Sobel Edge Detection Using FPGA-Artix®-7

Parth V. Parikh, Bhinjan A. Dalwadi, Ghanshyam D. Zambare

Abstract: Image processing requires extensive calculation. Image processing is conventionally done on CPU. Image processing on CPU is slow due to its sequential processing method. However FPGAs can be seen as an option to speed up image processing without increasing the clock speed because they allow parallel processing. In this project FPGA is used to implement the edge detection operation on image. The Sobel algorithm is used to detect the edges in an image. The Sobel algorithm is implemented on Artix®-7 Field Programmable Gate Array (FPGA) from Xilinx® using Hardware Description Language (HDL) and Intellectual Property (IP) cores.

Index Terms: Image Processing, Edge Detection, Sobel Operator, Sobel Top Module, VHDL, Xilinx® Artix®-7, VGA display.

1 INTRODUCTION

Image processing is a method to extract required information, enhance raw image data. Image processing is extensively used in industries for rapid quality testing, in security fields for threat monitoring, in entertainment field for making high quality media. Image processing is widely used in many applications. Edge detection is one of the gist of all these image processing applications.

Aspects of Image Processing:-

- Enhancement: Processing an image as per the application requirements. (Sharpening or smoothing an out of focus image, highlighting edges, improving image contrast or brightening an image, removing noise)
- Restoration: This may be considered as reversing the damage done to an image by a known cause. (Removing of blur caused by linear motion, removal of optical distortions)
- Segmentation: This involves subdividing an image into constituent parts or isolating certain aspects of an image. (Finding lines, circles, or particular shapes in an image, in an aerial photograph, identifying cars, trees, buildings, or roads.

Edge detection as a part of image enhancement and its results are essential for segmentation. It can be used as preliminary image processing step in object recognition, object tracking, face detection and recognition, biometrics, packaging etc. The edges of image are considered to be the most important image attributes. The edge detection in image processing is particularly used in the areas of feature extraction, algorithms which aim to identifying points in a digital image at which the image brightness changes sharply. The systems used for image processing, are mainly designed on the CPUs or GPUs or Application Specific Integrated Circuit (ASIC) for very high performance requirements. The implementation of image processing algorithms on CPU and GPU has limited options for parallel processing, which limits the maximum processing speed.

- *Parth V. Parikh, Bhinjan A. Dalwadi, Ghanshyam D. Zambare are currently pursuing bachelor's degree in Electronics Engineering in Birla Vishvakarma Mahavidhyalaya, Gujarat technological University, India*
- *Email: parthpower@gmail.com, bhinjan@gmail.com, ghanshyamzambare@gmail.com*

ASIC has parallel hardwired architecture optimized for one or more specific algorithms. ASIC has very limited options for future extensibility however it provides enormous speed-power performance compared to CPU and GPU due to its architecture. FPGA are completely reconfigurable devices which allows it to be as flexible as CPU/GPU based image processing algorithms and as fast as ASIC based design. The aim of this project is to design an FPGA based system that can perform neighborhood image processing faster than general CPUs and GPUs. The

2 EDGE DETECTION THEORY

Edge detection is method of detecting discontinuities or sharp intensity change. This is possible by simple derivation of intensity in 2D plane and detect maxima of derivative. There are several algorithms developed to do the same. There are several edge detection methods such as Sobel, Prewitt, Roberts, Canny. These methods have been proposed for detecting transitions in intensity in various parts of images. Early methods determined the best gradient operator to detect sharp intensity variations. Commonly used method for detecting edges is to apply derivative operators on images. Derivative based approaches can be categorized into two groups, namely first and second order derivative methods.

2.1 Sobel Operator [HYPERLINK "" \l "Xia10" 1]

The Sobel edge detection technique has two masks, one along the horizontal direction and the other in vertical direction. These masks are generally 3x3 matrix. The image is convolved with only two masks and this technique also has some smoothing effect to the random noise of the image. The image is convolved with the kernels estimating the gradient in the x-direction and the other in the y-direction. The two kernels of the Sobel edge detection technique are:

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \qquad G_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Every point in the image is convolved with these two kernels. One kernel out of the two has a maximum response to the horizontal edge while the other has the maximum response the vertical edge of the image. The maximum value out of the two convolutions is used as the output bit of the point, and the result is an edge detected image.

2.2 Image Gradient

An image gradient is a directional change in the intensity or color in an image. Image gradients may be used to extract information from images. There are two types of gradient:-

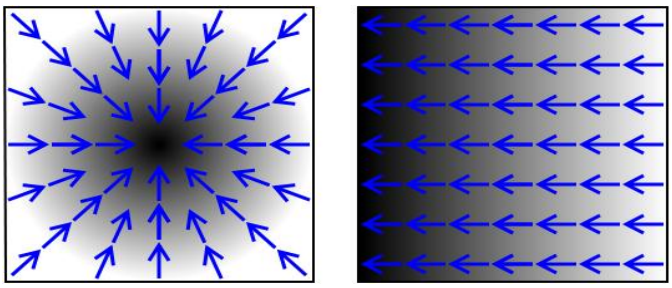

Fig.1. *Image Gradient*

An image gradient is the change in the intensity or the color in an image. The convolution of the pixel values of the image with the Sobel kernels will return the gradient in the X-direction and Y-direction. The Sobel gradient here is calculated using the equation below.

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

G is the Sobel gradient in this case and Gx and Gy are the gradient in X-direction and Y-direction respectively. There are two types of differentiation masks which are explained with help of processed images given as follows

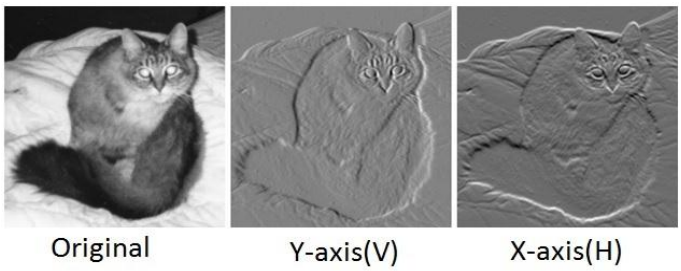

Fig.2. *Sobel Gradient in X & Y direction*

3 Implementation on FPGA using VHDL

In this implementation, a top module Finite State Machine (FSM) controls the whole process. Top module and all the sub modules are written and tested in Very High Speed Integrated Circuit Hardware Description Language (VHDL). IP cores from Xilinx® are used for generating clock used by the VGA interface and generating Block Memories for storing the image data.

3.1 System Archietecture

The system architecture design is highly focused on parallel processing and minimum clock cycle utilization. In this implementation Xilinx® Artix-7® XC7A100TCSG324-1 FPGA is used.

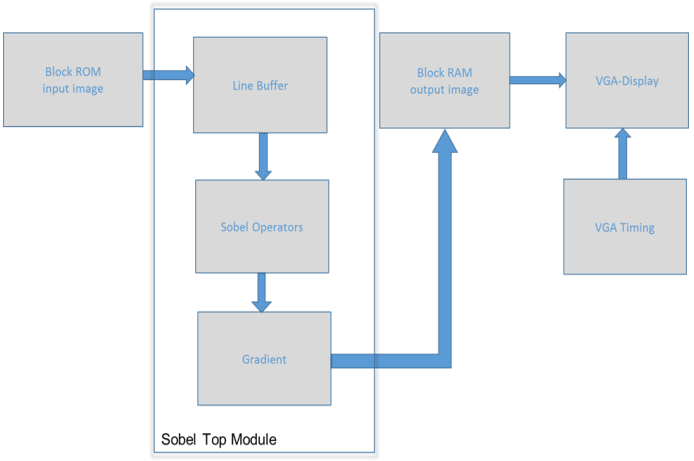

Fig.3. *Structural Block Diagram*

The structural block diagram show the 5 main blocks of the process which are Block ROM, Sobel top module, Block RAM, VGA display, VGA timing block. The Block ROM and Block RAM are generated using the Xilinx IP cores. The 8bit intensity level values of the grayscale image pixels are stored in BROM. The image pixel values are feed as input to the Sobel top module. The Sobel top module consists of, Line Buffer followed by Sobel Operator and later on Gradient. The Line Buffer are essentially shift registers with parallel output. It generates samples to perform convolution with the 3x3 kernel. These sample values of images are convolved with the Sobel X and Sobel Y operators and the resultant gradient is calculated to have the final output values of the edge detected image. In Sobel operator block, we have generated a combinational logic for performing arithmetic operation, which provides gradient of the image as a result i.e. output. These output values are then stored in the BRAM. The processed image is then displayed on a monitor through the VGA controller. A separate timing unit is designed for the control of the VGA controller.

3.2 Block Memory RAM/ROM

Block Memory are hardwired configurable memory blocks present in Xilinx® FPGAs. A (255*255) x 8bit Block RAM as well as Block ROM is used to store pixel data of input and output image of size 255 x 255 with 8bit grayscale resolution. The intensity data of the test image are converted to Coefficient (COE) file using Python program and dumped into the Input Block ROM. The Block RAM stores the gradient data of the input image. It is connected to the output of the Sobel Top Module and directly interfaced with RGB data lines of the VGA interface. Features:

- Low latency memory controller
- Separate read and write channel interfaces to utilize dual port FPGA BRAM technology
- Configurable BRAM data width (8-, 32-, 64-, 128-, 256-, 512-, and 1024-bit) (equals AXI slave port data width size)
- Supports memory sizes up to a maximum of 2 M Bytes (byte size 8 or 9)
- Performance up to 450 MHz
- Data widths from 1 to 4096 bits
- Memory depths from 2 to 128k

3.3 VGA Interface

VGA stands for Video Graphic Array, it is a standard video interface. A VGA video signal requires minimum 5 signals to completely display image:

- Horizontal Sync (Row Sync): Indicates when line is changed
- Vertical Sync (Frame Sync): Indicates when frame is changed
- Red (R)
- Green (G)
- Blue (B)

The RGB signals are the analog signals with two voltage levels i.e. 0-0.7v. Different colors can be produced by changing the analog levels of these color signals accordingly. The timing specification of Horizontal Sync (Hsync) and Vertical Sync (Vsync) signals depends on the resolution of the raster and the refresh rate. The Row Sync signal is high for the entire period necessary to display one line on the screen, when this signal goes low it marks the end of the previous line and the start of the next one. While the Row Sync signal is high the monitor displays one pixel for each rising edge of the pixel clock. The Vertical Sync signal marks the start and the end of a frame. This signal is high for the time necessary to display one entire frame on the screen. While this signal is high the monitor uses the Row Sync signal to determine when it has to display a new line. In both of the sync signals there are two intervals, called Front Porch and Back Porch, during which the pixel data sent to the display are not being displayed. This implementation uses VGA interface configured for 640x480 at the refresh rate of screen at 60 Hz. The timings for the synchronization signals are reported in the Table.1.

Resolution (pixels)	Refresh Rate (Hz)	Pixel Clock (MHz)	Horizontal (pixel clocks)				Vertical (rows)			
			Display	Front Porch	Sync Pulse	Back Porch	Display	Front Porch	Sync Pulse	Back Porch
640x480	60	25.175	640	16	96	48	480	10	2	33

Table.1 *VGA Specifications*

3.4 VGA Timing Generator

In the current implementation of the VGA Driver is configured for 640x480 resolution at 60 Hz refresh rate. Pixel Clock is generated using LogiCORE IP Clock Generator (v4.03a). The Clock Generator is configured for 25.175 MHz clock with Phase Lock Loop (PLL) configuration. A PLL is a voltage or current-driven oscillator that is constantly adjusted for stable clock frequency. PLL are which are hardwired in the FPGA fabric. According to the standard specifications for this configuration, sync pulses are generated with an FSM. Along with the sync pulses it gives the location of the current pixel position on the display, which is used in Block RAM to specify the address of the pixel to display.

3.5 VGA Controller

There are two blocks involved in the process of displaying the processed image as final output, one is VGA Controller and the second one is the Block RAM. As the image processing is carried out at master clock of 100 MHz whereas the displaying process on the VGA is done at 25.175 MHz, it is required to change the clock input of the Block RAM from master clock of 100 Mhz to 25.175 MHz for proper synchronization. The VGA Controller provides the Hsync and Vsync pulses directly to the display. The pixel location outputs from VGA Controller are used in an FSM to generate proper address location of the Block RAM for the current pixel location on the display. The data signals for the display wiz R, G and B signals are generates from the output data of the Block RAM. The R, G and B output signals from the FPGA are digital, which are converted into analog using DAC.

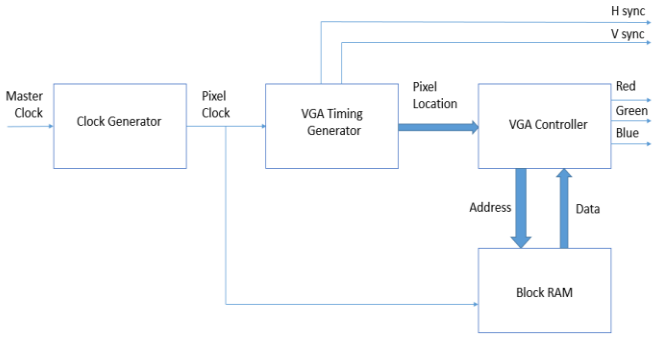

Fig.4 *VGA Controller*

3.6 Sobel Top Module

The Sobel top module is has four input signals and two output signal. Out of these four input line one is, 8bit data line that is used to provide image pixel data as input to Line Buffer. Line Buffer has 9 parallel outputs corresponding to a center pixel and its 8 neighborhood pixels. The output data from the Line Buffer is connected to a combinational logic – Sobel Operator Module - for applying two Sobel mask at a time. The Sobel Operator module is connected to a Gradient module which calculates the resultant gradient. The other three are the control lines – clk (clock input), rst (reset), en (enable). The clock signal is used for the synchronization, reset is used to reset the Line Buffer values to all zeros and the enable signal indicates write enable for the Line Buffer. The output of this Sobel top module is a 17 bit data that is scaled down by using most significant 8bits and stored in the Block RAM.

4 SIMULATION RESULTS

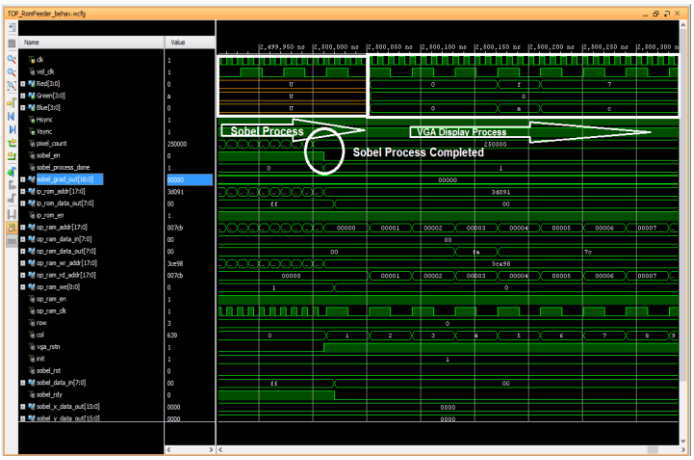

Fig.5 *Simulation Results*

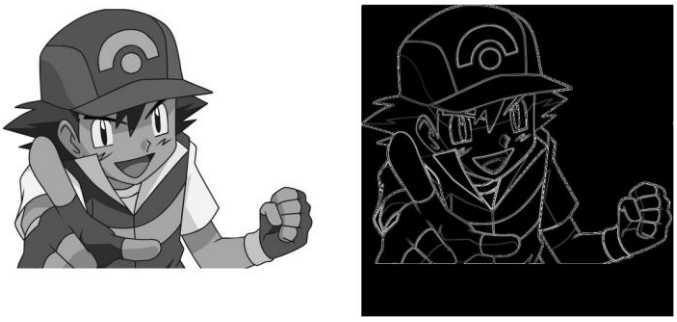

Fig.6. *Sobel Edge Detection of an image*

5 CONCLUSION

Displayed an image on screen through VGA. Image pixel data stored in block ram which was displayed. This is primary block for any dedicated image processing application. Data manipulation modules can be interfaced between the VGA output port and the image storage element. Such data manipulation modules are for image processing which are essentially digital signal processing blocks. These blocks incorporates image processing algorithms and processed image data can be buffered or directly given to output which is, in this case VGA.

References

[2] Raman Maini and Dr. Himanshu Aggarwal, "Study and Comparison of Various Image Edge Detection Techniques," International Journal of Image Processing, p. 12.

[3] Tsutomu Maruyama and Yoshiki Yamaguchi Shuichi Asano, "PERFORMANCE COMPARISON OF FPGA, GPU AND CPU IN IMAGE PROCESSING," IEEE, p. 6, 2009.

[4] Xilinx. (2011, January) Xilinx. [Online]. HYPERLINK "http://www.xilinx.com/support/documentation/ip_documentation/bram_block.pdf" http://www.xilinx.com/support/documentation/ip_documentation/bram_block.pdf

[5] Xilinx. (2012, January) Xilinx DS614 LogiCORE IP Clock Generator (v4.03a), Datasheet. [Online]. HYPERLINK "http://www.xilinx.com/support/documentation/ip_documentation/clock_generator/v4_03_a/clock_generator.pdf" http://www.xilinx.com/support/documentation/ip_documentation/clock_generator/v4_03_a/clock_generator.pdf

[6] Shahabuddin. Inamdar, "VHDL Coding Style Guidelines," University of South Florida, 2004.

[7] Xiaoguang Zhang, Lei Yang, and Huizhong Liu, "An improved sobel edge detection," 3rd IEEE international conference , pp. 67-71, 2010.