ISSN (Online): 1694-0814
www.IJCSI.org

A Novel Architecture for Real Time Implementation of Edge Detectors on FPGA

Sudeep K C[1] and Dr. Jharna Majumdar[2]

[1]Research Associate, Dept. of Computer Science, Nitte Meenakshi Institute of Technology, Bangalore 560 063, Karnataka, India

[2]Prof. and Dean, Dept. of Computer Science, Nitte Meenakshi Insitute of Technology, Bangalore 560 063, Karnataka, India

Abstract

With the introduction of reconfigurable platform such as Field Programmable Gate Arrays (FPGA) and advent of new high level tools to configure them, image processing on FPGA has emerged as a practical solutions for most of computer vision and image processing problems. This paper briefly explains the implementation of several edge detection algorithms like Sobel, Prewitt, Robert and Compass edge detectors on FPGA and makes a comparative study of their performance. The design utilizes powerful design tool System Generator (SysGen) and Embedded Development Kit (EDK) for hardware-software codesign and integrates the edge detection hardware as a peripheral to the Microblaze 32 bit soft RISC processor with an input from a CMOS camera and output to a DVI display and verified the results video in real time.

Keywords: *Field Programmable Gate Arrays (FPGA), Image processing algorithms, Edge detection, System Generator (SysGen), Embedded Development Kit (EDK), Real Time*

1. Introduction

An edge in an image is a contour across which the brightness of the image changes abruptly. However, image data is discrete, so edges in an image often are defined as the local maxima of the gradient. An edge detector is basically a high pass filter that can be applied to extract the edge points in an image. The goal of edge detection is to mark the points in a digital image at which the luminance changes abruptly. The mathematical representation for the same is a convolution sequence given as:

$$g(x,y) = \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f(x-k, y-l)\, h(k,l)$$

Where, f(x,y) is the input image, h(x,y) is filter mask and g(x,y) is the resultant image.

General purpose microcontrollers are proved to be less useful when it comes to the implementation of image processing algorithms on embedded scale. In certain instances, image processing algorithms are implemented on a dedicated Application Specific Integrated Circuits (ASIC) and more commonly on Digital Signal Processors (DSP). With the advent of Field Programmable Gate Arrays (FPGA), it has become an alternative for the implementation of Image processing algorithms on ASIC as it provides speed comparable to an ASIC and is easily reconfigurable.

The objective of this work is to develop a real-time edge detection system where the input comes from a live video acquired from a CMOS camera and the detected edges to be displayed on a DVI display screen.

2. The Setup

The setup for implementation consists of Video Starter Kit (VSK) consisting of Spartan 3A DSP XCSD3400A FPGA connected to a Micron CMOS camera of resolution 720 x 480 pixels delivering frames at 60 fps through a FPGA Mezzanine Card (FMC) Daughter card used for decoding the data arriving through the serial LVDS camera interface. The de-serialized input consists of V-Sync, H-Sync and 8 line data bus which serves as the input for the Edge detection model. The edge filter is applied in the Camera Processing block on the input signal arriving from the Camera In block. The output signal is Gamma corrected for the output DVI monitor and is driven by Display controller to the DVI output monitor. Video to VFBC and MPMC core helps us to store the image data and buffer them to the output screen.

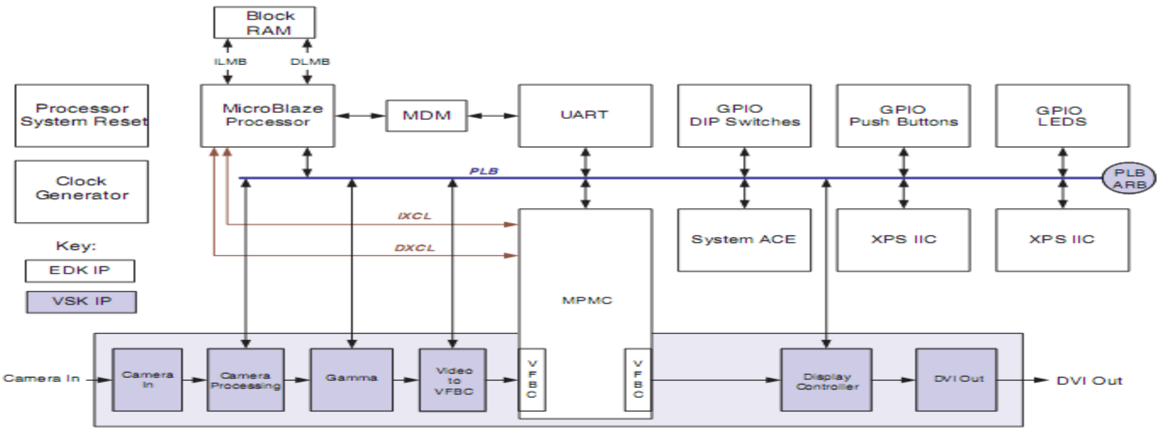


Fig. 1 Block Diagram of the complete setup

The status reporting and controlling of these blocks are carried out by Micro Blaze processor through Processor Local Bus (PLB). The block diagram of the setup is shown in Fig 1.

In the above setup a DVI display shows the output edge from the camera. However, for sake of convenience the edge detectors output is verified on few standard still images. The input is fed from a DVI video source and the edge is observed on the output display screen. The screen shot from both input and output is stored and is presented here.

3. Implementation

3.1 Sobel Edge Detector

The Sobel edge detection uses two masks, one for detecting image derivatives in horizontal direction and the other for detecting image derivatives in vertical direction. One mask is simply the other rotated by 90°. The Sobel kernels can also be thought of as 3 × 3 approximations to first-derivative-of-Gaussian kernels.

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

(a)

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

(b)

Fig. 2 Sobel Edge detector – (a) Horizontal and (b) Vertical Kernel

Gradient magnitude is given by:

$$G = \sqrt{{G_x}^2 + {G_y}^2} \approx |G_x| + |G_y|$$

Gradient direction is given as,

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

Where,

$$G_x = \frac{\partial f}{\partial x} = (-1) * f(x-1, y-1) + (-2) * f(x-1, y) + (-1) * f(x-1, y+1) + (0) * f(x, y-1) + (0) * f(x, y) + (0) * f(x, y+1) + (1) * f(x+1, y-1) + (2) * f(x+1, y) + (1) * f(x+1, y+1)$$

$$G_y = \frac{\partial f}{\partial y} = (-1) * f(x-1, y-1) + (0) * f(x-1, y) + (1) * f(x-1, y+1) + (-2) * f(x, y-1) + (0) * f(x, y) + (2) * f(x, y+1) + (-1) * f(x+1, y-1) + (0) * f(x+1, y) + (1) * f(x+1, y+1)$$

The implementation of the Edge detector consists of RGB to grayscale color space conversion and line buffers to synchronize H-sync, V-sync and Data-enable signals. The filter and buffer block consists of line buffer to hold the corresponding rows and edge detector block. The edge detector block consists of two convolution block which performs the operation specified by the kernel. Individual rows of each kernel are multiplied by the delayed elements which are stored in the line buffers to yield vertical and horizontal edges. These are added and gradient magnitude is found out. The gradient magnitude is thresholded using a manual threshold to obtain proper edges.

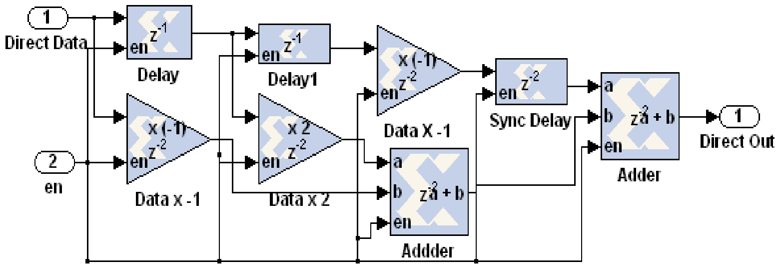


Fig. 3 Convolution block.

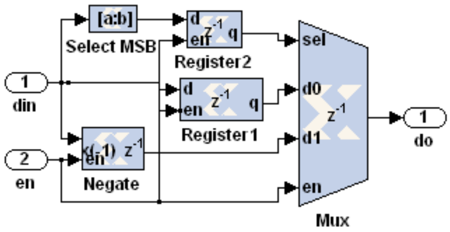


Fig. 4 Calculation of magnitude.

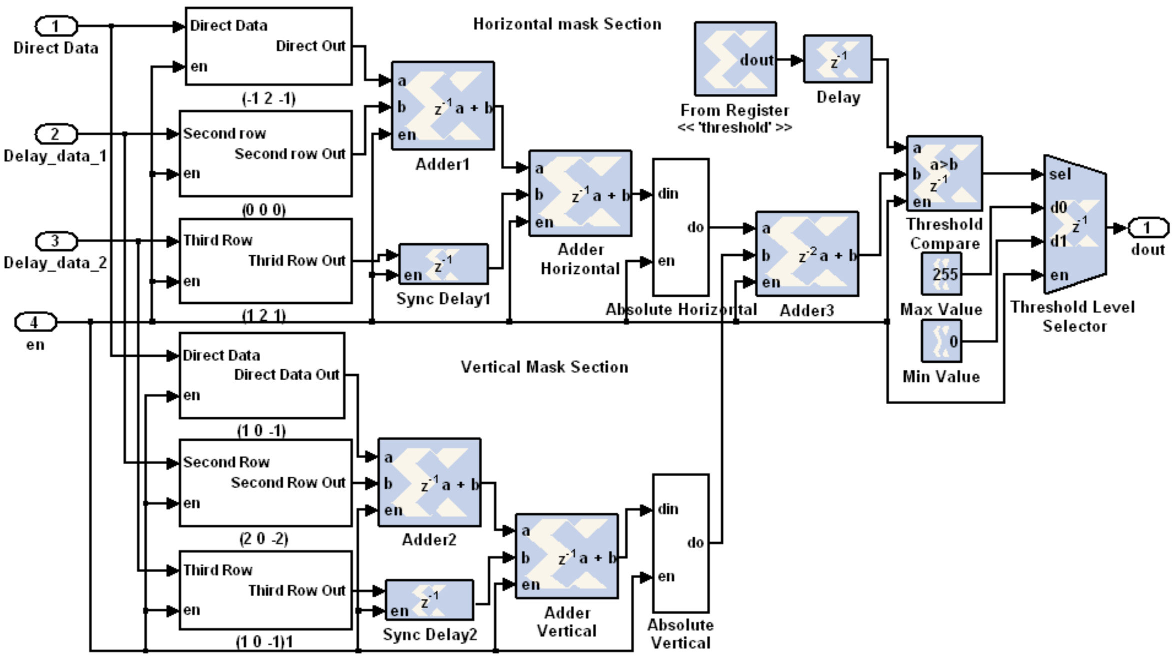


Fig. 5 Mask Implementation Block.

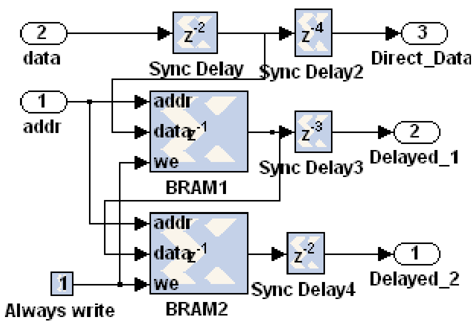


Fig. 6 Line Buffer.

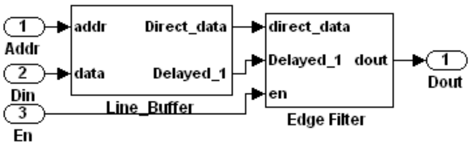


Fig. 7 Edge Detection Block.

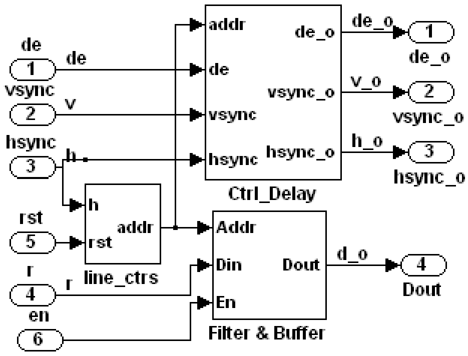


Fig. 8 Synchronization Block.

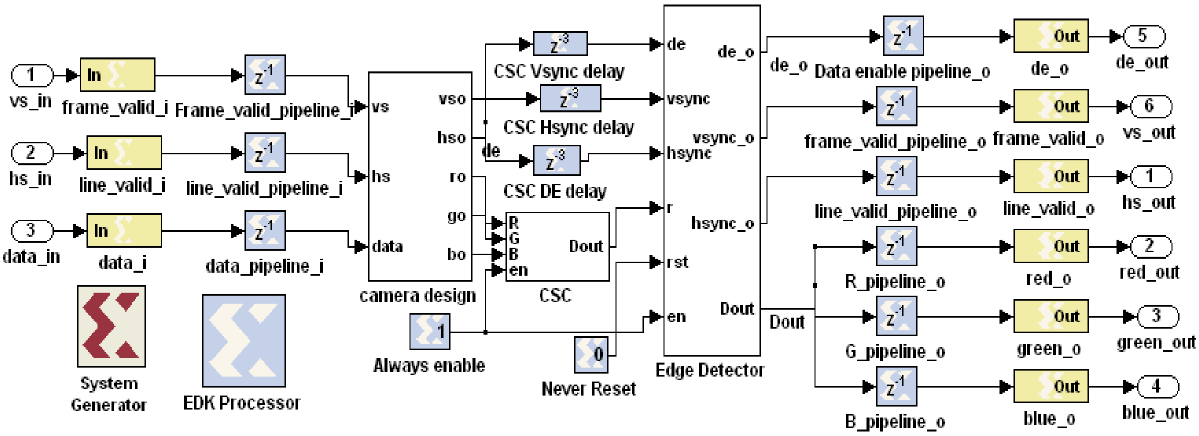


Fig. 9 Color Space Converter and Edge Detection Block.

3.2 Prewitt Edge Detection

The Prewitt edge detection is based on the idea of the central differences and give equal weightage to all pixels when averaging. Prewitt Cross operator performs a simple, quickly computable, 2-D spatial gradient measurement on the image.

These kernels are, however, sensitive to noise. We can reduce some of the effects of noise by averaging. This is done in the Prewitt kernels by averaging in y when calculating $\partial f/\partial x$ and by averaging in x when calculating $\partial f/\partial y$. Together, these kernels give us the components of the gradient vector.

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

(a)

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

(b)

Fig. 10 Prewitt Edge Detector – (a) Horizontal & (b)Vertical kernel.

The FPGA implementation is same as Sobel's edge detector, with the change in the mask values.

3.3 Robert Edge Detector

Roberts's operator consists of a pair of 2×2 convolution kernels which are orthogonal to each other. These masks are designed to respond maximally to edges running at 45° to the pixel grid, one mask for each of the two perpendicular orientations. The masks are shown below.

1	0
0	-1

(a)

0	1
-1	0

(b)

Fig 11. Robert Edge Detector (a) Horizontal & (b) Vertical Kernel.

Gradient magnitude is given by:

$$G = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

Gradient direction is given as,

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

Where,

$$G_x = \frac{\partial f}{\partial x} = 1 * f(x-1, y-1) + 0 * f(x-1, y) + 0 * f(x, y-1) + -1 * f(x, y)$$

$$G_y = \frac{\partial f}{\partial y} = 1 * f(x-1, y-1) + 0 * f(x-1, y) + 0 * f(x, y-1) + -1 * f(x, y)$$

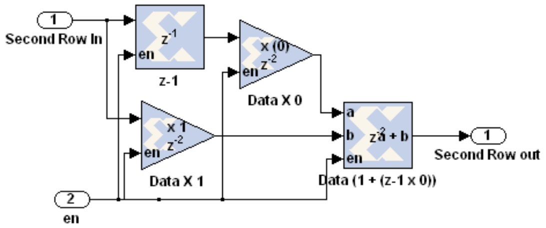


Fig. 12 Convolution Block for Robert Edge Mask.

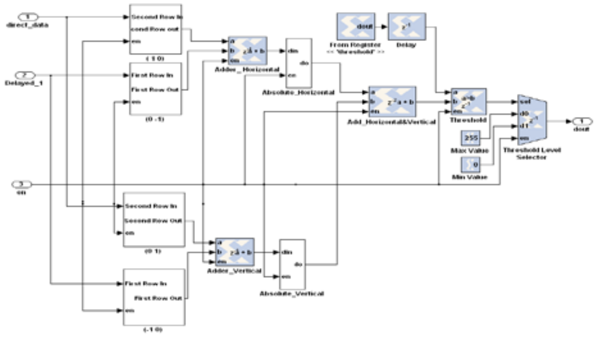


Fig. 13 Mask implementation Block.

3.4 Compass Edge Detector

Compass Edge Detectors are an alternative approach to the differential gradient edge detection. The edge magnitude and orientation of a pixel is determined by the template that matches the local area of the pixel the best. These operators comprise of eight kernels each. The whole set of *8* kernels of each operator are produced by taking one kernel of the respective type and by rotating its coefficients circularly. Each of the resultant kernels is sensitive to an edge orientation ranging from *0°* to *315°* in steps of *45°*, where *0°* corresponds to a vertical edge. In this paper, four types of compass edge detectors namely,

1. Compass,
2. Prewitt Compass,
3. Kirsch Compass,
4. Robinson Compass

are implemented

These kernels are convolved with the input image to obtain the gradients along the corresponding directions .G_S, G_{SE}, G_E, G_{NE}, G_N, G_{NW}, G_W, and G_{SW} represents gradients along South, South-East, East, North-East, North, North-West, West and South-West directions respectively.

The *edge magnitude* is given by the maximum of calculated gradients. Mathematically it can be expressed as,

$$G_{max} = max\ \{G_S, G_{SE}, G_E, G_{NE}, G_N, G_{NW}, G_W, G_{SW}\}$$

where, G_{max} is the gradient corresponding to maximum pixel value.

The local *edge orientation* is estimated with the orientation of the kernel that yields the maximum response. The values for the output orientation image lie between 0 and 7, depending on which of the *8* kernels produced the maximum response. These can be quantized in terms of angle ranging from 0 to (7*pi/4) radians differing by pi/4 in anticlockwise direction. Direction is perpendicular to the edge. Generally, direction points to the brighter side of the edge. Mathematically it can be expressed as,

$$\theta = \tan^{-1}\left(\frac{G_{max}}{G_s}\right)$$

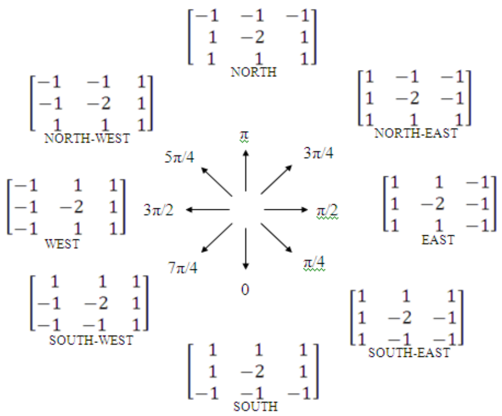


Fig. 14 Compass Masks and their orientation

The south-east direction masks for each type methods are shown below.

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

(a) Robinson (b) Prewitt

ISSN (Online): 1694-0814
www.IJCSI.org

$$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

(c) Kirsch

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

(d) Sobel

Fig. 15 Compass Edge Detector – South East Directional kernel

The FPGA implementation involved implementation of 8 Kernel operators which are similar to the Sobel kernel discussed above and a maximum block, to find the maximum of the calculated edge gradients.

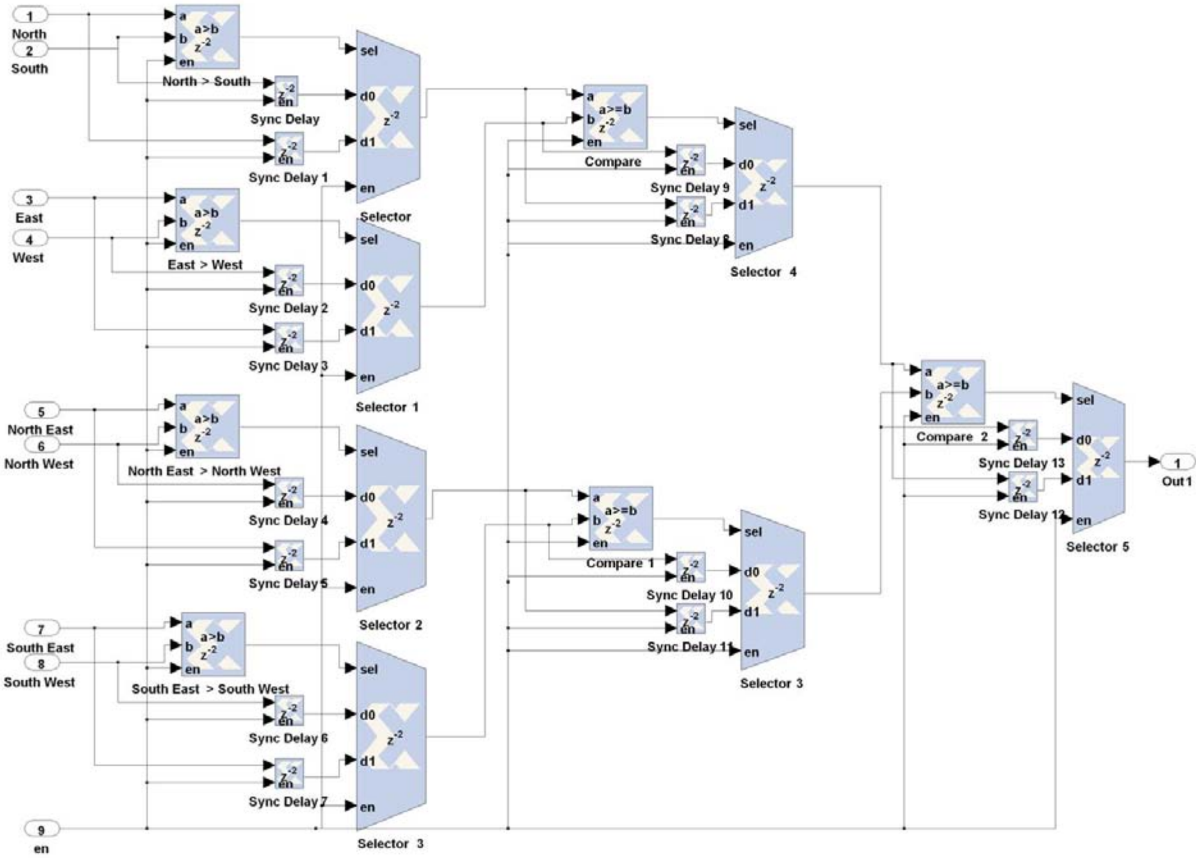


Fig. 16 Edge Maximum Block.

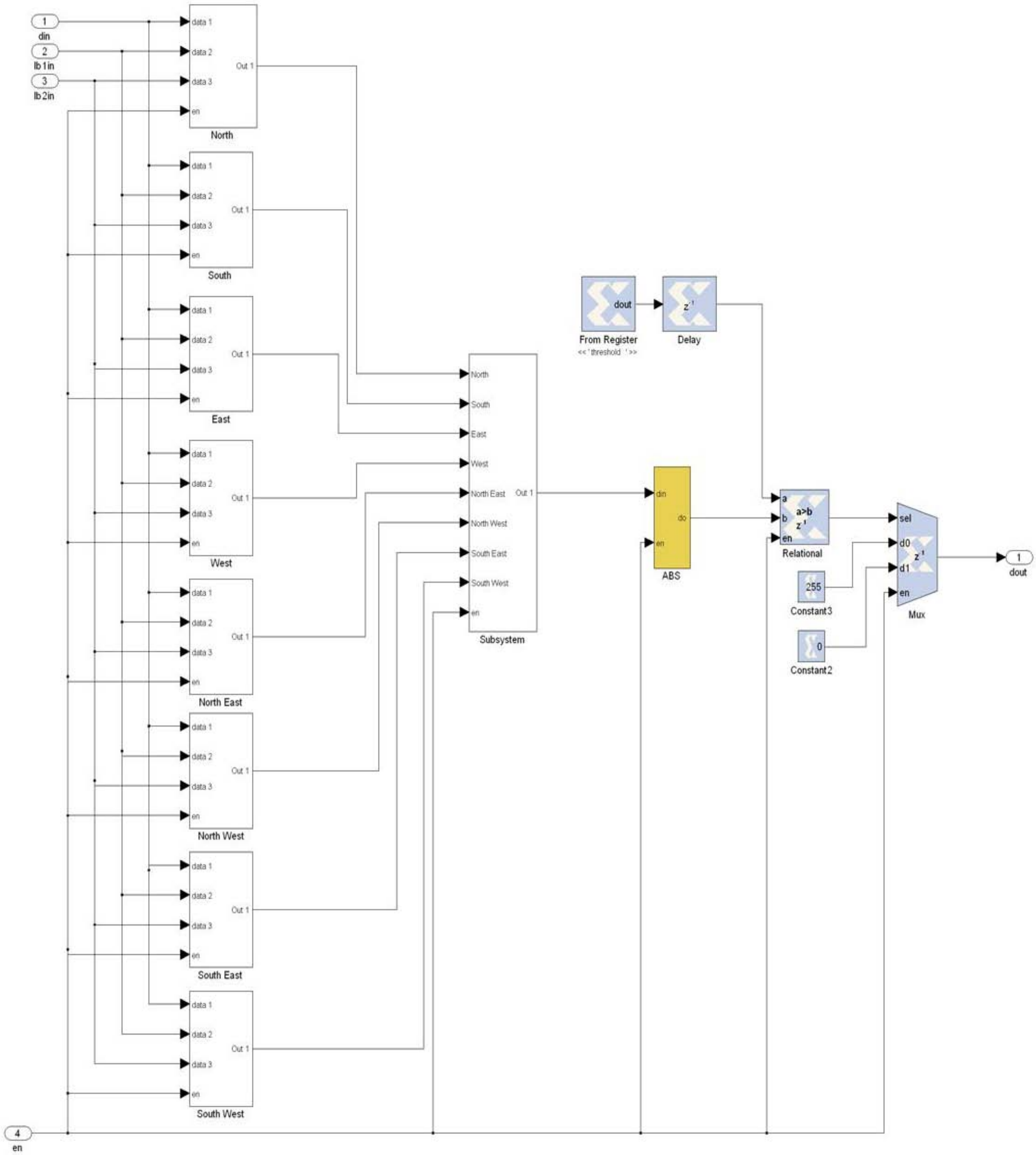


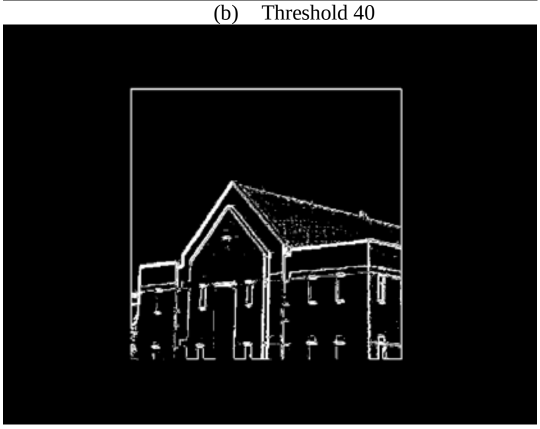
Fig. 17 System Generator Block for Compass Edge Detectors.

4. Results

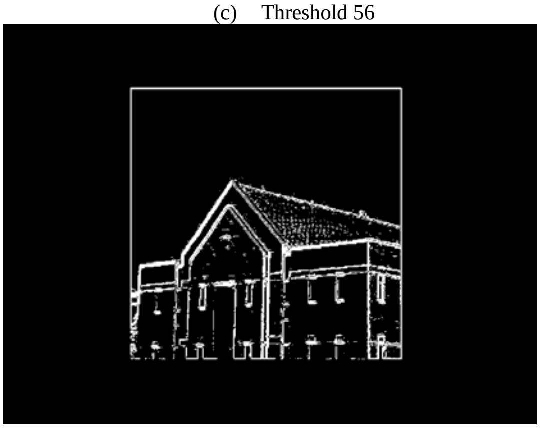
The output image for input images of resolution 800 x 600 discussed above are shown below.

Input Image (a) 800 x 600

(b) Threshold 40

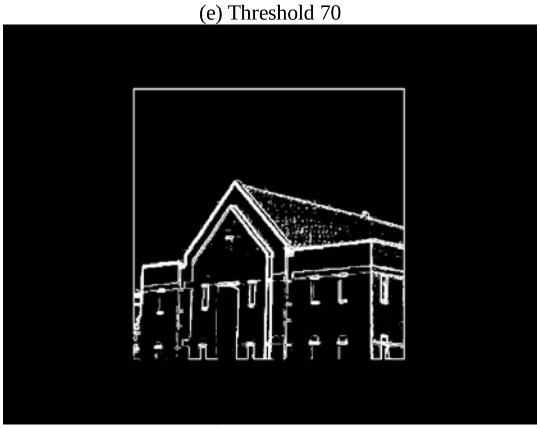


(c) Threshold 56

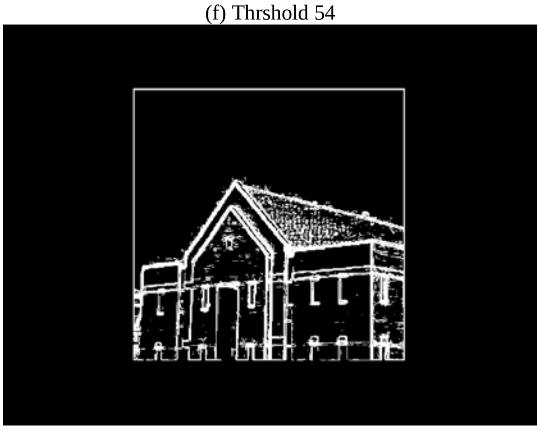


(d) Threshold 56

(e) Threshold 70



(f) Thrshold 54

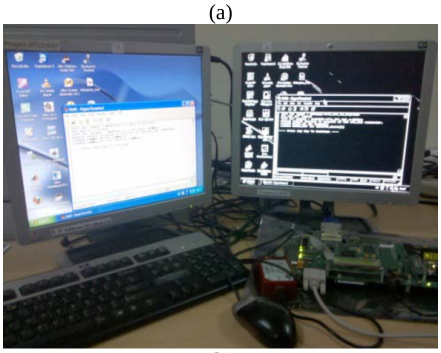


(g) Threshold 120

(h) Threshold 135

Fig. 18 (a) Input image of resolution 800 x 600 . Edge ouput using (b) Robert (c) Prewitt (d) Sobel (e) Robinson Compass (f) Prewitt Compass (g) Kirsch Compass Masks (h) Sobel Compass Masks.

(a)



(b)

Fig. 20 (a) Experimental setup for implementation of edge detection. Input is from CMOS camera and the ouput is on a DVI display (b) Setup for verification. Input is obtained from a DVI Video source and output is on a DVI display.

Edge detection methods investigated so far are further assessed by quality measures that give reliable statistical evidence to distinguish among the edge maps obtained. The absence of the ground truth edge map reveals the search for an alternative approach to assess and compare the quality of the edge maps resulted from the detectors exploited so far. The evidence for the best detector type is judged by studying the edge maps relative to each other through statistical evaluation. Upon this evaluation, an edge detection method can be employed to characterize edges to represent the image for further analysis and implementation.

Table 1: Relative frequencies of edge appearance for various edge detectors on edge output image

Operators	**Sobel**	**Roberts**	**Prewitt**
Sobel	1	0.7232	1.0014
Roberts	1.3827	1	1.3847
Prewitt	0.9986	0.7222	1
Robinson Compass	0.9976	0.7215	0.999
Prewitt Compass	1.0128	0.7325	1.0142
Compass Sobel	0.9977	0.7215	0.9991
Compass Kirsch	0.9928	0.781	0.9942

Operators	**Robinson Compass**	**Prewitt Compass**	**Sobel Compass**	**Kirsch Compass**
Sobel	1.0024	0.9874	1.0023	1.0072
Roberts	1.3861	1.3652	1.386	1.3927
Prewitts	1.001	0.986	1.0009	1.0058
Compass	1	0.985	0.9999	1.0048
Prewitt Compass	1.0152	1	1.0152	1.0201
Robinson Compass	1.0001	0.9851	1	1.0048
Kirsch Compass	0.9952	0.9803	0.9952	1

The above table summarizes the relative frequencies of edge pixel occurrence for various edge detectors. For each edge map, max (ndf) where ndf is the frequency f of occurrence for the filter is reported, and the ratio with respect to each other gives comparative statistics for the occurrence of edges. The Compass filter with Kirsch mask reports higher detected edge pixels.

Table 2: Resource estimation for various edge detectors on Spartan 3A DSP XCSD3400A FPGA

Edge Operator	**Slices**	**Flip Flops**	**LUT**	**IOBs**	**BRAMs**	**Max. Freq (Mhz)**
Sobel	4%	4%	3%	29%	4%	154.918
Prewitt	4%	4%	3%	29%	4%	154.918
Robert	2%	1%	1%	29%	3%	154.959
Compass operators	19%	16%	12%	29%	4%	137.646

5. Conclusion

Various edge detectors were implemented on FPGA at a rate of 60 fps for an input image of resolution 720x480. The design was tested for a resolution upto 1024 x 768 on a DVI input and output. The implementation can be extended to color video images as well for much faster frame rate.

Acknowledgments

The authors express their sincere gratitude to the Director and the Principal of Nitte Meenakshi Institute of Technology for providing Spartan 3ADSP Video Starter Kit to carry out the research

work. This work is inspired by the work done by Sneha et al[2].

References

[1] R. Gonzalaz, R. Woods, “Digital Image Processing”. New Jersey: Prentice-Hall 2002, ch 10.

[2] Sneha H L, Rashmi R, Sushma R P, Rajesh N, Dr. Jharna Majumdar, “FPGA Implementation of Real – Time Edge Detectors using Xilinx System Generator”, National conference for Emerging trends in Control, Communication, Signal Processing & VLSI Techniques

[3] Ownby, M.; Mahmoud, W.H.; , "A design methodology for implementing DSP with Xilinx® System Generator for Matlab®," System Theory, 2003. Proceedings of the 35th Southeastern Symposium on , vol., no., pp. 404- 408, 16-18 March 2003

[4] Hong Shan Neoh, Asher Hazanchuk, “Adaptive Edge Detection for Real-Time Video Processing using FPGAs”

[5] Zhang Zongcheng; Wang Hongyan; Liu Weixiao; Yu Ming; , "Real-time image video edge enhancement using hardware convolver," TENCON '93. Proceedings. Computer, Communication, Control and Power Engineering.1993 IEEE Region 10 Conference on , vol., no.0, pp.398-401 vol.3, 19-21 Oct 1993

[6] Mamta Juneja, Parvinder Singh Sandhu, “Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain”, International Journal of Computer Theory and Engineering, Vol. 1, No. 5, December, 2009

[7] Xilinx, Spartan 3A DSP FPGA Family Datasheet

Sudeep K C graduated from Nitte Meenakshi Institute of Technology in 2009 from Dept. of Electronics and Communication. He has been a part of On Board Computer subsystem of the STUDSAT – a student pico satellite program in 2009 and since then has been a part of Industrial Robot Automation program. His research areas include implementation of image and computer vision algorithms on embedded and reconfigurable hardware. He has been working in Dept. of Computer Science in Nitte Meenakshi Institute of Technology since 2009.

Dr. Jharna Majumdar Dr. Jharna Majumdar is currently working as Dean R & D and Professor and Head of Computer Science and Engineering (PG) at the NITTE Meenakshi Institute of Technology, Bangalore. Prior to this Dr. Majumdar served Aeronautical Development Establishment, Defence Research and Development Organization (DRDO), Ministry of Defence, Govt. of India from 1990 to 2007 as Research Scientist and Head of Aerial Image Exploitation Division, Bangalore. Dr. Majumdar has 37 years. of experience in R & D and Academics in the country and abroad. She has published large number of papers in National, International Conferences and Journals.Her research areas include Image and Video Processing for defense and non defense application, Robot Vision, Vision based autonomous guided systems, development of Computer Vision Algorithms in FPGA etc.. Recently as the Project Coordinator, she has provided leadership to a team of undergraduate students from seven engineering for the development of STUDSAT, the smallest STUDent SATellite of PICO Category (Less than 1 Kg), first time developed in the country.