International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-2 Issue-6, August 2013

Implementation of Canny Edge Detection Algorithm on FPGA and displaying Image through VGA Interface

Chaithra.N.M., K.V. Ramana Reddy

Abstract –Edge detection is one of the most important stages in image processing. The Canny edge detection algorithm is most widely used edge detection algorithm because of it advantages. In this paper we present canny edge detection algorithm implemented on Spartan 3E FPGA and developed VGA interfacing for displaying images on the screen. In this paper we have taken 128×128 Image and displayed same on the monitor through FPGA.

Keywords: Block Memory, Canny, FPGA, VGA Interface.

I. INTRODUCTION

Edge detection is the basic operation in image processing and has wide application in research area. Many edge detection algorithms have been proposed such as Robert detector, Prewitt detector, Kirsch detector, Gauss-Laplace detector and Canny edge detector. Due to its good performance Canny algorithm has been widely used in the field of image processing [1].

Edge detection is a very important area in the field of Computer Vision. Edges define the boundaries between regions in an image, which helps with segmentation and object recognition. They can show where shadows fall in an image or any other distinct change in the intensity of an image [2].

The recent studies on Canny edge detection algorithm shows that the traditional Canny edge detector has two shortcomings. The threshold of the algorithm needs to be set by manual. Secondly, the algorithm is very time consuming and cannot be implemented in real time. A new self-adapt threshold Canny algorithm is proposed and a pipelined implementation on FPGA is designed to overcome the above disadvantages. Compared with the implementation in a PC based system, pipelined implementation on FPGA takes much less implementation time and can therefore be used for the mobile robot vision system which is very strict for the real-time performance of its vision system [3].

Generally image processing algorithms are implemented on DSP kits. Image processing algorithms are repetitive in nature and require more computation. The alternative choice is to implement algorithm on the very expensive application specific integrated circuits (ASIC) or Field Programmable Gate Array (FPGA).

Since FPGAs offer the features like reprogram ability flexibility parallelism short development time and computational power these FPGAs are best suited to implement image processing algorithms [3, 4].

II. CANNY EDGE DETECTION ALGORITHM

Canny developed an approach to derive an optimal edge detector based on three criteria for edge detection [5].

i. Low error rate of detection. It should find all edges and nothing but edges.
ii. Localization of edges. The distance between actual edges in the image and the edges found by the algorithm should be minimized.
iii. Single response. The algorithm should not return multiple edge pixels when only a single edge exists[5].

The model was based on a step edge corrupted by additive white Gaussian noise. The original Canny algorithm [6] consists of following steps:

a. Smoothing the input image by Gaussian mask. This eliminates the high frequency components in the image. The output smoothed image is denoted as I(x, y).
b. Calculating the horizontal and vertical gradient Gx(x, y) and Gy(x, y) respectively at each pixel location by convolving the image I(x, y).
c. Computing the gradient magnitude G(x, y) and direction Θg(x, y) at each pixel location.
d. Applying non-maximum suppression (NMS) to thin the edges.
e. Computing the hysteresis high and low thresholds based on the histogram of the magnitudes of the gradients of the entire image.

Performing hysteresis thresholding to determine the edge map. In this paper VGA interfacing is done and the image is stored in block ROM on FPGA. Accessing the data from PC is a time consuming process and this problem can be overcome by storing the data/image on block ROM of FPGA and an independent system can be developed using this approach.

III. HARDWARE IMPLEMENTATION

The block diagram of the project is as shown in the Figure 3.1. The input image is gray scale image of size 128x128 each pixel of 8 bits. The Coefficient (coe) file of the input image is generated using MATLAB. This coe file is stored in Block ROM which is processed using the canny edge detection algorithm. The resulting image is displayed on monitor using VGA interfacing.

Manuscript published on 30 August 2013.
***** Correspondence Author (s)
Chaithra.N.M, VLSI Design and Embedded Systems, Visveswaraiah Technological University/ VTU Extension Centre, UTL Technologies Ltd. /Bangalore, India.
Asst.Prof K.V.Ramana Reddy, VLSI Design and Embedded Systems, Visveswaraiah Technological University/ VTU Extension Centre, UTL Technologies Ltd./ Bangalore, India.

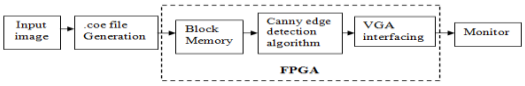

Figure 3.1. Block diagram of the canny edge detection algorithm with VGA interface

A. *Block Memory*

Spartan 3E FPGA consist of special Block memories. These Block memories help in speeding up the memory operations. Here single port Block ROM is used which can support up to 360Kbits. The width and depth of the design is user defined. The schematic for the block memory is shown in Figure 3.2.

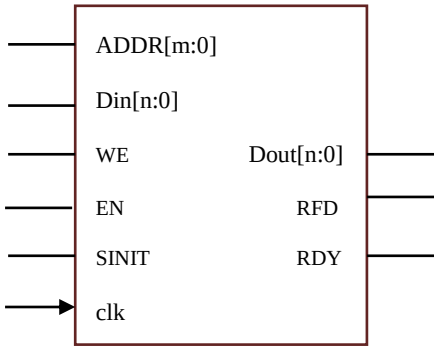

Figure 3.2: Core schematic symbol [7]

The memory operations take place on the active edge of the clock signal (CLK) when the block memory is enabled and when the block memory is not enabled the memory configuration and output remain unchanged. When WE is asserted the input data is stored in the memory at the location selected by the address [7].

B. *Canny Edge Detection Algorithm*

The block diagram of Canny edge detection algorithm is shown in figure 3.3

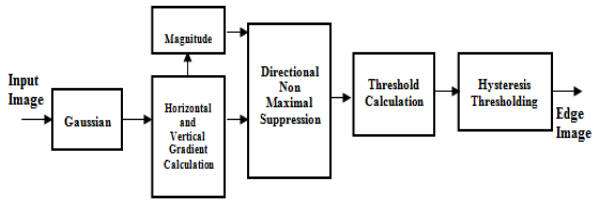

Figure 3.3: Block diagram of canny edge detection algorithm[1]

The block diagram is divided into following modules:

i. Image smoothing
ii. Gradient and Magnitude calculation unit
iii. Directional Non-Maximal suppression unit
iv. Threshold calculation
v. Hysteresis thresholding.

The input to the system is the color image where it is converted to gray scale image. A Coe (co-efficient) file is generated and stored in block memory. The pixel values stored in the Block Memory are given as the inputs to Canny edge detection algorithm. The design uses 3×3 convolution kernels, processing 128x128x8 Gray Scale Image which are internal operations inside the algorithm. Edge image is the final output of the canny edge detection algorithm. Structure and working of each module is explained in following section.

i.Image smoothing.

Smoothing of image is achieved by Gaussian convolution. The image is first passed through low pass filter (LPF) to reduce the noise, i.e. all high frequency components are eliminated. A 3x3 moving window Gaussian operator is used; two FIFO buffers with the depth of one row pixels of image are employed to access all the pixels in the 3x3 window at the same time [6]. Consider a two dimensional image of size 128x128 as input and it is represented as I(x,y) in spatial domain. The pixel values are obtained as array of values and these values are stored in a text file. This text file data is used as input during the smoothing process. Size of the filter has to be chosen very carefully because, as the size of the filter increases filter becomes less sensitive to noise.

Filtering mask is obtained by using the two dimensional Gaussian function， and the function is as follows

$$G(x,y) = \frac{1}{2\pi\sigma^2} \exp^{-(x^2+y^2)/2\sigma^2}$$

Where x and y represents horizontal and vertical values of a two dimensional image, sigma $'\sigma'$ is the standard deviation of the Gaussian function and G(x, y) represents the smoothened image. Sigma plays a major role in the selection of the coefficients of the Gaussian filter.

ii. Gradient Unit

Next step is the gradient calculation unit which is performed using gradient filter. The inputs to the gradient block are the output values of the smoothing unit. This step creates an additional 1-pixel border of invalid data around the input image. The calculation steps used in the algorithm uses Gx =I(x +1,y) - I(x -1,y) and Gy=I(x, y +1)- I(x, y -1) to calculate the horizontal gradient and vertical gradient respectively. Two different masks are used for calculating horizontal and vertical gradient of an image shown in table 3.1a and b respectively.

Table 3.1: Convolution Kernels

0	0.5	0
0	0.5	0
0	0.5	0

a

0	0	0
0.5	0.5	0.5
0	0	0

b

Using the masks mentioned above the gradient calculation is done using the following equations.

Gx= (P2+P5+P8)/2
Gy= (P4+P5+P6)/2

iii.Magnitude Unit:

The horizontal and vertical gradient values obtained in the gradient unit are the inputs to the magnitude block; these gradient values are used for finding the magnitude of the input image. Magnitude of the gradient is calculated using the formula Gmag = $\sqrt{Gx^2 + Gy^2}$.

iv. Directional Non-Maximum Suppression [NMS]:

The architecture of NMS block is shown in figure 3.4. It consists of window 3x3 unit, selector unit, arithmetic and comparator unit. Window 3×3 unit holds the 8 pixel values at a time and feeds the input to the selector unit and the middle value is given to the comparator unit.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

In the selector block angle is calculated at each pixel location simultaneously after the calculation of gradient of an image. Edge direction is taken by considering the inverse tangent function of vertical gradient to horizontal gradient. Θ = arctan (Gy/Gx). The angle Θg(x, y) is calculated using the gradients i.e. arc tan (gy/gx).

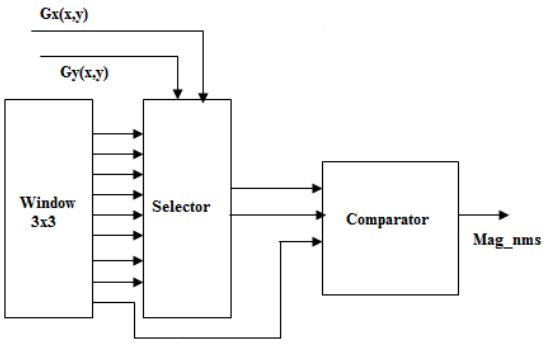

Figure 3.4 :Block diagram of Non Maximal Suppression Unit [1].

v. Hysteresis thresholding:

The architecture of pipelined thresholding is shown in figure 3.5. The output of the non maximum suppression unit contains some spurious edges. Hence the hysteresis thresholding is performed to reduce those effects. The threshold is calculated based on the gradient histogram i.e. we need the histogram of the image after the NMS operation. In non maximal suppression block the operation is performed to thin the edges as well as to detect all possible edges. To identify the true edges from output NMS image two thresholds are considered in this algorithm i.e. the high threshold and the low threshold.

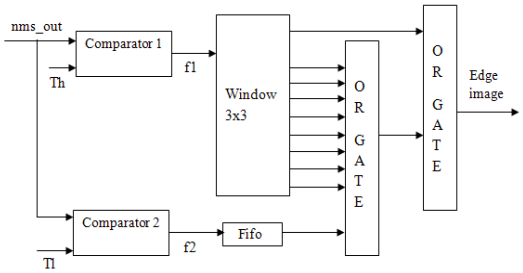

Figure 3.5: Block diagram of pipelined thresholding [1]

C. *VGA Interfacing*

VGA (video graphics array) is the standard video display interface used to connect the FPGA to monitor to display the edge detected image. The standard VGA monitor is of size 640x480. Individual pixels are turned on and off to display the image. Individual pixels turned on and off by scanning the entire column starting from 0 to 639 and row from 0 to 479 as shown in the Figure 3.6.

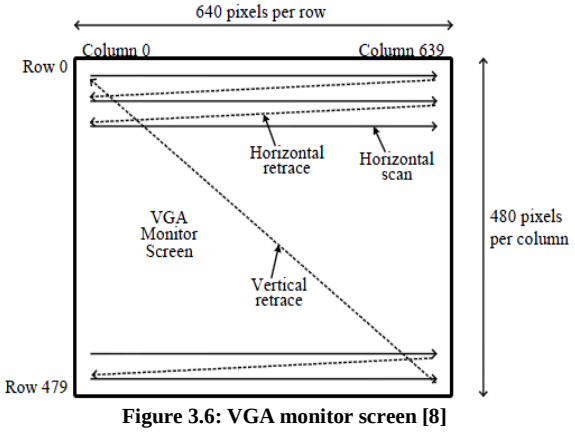

Figure 3.6: VGA monitor screen [8]

In order to control the monitor VGA monitor control circuit uses 5signals V-sync, H_sync, Red_out, Green_out, and Blue_out as shown in the Figure 3.7.

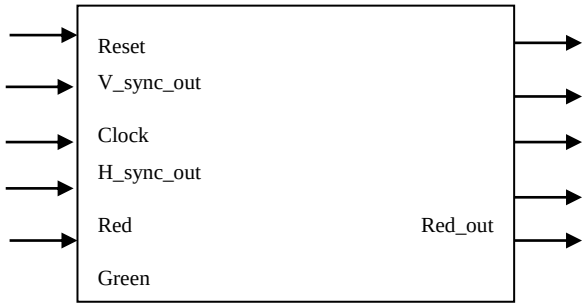

Figure 3.7: The logic symbol for the VGA monitor control [8]

The horizontal and vertical synchronization signals are used to control the scan rate of the monitor. The H_sync signal controls the time it takes to scan the row and V_sync signal is used to scan the entire screen. VGA control circuit works at 25.175MHz of clock. Red,Green and Blue signals are used to control the color of the image pixel displayed on the screen

IV. TESTING AND RESULTS

The design of the canny edge detection algorithm is done in verilog. Design and testing of individual module has been carried out. The simulation result of window 3×3 unit is shown in figure 4.1. The inputs to this block are the pixel values of an image.

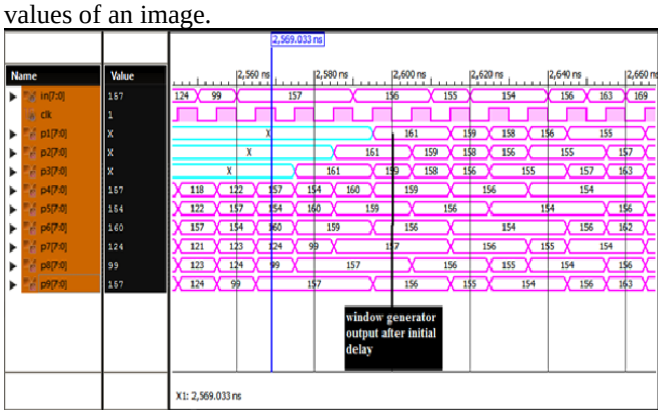

Figure 4.1: simulation result of 3x3 window generator.

The simulation result of gradient unit is shown in figure 4.2. The input to this block is from smoothing unit. These values are convolved with horizontal and vertical convolution masks.

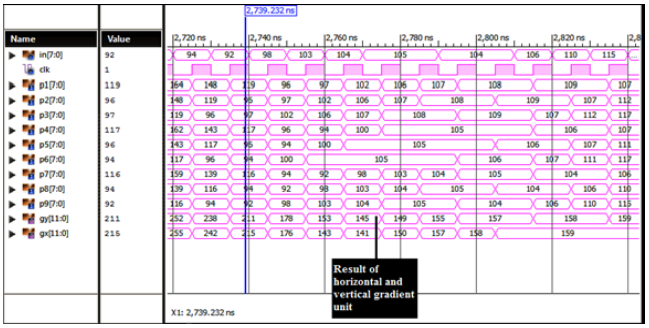

Figure 4.2: Simulation result of gradient calculation unit

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The simulation result for Magnitude calculation block is shown in Figure 4.3. The input to this unit is from gradient unit. Gradient is calculated at each point both in horizontal and vertical direction these values are taken as input to the magnitude unit.

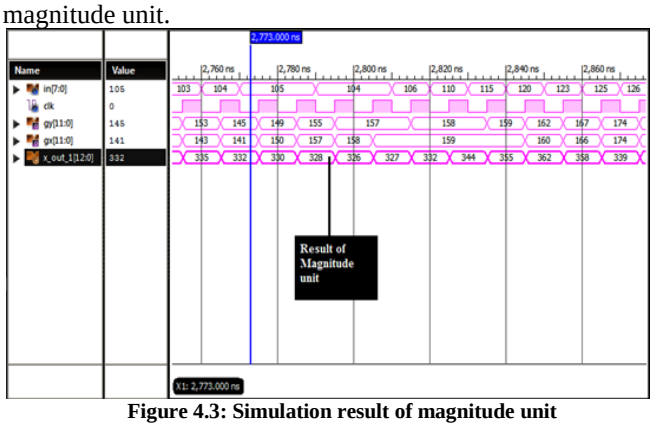

Figure 4.3: Simulation result of magnitude unit

The simulation result for Magnitude calculation block is shown in Figure 4.4. The input to this block is from gradient unit and also from magnitude unit. The obtained angle plays major role in the selection of magnitude from the magnitude unit. The simulation result is shown below.

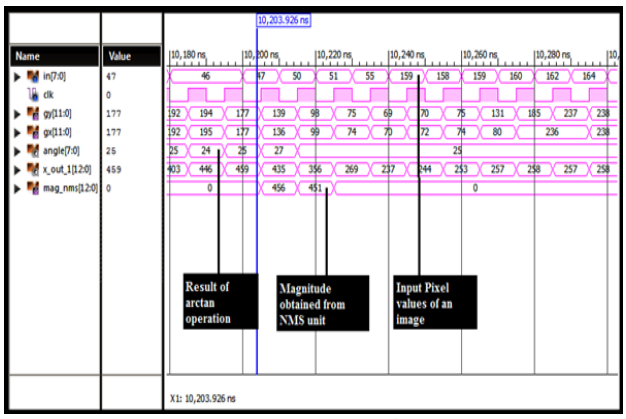

Figure 4.4: simulation result of NMS block.

The simulation result for Magnitude calculation block is shown in Figure 4.5. The output from NMS block is fed as input to this unit and is compared with high and low thresholds to get true and strong edges. The final result obtained after integration of entire block is as shown in below diagram.

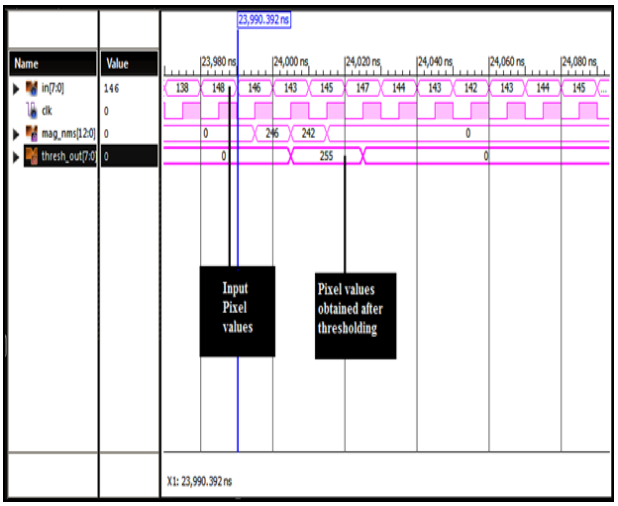

Figure 4.5 Simulation result of thresholding unit.

Different input images and their edge detected images are shown in Figure 4.6. The Figures A, C are the input images and Figures B, D are their respective edge detected results.

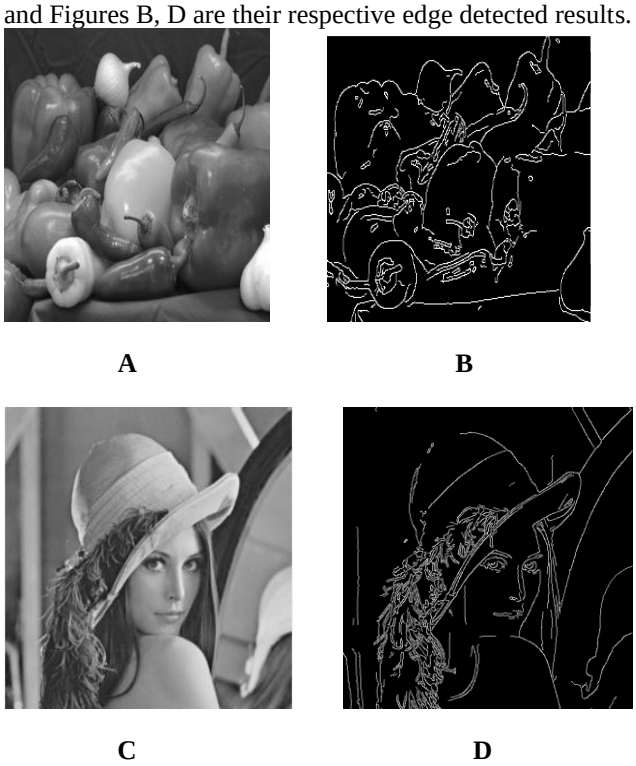

Figure 4.6: Simulation results

Initially we loaded an image of size 128×128 in block memory. Canny edge detection algorithm has been implemented on FPGA and then VGA interfacing is developed. The interfacing of Spartan 3E FPGA with VGA monitor for displaying obtained edge detected image is shown in figure 4.7.

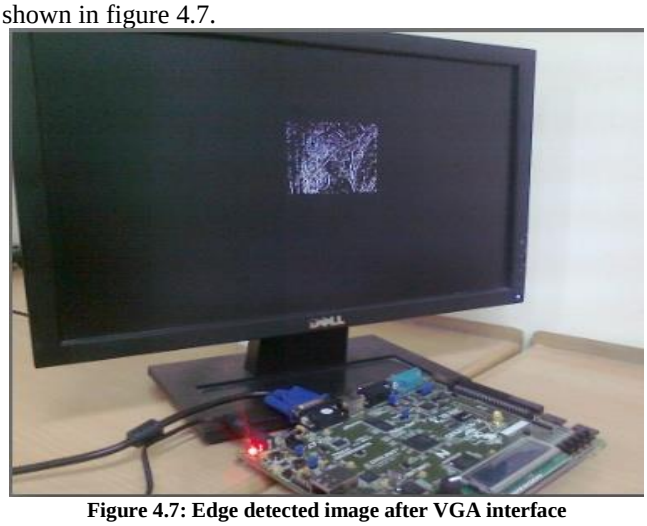

Figure 4.7: Edge detected image after VGA interface

V. CONCLUSION

In this paper we implemented Canny Edge Detection algorithm on Spartan 3E FPGA. VGA interfacing is developed for displaying the images on the monitor. An image of size 128×128 is first stored in block Rom on FPGA and then processed through Canny edge detection algorithm and displayed on VGA monitor.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The entire system is developed simulated and synthesised using Spartan 3E FPGA board.
In future videos can be stored in the memory and video edge detection can be performed camera interfacing can be done to take real time images and the system can be used for security purposes

ACKNOWLEDGMENT

I acknowledge Dr. V. Venkateswarlu, Principal, VTU Extension Centre, UTL Technologies Ltd., Bangalore for his guidance and suggestion and UTL Technologies Ltd., Bangalore for providing lab facility during the design and implementation.

REFERENCES

[1]. Qian Xu, Chaitali Chakrabarti and Lina J. Karam "A Distributed Canny Edge Detector and Its Implementation On FPGA" School of Electrical, Computer and Energy Engineering, Arizona State University, IEEE, 2011, pp. 500-505.
[2]. Parvinder Singh Sandhu, Mamata Juneja and Ekta Walia "Comparative Analysis of Edge Detectin Techniques for extracting Refined Boundaries" 2009 International Conference on Machine Learning and Computing ,IPCSIT vol 3, 2011.
[3]. Wenhao He and Kui Yuan "An Improved Canny Edge Detector and its Realization on FPGA" IEEE Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China, June 25 - 27, 2008, pp. 6561-6564.
[4]. Osman Z.E.M; Hussin ;Ali, N.B.Z "Optimization of Processor Architecture for Image Edge Detection Filter" IEEE transaction on Computer Modelling and Simulation, 2010, pp 648-652.
[5]. Alasdair Mc Andrew. "Introduction to Digital Image Processing with MATLAB".
[6]. Gao Jie and Liu Ning "An improved adaptive threshold canny edge detection algorithm", IEEE International Conference on Computer Science and Electronics Engineering, 2012, pp. 164-168.
[7]. Muralikrishna, B.; Gnana Deepika ,K.; Raghu Kanth, B.; Swaroop Vemana, V.G.; "Image Processing using IP Core Generator through FPGA", International Journal of Computer Applications, vol 46-No.23, May 2012,pp. 48-52.
[8]. Enoch Hwang, "Build a VGA Monitor Controller", Circuit Cellar, Issue 172 , November 2004,pp. 12-17.
[9]. Rafael C. Gonzalez, Richard E. Woods. "Digital Image Processing", Prentice Hall, 2nd edition (January 15, 2002).
[10]. S. Varadarajan, C. Chakrabarti, L. J. Karam, and J. M.Bauza, "A distributed psycho- visually motivate Canny edge detector", IEEE ICASSP, March 2010, pp. 822 –825.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)