HARDWARE SOFTWARE CO-SIMULATION OF EDGE DETECTION FOR IMAGE PROCESSING SYSTEM USING DELAY BLOCK IN XSG

Aniket A. Ingle[1], Vrushali G. Raut[2]

[1]*Dept. of Electronics and Telecommunications, Sinhgad College of Engineering, Pune, India*
[2]*Dept. of Electronics and Telecommunications, Sinhgad College of Engineering, Pune, India*

Abstract

This paper implement an image processing algorithm applicable to Edge Detection for still image in a Xilinx FPGA using System Generator. We prefer sobel algorithm which is most reliable and gives us an efficient output. If we prefer to write HDL code for such algorithm in Xilinx FPGA then it's too bulky and time consuming. We design this system with use of Xilinx System Generator blocks. Its tool with a high- level graphical interface under Matlab environment Its Simulink based blocks which makes it very easy to handle with respect to other software for hardware description.

Keywords— *Matlab, Xilinx System Generator, FPGA, Edge detection algorithm.*

----------***----------

1. INTRODUCTION

For human interpretation image processing improve quality of image. There are lots of examples where image processing helps to analyze, infer and make decisions. Image processing work pixel by pixel

In image processing before feature extraction and object segmentation we have to detect edges from frame. This edge detection process detects outlines of an object. Object and background in image is isolated by edge detection feature. A sharp discontinuity in image is located by edge detection. The boundary of object in any image is characterized by discontinuity which gives instant change in pixel intensity. This process compress image without losing any important feature of that image. Edge detection is initial stage of image processing but it corrupted with white noise always. These edge detection methods are classified into two categories, gradient and Laplacian. Matrix area gradient operation is basic Edge detection operator. By putting mask on detected pixel calculate value of detected point using edge detector operator. If the value is greater than threshold value then it considered as edge point. Robert, Prewitt and Sobel are example of gradient based edge detection algorithm .Kernel operator play a vital role in gradient based edge detection algorithm which calculates the slope in directions that are perpendicular to each other.

The goal of this paper is to implement an image processing algorithm applicable to Edge Detection system in a Xilinx FPGA using System Generator with a focus on achieving low cost and short development time. Edge detection algorithms are implemented on software. But now days hardware approach has become an alternative where complex computation reduce and with help of FPGA or Reconfigurable device we can use parallelism and pipelining concept which easily increase speed. By Implementing image processing on reconfigurable hardware like FPGA which reduce the time of production cost, enables rapid ASIC prototyping of complex algorithm .It also simplifies debugging and verification.

2. LITERATURE REVIEW

A lot of work done on edge detection algorithm to detect edge of an object .On the basis of edge detection algorithm we can improve quality of image for human interpretation. Image processing used in various field now days such as in medical application, for digital aerial image detection from satellite, for vehicle detection etc.

There are broadly three methods to detect edges
1) First order derivative (Gradient Method) Method.
   Example: A) Robert Operator
   B) Prewitt Operator
C) Sobel Operator
2) Second order derivative Method.
   Example: A) Laplacian
   B) Laplace of Gaussian
   C) Difference of Gaussian
3) Optimal edge detection method.
   A) Canny edge detection.

The derivative operators are used for image enhancement or to enhance the details present in the image and these derivatives operations can be used for detection of edges present in the image.

In paper [1] represent how to implement an image processing algorithm applicable to Edge Detection system in a Xilinx FPGA using System Generator for still image, with a focus on achieving overall high performance, low value and short development time to come in market. The design of edge detection is demonstrate with help of Xilinx system generator block in Simulink environment. Spartan 3A board use to implement this design.

In this paper [2] use System Generator tool in developing vehicle image processing edge detection algorithms which is developed by Xilinx based on MATLAB. Edge detection algorithm model and design are finished in MATLAB Simulink, preparation of top-level file in ISE 10.0 environment then achieve a System Generator functions and other modules instantiated. Import the hardware design which generate by System Generator into the paper, and then the paper should be simulated, synthesis, finally completed the hardware-based of the algorithm. And display the processing image through VGA.

This paper prefers first order derivative method over second order derivative method for edge detection. First derivation can be computed by using gradient operators .The second order derivative is very sensitive to noise present in the image and that is the reason second derivative operators are not usually used for edge detection operation but the second derivative operators are gives some secondary information, sign determine whether the point is lying on the darker side of the image or a point is lying on the brighter side of the image.

This paper prefer Sobel operator over Prewitt and Robert Operator. Robert filters have shortest support and more vulnerable to output noise. The Prewitt operator is based on the idea of central difference and is much better than Roberts's operator. Prewitt's operator has longer support and is less vulnerable to noise. The Sobel operator is also a central difference with more weights to the central pixels where averaging as given by equation. It has improved noise suppression than Prewitt's operator.

In third section we describe about edge detection, fourth section gives idea about proposed design in which we describe flow of Xilinx system generator(XSG),sobel edge detector operator and at last we discuss on implementation and application which can be implemented and use in practical world with help of our proposed design.

Canny edge detection is optimal edge detection method. According result of several research papers canny gives us finest edge detection as compare to First order derivative method or second order derivative method.

XSG (Xilinx System Generator) is tool with high level graphical interface under matlab environment with Simulink based block. This is finest way for hardware approach. It provides easier hardware verification & implementation compare to HDL based approach. Our goal to achieve high performance, low cost, short development time using Xilinx system generator (XSG) it's easily fulfils. It directly generate .UCF file of VHDL or VERILOG code which we can burn directly on FPGA board.

3. SOBEL EDGE DETECTION ALGORITHM

The Sobel operator is type of first order edge detection operator. It computes the gradient of image intensity function.

At every point in the image the resultant gradient at this point is given by Sobel norms. There are only 0 and 90 degree convolution kernel used by sobel operator.

The magnitude of gradient at each point is find out by combining these individual kernel. The gradient magnitude is given by:

$$|H| = \sqrt{Hx^2 + Hy^2} \qquad (1)$$

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

Hx (Convolution kernel in x direction)

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

Hy (Convolution kernel in y direction)

Fig.3.1: Convolution Kernel [1].

The magnitude of gradient at each point given by:

$$|H| = |Hx| + |Hy| \qquad (2)$$

This is much faster to compute. The sobel operator has advantage of simplicity in calculation. Edge is detected with help of two convolution kernel that's reason it has low accuracy. This convolution kernel shown in figure 3.1 [1].

4. PROPOSED WORK

The entire operation of edge detection proposed using Simulink and Xilinx blocks goes through 3 phases,
A) Image pre-processing blocks.
B) Edge detection using XSG.
C) Image post-processing blocks

4.1 Image Preprocessing Blocksets

The model based design used for image pre-processing is shown in figure 4.1[5]. The blocks utilized here are discussed below. Input images which could be colour or grayscale are provided as input to the File block. A colour space conversion block converts three channel RGB image into two channel grayscale image.

This data which is in 2D is to be converted to 1D for further processing. Frame conversion block sets output signal to frame based data of particular size and provided to unbuffer block which converts this frame to scalar samples output at a higher sampling rate.

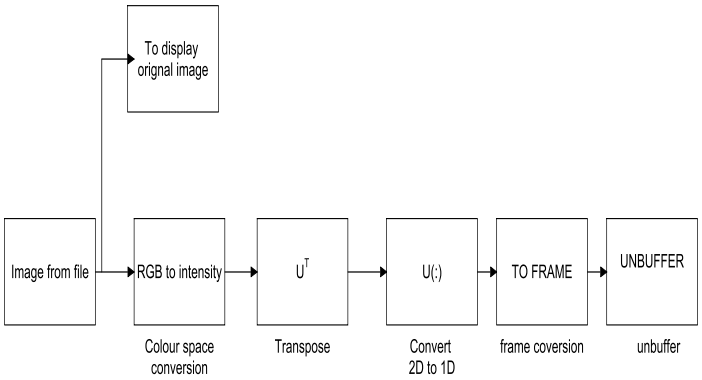

Fig. 4.1: Image Pre-Processing.

4.2 Edge Detection using Xilinx Blocksets

The model based design using Xilinx blocksets for processing the input image for edge detection is shown in figure 4.2, Xilinx fixed point type conversion is made possible by Gateway In block. To perform the edge detection a convolution operation of the input image with vertical and horizontal mask which is made up of delay block, adder and substractor. This is followed by certain arithmetic blocks to merge all the processed data's.

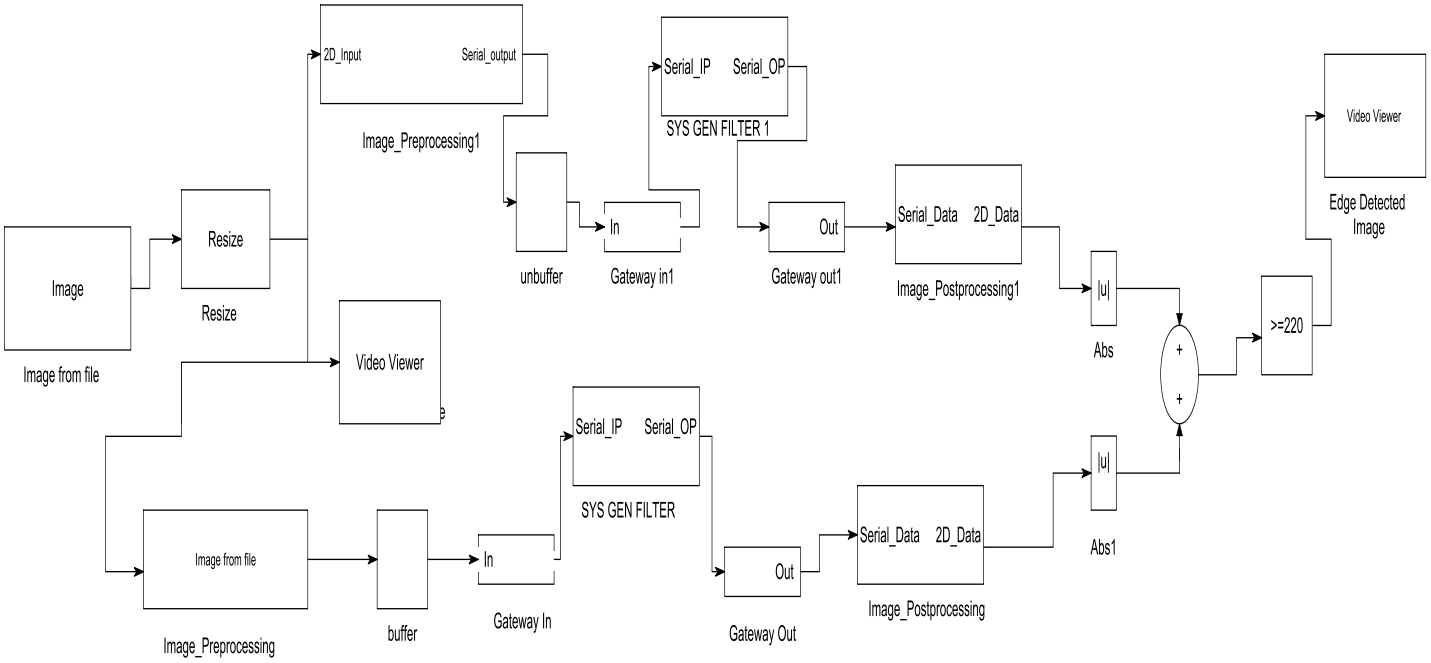

Fig.4.2: XSG Model for edge detection

4.3 Image Post Processing Blocksets

The post-processing blocks which are used to convert the image output back to floating point type we get 2d image at output is shown in figure 4.3. It compare with threshold value and finally we get fine edge detected image which we can see in Simulink environment.

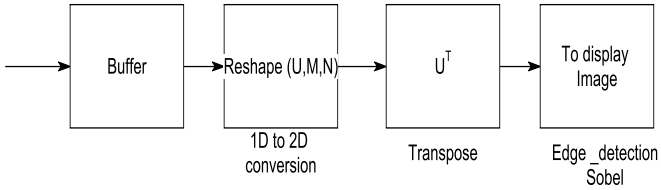

Fig.4.3: Image Post-Processing.

5. HARDWARE IMPLEMENTATION

For implementation of this design in a FPGA board the entire module should be converted to FPGA synthesizable format. For that purpose main module for edge detection is converted to JTAG hardware co-simulation, this is done with the help of System generator block specially its system generator token. This block is configured according to the target platform and a bit stream (*.bit) file is generated. After the bit stream file is generated, hardware co-simulation target is selected and in this work, Spartan 3E starter kit (XC3S500E-FG320) is used for board level implementation. The entire architecture with the hardware and software co-simulation design is shown in figure 5.1.

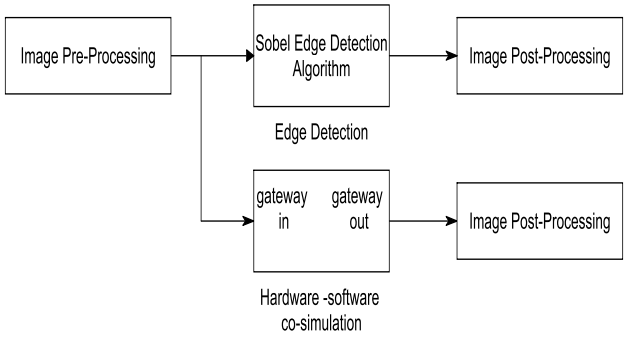

Fig.5.1: FPGA Hardware Implementation.

6. APPLICATION

There are various application in image processing which needs edge detection and where we can easily apply our proposed hardware software co-simulation technique such application are given below:

1. It use in a Biometrics.
2. Edge detection of Digital aerial image.
3. Vehicles detection at check post.
4. For machine vision also we can use it.

7. RESULTS AND COMPARISONS

We use image preprocessing blocks for converting our RGB image into grayscale then this 2D image is again converted into 1D for hardware approach. Sobel Edge detection model made with help of XSG blocks in Simulink environment. Our edge detected image finally goes through image post processing block where it again converted into 2D image. Edge detected image obtain by this approach is given below in figure 7.2.We use targeted board for implementation is Xilinx Spartan 3E XC3S500e.

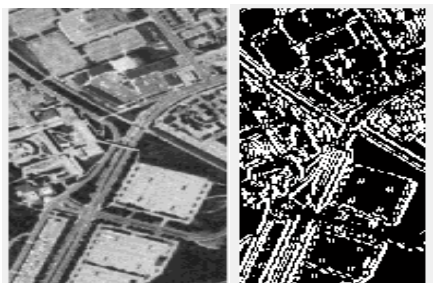

Fig.7.1: Original image **Fig.7.2:** Edge detected image

Table 1 Proposed work for XSG model of Edge Detection

Parameter	**Proposed Work**	
Slices	51	1%
Slice Flip Flop	97	1%
Slice LUTs	8	0 %
Block RAM	0	0 %
IOB's	39	16%

Table 2 Comparisons of proposed with existing design

Parameter	**Proposed Design**	**[1]**	**[2]**
Platforms	Xilinx Spartan 3E XC3S500e	Xilinx Spartan 3E XC3S500e	Xilinx Spartan 3E XC3S500e
Slices	51	163	326
Slice Flip flop	96	116	232
LUT	8	130	260
IOB	39	49	38
Frequency (MHz)	249.128	-	220.7

8. CONCLUSIONS

Edge detection using software is not tough job but when we are going to implement it on hardware we have to face challenges like total VHDL code or Verilog code actually becomes very bulky it's near about 5000 lines. To shrink it we use Xilinx system generator. Simulation speed increase by this hardware software co-simulation technique .We can easily go for ASIC prototype by this approach. This design is implemented in the Xilinx FPGA Development kit.

REFERENCES

[1] Yahia Said, Taoufik Saidani, Fethi Smach and Mohamed Atri "Real Time Hardware Co-simulation of Edge Detection for Video Processing System", 2012 IEEE.

[2] Zhang Shanshan, Wang Xiaohong "Vehicle Image Edge Detection Algorithm Hardware Implementation on FPGA", 2010 International Conference on Computer Application and System Modeling (ICCASM 2010).

[3] Kobzili EI Houari, Benbouchama Cherrad, "A Software-Hardware Mixed Design for the FPGA Implementation of the Real-Time Edge Detection", 2010 IEEE

[4] Gonzalez, Rafael C, 2008. "Digital Image Processing", Pearson Education, Inc., publishing as Prentice Hall.

[5] Mohamed Nasir Bin Mohamed Shukor, Lo Hai Hiung, Patrick Sebastian3, 2007. "Implementation of Real-time Simple Edge Detection on FPGA" pp. 1404-1405,IEEE

[6] Nick Kanopoulos, NageshVasanthanada, Robert L.Baker, 1988. "Design of an Image Edge Detection Filter Using the Sobel Operator" pp. 359, IEEE.

[7] J.F. Canny, "A computation approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no.6, pp. 769-798, Nov 1986.

[8] D. Marr and E.Hildreth,"Theory of edge Detection," Proc. Royal Soc. of London, series B, vol. 207, pp. 187-217, 1980.

[9] D. Demigny, and T. Kamle," A discrete expression of Canny's criteria for step edge detector performances evaluation", IEEE Transactions on

Pattern Analysis and Machine Intelligence, pp. 1199-1211, 1997.

[10] H. Neoh, A. Hazanchuk, "Adaptive Edge Detection for Real-Time Video processing using FPGAs", Global Signal Processing (2004)

[11] SHIGERU.A, "Consistent Gradient Operators", IEEE Transactions on Pattern Analysis and Machine Intelligence, 22 (3), 2000.

[12] Dong, Q., Song, C., Ben, C., Quan, L., "A fast sub pixel edge detection method using Sobel-Zernike moments operator", Image and Vision Computing, Vol.23, pp.11-17, 2005.Minimal Resources", Proc. International Conference on Intelligent Systems and Signal Processing, pp. 338-343 2013.