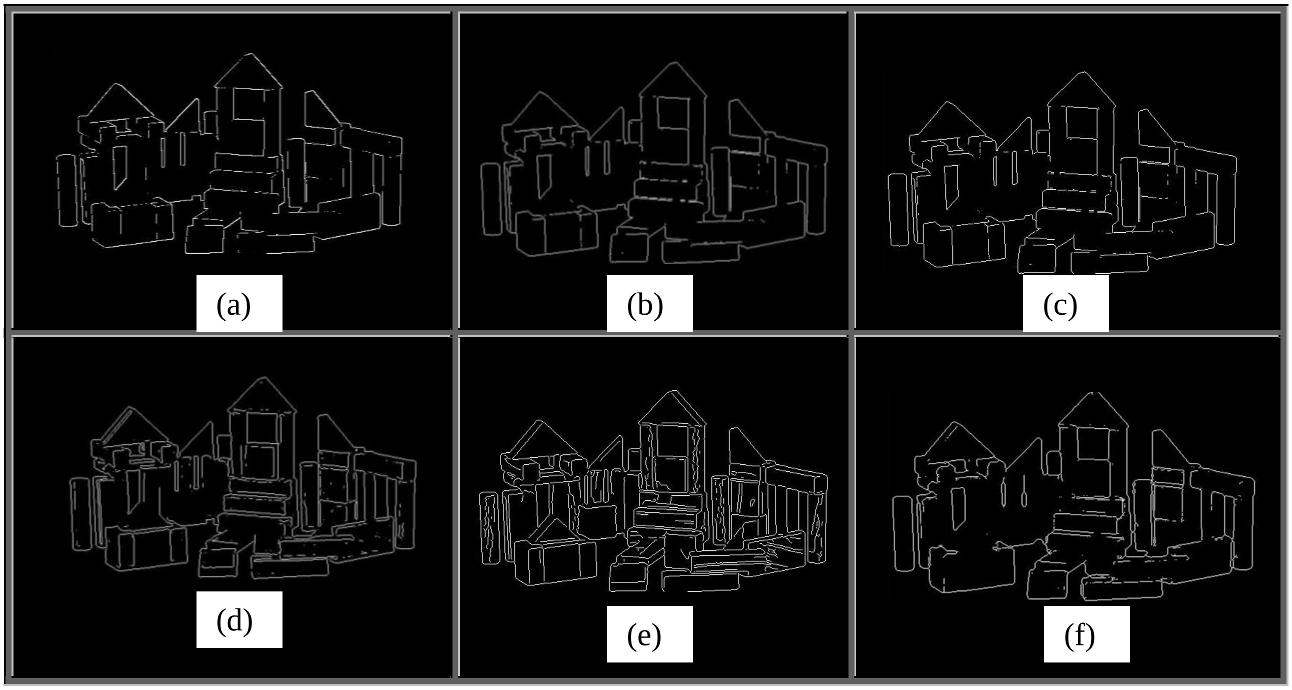

Figure 23. The results of different edge detectors on Sunflower image

(a) Roberts (b) Sobel (c) Prewitt (d) Laplacian of Gaussian (e) Canny (f) Proposed

Here, clearly it can be concluded that the proposed method performs better than Roberts edge detector, Sobel edge detection operator and Prewitt edge detection operator but as it is well known fact that, the choice of edge detection method also strongly depends upon the application domain as well[1]. For example, in the Figure 21 i.e the Boxes image, the proposed method, the Sobel and Prewitt edge detection operator have outperformed Laplacian of Gaussian and Canny edge detector as the latter operators are detecting even the false edges inside the true edge surface. Also, in the Figure 20 i.e. the Glass Image, the proposed method, has even outperformed Laplacian Of Gaussian and Canny edge detector for the same reason.

5.2 QUANTITATIVE ASSESSMENT

Since we know that binary edge maps are the output from edge detection techniques, so for the evaluation of quality of these resulting edge maps, several assessment techniques are proposed in the literature. In order to make quantitative comparisons, two quantitative measures are used: Cohen's Kappa [2] and Shannon's Entropy [20, 6].

Cohen's Kappa

Cohen's kappa, κ takes into account the agreement occurring by chance. The equation for κ is –

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \quad (25)$$

where is $\Pr(a)$ is observed percentage of agreement and $\Pr(e)$ is expected percentage of agreement.

The comparison of results using Cohen's Kappa is shown in Table 1.

Table 1
COHEN'S KAPPA VALUES

Image	Roberts	Sobel	Prewitt	LoG	Canny	Proposed
Duckling	0.3871	0.3171	0.3183	0.4422	0.4702	0.4586
Fruits	0.3241	0.2654	0.2631	0.4194	0.4970	0.4071
Boxes	0.7994	0.1414	0.1355	0.2860	0.1532	0.2526
Lego	0.4377	0.2758	0.2755	0.4251	0.4909	0.3886
Sunflower	0.3513	0.2552	0.4063	0.1753	0.1030	0.5685
Glass	0.7260	0.1551	0.1491	0.2674	0.1670	0.3463
Toy	0.4908	0.2474	0.2479	0.4013	0.4598	0.3725
Blocks	0.3675	0.3087	0.3077	0.5286	0.4846	0.3810
Boy	0.2813	0.2561	0.2574	0.4218	0.4614	0.4358
Dalmatian	0.3758	0.2877	0.2880	0.4211	0.4638	0.4433
Duck	0.3167	0.2499	0.2414	0.4006	0.5469	0.3752
Exina	0.2330	0.2167	0.2148	0.3784	0.3787	0.4905
Cub	0.2686	0.2485	0.2493	0.4400	0.4558	0.4136
Hotairballoon	0.5974	0.2619	0.2631	0.3187	0.2547	0.4783
Hotairballoon2	0.6894	0.1964	0.2093	0.3235	0.2978	0.2592
Toytruck	0.5771	0.1444	0.1435	0.3496	0.4491	0.2361

Shannon's Entropy

Shannon's entropy H is given by the formula

$$H(X) = -\sum_{i=0}^{N-1} p_i log_2 p_i \quad (26)$$

where p_i is the probability of a given symbol.

The comparison of results using Shannon's entropy is shown in Table 2.

Table 2
SHANNON'S ENTROPY VALUES

Image	Roberts	Sobel	Prewitt	LoG	Canny	Proposed
Duckling	0.0991	0.1137	0.1129	0.1488	0.1662	0.1209
Fruits	0.1375	0.1522	0.1499	0.2332	0.2786	0.1975
Boxes	0.1024	0.0931	0.0924	0.1410	0.1597	0.1037
Lego	0.1403	0.1423	0.1423	0.1933	0.2364	0.1493
Sunflower	0.1289	0.1432	0.1428	0.2121	0.2599	0.1224
Glass	0.1014	0.0965	0.0963	0.1569	0.1921	0.1019
Toy	0.1157	0.1057	0.1048	0.1333	0.1537	0.1142
Blocks	0.0914	0.0931	0.0929	0.1322	0.1481	0.1010
Boy	0.1342	0.1809	0.1802	0.2758	0.3129	0.2566
Dalmatian	0.1487	0.1377	0.1371	0.1874	0.2191	0.1832
Duck	0.1174	0.1393	0.1386	0.2170	0.2771	0.1401
Exina	0.1361	0.2126	0.2076	0.3866	0.4011	0.4058
Cub	0.1261	0.1745	0.1727	0.2943	0.3230	0.2757
Hotairballoon	0.0894	0.0970	0.0962	0.1508	0.1748	0.1166
Hotairballoon2	0.0475	0.0440	0.0440	0.0861	0.1691	0.0342
Toytruck	0.0839	0.0751	0.0743	0.1644	0.2181	0.0748

Chapter 6

CONCLUSION

A new method for detecting edges in images was proposed by using the universal concept behind the Pythagorean Theorem and is based on the applications of geometric surface two-sheeted hyperboloid. The proposed method uses a triangular kernel which computes values for every possible direction first, followed by mutually subtracting each resulting image. After this, thresholding is applied in resultant images to generate binary image. Morphological operation thinning is applied as post processing step for refining the edges furthermore. Final results show that the proposed method has comparatively consistent performance on various test images, which is competitive with traditional edge detecting algorithms. It was also proved that use of edge detection method also strongly depends upon the application domain as well.

REFERENCES

[1] R.C. Gonzalez, R.E.Woods, *"Digital Image Processing"*, Pearson Education, 2002.

[2] Berry, K.J. and P.W. Mielke, *"A generalization of Cohen's kappa agreement measure to interval measurement and multiple raters"*, Educational and Psychological Measurement, 1988. Vol. 48, No. 4, p. 921-933.

[3] Prewitt, J.M.S., *"Object Enhancement and Extraction"*, Picture processing and Psychopictorics, Academic Press, 1970.

[4] Sotak Jr, G.E. and K.L. Boyer, *"The laplacian-of-gaussian kernel: A formal analysis and design procedure for fast, accurate convolution and full-frame output"*, Computer Vision, Graphics, and Image Processing, 1989. 48(2): p. 147-189.

[5] Canny, J., *"A Computational Approach To Edge Detection"*, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. Pami-8, No. 6, November 1986, p. 679–698.

[6] Sharma, B.D. and I.J. Taneja, *"Entropy of type (α, β) and other generalized measures in information theory"*, Metrika, 1975. 22(1), p. 205-215.

[7] Sugata Ghosal and Rajiv Mehrotra, *"Detection of Composite Edges"*, IEEE Transactions On Image Processing, Vol. 3. No. I, January 1994, p. 14-25.

[8] Russell C. Hardie and Charles G. Boncelet, *"Gradient-Based Edge Detection Using Nonlinear Edge Enhancing Prefilters"*, IEEE Transactions On Image Processing, Vol. 4, No. 11, November 1995, p. 1572-1577.

[9] Alberto De Santis and Carmela Sinisgalli, *"A Bayesian Approach to Edge Detection in Noisy Images"*, IEEE Transactions On Circuits And Systems—I: Fundamental Theory And Applications, Vol. 46, No. 6, June 1999, p. 686-699.

[10] C. Grecos, J. Jiang and E. A. Edirisinghe, *"Two Low Cost Algorithms For Improved Diagonal Edge Detection"*, JPEG-LS IEEE Transactions on Consumer Electronics, Vol. 47, No. 3, August 2001, p. 466-473.

[11] Scott Konishi, Alan L. Yuille, James M. Coughlan, and Song Chun Zhu, *"Statistical Edge Detection: Learning and Evaluating Edge Cues"*, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 25, No. 1, January 2003, p.57-74.

[12] Zhang, Guoliang Zhao and Li Su, *"A New Edge Detection Method in Image Processing"*, Proceedings of ISCIT, 2005, p.430-433.

[13] J.-A. Jiang, C.-L. Chuang, Y.-L. Lu and C.-S. Fahn, *"Mathematical-morphology-based edge detectors for detection of thin edges in low-contrast regions"*, IET Image Processing, Vol.1, No. 3, 2007, p. 269–277.

[14] Olivier Laligant and Fre´de´ ric Truchetet, *"A Nonlinear Derivative Scheme Applied to Edge Detection"*, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 32, No. 2, February 2010, p.242-257.

[15] Sos S. Agaian, Karen A. Panetta, Shahan C. Nercessian, and Ethan E. Danahy, *"Boolean Derivatives With Application to Edge Detection for Imaging Systems"*, IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 40, No. 2, April 2010, p.371-382.

[16] Yang Chen, *"Edge Detection Using Logarithm Function"*, International Congress on Image and Signal Processing (CISP), 2010 , p.1378-1381.

[17] Judith D. Sally and Paul Sally, *"Chapter 3: Pythagorean triples- Roots to research: a vertical development of mathematical problems"*, American Mathematical Society Bookstore, ISBN 0-8218-4403-2, 2007

[18] http://betterexplained.com/articles/surprising-uses-of-the-pythagorean-theorem

[19] https://en.wikipedia.org/wiki/Metcalfe%27s_law

[20] Daróczy, Z., *"Generalized information functions"*, Information and control, Vol. 16, No.1, 1970, p. 36-51.

[21] Iva Bogdanova, Xavier Bresson, Jean-Philippe Thiran, *"Scale Space Analysis and Active Contours for Omnidirectional Images"*, IEEE Transactions On Image Processing, 2007, p.1-14.

[22] http://www.mathworksheetscenter.com/mathtips/logarithms.html

[23] Davis, L.S., "*A survey of edge detection techniques",* Computer Graphics and Image Processing, Vol. 4, No.3, 1975, p. 248-270.