

FPGA Implementation of Image Enhancement Techniques for Image Processing: A Review

Om Parkash, Rajni

Department of Electronics and Communication
Deenbandhu Chhotu Ram University of Science & Technology
Murthal, Sonipat, India

Abstract— The techniques used here for Image Enhancements are to improve the quality of image for other purposes. In Image Enhancement, they have been processing an image in order to get more suitable result for specific application than the real image. A real time configurable system for DIP has been presented in Verilog-HDL description and has also focused on the techniques for image enhancement like Brightness adjustment, Contrast adjustment, Negative transformation, thresholding, Edge detection and filtering on field programmable gate array (FPGA). The solution provides an innovative hardware implementation, simulation and synthesis on FPGA. These Techniques are successfully done on real image in Verilog language using Xilinx ISE, Matlab, Vivado software.

Keywords: Verilog, FPGA, Image Enhancement, Real time configurable system, Xilinx.

I. INTRODUCTION

When analysis and manipulation is done of the telecom data digital image processing plays a vital role. We have been able to fetch the digital data from medical and satellite remote sensing in particular, with the help of digital image processing. There are two important components of digital signal processing, namely, Image enhancement and information extraction.

The visibility of any part of the image or feature can be improved using the image enhancement technique. There are a number of different techniques used in image enhancement processes in order to improve the visual appearance of an image. In the meantime, image enhancement helps in improving the appearance of an image by raising the dominance of some pictures or lowering ambiguity in the middle of different areas of the image. We have two categories of Enhancement methods:

1. Spatial domain methods
2. Frequency domain methods

Spatial domain techniques, deal with the image pixels. In order to achieve the desired increase, pixel values are manipulated. Talking about frequency domain methods, the image is transferred to the first instance domain. This implies that firstly, the Fourier conversion of the image is calculated. Once the Fourier transformation is done then only all the enhancement operations are done after that consequent Fourier transform is done in order to get the resulting image. The 8-bit digital grey image can have pixel value ranging from 0 to 255.

In large and expanded areas, we use digital image processing to cover applications in art, multimedia services, medical, monitoring, space exploration certification, automated industry inspection and many other areas. S.Vamya, et.al related to the implementation of the image enhancement algorithm such as Brightness Control, Contrast Stretching and Histogram Equalization on FPGA. The minimum Period for the implemented algorithm of an image size of 100x100 is 5ns. Nitin Sachdeva et.al gives a design of real-time histogram equalization circuit to increase the images using FPGA. Varsha S .et.al, tells about the implementation of hardware of negative changes, thresholding and Contrast stretching with the help of Spartan-3E boards. And the Final images shown on the VGA monitor. Terek M. Bittabassi et.al Proposes the different algorithm for image enhancement using FPGA. Like median filter, contrast stretching, histogram equalization, negative image change and Power law change. All these design algorithms were applied to 100x100 grayscale 'lena' image on cyclone-III FPGA-based hardware. Enhancement algorithm with the help of (XSG) Xilinx system generator tool.

First of all, MATLAB is used for converting the image here into a text or coefficient file. Digital design is used to process this data and the output which we get after processing is then converted into an image. Verilog HDL is used for the implementation of various digital image enhancement algorithms.

The proposed algorithms are shown through the block diagram given in Figure1.

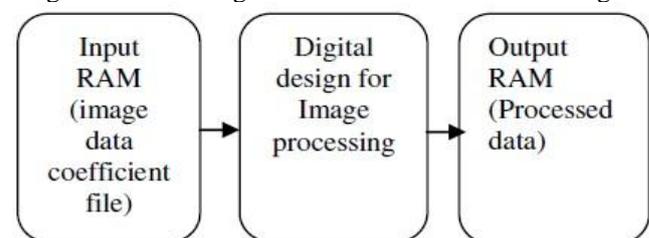


Figure 1: Block diagram of system

Implementation of image enhancement algorithms in this paper includes: brightness control, negative transformation, contrast stretching, thresholding and filtering techniques. Why is the need to Enhance the image, one of the examples is that for a medical imaging a picture of a retinal fundus image can be taken that is important for analyzing eye-related diseases and early detection. Enhancement of such an image is necessary for exposing features to detect abnormal eyes.

Therefore, in order to diagnose and treat diseases like diabetic retinopathy, glaucoma etc, we use it.

Techniques

II METHODOLOGY

1) Brightness Control, 2) Contrast Stretching, 3) Negative Transformation, 4) Thresholding, 5) Filtering Techniques, 6) Edge detection. The above are the image enhancement algorithms explained briefly under this section.

A. Brightness Control

This is the process where a constant value is added to image pixels which have poor brightness to enhance the grey level of every pixel. Due to the poor brightness of digital image, the objects in the image doesn't appear clearly. This happens because the image has been captured under low light conditions. In order to tackle this issue, the digital image's brightness level can again be increased and the image can become more tempting. A dark image with low brightness level can have an increased brightness level, if a constant is added to the grey value of every pixel. By this add operation, the histogram will be shifted towards the brighter side with a constant factor since a constant is added. Before applying this method in order to enhance the brightness level of the digital image, the constant should be wisely chosen so that the complete range for the grey values lies between 0 to 255. Suppose we have the final grey value more than 255 then we will lose the information.

This all works in an algorithm and the algorithm works as follows:

$$J(e) = I(e) + g, \text{ if } I(e) + g \leq 255$$

$$J(e) = 255, \text{ if } I(e) + g > 255$$

g -should be greater than 0 and is a constant value. $I(r)$ -is the input pixel(r) of grey level and $J(r)$ is the output pixel (r) of the grey level after we perform the process of increasing the brightness.

B. Contrast Stretching

Contrast stretching makes an attempt to improve the quality of the image by elaborating or stretching the intensity value's range that it contains to utilize fully all the possible values. The process of stretching has been restricted to linear mapping and also to the linear mapping of input to the output values. Suppose we have low light conditions and we are getting a low contrast image due to that, then the contrast stretching process and the absence of dynamic range of the sensor of camera provides a good quality image. In order to change the contrast or stretching the contrast we basically are increasing the dynamic range of the images grey values which are being processed. To achieve contrast stretching we need to use a piecewise transformation function. Contrast stretching algorithm operate as follows:

$$J(e) = \{I(e) - f_{min}\} \frac{(max - min)}{(f_{max} - f_{min})} + min$$

$I(e)$ is the input pixel's (e) grey level and $J(e)$ is the output pixel's(r) grey level after the operation of contrast stretching has been performed. The maximum and minimum grey level values of the input are f_{max} and f_{min} respectively. The maximum and minimum grey levels determines the grey

level range of the output image and are denoted by max and min respectively.

C. Negative Transformation

Several applications are available in which the digital image's negatives are quiet useful. For instance, when we are displaying the medical images and at the time of capturing a screen with monochrome positive film by using the negatives and not the normal slides. The transformation function has been used to get the desired negatives of the digital image:

$$J(e) = (L-1)-I(e)$$

Here L is used to show the number of grey levels. The grey level of the input pixel is $I(e)$ and the grey level of the transformed output is $J(r)$. The main idea or strategy behind decreasing the intensity of the output image by increasing the intensity of the input is to flip the order from black to white.

D. Thresholding

When we transform all the pixels of an image into two values only then it is called as Thresholding. Here we are performing a unique form of quantization where we have compared the pixel values and the threshold value. When Thresholding is performed, the output image for 8bit grey level comprises of only two values that is 0 and 255.

$$J(e) = 0, \text{ if } I(e) \leq T$$

$$J(e) = 255, \text{ if } I(e) > T$$

The results from a thresholded image is an image that is white and black, and we get only two grey level values.

E. Filtering techniques

We have the coordinates of each pixel as (x,y) and in the enhanced images the value for these pixels is the resultant of the operations being performed on the pixels in the neighbourhood of (x,y) in the input image $I(x,y)$. There can be any random shaped Neighbourhoods, mostly they are rectangular. The noise of the input image can be removed using the filtering operations.

(i)Image smoothing: The objectives of image smoothing are: to diminish camera noise effects, missing pixel values, spurious pixel values etc. we have several ways for image smoothing. We will be considering edge-preserving smoothing and neighbourhood averaging.

a. Neighbourhood averaging (Mean filter): From average pixel value in the neighbourhood of (x,y) in the input image gives rise to each pixel in the smoothed image $J(x,y)$ is resultant from the average pixel value in a neighbourhood of (x,y) in the input image.

For example, for 3X3 neighbourhood found around each pixel, the following mask is utilized: $1/9 \ 1/9 \ 1/9$

$$1/9 \ 1/9 \ 1/9$$

$$1/9 \ 1/9 \ 1/9$$

Here $1/9$ is multiplied by each pixel, added, and then the result produced after this multiplication has been placed in the output image. The mask is made to move across the image unless and until all the pixels are covered. This implies that the image has been convolved for the purpose of image soothing, the image has been convolved with the soothing mask.

b. Edge preserving smoothing (Median filter): Neighbourhood averaging leads to blur images and which is

indirectly the result of the attenuated high frequencies of the image. We can use an alternating approach in order to solve this issue, the alternative is median filtering. Under this approach, the median of the pixel values has been set to the grey level in the neighbourhood of that pixel. For instance, if the pixel values in a 3x3 neighbourhood are (10, 20, 20, 15, 20, 20, 20, 25, 100). By sorting these values, we get (10, 15, 20, 20, 20, 20, 25, 100) and hence the value of median is 20.

F. Edge Detection

The fundamental tool for image segmentation is Edge detection. Under this method, the transformation of original images into edge images, this gets benefitted from the changes in the image when the grey levels are toned in the image. When it comes to the concept of processing the image mainly in computer vision, the localization of the important variations of the grey level image is treated through the process of edge detection and then there goes a process which is for analyzing the detection of the geometrical and the physical properties of objects in the scene. All this is working as the fundamental and basic process which detects as well as provides the outlining of the object and boundaries among the objects along with the background in the image. We can detect the significant discontinuities, by using the Edge detection which is the most familiar and basic approach, in intensity values.

The local changes in the intensity of the image are called the Edges. Suppose there are two regions, then at the boundary of those two regions will occur the edges. We can extract the main features from the edges of an image. The image analysis can be more easy through a special feature found in Edge detection. The access to these features is possible via advanced computer vision algorithms. The process of edge detection is used for several applications as in biometrics, medical image processing etc. since edge detection is used for object detection. The edge detection is the active area for research since it provides higher levels for analysis of image. There are 3 distinct types of discontinuities found in grey level and they are point, edges and line. We can use the spatial masks for detecting these discontinuities of the image. In literature there exists so many techniques for edge detection for the process of image segmentation. Here we have reviewed the most rarely used techniques for edge detection based on discontinuity. These techniques are as follows: Sobel Edge Detection, Roberts edge detection, Kirsh Edge Detection, Marr-Hildreth Edge Detection, Prewitt edge detection, Canny Edge Detection, Robinson edge detection and LoG edge detection.

a) Sobel Edge Detection

Sobel introduced the Sobel edge detection method in 1970 (Rafael C. Gonzalez (2004)). The Sobel edge detection method helps in finding the edges by using a method of Sobel approximation to the derivative. Here the edges only on the points where we get the highest gradient. The technique used here is helpful in performing a 2-D spatial gradient quantity over an image also providing us with some highlighted regions where the spatial frequency is high corresponding to the edges.

In an n input greyscale image on every point, this method is helpful in determining the approximated absolute gradient magnitude. In a hypothesis, operator have paired 3x3 complication kernels at least and they are tabulated as follows. Every next kernel is 90 degree rotated by the preceding one. This method is not similar with that of Roberts cross operator method.

b) Roberts Edge Detection

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

 G_x

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

 G_y

Lawrence Roberts (1965) introduced the Roberts Edge detection technique and that's the reason why it has been named as Roberts edge detection. It can be used to perform 2-d spatial gradient measurements as well as simple and quick to compute measurements. This method can be used to focus the areas of high spatial frequencies like the edges. One of the most common utilization of this is the similarity of the grey scale image given as input to the operator and output. The approximated complete magnitude taken of the spatial gradient of image at the input of a given point is represented by the pixel values at that point in the output.

-1	0
0	+1

 G_x

0	-1
+1	0

 G_y

c) Kirsch Edge detection

The kirsch edge detection method was proposed in 1971 by Kirsch. Under this technique a masks are used but they are considered as one mask and there are eight compass directions where this mask is rotated and the eight directions are as follows: North, South, East, West, Northwest, Southwest, Southeast and Northeast. The masks have been differentiated as follows:

$$\begin{array}{cccc}
 k_0 & k_1 & k_2 & k_3 \\
 E = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & NE = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} & N = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & NW = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\
 k_4 & k_5 & k_6 & k_7 \\
 W = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & SW = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} & S = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & SE = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}
 \end{array}$$

When each mask is convoluted with the image, the maximum value that we get is the edge magnitude. And in order to define the direction, the mask that produces the highest magnitude is considered. For instance, if we have two

masks, one corresponding to a vertical edge and other corresponding to a diagonal edge and they are k_0 , k_5 respectively. Note that there the last four masks have been considered exactly equal or same as the first four but there is a central axis to which they have been flipped.

d) Prewitt Edge Detection

In 1970, Prewitt (Rafael C. Gonzalez) proposed Prewitt Edge Detection. It is a right way to calculate the orientation of an edge and its magnitude.

The different edge detection, even though needs a time taking calculation while estimating direction from magnitude in x and y directions, the direction is directly obtained from kernel that too in highest response by compass edge detection. Its limitation is for 8 possible directions; however, studies say that the most direct directional estimation is not much perfect. 3×3 neighbourhood is used for estimating gradient based edge detector for eight directions.

Once the eight convolution masks get calculated, one of the complication mask is selected, namely having purpose of biggest module. Although Prewitt detection seems to be simpler in computational implementation than Sobel detection but the result is somewhat noisier.

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

 G_x

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

 G_y

e) Canny Edge Detection Algorithm

The Canny edge detection algorithm is also called as the optimal edge detector. Canny wanted to increase the performance of the edge detectors which were there in market when he initiated his work. And the efforts he made were not wasted, he got success and could accomplish his goals, his ideas as well as the methods can be very clearly seen in his paper, "A Computational Approach to Edge Detection"[11]. He followed a list of agenda's in his paper in order to work for the enhancement of existing techniques for the edge detection. Firstly, he started with the low error rate. It was mandatory that there should be not just any response at all to the non-edges and all the edges of an image should be covered without any edge getting missed. And the next concept used was that the edge points needs to be localized. We can also call it as, the distance calculated by detector between different edge pixels and the real edge needs to be at minimum. The third concept as per the Canny's theory is that there should exist only single response for a single edge. This idea was utilized because the other two ideas were not that considerable enough in order to fully remove the chances of having numerous acknowledgements to an edge. Depending upon this concept, firstly the canny edge detector is utilized in order to provide a smoothen the image and remove the noise from the image. Then the image gradient needs to be searched in order to put light on regions where there are high spatial derivatives. The set of rules then blocks

along these areas where there are high spatial derivatives and the pixel which is not at the maximum is defeated. This can also be termed as non-maximum suppression. A series of gradient needs to be further reduced via hysteresis. Hysteresis is a way which can be utilized to track through the pixels left and which have not been defeated. There are two thresholds getting used in hysteresis, suppose the magnitude of the second is less than the first threshold, the value is set as zero and it makes a non-edge. But suppose the magnitude is more than the high threshold, then an edge is made. Also if we have the magnitude in the middle of the two thresholds, hence the value is set as zero otherwise there exists a path from this very pixel to the pixel whose gradient value is more than T_2 .

For the implementation of canny edge detector method, it follows a sequence of rules or steps that are explained as follows.

Step 1: -

Firstly, the idea is to eliminate the noise from the original image before the attempt is made for locating and detecting the edges. Since the Gaussian filter can be calculated via a simple mask, this is utilized only in this canny algorithm. When a perfectly suitable mask is computed then only the Gaussian smoothing is done by making use of the standard methods of convolution. The actual image is bigger than the convolution mask. This results the mask is moved smoothly on the image and changing a square of pixels at an instant. The width of Gaussian mask and detectors sensitivity to noise are inversely related, more the width, lesser is the detector's sensitivity towards noise. With the rise in the Gaussian width, the localized error in the detected edges is also increased.

Step 2: -

Once we smoothen the image and the noise is eliminated, then we are left with finding the strength of edge with the help of calculating the gradient of the image. As we already know that, the Sobel operators are helpful if we want to measure on 2-D spatial gradient of an image. After that it is easy to find the estimated absolute gradient magnitude or the edge strength at any point. When we talk about the Sobel operator [a], we actually know that it is making use of a tong of 3×3 convolution masks, one of them is used for analysing the gradient of column(x-direction) and the other one is used for measuring the gradient in the rows(y-direction). This can be explained as follows:

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

 G_x

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

 G_y

We can approximate the magnitude of the gradient or the edge strength by making use of the following formula:

$$|G| = |G_x| + |G_y|$$

Step 3: -

By making the usage of gradient in x and y directions, the edge directions can be calculated. However, if the sum X is zero, it will introduce errors. Hence when it comes to code,

there needs to set a restriction when such things take place. Based on the value of gradient we have in y-direction, every time we get the value of gradient in x-direction as zero, the direction of the edges will have to be equal to 90 or 0 degrees. And whenever the gradient value in in y-direction is zero, the direction of edge will be 90 degrees. We use the given formula when we want to find out the direction of edges: $\text{Theta} = \text{invtan} (G_y / G_x)$

Step 4: -

Suppose that we are known to the direction of the edge, then the next is to relate the direction of the edge with the direction that can be traced by looking at the image. Therefore, if we think of the pixels in a 5x5 image, they can be aligned as below:

```

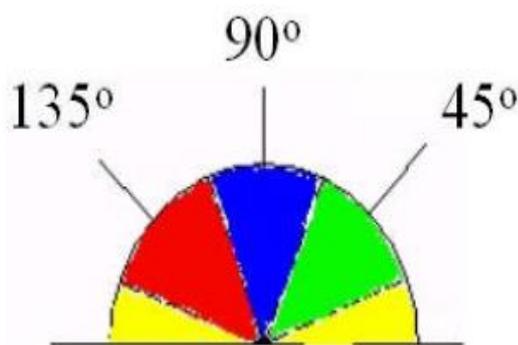
X   X   X   X   X
X   X   X   X   X
X   X   a   X   X
X   X   X   X   X
X   X   X   X   X

```

From this, we can look at the pixel "a", the surrounding pixels can be described by making use of the available and possible 4 directions –

- 0 degrees (in the horizontal direction)
- 90 degrees (in the vertical direction)
- 45 degrees (along the positive diagonal)
- 135 degrees (along the negative diagonal)

Hence the edge orientations need to be one among the four towards the direction it is more close to (i.e. if we have the angle of orientation as 3 degrees then change it to zero degrees). Now in order to understand it in much better way, assume that there exists a semicircle divided into 5 regions.



Hence, if any of the direction of edges falls in to the range of yellow (0 to 22.5 & 157.5 to 180 degrees) is made to be at zero degrees. The direction of edges falling into the blue range (67.5 to 112.5 degrees) is made to be 90 degrees. If there is any direction of the edge that falls in the range of green (22.5 to 67.5 degrees) is made to be 45 degrees. And at last but not the least, if any of the edges falls into the range of red (112.5 to 157.5 degrees) is made to be 135 degrees.

Step 5: -

When we have the information regarding the direction of edges, then the non-maximum suppressions need to be applied. We use these suppressions in order to trace the edges along the direction of edges and hence can suppress the value

of any pixels (sets it equal to 0) which is not assumed to be an edge. We get fine lines as the output image.

Step 6: -

At the final stage, hysteresis [12] process can be utilized in order to remove the streaking. There occurs a fluctuation above and below the threshold in the operators output resulting into smashing up of the outlines of the edges and this entire thing is termed as Streaking. If there exists only one threshold i.e. T1 and is applied to an image and the average strength of the image was also equal to T1, hence noise will lead to some instances where edge immerses lower than threshold. And as much as it is immersed lower the threshold the same it will be more than the threshold and creating an edge which will form a dashed line. In need of avoiding this, in the process of hysteresis, two thresholds are used, one is a low and another is high. The pixels in the image whose value is more than the threshold value T1 is expected to be the pixel on the edge and is pointed as such instantly. In the similar manner, if there is any other pixel that has the value more than the threshold value T2 are also marked as edge pixels. And supposing that we want to follow the edge then there is requirement for gradient T2 to begin with but this only discontinues when it hits a gradient lower than T1.

f) Robinson Edge detection

Much like the Kirsch mask, is Robinson Edge detection (Robinson 1977), but it is much easier in implementation as it is dependent only on the coefficients of 0, 1 and 2. The masks are similar at their zero and directional axis.

$$\begin{array}{cccc}
 r_0 & r_1 & r_2 & r_3 \\
 E = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} & NE = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} & N = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} & NW = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \\
 r_4 & r_5 & r_6 & r_7 \\
 W = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} & SW = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} & S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} & SE = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}
 \end{array}$$

By computing the results of four masks one can easily negate and obtain the results of other four masks. We show the masks as given below:

The largest value obtained when every mask is applied to pixel neighbourhood is known as magnitude of gradient, while the mask which yields maximum response, its angle of line zeroes is approximated to calculate angle of gradient.

g) LoG edge detection

In LoG edge detection method, LoG stands for Laplacian of Gaussian. And this method of edge detection was introduced in 1982 by Marr. The second order derivative is the Laplacian of Gaussian of an image $f(x, y)$ and is defined as,

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

This method has its two types of effects, one is that it gives smoothness to the image and another is that it helps in calculating the Laplacian, giving rise to a double edge image. In order to locate the edges, we need to search for the zero crossings in the mid points of these double edges. We can implement the Laplacian function digitally by utilizing the masks given below,

0	-1	0
-1	4	-1
0	-1	0

 G_x

-1	-1	-1
-1	8	-1
-1	-1	-1

 G_y

By using the Laplacian, we get help in finding things like where is the pixel located, is it on the darker side or on the lighter side of an edge.

III Results

This section presents a performance relative to various edge identification technologies like Sobel edge Detector, Roberts Edge detector, Prewitt Edge detector, LoG edge detector, Kirch, Robinson and Kenny Edge detector. Using the MATLAB R2017a, the edge recognition technology was implemented, and was tested with an image (University of Bharathiar). The objective is to create a clean edge map to remove the major edge features of the image. The image received by using the original image and various edge detection techniques is given in Figures



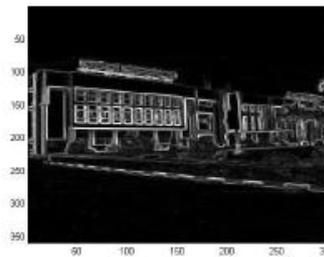
Original



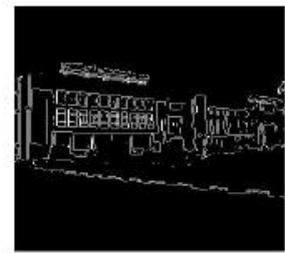
Sobel



Roberts



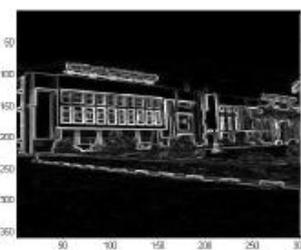
Kirsch



Prewitt



Canny



Robinson

Roberts, Sobel and Prewitt results actually wander from others. LoG and Canny show a nearly identical edge map. The Kirsch and Robinson edge maps are almost the same. From figures we can see, Canny's result is far better than other results.

CONCLUSION

According to the computer vision, the fastest moving field is image processing. By making use of technological process in digital imaging, in computer processor, also in mass storage devices, the growth of image processing has been increased. Here by using this review, an attempt is made to review the edge detection technique which is dependent on the stages of imbalance intensity. Using the MATLAB software, the comparative display of different edge detection techniques is done with an image. In order to get this result, we observe the comparative displays by generating the Canny and Log edge detectors almost the same edge map. When the result of the Canny's detector is upper in level to everyone for any selected image, because the dismantling of different edges works better under different conditions. However, there is a lot of advancement techniques available in the literature, because for the communities involved in research, finding exactly the image without any noise in the original image is a daunting process.

REFERENCES

- [1.] Abdul Manan" Implementation of Image Processing Algorithm on FPGA" AKGEC JOURNAL OF TECHNOLOGY, Vol. 2, No. 1
- [2.] Nirmala S. O, T. D. Dongale, R. K. Kamat" Review on Image Enhancement Techniques: FPGA Implementation perspective" International Journal of Electronics Communication and Computer Technology (IJECC) Volume 2 Issue 6 (November 2012)
- [3.] K.Surya Anush, B Likhita, I Rohini, D.Chiranjeevulu" MEDICAL IMAGE ENHANCEMENT USING HARDWARE DESCRIPTION LANGUAGE" International

journal of advanced research in computer science vol. no 6,
issue no 4, April 2017

[4.] Qian Xu, Srenivas Varadarajan, Chaitali Chakrabarti” A Distributed Canny Edge Detector: Algorithm and FPGA Implementation” IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 23, NO. 7, JULY 2014

[5.] IULIANA CHIUCHIȘAN, MARIUS CRISTIAN CERLINCĂ” Implementation of Real-Time System for Medical Image Processing using Verilog Hardware Description Language” Recent Researches in Medicine, Biology and Bioscience

[6.] Praveen vanaparthi, Sahitya.G, Krishna Sree and Dr.C.D.Naidu” FPGA IMPLEMENTATION OF IMAGE ENHANCEMENT ALGORITHMS FOR BIOMEDICAL IMAGE PROCESSING” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 11, November 2013

[7.] Iuliana Chiuchisan” A New FPGA-based Real-Time Configurable System for Medical Image Processing” The 4th IEEE International Conference on E-Health and Bioengineering - EHB 2013

[8.] www.altera.com

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

[10.] Avinash G. Mahalle, A. M. Shah” FPGA Implementation of Gradient Based Edge Detection Algorithms” International Journal of Innovative Research in Computer and Communication Engineering Vol. 5, Issue 5, May 2017

[11.] Muthukrishnan.R and M.Radha” EDGE DETECTION TECHNIQUES FOR IMAGE SEGMENTATION” International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 6, Dec 2011

[12.] Nikhat Fatema Sheikh, Prof. Ruhina Quazi “FPGA Implementation for Image Edge Detection using Xilinx System Generator” International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 5 Issue: 7

[13.] K.NARESH, M.MAHENDE” FPGA Based Implementation of Image Edge Detection using Canny Edge Detection Algorithm” International journal of advanced research in computer science ISSN 2319-8885 Vol.03,Issue.29 October-2014

[14.] Avinash G. Mahalle” A Review on FPGA Implementation of Edge Detection Algorithms” International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064