www.ijrde.com Vol.2: Issue.1, June-July 2013 pp- 22-32

FPGA Based Edge Detection Using Modified Sobel Filter

Dr. Abdulsattar M. Khidhir[1], Nawal Younis Abdullah[2]

*(Computer Systems Department, Technical Institute/ Foundation of Technical Education, Iraq)
Email: abdulsattarmk2@yahoo.com)
** (Computer Engineering Department, Technical College/ Foundation of Technical Education, Iraq
Email: nawalalhadedee@yahoo.com)

ABSTRACT

In recent years, edge detection technology has gradually been widely used. This paper presents one of the classical edge detection operators, Sobel edge detector. Field Programmable Gate Array (FPGA) technology becomes an alternative for the implementation of software algorithms. This paper presents FPGA based architecture for Sobel operator using Virtex-5 ML506 board to find the edges for grayscale images. Firstly, the standard Sobel operator is used to detect the edges in grayscale images. Then the Sobel operator is modified to find the edges for the images with noise reduction. The system for edge detection is done with the combination of EDK(Embedded Development Kit) and Matlab environments.

Keywords **- -Edge Detection, EDK, Sobel Operator, Virtex-5**

I. INTRODUCTION

Edge detection is a fundamental tool used in most image processing applications to obtain information from the frames as a precursor step to feature extraction and object segmentation. This process detects outlines of an object and boundaries between objects and the background in the image [1].

Edges are significant local changes of intensity in an image. Various physical events cause intensity changes: Geometric events and Non-geometric events. Geometric **EVENTS** include: object boundary (discontinuity in depth and/or surface color and texture) and surface boundary (discontinuity in surface orientation and/or surface color and texture). Non-geometric events include: specularity (direct reflection of light, such as a mirror), shadows (from other objects or from the same object) and inter-reflections [2].

Edge detection is one of the most commonly used operations in image processing, which is the subject of research for many researchers, for example, Priyanka S. Chikkali et al. proposed a work of FPGA based architecture for edge detection using Sobel operator and uses histogram method for segmentation[3].

In [4] Sanjay Singh et al. described the real-time implementation of Sobel operator based color image edge detection using FPGA.

In [5] Manoj T H et al. made a Survey and Evaluation of Edge Detection Operators. A combination of different edge detection algorithm is described to extract the text from natural images. Combine Edge Detection method locates the edges better compare to other classical edge detectors when extraction of connected component.

In this paper, the system is implemented on XC5VSX50T chip of Xilinx Virtex-5 family using Xilinx ML506 development board. The implementation is done through the collaboration of MATLAB and EDK.

1.1 SOBEL EDGE DETECTION

The Sobel operator is a classic first order edge detection operator computing an approximation of the gradient of the image intensity function. At each point in the image the result of the Sobel operator is the corresponding norm of this gradient vector. The Sobel operator only considers the two orientations which are 0°and 90°convolution kernels as shown in Fig.(1) and Fig.(2). The Sobel operator has the advantage of simplicity in calculation. But the accuracy is relatively low because it only used two convolution kernels to detect the edge of image [3].

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

Figure (1):Horizontal operator Figure (2): Vertical operator

Equation (1) shows convolution of input image with horizontal mask and Equation (2) shows convolution of image with vertical mask.

$$G_x = \{f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1)\} - \begin{Bmatrix} f(x-1,y-1) + 2f(x-1,y) + \\ f(x-1,y-1) \end{Bmatrix} \quad \dots\dots(1)$$

$$G_y = \{f(x-1,y-1) + 2f(x,y-1) + f(x+1,y-1)\} - \begin{Bmatrix} f(x-1,y+1) + 2f(x,y+1) + \\ f(x+1,y+1) \end{Bmatrix} \quad \dots\dots(2)$$

Size of gradient can be calculated from gradient vectors as shown in (3).

$$|G| = \sqrt{G_x^2 + G_y^2} \quad \text{.....(3)}$$

Gradient can be written as shown below in (4).

$$G(x,y) = |G_x| + |G_y| \quad \text{.....(4)}$$

If Sobel operator is used to detect the edge of image f, then the horizontal template M_x and vertical template M_y shown in Fig. (1) and Fig.(2) must be used to get convolution with image, without taking into account the border conditions. It may get the same size of two gradient Matrix G_x and G_y as the original image. Then the total gradient value G can be obtained by adding the two gradient matrices. Finally, the

edge can get by applying threshold. If G is greater than threshold then pixel should be identified as edge. The flow chart for Sobel operator is shown in Fig. (3).

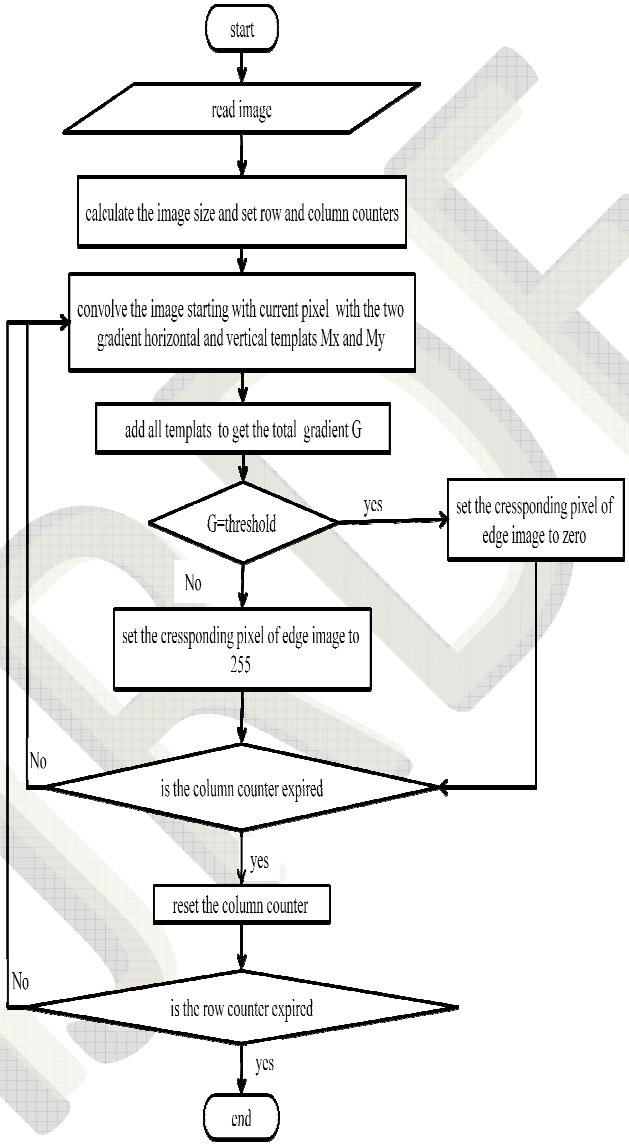


Figure (3) Flow Chart for Sobel Operator

The Sobel operator is used mostly although it is slower than the Roberts cross operator, because its horizontal and vertical kernels smooth the input image and makes operator less sensitive to noise. The reason for using Sobel operator is that it has relatively small masks compare to other operators [6].
In this paper, the hardware implementation of edge detection firstly applied using classical Sobel operator illustrated in Fig.(1) and Fig.(2). Then a modified Sobel operator is applied to suppress noise in the output image. The horizontal mask and vertical mask of the modified Sobel operator are shown in Fig. (4) and Fig. (5).

$$M_X = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad M_Y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Figure (4): Horizontal Modified Mask

Figure (5): Vertical Modified Mask

1.2 EMBEDDED DEVELOPMENT KIT (EDK) AND ML506 BOARD

The Xilinx Embedded Development Kit (EDK) is a suite of tools and IP (Intellectual Property) that enables you to design a complete embedded processor system for implementation in a Xilinx Field Programmable Gate Array (FPGA) device [7]. Microblaze Soft IP core in EDK is used to customize CPU and peripherals reasonably. This platform supports C and C++ programming language. Thus the FPGA's function of process image data can be implemented rapidly [8].
The design kit board comprises one Xilinx Virtex5-SX50T FPGA clocked at 100 MHz, whose main characteristics are: 32640 slices, 132 Blocks RAM of 36 kb (4752 kb), 288 DSP blocks, global bitstream size: 2.5 MByte and 256 MB DDR2 SODIMM.

II. SYSTEM DESIGN AND ARCHITECHTURE

The entire system is implemented on Xilinx Virtex5-XC5VSX50T using Xilinx ML506 development board. The edge detection algorithm is implemented on Xilinx Microblaze Soft core processor running at clock rate of 125 MHz. The XC5VSX50T FPGA chip is selected as the main processor. Other devices are also used as peripherals such as Micron's DDR2 SDRAM memory MT4HTF3264HY, UART(Universal Asynchronous Receiver Transmitter) used for receiving data from MATLAB and Xps-TFT(Thin Film Transistor) controller as a hardware display controller to display the processed image on a monitor through DVI (Digital Visual Interface) port.

System block diagram is illustrated in Fig. (6). The system has combined the MATLAB and EDK to implement the design. In MATLAB environment, the image is read in the form of a data matrix with class type of uint8 and sent from MATLAB to FPGA using fwrite () function provided by MATLAB R2011b.
The image data are sent via serial port with 230400 baud rate, 8 data bits, one Stop bit, and no Parity bit. In EDK environment, the image data are received by a processer system via UART peripheral with the same baud rate used in MATLAB. The received operation is done with the C++ function XUartNs550_RecvByte() to perform data acquisition by the embedded processer system. The data are received after configure the UART to the same parameters as in MATLAB. The image data are stored in Block RAM as a matrix of 128×128 pixels, and then read out the image data from Block RAM for the purpose of applying the Sobel operator to find the edges of the image. The data of the processed image are stored in a buffer created in DDR2 SDRAM. The buffer has the same size of image. Finally, the result of edge detection is displayed on a monitor.

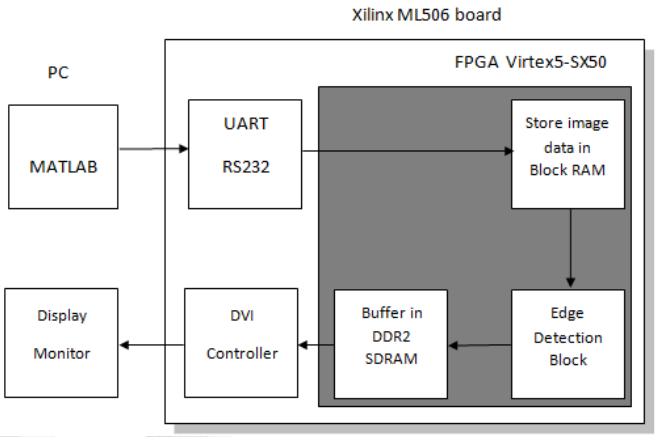


Figure (6): System hardware block diagram

III. SOFT-CORE PROCESSOR SYSTEM

The hardware part of the designed system in EDK is shown in Fig. (7).

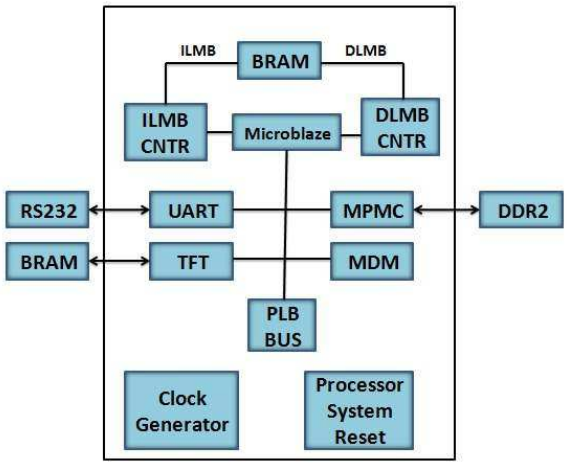


Figure (7) The block diagram of the hardware part

Wizard Screen	System Property	Setting or Command to Use
Create New Project	• Project Name • Project Location • Top-level source type	•myproject.xmp .D:\ myproject .xmp • HDL .
Welcome to the Base System Builder	Project type options	I Would like to create a new design.
Select Board	Board vendor ,name and version	Xilinx,Virtex-5 ML506 Evaluation Platform, Board Revision 1. . Click Next
Select Processor	Processor type	. Microblaze processor . Click Next
Configure MicroBlaze processor	• Clock frequencies • Processor configuration • Local Memory	100 Mhz Processor,125Mhz BRAM 8KB . Click Next
Add Internal		RS232_Uart, DDR2_SDRAM

Peripherals		• Click Next.
Cache window		Don't use any type
System Created	System summary page	Click Finish.

Table (1): The components of the designed system using BSB

The Soft_Core processor can be built using embedded design techniques employing platform studio available in the Xilinx embedded design kit. The Base System Builder (BSB) is an important tool in Xilinx Platform Studio (XPS) to start creating the design system. Table (1) shows the steps for designing system.

After finished creating the system, adding XPS_TFT IP core from IP catalog tab, IO Modules, XPS TFT V2.01a to the project. This core is capable of configuring Chrontel CH-7301 DVI Transmitter chip through I2C interface. Fig. (8) shows the designed embedded system components and their connections. In this system serial RS-232 controller, 256 MB memory size of the DDR2 SDRAM controller, clock generator and XPS-TFT controller IP cores are plugged in the PLB (Processor Local Bus) bus. All the peripheral controllers plugged in the PLB bus are needed to be configured. The ports' properties, the connection between each other and the FPGA's physical ports are configured as external ports to connect the peripherals directly in the XPS 12.1 graphical interface.

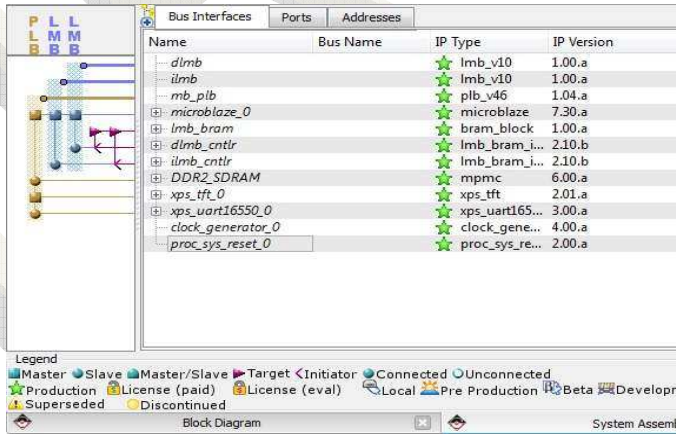


Figure (8) The designed embedded system components and their connections

The block diagram image of the soft-core processor generated in XPS (Xilinx Platform Studio) is shown in Fig. (9).

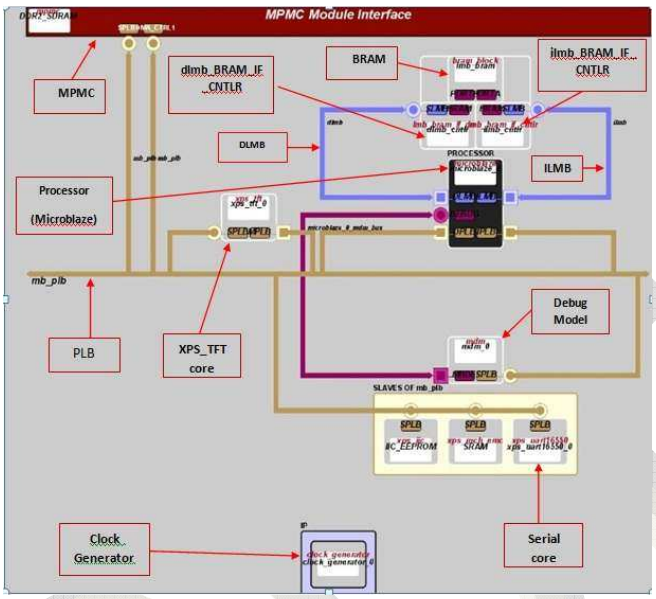


Figure (9) Block Diagram Image

3.1 CONFIGURATION OF IPS USED IN SYSTEM DESIGN

Each IP added in the Soft_Core Processor must be configured with the parameters that used for designing the system. The following figures illustrate the configuration of each IP core. The UART is configured in XPS (Xilinx Platform Studio) as shown in Fig. (10). The UART 16550 is initialized in software (C source file) with the parameters: 230400 Hz baud rate, 8 data bits, one stop bit and no parity bit.

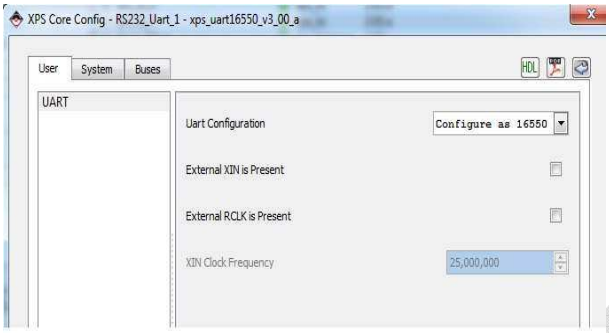


Figure (10) Configuration of UART

While DDR2_SDRAM is configured as shown in Fig. (11). Set the base address of PLB attached video memory to 0x90000000.

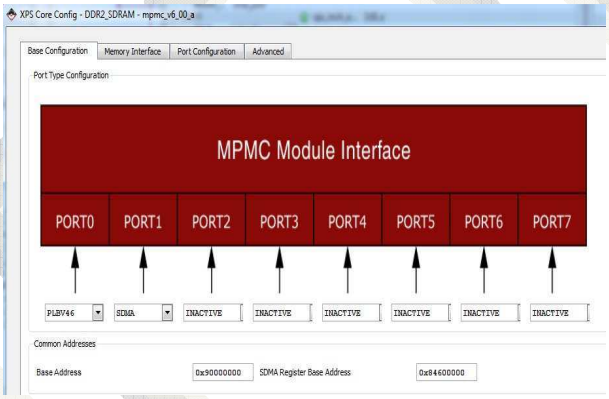


Figure (11) Configuration of DDR2_SDRAM

The XPS_TFT IP also configured as shown in Fig. (12) by set the address of Chrontel DVI to 0x76 and the base address of the attached video memory to 0x90000000.

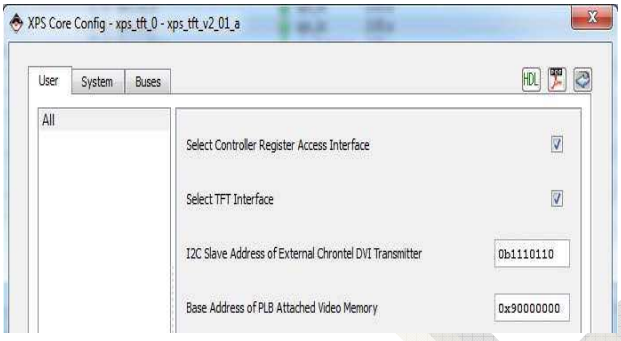
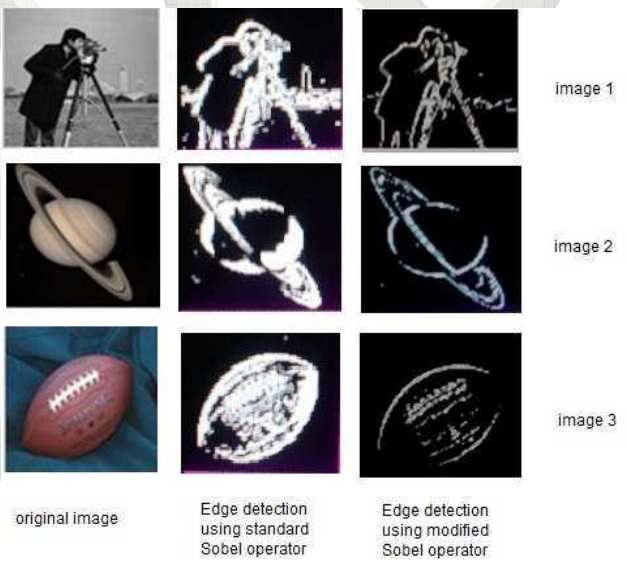


Figure (12) Configuration of XPS_TFT IP Core

The edge detection algorithm is written as C source file with the system project in XPS. After building the software project and generate libraries and bitstream, the bitstream file is downloaded to the Virtex-5 FPGA initializing the UART to receive image data from MATLAB. Then the edge detection algorithm is applied on the received image data and finally displays the result on a monitor.

VI. RESULTS

The experimental results for image edge detection in FPGA using both the standard and modified Sobel operator are shown in Fig. (13). The standard and modified Sobel operators are applied on many images and Fig. (13) shows the results of three samples of images. The result image is shown on a monitor.



(a) (b) (c)

Figure (13) Results for Sobel Edge detection

Fig. (13.a) shows the original image. Fig. (13.b) shows edge detection result appeared on a monitor by applying standard Sobel operator in FPGA .Figure (13.c) shows edge detection result appeared on a monitor by applying by applying modified Sobel operator in FPGA.
The value of threshold used in image1 for edge detection is 70.While in images (2and 3), the value of threshold used is 40.

V. CONCLUSION

In hardware design, it's better to minimize memory size in FPGA. The modified Sobel operator for edge detection gave results better than the results obtained from the standard Sobel operator. When using standard Sobel operator, the results appeared with little noise (distortion edges). While when using the modified Sobel operator, the edges appeared enhanced (sharper edges).

REFERENCES

1. Hong Shan Neoh and Asher Hazanchuk, "Adaptive Edge Detection for Real-Time Video Processing using FPGAs", Copyright © Altera Corporation, 2005.
2. Trucco, Chapt4 AND Jain et al., Chapt5 "Edge Detection" , http://www.cse.unr.edu/~bebis/CS791E/Notes/EdgeDetection.pdf
3. Priyanka S. Chikkali and K. prabuhushetty, "FPGA based Image Edge Detection and Segmentation*", International Journal of Advanced Engineering Science and Technologies*, *9 (2)*, 2011, pages (187– 192).
4. Sanjay Singh, Anil Kumar Saini and Ravi Saini, "Real-time FPGA Based Implementation of Color Image Edge Detection", *International Journal of Image, Graphics and Signal Processing*, Published online in MECS (http://www.mecs-press.org/) ,2012, pages(19-25).
5. Manoj T. H. and A. Santha Rubia ," A Survey and Evaluation of Edge Detection Operators: Application to Text Recognition", *International Journal of Computer Technology & Applications*, *3 (4)*, 2012, pages(1481-1484).
6. Rajesh Mehra and Rupinder Verma , "Area Efficient FPGA Implementation of Sobel Edge Detector for Image Processing Applications", *International Journal of Computer Applications*, *Vol.5(16)* , 2012, pages(7 –11).
7. Xilinx Inc, EDK Concepts, Tools and Techniques, A Hands-On Guide to effective Embedded System Design, UG683 EDK 12.2, http://www.xilinx.com,2010.
8. Wu Jianping, Ye Yutang, Liu Lin, Huang Bingquan and Guo Tao, "High-speed FPGA-based SOPC Application For Currency Sorting System", *The Tenth International Conference on Electronic Measurement & Instruments,* 2011, pages (85-89).