Implementation of Edge Detection Algorithm Using FPGA

Harshada Shimpi¹, Nisha Gaikwad², Meghana Dhage³, Prof. A.S. Pawar⁴
UG Student, Dept. of E&TC Engineering, PCCOE, Pune, Maharashtra, India ¹,²,³
A.P. Dept. of E&TC Engineering, PCCOE, Pune, Maharashtra, India ⁴

ABSTRACT: As in today’s generation processing of images plays a very important role. So this paper describes the implementation of edge detection algorithm. The algorithm used is SOBEL. It is used in various applications as medical image processing, object detection, security etc. The main aim behind this is to process the image and use it in various applications using FPGA. After reading the pixels of an image the algorithm is applied in VERILOG. After processing the image on FPGA the edge detected image is displayed on monitor. The FPGA used is ‘SPARTAN-6’. The entire simulation of the above process is done VERILOG using ‘XILINX-14.1’. And to display input and output image MATLAB is used.

KEYWORDS: FPGA, Verilog, Spartan-6, Edge detection, Sobel operator.

I. INTRODUCTION

Image processing is widely used in many applications. It is having many advantages. The heart of all these image processing applications is the edge detection. Edge detection can be used in various medical applications such as tumour detection in human body. So using only edge detection algorithm the location of the tumour is detected. Also it can be used in security purpose, object recognition, object tracking, face identification etc.

This paper shows the results of SOBEL OPERATOR. The parallel processing capability of FPGA makes an advantage of using it for image processing. The main concept is to apply the fixed mask values of SOBEL OPERATOR on image. This reduces the complexity of algorithm. FPGA is a reconfigurable device and because of use of such devices the time to market cost reduces. Also it becomes easy for verification and debugging.

II. LITERATURE SURVEY

During literature we came across different papers. We referred the paper ‘Implementation of Edge Detection using FPGA and Model Based Approach’, ICICES-2014 Paper by Prof. S.S Pujari, Miss. Sofia Nayak. In this paper they have described the Simulink models for SOBEL and PREWITT Operators for edge detection. From that paper we understood the design flow using the Simulink module. Then we referred ‘DIGITAL IMAGE PROCESSING’ book by Gonzalez and Woods and we studied three gradient based operators i.e. ROBERT, PREWITT and SOBEL. We implemented these algorithms into MATLAB. By comparing the results obtain from MATLAB we decided to implement SOBEL OPERATOR as it gives sharper edges and minute details also. In paper [9] it represent how to implement an image processing algorithm applicable to Edge Detection system in a Xilinx FPGA using System Generator for still images. The design of edge detection is explained in Simulink environment. We referred paper [8] in this paper we have got an idea about execution of the algorithm on hardware. We got an idea about converting the image into text format reduces the complexity as VERILOG cannot support the standard image formats.
III. BLOCK DIAGRAM

The block diagram of the Edge Detection system is as shown in figure [1]. It gives the idea of the overall functioning of the system. There are three main blocks of the system: Input Image, FPGA Board and Monitor. The fig.[2] shows the design flow for hardware implementation of the system. Firstly the image is taken from the computer. Then the image is given to the pre-processing unit. The pre-processing unit is shown in fig. [3]. In this unit the image is converted into 128*128 size image. Then the RGB image is converted into the grey scale image. As the VERILOG cannot handle the standard image formats the grey level values of the intensities are converted into binary value and it is written on the text file and the text file is given to the Edge detection module which is coded in VERILOG. The pre-processing and post-processing on the image are done in MATLAB. After that the entire code is processed on FPGA Board. The processed data is then given to the monitor using UART protocol. And the resultant edge detected image is observed on monitor. The block diagram of the system as follows:

![Image](image_url)

**Fig. 1 Block Diagram of the system.**

![Image](image_url)

**Fig. 2 Design Flow for hardware implementation**

![Image](image_url)

**Fig. 3 Image pre-processing unit**

IV. FLOW CHART

The Flow Chart of the system is shown in below figure. It shows the stepwise working of the algorithm.

FLOW CHART DESCRIPTION: Initially after START the image is read i.e. its pixel values are read. Then that image is convolved with the filter. After the horizontal and vertical mask of the operator are convolved with the original image. Let the horizontal and vertical convolution be Gx and Gy respectively. Then one threshold value is set as say T. Calculate gradient G. It is calculated as \( \sqrt{(Gx^2)+(Gy^2)} \). Then consider first pixel say ‘M’. For M if G is greater than T then consider the next neighbouring pixel and continue the procedure. When G is less than T mark that pixel as it locates the edge. The mask values for Gx and Gy of SOBEL OPERATOR are:

\[
Gx = \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 \\
\end{bmatrix} \quad \text{Gy} = \begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1 \\
\end{bmatrix}
\]
Flow chart for the system:

START

Read the image and convolve with filter.

Convolve the resultant image with chosen Operator’s gradient mask in i axis

Convolve the resultant image with chosen Operator’s gradient mask in j axis.

Set a threshold value, T.

For a pixel say M (i, j).

Compute the gradient Magnitude say G.

IS G>T

Mark pixel as an “edge”.

END

Consider the next Neighbor pixel.
V. RESULTS

1) The comparison of different operators: The fig. [4] shows the comparison of different operators. Image in the first column shows the original image which we want to process. Here the image is grey scale image. The first row shows the result of Robert Operator, second row shows the result of Sobel operator and third row shows the result of Prewitt operator. We can conclude from the below result that the ‘Sobel operator’ gives more sharp results.

2) The Square calculation of one pixel i.e. Gx^2: The fig.[5] shows the calculation of square of gradient in x-direction. It shows the square of one pixel. For the calculation of resultant magnitude we require the square of the gradient in x-direction and in y-direction. Here the ‘gx1t = 84681’ is the value of the gradient.

3) Resultant gradient calculation: The fig.[6] shows the resultant gradient calculation. Resultant gradient is calculated as: sqrt((Gx^2)+(Gy^2)). Here the ‘root_val’ parameter corresponds to the resultant gradient of each pixel. It shows that the simulation of Sobel operator in Xilinx giving accurate results.
CONCLUSION

We have successfully compared all the operators in MATLAB. And by implementing this system we conclude that the SOBEL OPERATOR gives more accurate results. The hardware used is Spartan-6(XC6SLX16). We have calculated the gradient in VERILOG. We also conclude that the testing and changing of the parameters became easy. One can easily change the different mask values observe the changes.

REFERENCES


[8] FPGA IMPLEMENTATION of SOBEL EDGE DETECTOR under V. KamatchiSundari& M. Manikandan, P. Prakash in International Journal of Advances in Science and Technology (IJAST)