Asian Journal of Scientific Research

ISSN 1992-1454

Asian Journal of Scientific Research 7 (2): 188-198, 2014
ISSN 1992-1454 / DOI: 10.3923/ajsr.2014.188.198

Hardware Implementation of Image Edge Detection Using Xilinx System Generator

[1]C. Sujatha and [2]D. Selvathi

[1]Department of Electronics and Communication Engineering, Sethu Institute of Technology, Kariapatti, Tamil Nadu, India

[2]Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

Corresponding Author: C. Sujatha, Department of Electronics and Communication Engineering, Sethu Institute of Technology, Kariapatti 626 115, Tamil Nadu, India Tel: 9442039061

ABSTRACT

Image edge detection plays a vital role in computer vision and image processing. Edges are most important properties of an image which are mainly used in image analysis and processing. Hardware implementation of image edge detection is essential for real time applications and it is used to increase the speed of operation. Field Programmable Gate Array (FPGA) plays a vital role in hardware implementation of image processing application because of its re-programmability and parallelism. The Simplified Gabor Wavelet (SGW) based edge detection provides better result than other conventional edge detection methods. The proposed work of this paper is FPGA implementation for this Simplified Gabor Wavelet based image edge detection approach. The proposed hardware implementation of Simplified Gabor Wavelet based method is done using Xilinx System Generator (XSG) tool. The hardware implementation result of Simplified Gabor Wavelet based edge detection method is compared with all other methods and proves that the proposed method is optimal for image edge detection. This architecture is implemented on Virtex 5 Field Programmable Gate Array platform.

Key words: Image edge detection, simplified gabor wavelet, xilinx system generator, field programmable gate array

INTRODUCTION

The quality of image is enhanced by using the various image processing technique which is widely used in many areas such as medical, video surveillance, target recognition and robotics application.

Edges are the pre-dominant features of an image which is mainly used to analyze the images and it plays a vital role in many applications of image processing. The performance of edge detection results affects the further processing such as object recognition and object classification etc. Many image processing applications are subject to a real time constraint. The requirement of real time image processing is not satisfied by software implementation on a computer, a dedicated hardware is needed to fulfill this real time requirement. Even though the general purpose processors are getting faster every day, image and video processing applications still need more computational powers. As an alternative to general purpose processors, such systems can be implemented on Digital Signal Processors (DSP), on the very expensive Application Specific Integrated Circuit (ASIC) or on FPGA. The advantages of FPGA such as high reprogram ability, parallelism, flexibility and computational power prove that the FPGA implementation is better choice than DSP and ASIC based implementation (Draper *et al*., 2003). FPGAs are generally programmed using Hardware Description Language (HDL) which uses a low level, hardware-oriented programming model to fully exploit their potential performance (Kiran *et al*., 2008). But programming FPGAs using HDL is time consuming process and more complicated for large system design. This situation has been changing over with high level programming tools such as Handel-C, AccelDSP and Xilinx System Generator for DSP, etc.

Xilinx System Generator is a high-level software tool that enables the use of MATLAB/Simulink environment to create and verify hardware designs for Xilinx FPGAs quickly and easily (Elamaran *et al*., 2012). It provides a library of Simulink blocks, memories and DSP functions. It also includes a code generator that automatically generates HDL code from the created model. Generated HDL code can be synthesized and implemented in the Xilinx FPGAs. The Simulink blocks can operate in continuous time and floating point format.

But the XSG blocks can operate only in discrete-time and fixed-point format (Basu *et al*., 2012). Design entry is the first step for FPGA design flow, this design entry may be either using HDL language or Schematic based approach in conventional FPGA design. The design entry of XSG based FPGA implementation is block diagram representation of the entire system. Thus by using XSG, complex systems can be implemented easily. The FPGA based hardware implementation of image edge detection using Simplified Gabor Wavelet is proposed in this paper. This system is implemented on FPGA board using a combination of MATLAB Simulink and Xilinx System Generator prototyping environment. Conventional image edge detection algorithm such as Sobel, Prewitt, Robert and LoG are also implemented in FPGA using XSG. The proposed implementation result is compared with conventional approach.

There are many works already implemented on FPGA using Xilinx System Generator tool. Some of that works are listed in this section. Using Xilinx System Generator, Christe *et al*. (2011) proposed FPGA architecture for MRI image analysis and tumor characterization and experimental results for various filtering algorithms are given in this work. This architecture is implemented in XC3S500E-FG320 and only 50% of the resources are utilized for this implementation. A system for image reconstruction of Computer Tomography (CT) and its FPGA based hardware implementation is done by Que *et al*. (2010). This algorithm is co-simulated on both Hardware/Software using simulink and XSG block. This system is implemented on Virtex-2 processor. FPGA based architecture for image edge detection is presented by Sudeep and Majumdar (2011). In this architecture, Sobel, Prewitt and Robert algorithms are implemented on Spartan-3A FPGA device using XSG. FPGA based architecture for feature extraction of digital aerial images is presented by Harinarayan *et al*. (2011). This design is implemented on the Virtex-4 processor using XSG. Sobel and Prewitt based edge detection methods are used for this feature extraction.

Shanshan and Xiaohong (2010) implemented FPGA architecture for image edge detection using Xilinx System Generator tool. This architecture mainly used to identify the edge features of vehicle images. This system is implemented on Spartan-3E board and resource utilization summary is tabulated. The design and implementation of a real time DSP application using prototyping tools such as Matlab-Simulink and XSG is described by Basu *et al*. (2012). The design is implemented on a spartan-3A board and device utilization summary is given. FPGA implementation of an image processing system with the help of XSG is presented by Said *et al*. (2012). This architecture is implemented sobel based for edge detection algorithm. The design was implemented on Spartan 3A and Virtex-5 FPGA devices and their utilization summaries are compared.

FPGA based implementation of conventional edge detectors such as Sobel, Prewitt, Robert and Laplacian of Gaussian (LoG) are already implemented using XSG (Sudeep and Majumdar 2011; Said *et al.*, 2012). But the hardware implementation of Simplified Gabor Wavelet based edge detection using XSG is not available so for. Thus the proposed work of this paper is implementation of the SGW based edge detection using XSG and this proposed work is compared with other conventional method.

ALGORITHM OF SIMPLIFIED GABOR WAVELETS BASED EDGE DETECTION

In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave as follows in Eq. 1 (Jiang *et al.*, 2009):

$$G(x, y) = \exp\left[\frac{x^2 + y^2}{2\sigma^2}\right] \exp\{j\omega(x \cos\theta + y \sin\theta)\} \tag{1}$$

where, σ is the standard deviation of the Gaussian function in the x- and y-directions and ω denotes the angular frequency. Family of Gabor kernels can be obtained from Eq. 1 by selecting different angular frequencies (ω) and orientations (θ). These kernels are used to extract features from an image. The most useful application of the Gabor Wavelets is edge detection (Mehrotra *et al.*, 1992). For given an input image I(x, y), the Gabor Wavelet features are extracted by convolving I(x, y) with G(x, y) as in Eq. 2:

$$\phi(x, y) = G(x, y) \otimes I(x, y) \tag{2}$$

where, ⊗ denotes the 2-D convolution operation. The computation required for Gabor Wavelet based feature extraction is very intensive. This in turn creates a bottleneck problem for real time processing. Hence, an efficient method for extracting Gabor features is needed for many practical applications. Jiang *et al.* (2009) have proposed that the imaginary part of a Gabor filter is an efficient and robust means for face and edge detection. The imaginary part of a GW is as in Eq. 3:

$$S(x, y) = \exp\left[\frac{x^2 + y^2}{2\sigma^2}\right] \sin\{\omega(x \cos\theta + y \sin\theta)\} \tag{3}$$

Set of Simplified Gabor kernels can be obtained from Eq. 3 by selecting different center frequencies and orientations. These kernels are used to extract features from an image. This method is known as Simplified Gabor wavelet (SGW). The number of orientations used in this work is four, i.e., $\theta_k = k\pi/4$ for k = 0, 1, 2, 3. As edges are much localized feature of an image, the value of angular frequency (ω) should be small (Jiang *et al.*, 2009). So edges can be detected efficiently by setting ω = 0.3 and 0.5π. Thus edge detection can be done efficiently by using SGWs of two different frequencies (ω) and four different orientations (θ). The convolution of SGW kernels with the image I(x, y) generates the SGW features and is denoted as $\phi'_{\omega,\theta}(x_c, y_c)$. The resulting SGW feature at a pixel position is equal to the absolute maximum of the eight SGW features as shown in Eq. 4:

$$\phi''(x_c, y_c) = \max\{\phi'_{wi,\theta j}(x_c, y_c), i = 0, 1 \text{ and } j = 0, 1, 2, 3\} \tag{4}$$

where, $\omega_0 = 0.3\pi$, $\omega_1 = 0.5\pi$ and $\theta j = j\pi/4$, for $j = 0,..., 3$. Thus eight different SGW kernels are used to detect the edges. Maximum of these eight convolution results gives the edge map of an input image. Finally thresholding function is applied to get final binary edge output.

METHODOLOGY OF PROPOSED HARDWARE IMPLEMENTATION OF SGW BASED EDGE DETECTION METHOD

The image edge detection methods need to be implemented in hardware in order to meet the real time applications. FPGA implementation can be performed using prototyping environment using Matlab/Simulink and Xilinx System Generator tool. The design flow of hardware implementation of image edge detection using XSG is given in Fig. 1. Image source and image viewer are simulink block sets by using these blocks image can give as input and output image can be viewed on image viewer block set. Image pre-processing and image post-processing units are common for all the image processing applications which are designed using Simulink blocksets. Edge detection modules are different for each and every edge detection algorithms which is implemented using Xilinx System Generator blocksets.

Image pre-processing unit: The gray image is in 2D array size such as R*C where R, C represent the row and column of an image respectively. For XSG implementation, the image must be converted into 1D data array. Image pre-processing blocks is used to convert the 2D image data into 1D data array which is shown in Fig. 2. Image Pre-processing block includes the Transpose, Convert 2-D to 1-D, Frame conversion and Unbuffer block. The Transpose block transposes the R*C input image matrix into C*R sized matrix. Convert 2-D to 1-D block reshapes a C*R matrix input to a 1-D vector. Frame conversion block set the output signal to frame based data and provided to unbuffer block which converts this frame to scalar samples output at a higher sampling rate.

Image post-processing unit: The image post processing block is needed to reconstruct 1-D array into 2-D image which is shown in Fig. 3. This block includes Buffer, Convert 1-D to 2-D block,

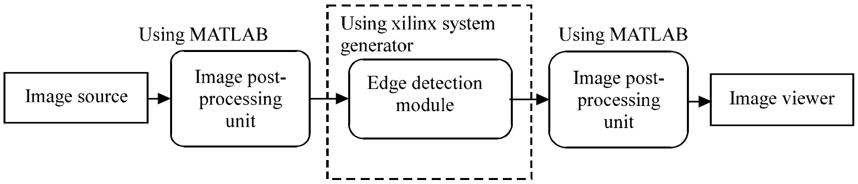

Using MATLAB
Using xilinx system generator
Using MATLAB
Image source
Image post-processing unit
Edge detection module
Image post-processing unit
Image viewer

Fig. 1: Design flow for hardware implementation of image edge detection

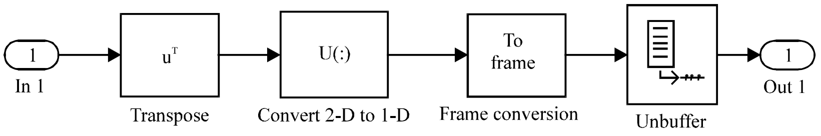


Fig. 2: Image pre-processing unit

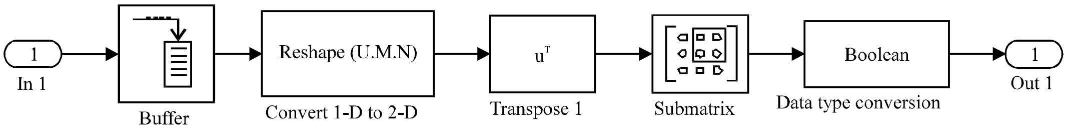


Fig. 3: Image post-processing unit

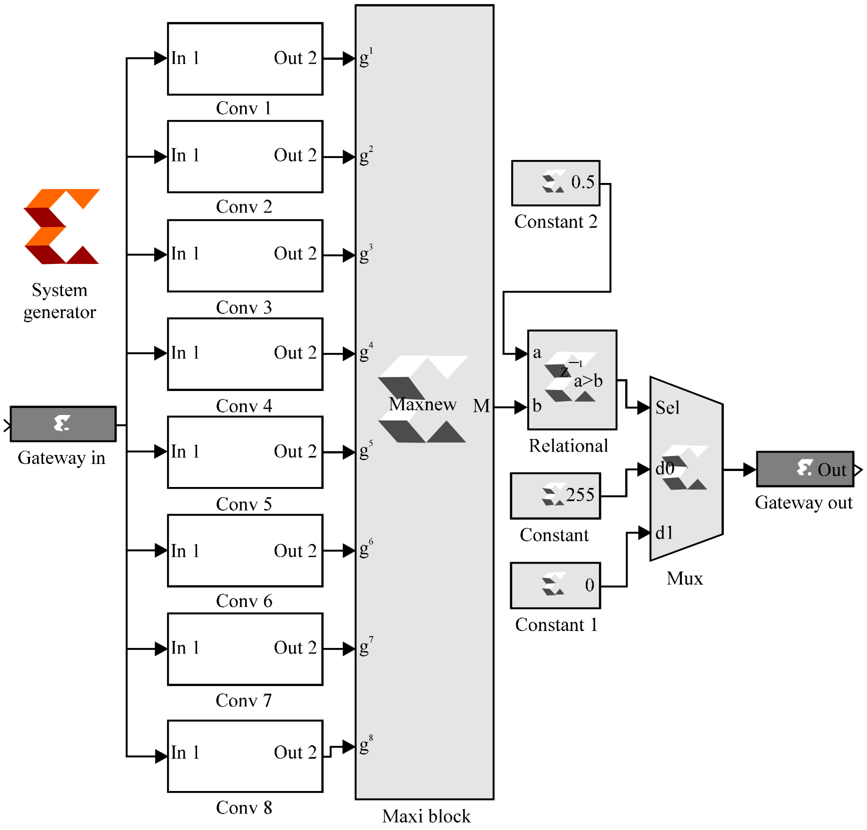


Fig. 4: Block diagram of SGW based edge detection using XSG

Transpose, Submatrix, Data Type Conversion blocks. Buffer changes input sequence into smaller or larger frame size. The Buffer block redistributes the input samples to a new frame size. The Convert 1-D to 2-D block changes 1-D vector data into 2-D data of the size of C*R matrix. The Transpose block transposes the C*R input image matrix into size R*C. The data type is converted using data type conversion block.

Edge detection module for SGW based image edge detection method: The proposed FPGA implementation for Simplified Gabor Wavelet based edge detection is done by using simulink and XSG block. The proposed architecture consists of image pre-processing, image post-processing and edge detection modules as shown in Fig. 1. The image pre-processing and image post-processing blocks are already shown in Fig. 2 and 3. The image edge detection module for Simplified Gabor Wavelet based approach is shown in Fig. 4. By using image pre processing block, 2-D image is converted into 1-D signal.

The eight SGW kernels are implemented using convolution sub systems and detailed block diagram of convolution subsystem is shown in Fig. 5. The coefficient for these convolution blocks are obtained from Matlab implementation of SGW mask which is discussed in previous chapter. Eight SGW convolution subsystems are used to get edge features. The maximum of these eight outputs is found out using M-code subsystem. Thresholding function can be implemented using Mux block, edge output of image can be displayed on video viewer block.

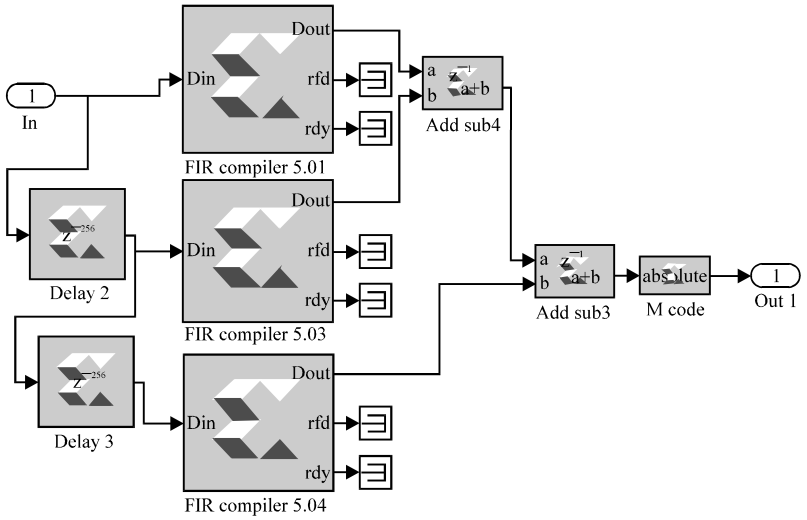


Fig. 5: Detailed block diagram of convolution module in SGW based edge detection

Hardware implementation of conventional image edge detection method: Conventional edge detection method such as Sobel, Prewitt, Robert and LoG can be implemented in FPGA using XSG. In these methods edges are detected by convolving the images with the following respective kernels (Gonzalez and Woods, 1992).

Operators	Gx	Gy
Sobel	$\begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix}$	$\begin{bmatrix}-1 & 0 & -1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}$
Prewitt	$\begin{bmatrix}-1 & -1 & -1\\ 0 & 0 & 0\\ 1 & 1 & 1\end{bmatrix}$	$\begin{bmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix}$
Robert	$\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}$	
LoG	$\begin{bmatrix}-1 & -1 & -1\\ -1 & 8 & -1\\ -1 & -1 & -1\\ & G & \end{bmatrix}$	

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid. The gradient magnitude is given as Eq. 5:

$$|G| = |Gx|+|Gy| \tag{5}$$

Thresholding is applied to identify the edge pixels. If gradient is greater than threshold then that pixels are kept as edge pixel. Otherwise it is noted as non edge pixels.

These conventional edge detection methods are implemented on FPGA using Xilinx System Generator. The design flow is as shown in Fig. 1. By using Image pre-processing block image is

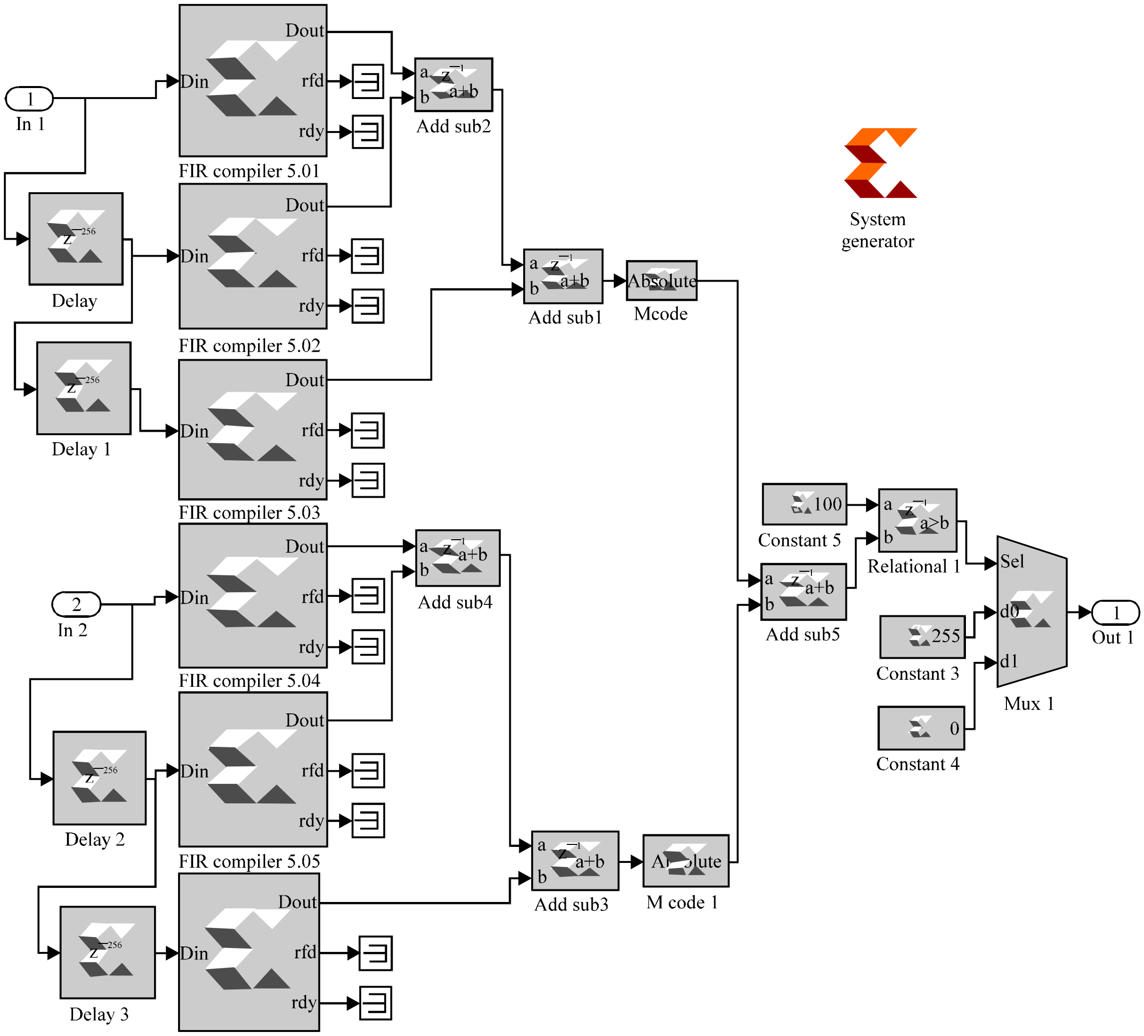


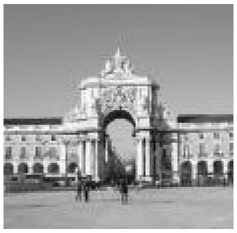
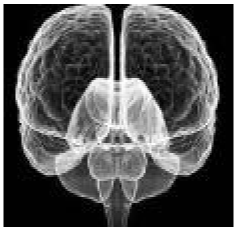
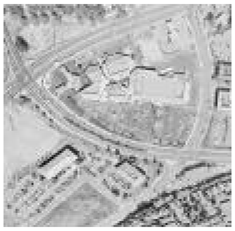
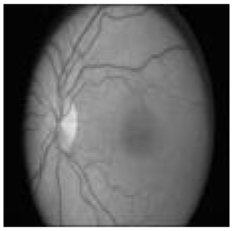
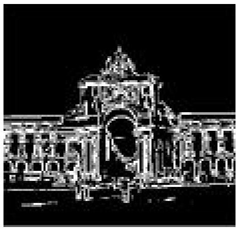
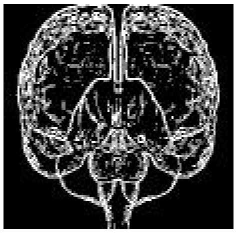
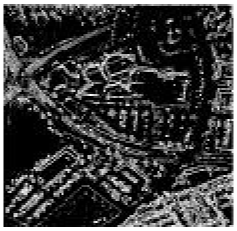
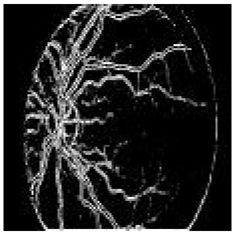
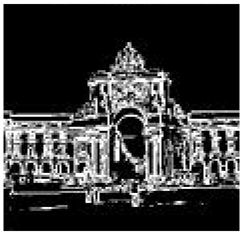
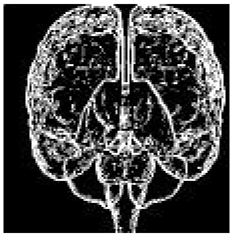
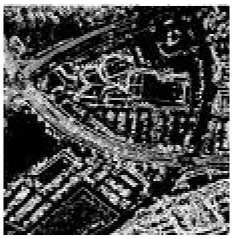
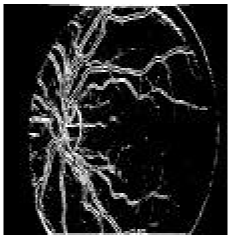
Fig. 6: Edge detection module for hardware implementation of conventional methods

converted into 1D data array. The image edge detection module for this conventional approach is shown in Fig. 6. The main block is edge detection module which is used to perform the arithmetic operation for edge detection method. To implement conventional edge detection method such as Sobel, Prewitt, Robert and LoG, the co-efficient of FIR filter is changed according to the corresponding kernels.

RESULTS

The proposed FPGA implementation for Simplified Gabor Wavelet based image edge detection method was done using MATLAB-Simulink and Xilinx System Generator tool. The SGW based edge detection method was also implemented on MATLAB. Both hardware and software implementation of Simplified Gabor Wavelet based edge detection method for various images was tabulated in Table 1. The conventional edge detection methods such as Sobel, Prewitt, Robert and LoG are also implemented using Xilinx System Generator. The results of proposed hardware implementation of simplified Gabor based approach was compared with conventional methods and this comparison was shown in Table 2.

Table 1: Comparison of hardware and software implementation of SGW based edge detection

Original image				
Software implementation				
Proposed hardware implementation				

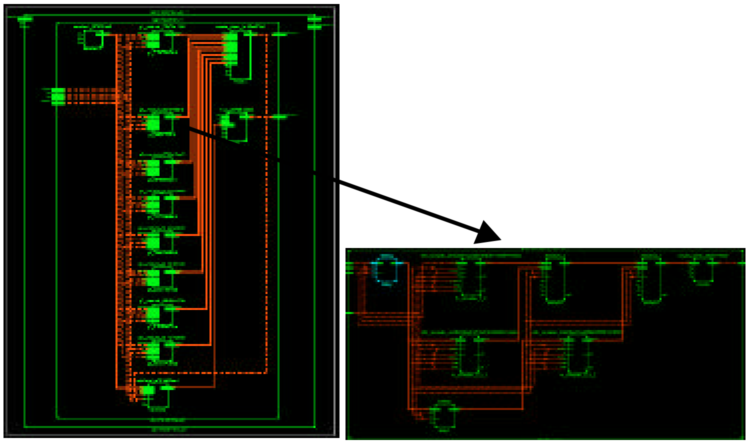


Fig. 7: Top level RTL schematic of proposed architecture and detailed internal RTL schematic of single convolution block

After this simulation result, the entire design was synthesized into Vertex-5 device (5vsx240tff1738-2) and bit stream of the entire system was generated. The generated bit stream was verified by downloading it onto the FPGA board and ensured it ran successfully. The generated top level RTL schematic of the proposed SGW based edge detection system and the internal schematic of single convolution block are given in Fig. 7. The device utilization summary for

Table 2: Comparison of proposed hardware implementation of SGW result with other methods

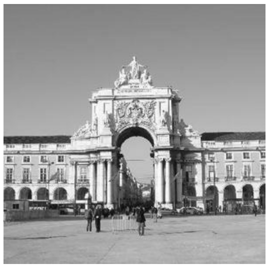
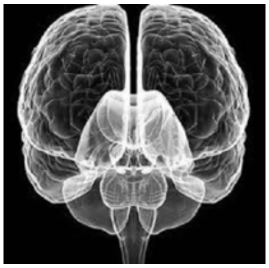
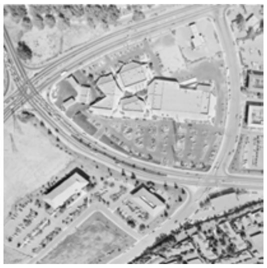
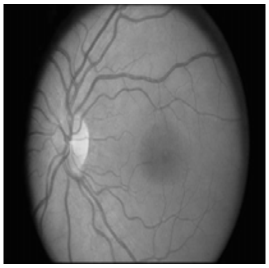
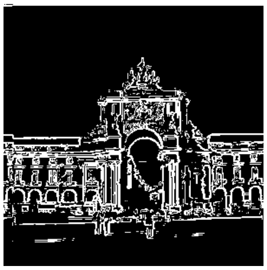
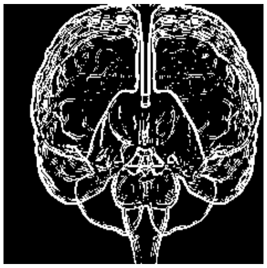
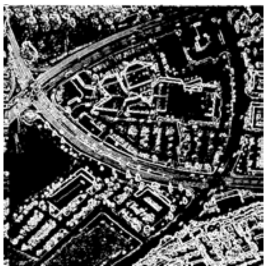
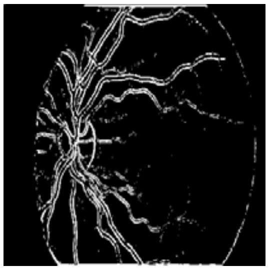
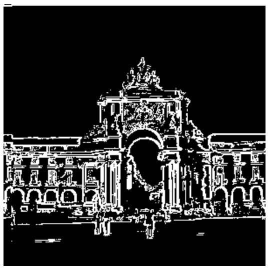
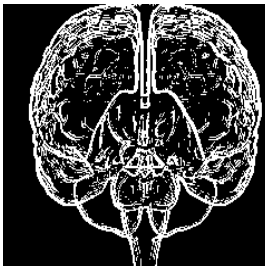
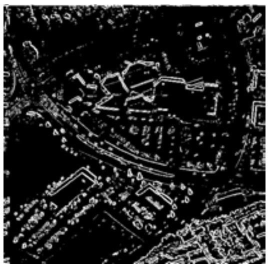
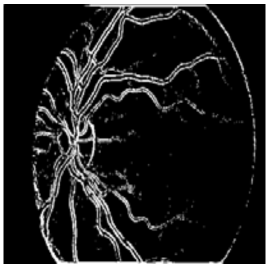
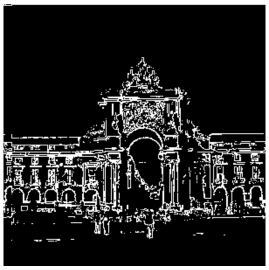
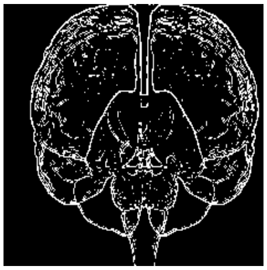
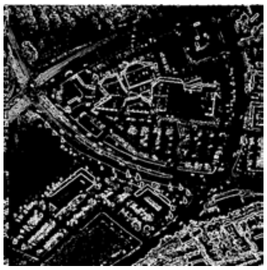
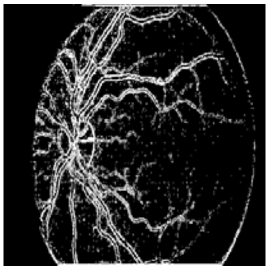
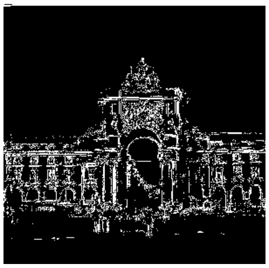
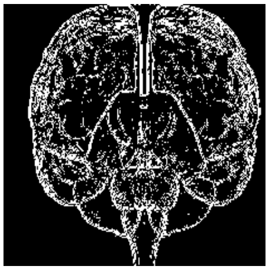
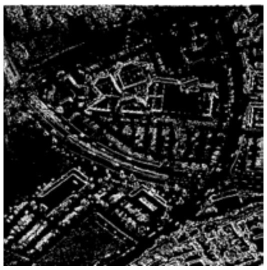
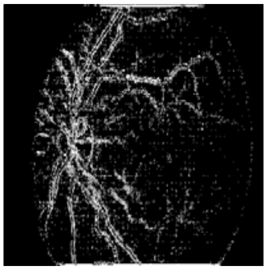
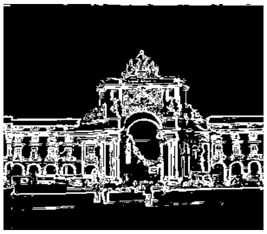
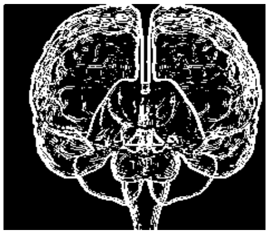
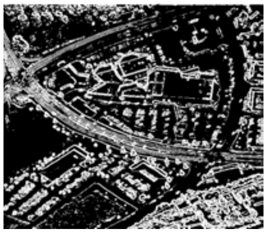
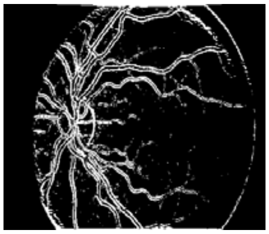
Original image				
Sobel				
Prewitt				
Robert				
LoG				
SGW				

Table 3: Comparison of device utilization summary for proposed FPGA implementation of SGW method with other methods

Edge detection method	Sobel			Prewitt			Robert			LoG			Proposed SGW		
	Used	Available	Usage (%)	Used	Available	Usage (%)	Used	Available	Usage (%)	Used	Available	Usage (%)	Used	Available	Usage (%)
Slices	159	37440	1	162	37440	1	188	37440	1	164	37440	1	2126	37440	5
Slice Rogisters	461	149760	1	461	149760	1	420	149760	1	495	149760	1	5944	149760	3
Slice LUTs	282	149760	1	282	149760	1	350	149760	1	303	149760	1	5613	149760	1
Bonded IOBs	17	960	1	17	960	1	39	960	4	17	960	1	55	960	5
DSP48Es	5	1056	1	5	1056	1	8	1056	1	5	1056	1	72	1056	6
BUFG	1	32	3	1	32	3	1	32	3	1	32	3	1	32	3

Table 4: Comparison of timing summary for proposed FPGA implementation of SGW method with other methods

	Sobel	Prewitt	Robert	LoG	SGW
Speed Grade	-2	-2	-2	-2	-2
Minimum Period ns	2.481	2.481	2.502	2.481	6.482
Maximum Frequency MHZ	403.088	403.088	399.736	403.088	154.269
Minimum arrival time before clock ns	0.98	0.98	1.287	0.98	0.98
Maximum output required time after clock ns	2.826	2.826	3.356	2.826	2.826

proposed FPGA implementation of SGW based method was compared with conventional method implementation and this comparison was tabulated in Table 3. The timing summary for proposed FPGA implementation of SGW method was compared with other methods and this summary was tabulated in Table 4.

DISCUSSION

FPGA implementation of conventional edge detection methods are already done in some other work. But FPGA implementation of SGW based edge detection algorithm is not yet implemented so for. So this proposed FPGA implementation of SGW based edge detection work is new in this area. The comparison result of proposed hardware and software implementation of SGW based method proves that proposed hardware implementation provides better result than software implementation. Proposed SGW based hardware implementation result is compared with conventional approaches. This comparison proves that SGW based approach provides optimal result than other conventional approaches. The device utilization summary shows that the device requirement for all those methods also very less. The proposed hardware implementation of Simplified Gabor Wavelet based image edge detection gives efficient edge map for all types of images.

CONCLUSION

This proposed architecture is constructed using a prototyping environment which consists of MATLAB-Simulink and Xilinx System Generator tool. The result given in this work proves that the proposed hardware implementation of Simplified Gabor Wavelet based image edge detection gives optimal result for all kind of images such as biomedical image and satellite image. Thus this proposed architecture is very well suited for real time image edge detection applications.

REFERENCES

Basu, A., T.S. Das and S.K. Sarkar, 2012. SysGen architecture for visual information hiding framework. Int. J. Emerg. Technol. Adv. Eng., 2: 32-40.

Christe, S.A., M. Vignesh and A. Kandaswamy, 2011. An efficient FPGA implementation of MRI image filtering and tumour characterization using Xilinx system generator. Int. J. VLSI Des. Commun. Syst., 2: 95-109.

Draper, B.A., J.R. Beveridge, A.P.W. Bohm, C. Ross and M. Chawathe, 2003. Accelerated image processing on FPGAs. IEEE Trans. Image Proc., 12: 1543-1551.

Elamaran, V., A. Aswini, V. Niraimathi and D. Kokilavani, 2012. FPGA implementation of audio enhancement using LMS filters. J. Artif. Intell., 5: 221-226.

Gonzalez, R.C. and R.E. Woods, 1992. Digital Image Processing. Addison-Wesley, Reading, MA., USA.

Harinarayan, R., R. Pannerselvam, M.M. Ali and D.K. Tripathi, 2011. Feature extraction of digital aerial images by FPGA based implementation of edge detection algorithms. Proceedings of the International Conference on Emerging Trends in Electrical and Computer Technology, March 23-24, 2011, Tamil Nadu, pp: 631-635.

Jiang, W., K.M. Lam and T.Z. Shen, 2009. Efficient edge detection using simplified gabor wavelets. IEEE Trans. Syst. Man Cybern. Part B: Cybern., 39: 1036-1046.

Kiran, M., K.M. War, L.M. Kuan, L.K. Meng and L.W. Kin, 2008. Implementing image processing algorithms using hardware in the loop approach for Xilinx FPGA. Proceedings of the International Confereence on Electronic Design, December 1-3, 2008, Penang, Malaysia, pp: 1-6.

Mehrotra, R., K. Namuduri and N. Ranganathan, 1992. Gabor filter-based edge detection. Pattern Recognit., 25: 1479-1494.

Que, Z., Z. Yongxin, W. Xuan, Y. Jibo and H. Tian *et al*., 2010. Implementing medical CT algorithms on stand-alone FPGA based systems using an efficient workflow with SysGen and simulink. Proceeding of the IEEE International Conference on Computer and Information Technology (CIT 2010), June 29-July 1, 2010, Bradford, UK., pp: 2391-2396.

Said, Y., T. Saidani, F. Smach and M. Atri, 2012. Real time hardware co-simulation of edge detection for video processing system. Proceedings of the Mediterranean Electrotechnical Conference, March 25-28, 2012, Yasmine Hammamet, pp: 852-855.

Shanshan, Z. and W. Xiaohong, 2010. Vehicle image edge detection algorithm hardware implementation on FPGA. Proceedings of the International Conference on Computer Application System Modeling, Volume 14, October 22-24, 2010, Taiyuan, pp: 184-188.

Sudeep, K.C. and J. Majumdar, 2011. A novel architecture for real time implementation of edge detectors on FPGA. Int. J. Comput. Sci., 8: 193-202.