

# Implementation of Mainly Used Edge Detection Algorithms on FPGA

Yaser Icer <sup>\*1</sup>, Mustafa Turk <sup>2</sup>

Accepted 3rd September 2016

**Abstract:** Edge detection has important applications area in image processing field. Today, it is a fact that the image processing used in many fields. Therefore, the applicability of edge detection process in the field is also has great importance. In this study, mainly used edge detection algorithms in the literature; I.e. Sobel, Prewitt and Canny algorithms is provided using the verification and inspection on FPGA (Field Programmable Gate Arrays). Program files required for FPGA is prepared by Xilinx System Generator DSP blocks, which can work integrated with Matlab/Simulink. For this study; gray format images, which is stored on the computer has been sent to FPGA with USB configuration port interface on FPGA. Edge detection process is realized by moving subject images from the computer with the same connection to FPGA and then, Sobel, Prewitt and Canny algorithms are applied to the images on FPGA respectively. Edge detection process for the same images are performed by Simulink and FPGA board at the same time and then, edge detected images obtained from these two environment are compared and also it has been observed on the FPGA resource usage.

**Keywords:** FPGA (Field Programmable Gate Arrays), Sobel, Prewitt, Canny, Matlab/Simulink and Xilinx System Generator, DSP (Digital Signal Processing)

## 1. Introduction

Today, with the development of image processing technology it is also used quite a lot. Most of our lives intertwined with a technological structure is wrapped in an image processing device is configured on. The algorithms used in this field of technology with the speed proportional manner are developing very rapidly. Work is done in many areas, I.e.; Security systems, traffic warning systems, microscopy and medical imaging systems, making use of image processing techniques such as remote sensing systems.

One of the most important areas in the image processing is an edge detection process. Edge detection is one of the most important task of finding the image. Edges is the most important information in the image. Human visual system is based on the direct detection of edges [1]. To distinguish objects in the images, the display of details which can not be noticed with the naked eye, in situations such as edge detection algorithms to compare the two different patterns are used quite extensively. Haralick, found that the gradient value using the derivative value interpolation equation is found through applied on pixels in the direction of the gradient on the second derivative method caller has provided a transition to zero [2]. Canny, the implementation of the operator derived from the Gaussian mask derivative based proposed a method [3]. Roberts is the first method used in image processing, Prewitt and Sobel operators are an example to the most appropriate operators [4]. Because image processing and edge detection processing speed for applications that require processing typically computer-based applications, DSP-based methods and FPGA's are used.

We can sort out some work done using image processing FPGA said. Chris and his friends for the detection of tumors in the brain images taken by MRI device, brands have benefited from Xilinx FPGA development board. Implemented on FPGA determined the tumor site with different filters. Create a system in which they work in a way that is integrated with Matlab, compared to other systems available in the literature, 50% less use of resources has been realized [5]. Rodriguez and colleagues while running real-time median filter design on FPGA, have created a system that enables visual identification of errors in production centers [6]. Nelson, a team using different FPGA hardware produced by the master's thesis on Xilinx and Altera company that performs morphological operations and convolution processing, receives the results were compared with the results produced by using MATLAB [7]. Nana and her friends while improving the resolution of an image taken using DSP with FPGA hardware have gained a clearer picture. In this process; They used the Xilinx Spartan-3 FPGA hardware configuration of the image and they have developed the resolution with DSP [8].

This article made a thorough research on the FPGA programmable logic devices and technologies and the basic operators used in edge detection Sobel, Prewitt and Canny operator of the implementation on FPGA and is provided to be examined. In this study, Matlab / Simulink and Xilinx System Generator can run Xilinx integrated DSP blocks in Simulink and Simulink blocks with the tool is used. Due to prepared system 256\*256 gray formatted images on computer sent to Simulink and FPGA at the same time and Sobel, Prewitt and Canny operators were applied to the images. In prepared system, data transmission between computer and FPGA is provided with USB configuration port. FPGA's need to work \*.bit extension program file has been created with the Xilinx System Generator. In practice, Spartan3E XC3S1600 belonging to the company Xilinx FPGA device is used.

The following sections of this article consists of more than four sections. Scope of issues examined in this section are as follows:

<sup>1</sup> Diyarbakır Technical Sciences Vocational School, Dicle University, Campus, 21080, Diyarbakır/Turkey

<sup>2</sup> Department of Electrical and Electronic Engineering, Firat University, Campus, 23200, Elazığ/Turkey

\* Corresponding Author: Email: [yaser.icer@dicle.edu.tr](mailto:yaser.icer@dicle.edu.tr)

Section 2 in this article; edge detection, edge detection methods and general information about the basic edge detection operators are given. At Section 3; information about Xilinx System Generator, which used for programming of FPGA board and Simulink DSP blocks are given. Section 4 in this article; implementation of Sobel, Prewitt and Canny edge detection operator on FPGA and Simulink side is explained in detail. Results and conclusion parts are shown in sections 5 and 6 respectively.

## 2. Edge Detection and Operators

Visual system in humans is first trend detection edges in the image of the object located on any image. This is because the stationary location typically does not contain information. In an image, the image improvement is of great importance how important is the determination of the edges. Generally such high levels of image processing in the edge detection operators are implemented as a precursor [9]. Different features depending on optical and shape of an image leads to the formation of the edges. shape-based features in the appearance of the edges; changes in the direction of surface texture or color, object, borders, colors observed two objects sourced from the top are differences in the appearance and texture [10]. Edges which some of their optical properties are; the direct reflection light, shadows caused from a part of the same object or other objects, reflections arising from the intermediate part of the same object or other objects, texture or color variation [11].

One of the most efficient methods to detect edges in the image that exists is to identify instant gray value changes on the image. One of the main approaches used by many edge detection method which involves calculating the derivative locally. In the first derivative of the edges of the image locally reached the highest value, while the second derivative image in the edge of regional zero. 1st and 2nd derivatives are calculated for the image as a result of this, a local maximum and edges with the zero crossing points for the examination of the image is determined. Fig. 1 shows derivative values of image function at edge points.

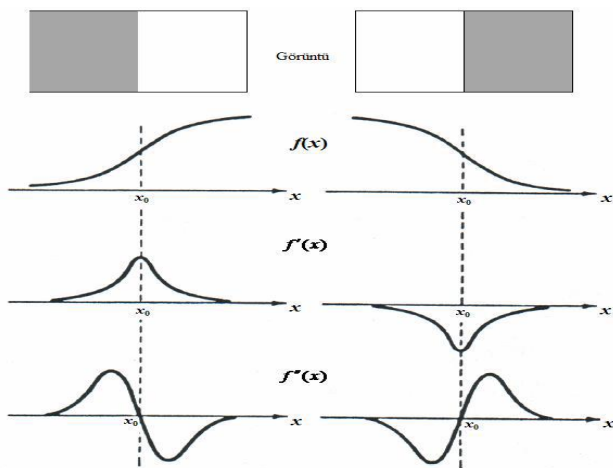


Figure 1. Derivative of edge transition of edge points

### 2.1. The First Derivative(Gradient) Based Methods

Gradient based single variable is expressed as the first derivative of a function. This operation is a function which depends on two variables calculated by applying the process according to the 1st derivative of each variable. As an image is expressed as a function of two changes, the necessary information about the

direction of the gradient edge angle as can be seen by looking at the size gradient to be applied to the edges in an image display function will give us.  $f(x,y)$  as an equation for an image display function, gradient magnitude and gradient angle is calculated by equations below.

$$G[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x,y)-f(x,y)}{\Delta x} \\ \lim_{\Delta y \rightarrow 0} \frac{f(x,y+\Delta y)-f(x,y)}{\Delta y} \end{bmatrix} \quad (1)$$

$$G = \sqrt{G_x^2 + G_y^2} \quad , \quad \theta = \tan^{-1} \frac{G_y}{G_x} \quad (2)$$

Roberts, Sobel, Prewitt and Canny 1st derivative (gradient) based methods are the most well-known among operators.

#### 2.1.1. Roberts Operator

Roberts's operator on the image of a mask with a drizzle of 2x2 at every point allows for the calculation of the approximate gradient vector. The magnitude of the gradient vectors is calculated, it is placed on the edges of a pixel matrix of image size designated image can be obtained. Gradient operation on the image convolution operation result is calculated using two different masks image horizontally and vertically. The result of using the first derivative of the convolution equation gives two masks, which is given below, (3).

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad , \quad G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (3)$$

#### 2.1.2. Sobel Operator

Another fundamental edge detection operator is Sobel. This provides operators with the direction to make an efficient assessment as more independent gradient calculation based on more extensive use operators like Robert [12]. The basic idea underlying the creation of Sobel operator edge can be explained as follows. The total value of the gradient of a digital image to 3x3 neighbourhood center and 4 formed in the direction vector is based on a function that is considered. Sobel horizontal and vertical masks are given below.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad , \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (4)$$

#### 2.1.3. Prewitt Operator

Prewitt edge operator is an operator that calculates the gradient vector at each point on the image pixels. Found the gradient vector magnitude edge image that made improvements, while the direction of the gradient vector gives the edge direction [13]. Prewitt operator edge gradient values using the two different masks in horizontal and vertical pixel on the image is calculated by convolution operation. The masks are used during this process include using difference equations of the first derivative. Prewitt horizontal and vertical masks are given below.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad , \quad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (5)$$

### 2.1.4. Canny Operator

Canny has been working for optimum edge detection algorithm [3]. Determining this algorithm has been designed to provide the following features:

- Developing a great possibility that when determining the value of the algorithm places where the edges, the edges are not able to determine locations with a small probability value.
- To be defined edges should be on the edge to be the closest to real edge.
- Real edge should be detected only once.

Canny edge detection algorithm work consists of five steps. These steps are as follows, respectively:

- Smoothing
- Gradient Calculation
- Non-maximum Suppression
- Double Thresholding
- Edge Tracking with Hysteresis

All of the amount of noise present in the images obtained by the camera is inevitable. During the edge detection process on the noise image to image will cause errors need to be cleaned out. In order to cleaning Gaussian filter is applied to the image. This process is for smoothing and equations given below is for Gaussian filter.

$$G_{\sigma} \equiv \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (6)$$

$$G = \frac{1}{159} \cdot \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (7)$$

For calculation of gradient magnitude and gradient angle Sobel filter are applied to the images and equation (2) is used.

The main objective of making the process non-maximum suppression is sharpen the edge of the image with a blurred gradient magnitude. This process is mainly performed by making all pixel values of the non-local maximum points in the gradient image to zero.

After non-maximum suppression process some of pixel values is not real edge pixels due to noise and color variation. The aim of double thresholding is elimination of these pixels. Canny edge detection algorithm uses two threshold, including high and low thresholds. Edge pixel density greater than high threshold pixels that "strong" as classifying edge pixels which are printed below the lower threshold pixel density. Edge pixel value between the low and high threshold in case the edges of the pixels are "weak" as classified.

After double thresholding process the pixels, which are classified "strong" as determined edge pixels precisely. If the pixels, which outside of that and classified "weak" is neighbourhood of "strong" pixels remain otherwise value of them became zero.

### 2.2. The Second Derivative(Laplacian) Based Methods

In this method when we take the first derivative of image function, at the edge points value of derivative become

maximum and minimum. After this we take second derivative and it will give us edge pixels at zero values of image function. Noise sensitivity of this method is higher than the first derivative method. Marr-Hildreth(LoG) operator is well known operator for the second derivative based methods [1].

#### 2.2.1. Marr-Hildreth(LoG) Operator

Marr and Hildreth applying Laplace filter to image the function passed through the filter Gaussian filters were developed in 1980, allows the presence of edges on the image. High frequency noise on the image cleaning with a Gaussian filter operation, and applying Laplace filter by taken second derivative of image function gives edge detection of image. Equations given below is about Laplacian and Gaussian of image function respectively.

$$h(x, y) = \nabla^2[g(x, y) * f(x, y)] = [\nabla^2 g(x, y)] * f(x, y) \quad (8)$$

$$\nabla^2 g(x, y) = \left(\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right) \cdot e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (9)$$

## 3. Programming FPGA with XSG

Xilinx System Generator (XSG) is produced by the Xilinx DSP design tool. This software tool MATLAB model-based design environment that enables the use of Simulink for FPGA. System Generator using the design is carried out using hosts Xilinx special design blocks in DSP functions in Simulink. FPGA program file that is necessary for creating synthesis and placement/ routing steps are automatically executed in this environment.

System Generator is working according to the Simulink model-based design methods. Usually, executable functions are created using standard Simulink blocks. This feature of System Generator is designed using floating point number system without function hardware details. System Generator uses the Xilinx DSP blocks in Simulink and automatically run to perform the core functions of a high degree of conformity of the connection required for the DSP blocks.

The Xilinx DSP block are accessed through the library operated with Matlab Simulink tool. Blocks can be made easily for the search process is divided into sub-categories. "Index" is keeping all blocks and sub-categories shortcuts to frequently used blocks, offers ease of use possibilities. DSP building blocks over 90 in the System Generator to design a DSP system is located. System Generator DSP blocks are located in Matlab/Simulink and its image windows is given in Fig. 2.

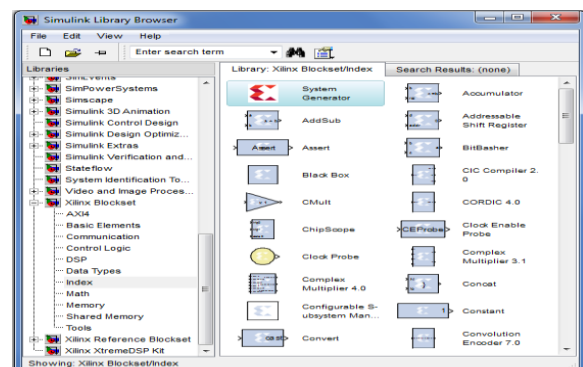


Figure 2. System Generator DSP blocks in Matlab/Simulink

System Generator works with standard Simulink models. "Gateway In" and "Gateway Out" is defined as two blocks in Simulink simulation model and they define the input and output of the FPGA device is used. "Gateway In" block, converting floating point to fixed-point number type. "Gateway Out" is transforming the FPGA block out the double-precision number type. After FPGAs limits determined by the gateway blocks, DSP designs using the Xilinx DSP block can be realized. Standard Simulink blocks is not used between the "Gateway In" and "Gateway Out" blocks.

#### 4. Proposed Design and Implementation

The implementation of edge detection algorithms on FPGA realized in three steps, which are image pre-processing, image post-processing and edge detection with XSG. Addition to this, hardware implementation is realized by using XSG for preparation of programming file (\*.bit) for FPGA. Fig. 3 shows system diagram for implementation of edge detection algorithms on FPGA.

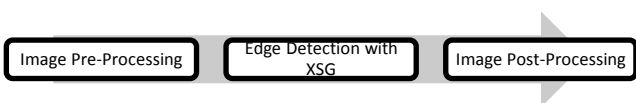


Figure 3. System diagram for edge detection algorithms

##### 4.1. Image Pre-Processing

During this process the data entry format of the "Gateway In" the blog entry block for FPGA card in gray format image on a computer and converted into serial digital data format. Simulink blocks used in image pre-processing steps are given in Fig. 4.

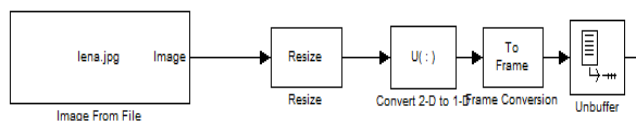


Figure 4. Image pre-processing blocks

"Image from File" block converts image data to M by N matrix format, "Resize" block changes image matrix size (in this study it is 256\*256), "Convert 2D to 1D" block converts two dimension matrix image data to one dimension, "Frame Conversion" block converts one dimension image data to frame by frame serial data, and "Unbuffer" block converts coming data to digital samples with high sampling rate.

##### 4.2. Image Post-Processing

FPGA card with "Gateway Out" block of the output block to output data series because it is in digital format and can be monitored on a digital display that data in a serial format and the image data is converted to two-dimensional image format. Simulink blocks used in image post-processing phase is given in Fig. 5.

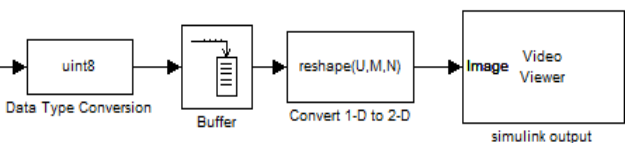


Figure 5. Image post-processing blocks

"Data Type Conversion" block converts digital data to 8 bit unsigned integer format, "Buffer" block converts digital data to one dimensional serial data with low sampling rate, "Convert 1D to 2D" block converts one dimensional image data to two

dimensional image matrix, and "Image Viewer" shows two dimensional image matrix as an image.

#### 4.3. Hardware and Software Implementation

Hardware implementation is realised by using XSG block, which can be used for creating FPGA equivalent block on Simulink side. Software implementation is realised by using Xilinx DSP blocks on Simulink. Both hardware and software implementation is run at the same time on Simulink and FPGA. Fig. 6 shows hardware and software realisation of proposed design.

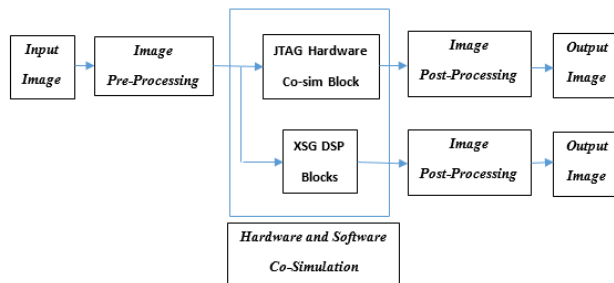


Figure 6. Hardware and software realisation of proposed design

#### 4.4. Edge Detection with XSG

In order to implementation of edge detection on FPGA, XSG DSP blocks on Simulink is used. For implementation of Sobel operator, gradient calculation and thresholding operation is done XSG DSP blocks, which is shown in Fig. 7 and Fig. 8 respectively. Gradient calculation blocks give the convolution of Sobel mask coefficient with original image pixels and thresholding blocks give the binary edge detected image with respect to a threshold value.

Implementation of Prewitt operator is done same as Sobel operator. Because only the convolution coefficient of Prewitt mask is different from Sobel operator.

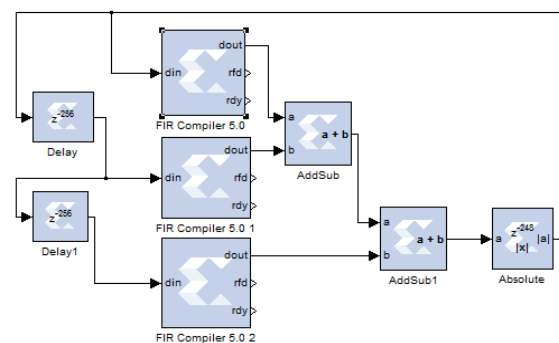


Figure 7. Gradient calculation blocks

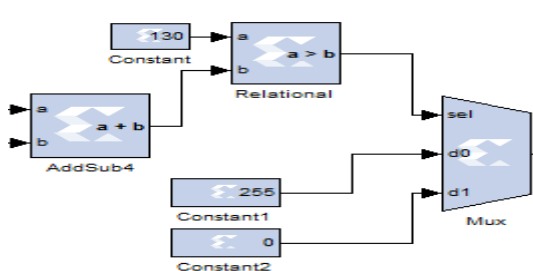


Figure 8. Thresholding blocks

For implementation of Canny algorithm; first of all, Gauss filter is used to remove noise from original image. Fig. 9 shows Gauss filter blocks for XSG.

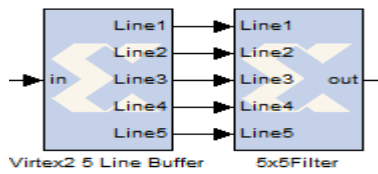


Figure 9. Gauss filter blocks for XSG

After that, Sobel operator is applied to image pixels for gradient magnitude and gradient angle calculation. And then, NMS (non-maximum suppression) is applied to image pixels, which is shown in Fig. 10.

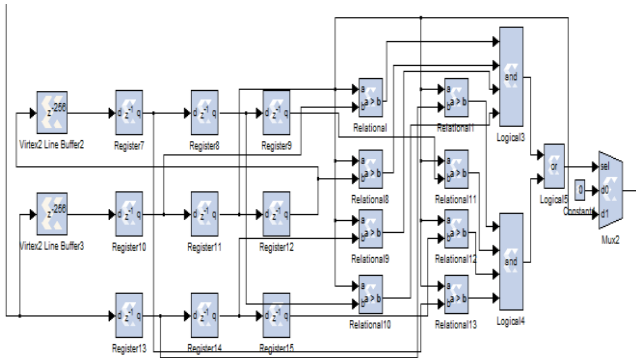


Figure 10. NMS blocks for XSG

Finally, double thresholding is applied to image pixels and binary edge detected image is obtained. Double thresholding XSG blocks are shown in Fig. 11.

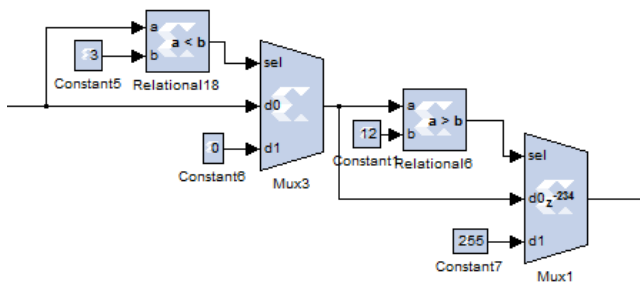


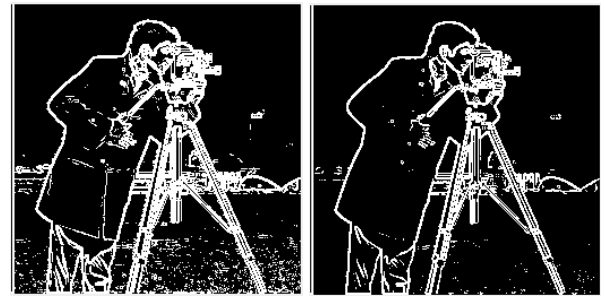
Figure 11. Double thresholding blocks for XSG

## 5. Results

The proposed work is implemented by using Matlab and XSG (Xilinx System Generator) simulink blocksets. This method has been tested on some standard test image like 'Cameraman'. Fig. 12 shows edge detected images, which are detected by Matlab/Simulink and FPGA board for Sobel, Prewitt and Canny algorithms.

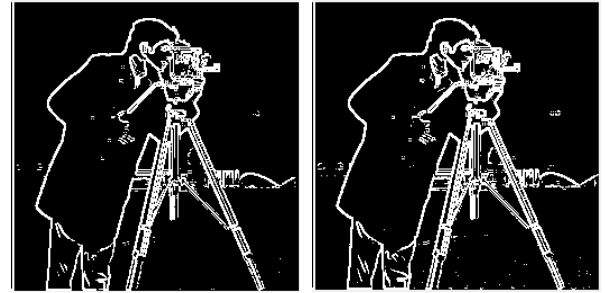


(a)



(b)

(c)



(d)

(e)



(f)

(g)

Figure 12. (a) Original image (b) Sobel Simulink result (c) Sobel FPGA result (d) Prewitt Simulink result (e) Prewitt FPGA result (f) Canny Simulink result (g) Canny FPGA result

FPGA resource usage of Sobel-Prewitt and Canny algorithms on Spartan3E XC3S1600E is given Table 1 and Table 2 respectively. Sobel and Prewitt resource usage is given in one table, because they use the same resource usage on FPGA.

Table 1. FPGA Resource Usage for Sobel or Prewitt

Resource Utilization Summary (Sobel or Prewitt)			
Resources	Used	Available	Utilization
Number of Slice Flip Flops	1,404	29,504	4%
Number of 4 input LUTs	2,201	29,504	7%
Number of occupied Slices	1,401	14,752	9%
Total Number of 4 input LUTs	2,369	29,504	8%
Number of bonded IOBs	17	250	6%
Number of BUFGMUXs	1	24	4%
Number of MULT18X18SIOs	18	36	50%

**Table 2.** FPGA Resource Usage for Canny

Resource Utilization Summary (Canny)			
Resources	Used	Available	Utilization
Number of Slice Flip Flops	916	29,504	3%
Number of 4 input LUTs	603	29,504	2%
Number of occupied Slices	654	14,752	4%
Total Number of 4 input LUTs	688	29,504	2%
Number of bonded IOBs	17	250	6%
Number of RAMB16s	13	20	65%
Number of BUFGMUXs	1	24	4%
Number of MULT18X18SIOs	5	36	14%

The proposed designs use low resources on FPGA. Sobel and Prewitt resource usage is same value, because only their convolution mask coefficient is different from each other. Resource usage of Canny algorithm is greater than Sobel or Prewitt, because Canny algorithm includes more processing steps from Sobel or Prewitt.

The performance of edge detected image is analysed by using Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR). Equation (10) and (11) show calculation of MSE and PSNR respectively.

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M \times N} \quad (10)$$

$$PSNR = 10 \log_{10} \left( \frac{R^2}{MSE} \right) \quad (11)$$

For MSE value;  $I_1$  is pixel value of original image,  $I_2$  is pixel value of edge detected image, M and N is horizontal and vertical pixel size of original image. R is taken 255 in this work, because original image data format is 8-bit unsigned integer. Table 3 shows quantitative results of MSE and PSNR values for proposed works.

**Table 3.** Quantitative Results of MSE and PSNR

Algorithms / Platform		MSE	PSNR (dB)
Sobel	Simulink	29327.8	3.458
	FPGA	30149.5	3.338
Prewitt	Simulink	31137.2	3.198
	FPGA	31345.8	3.169
Canny	Simulink	31447.1	3.155
	FPGA	31497.8	3.148

MSE gives mean square difference between original image pixel and edge detected image pixel. If this difference is high, this means that similarity between images is low [14]. For edge detection process, high MSE means that edge detection performance is better. Therefore low PSNR value gives better edge detection performance [15].

## 6. Conclusion

In this proposed work, edge detection algorithms used in literature is introduced. Implementation of Sobel, Prewitt and Canny edge detection algorithms are done in Matlab/Simulink and FPGA board at the same time. When we look at the resource usage on FPGA board, proposed designs used small part of FPGA resources. If we analysis edge detection performance due to Matlab/Simulink and FPGA platform, we can see that MSE values of edge detected images from FPGA is greater than Simulink and therefore PSNR values of edge detected images from FPGA is less than Simulink platform. Because of this, results have shown edge detection performance of FPGA is better than Matlab/Simulink. Another performance investigation is that algorithms, which used for edge detection. That is Prewitt operator gives better results than Sobel operator due to edge detection performance because of its MSE and PSNR values. Canny algorithm has the best edge detection performance than Sobel and Prewitt operators. Because, it has highest MSE and lowest PSNR values comparing with them. As a result, more complex system can be implemented on FPGA, and advantages of edge detection implemented on FPGA is that; it is field programmable and economic for design, it has flexible structure and parallel processing capability.

## References

- [1] Marr, D., Hildreth E., "Theory of edge detection", Proc. R. Soc. Lond. A, Math. Phys. Sci., B 207, 187–217, 1980.
- [2] Haralick, R.M., "Digital step edges from zero crossing of second directional derivatives", IEEE Trans. Pattern Anal. Mach. Intell 58–68, 1984.
- [3] Canny, J F., "A Computational Approach to Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 679-698, 1986.
- [4] Bovik A., "Handbook of Image and Video Processing", Academic Press, 1384, 2010
- [5] Christe, S.A., Vignesh, M., Kandaswamy, A., "An efficient FPGA implementation of MRI image filtering tumour characterization using Xilinx system generator", International Journal of VLSI Desing & Communication Systems, 2 (4): 95-109, 2011.
- [6] Vega-Rodríguez, M.A., Sánchez-Pérez, J.M., Gómez-Pulido, J.A., "An FPGA-based implementation for median filter meeting the real-time requirements of automated visual inspection systems", 10th Mediterranean Conference on Control and Automation, Lisbon-Portugal, 2002
- [7] Nelson, A.E., "Implementation of image processing algorithms on FPGA hardware", Master of Science Thesis, Faculty of the Graduate School of Vanderbilt University, Nashville, TN-USA, 2000.
- [8] Nana, L., Weixing, Z., Shiyang, M., Wen, H., "Super resolution video reconstruction in DSP+FPGA based on lifting wavelet", The Tenth International Conference on Electronic Measurement & Instruments, China, 101-104, 2011.
- [9] Davies, E. R. , *Machine Vision*, Academic Press, 1997.
- [10] Aarnink, R. G., Rosette, J. M. C. H., Feitz, Wouter F. J., Debryne, F. M. ve Wijkstra, H., *A Preprocessing Algorithm for Edge Detection with Multiple Scales of Resolution*, European Journal of Ultrasound, 5, 113-126, 1997.
- [11] Ziou, D. ve Tabbone, S., *Edge Detection Techniques - An Overview*, Technical Report, No. 195, Dept. Math & Informatique, Universit de Sherbrooke, 1997.

- [12] Sobel, I., *An Isotropic 3x3 Gradient Operator*, Machine Vision for ThreeDimensional Scenes, Freeman, H., Academic Pres, NY, 376-379, 1990.
- [13] Prewitt, J., *Object Enhancemet And Extraction*, Picture Processing and Psychopictorics (B. Lipkin ve A. Rosenfeld, editör), NY, Academic Pres, 1970.
- [14] P. Vidya, S. Veni and K.A. Narayanankutty, “*Performance Analysis of Edge Detection Methods on Hexagonal Sampling Grid*”, International Journal of Electronic Engineering Research, Volume 1, pp. 313–328, 2009.
- [15] Peter Kellman and Elliot R. McVeigh, “*Image Reconstruction in SNR Units: A General Method for SNR Measurement*”, Magn Reson Med. Author manuscript, Magn Reson Med. Volume 54(6) pp. 1439–1447, 2005.