Edge Detection in VHDL

[1]M.Rathod

PG Student, ME Communication System,
L.J. Institute of Engineering and Technology.
Mayur.74625@gmail.com

Abstract- **Edge is a basic feature of an image. edge is a collection of the pixels it refers to the part where the brightness of the image local area changes significantly. Real time edge detection is required in many embedded system where execution speed is critical. The VHDL is an appropriate Hardware Description Language (HDL) for providing hardware models of practical image processing algorithms. This paper present study on various edge detectors. Introduction and Comparison between different edge detection algorithms is presented. In this project my Aim is to implement edge detection algorithm on FPGA. The FPGA technology has the advantage of high-performance for digital image processing and low cost. The FPGA provides the necessary hardware for image processing algorithms with flexibility to support edge detection algorithm. HDL Coder is used for MATLAB to vhdl conversion. Simulation result and also edge detected result for sobel and prewitt algorithm using HDL Coder and hand code is presented..**

Keywords-Gradient Operator, Edge detection, FPGA, Matlab,HDL coder

I. INTRODUCTION

Edge detection is an essential preprocessing step in image processing. In digital image processing, the term edge is a collection of the pixels it refers to the part where the brightness of the image local area changes significantly. An edge detector is basically a high pass filter that can be applied to extract the edge points in an image. The goal is to mark the points in an image at which the luminance changes abruptly. Digital Image Processing proposed the general methods of edge detection are First Order Derivative (gradient method), Second-Order Derivative and Optimal Edge Detection to detect the edge. A lot of edge detection algorithms, such as Robert detector, Prewitt detector, Kirsch detector, Gauss-Laplace detector and Canny detector have been proposed[1][4][5][15]. Among these algorithms, Canny algorithm has been used widely in the field of image processing because of its good performance. The Canny edge detector is predominantly used in many real-world applications due to its ability to extract significant edges with good detection and good localization performance [5]. Unfortunately, the Canny edge detection algorithm contains extensive pre-processing and post-processing steps and is more computationally complex than other edge detection algorithms. Software implementation of edge detection algorithm is easy but not good in terms of speed and memory requirement. Implementing image processing algorithms on reconfigurable hardware minimizes the time-to-market cost, enables rapid prototyping of complex algorithms and simplifies debugging and verification [6] [13]. There are two types of technologies available for hardware design. Full custom hardware design also called as Application Specific Integrated Circuits (ASIC) and semi custom hardware device, which are programmable devices like Digital signal processors (DSP‟s) and Field Programmable Gate Arrays (FPGA‟s). Full custom ASIC design offers highest performance, but the complexity and the cost associated with the design is very high. The ASIC design cannot be changed; time taken to design the hardware is also very high. Hardware design techniques such as parallelism and pipelining techniques can be developed on a FPGA,not possible in dedicated DSP designs [12]. So FPGAs are ideal choice for implementation of real time image processing algorithms.

Variable Involved In The Selection Of An Edge Detection Operator [12]

1. Edge Orientation: The geometry of the operator determines a characteristics direction in which it is most sensitive to edges.
2. Noise Environment: Edge detection is difficult in noisy images, since both the noise and edges contains high frequency content.
3. Edge Structure: Not all edge involves a step change in intensity. Gradual change is also there in images.

II. METHODS

1. Robert algorithm: 2 * 2 gradient operator
2. Sobel algorithm: 3 * 3 gradient operator
3. Prewitt algorithm: 3 * 3 gradient operator
4. Canny algorithm: non maxima suppression
5. Zero crossing: Laplacian of Gaussian operator

There is various type of operator available for edge detection. In first order derivative the input image is convolved by an adapted mask to generate a gradient In which edge are detected by thresholding.

Mask for various edge detection operator is given below.

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

(1)

-1	0	+1
-1	0	+1
-1	0	+1

Gx

+1	+1	+1
0	0	0
-1	-1	-1

Gy

(2)

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

Gx

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

Gy

(3)

Figure 1: Kernel Matrices for Robert, Prewitt and Sobel algorithms.

Strength of edge at each pixel can be calculated by Magnitude calculation which can be given by equation.

$$|G| = |Gx| + |Gy|$$

Direction of edge is given by finding angle by eq.

$$\theta = \tan\text{-}1\ (Gy / Gx)$$

Canny algorithm: it is the optimal edge detector because of low error rate. It is the algorithm based on non-maxima suppression.

III. CANNY ALGORITHM

The Canny edge detection algorithm is known to many as the optimal edge detector. The first and most obvious is low error rate. It is the important that edges occurring in images should not be missed and that there are no responses to non-edges. The second criterion is that the edge points be well localized. A third criterion is to have only one response to a single edge.

Figure for canny edge detection is shown. Steps for canny edge detection and details of each step are also presented [5].

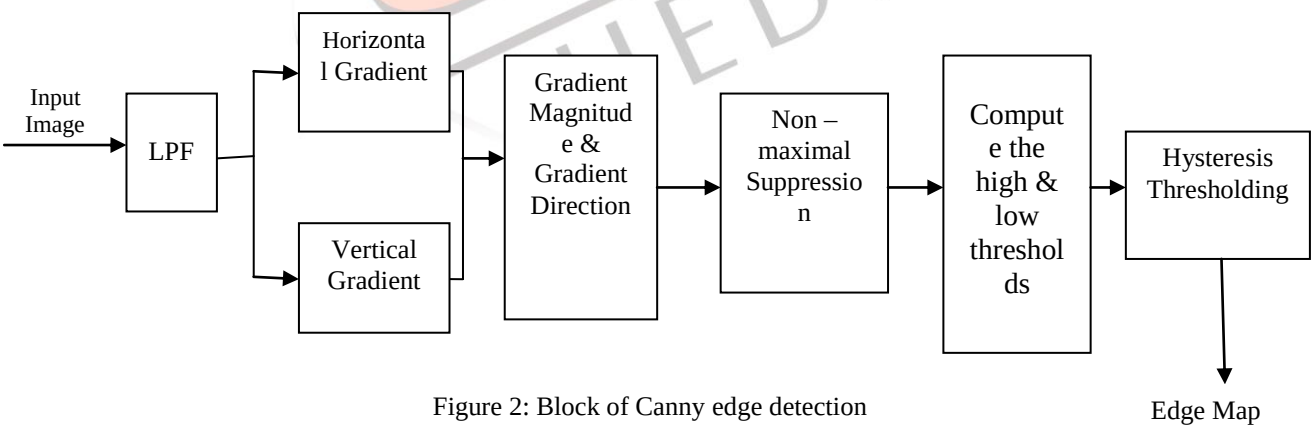

Figure 2: Block of Canny edge detection

The canny edge detector is carried out in following four steps:

1. Image smoothing
2. Gradient calculation and directional Non-Maximum suppression
3. calculating thresholds
4. Thresholding with hysteresis

1. Smooth the image with a 2 dimensional Gaussian. In most cases the computation of a 2 dimensional Gaussian is expensive, therefore it's approximated by 2 one dimensional Gaussians, one within the x direction and also the alternative within the y direction.

2. Take the gradient of the image. This shows changes in intensity, that indicates the presence of edges. This truly provides 2 results, the gradient within the x direction and also the gradient within the y direction.
3. Non-maximal suppression. Edges can occur at points the wherever the gradient is at a most. Therefore, all points not at a most should be suppressed. So as to try and do this, the magnitude and direction of the gradient is computed at every picture element. Then for every picture element check if the magnitude of the gradient is larger at one pixel's distance away in either the positive or the negative direction perpendicular to the gradient. If the picture element isn't larger than each, suppress it.
4. Edge Thresholding. The strategy of thresholding utilized by the cagey Edge Detector is referred to as "hysteresis". It makes use of each a high threshold and an occasional threshold. If a picture element includes a price on top of the high threshold, it's set as a position picture element. If a picture element includes a price on top of the low threshold and is that the neighbour of a position picture element, it's set as a position picture element further. If a picture element includes a price on top of the low threshold however isn't the neighbour of a position picture element, it's not set as a position picture element. If a picture element includes a price below the low threshold, it's ne'er set as a position picture element.

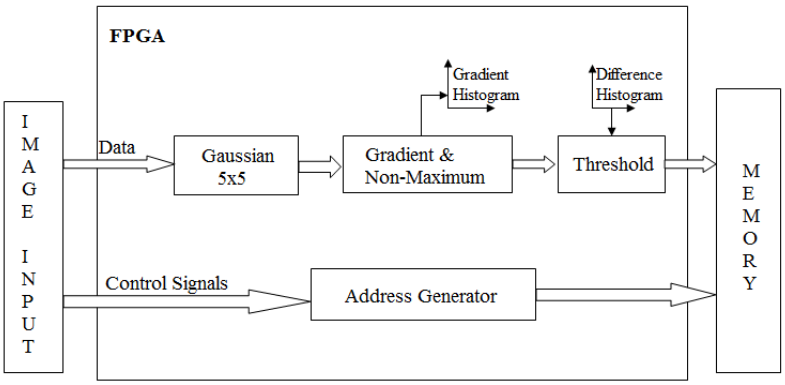

Figure 3: Design flow of Canny edge detection[15]

Comparative study of various edge detection algorithm is given in table.

OPERATOR	ADVANTAGE	DISADVANTAGE
Sobel, prewitt, robbert	Simplicity, Detection of edges and their orientations.	Sensitivity to noise, Inaccurate
canny	Using probability for finding error rate, Localization and response. Improving signal to noise ratio, Better detection specially in noise conditions.	Complex Computations, False zero crossing, Time Consuming.
Laplacian of Gaussian	Finding the correct places of edges, Testing wider area around the pixel.	Sensitivity to noise, Not finding the orientation of edge because of using the Laplacian filter.

Table I: Comparison of algorithm.

IV. 4. PROPOSED MODEL

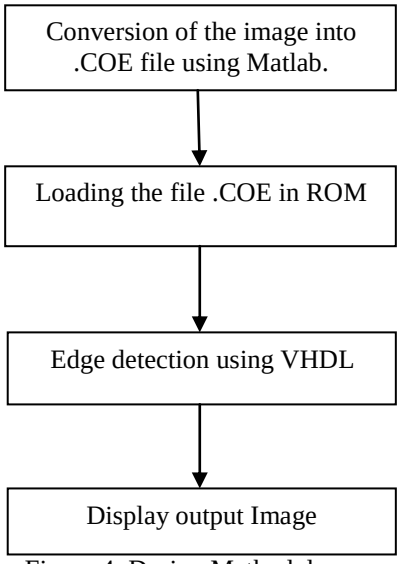

Figure 4: Design Methodology.

The proposed work is represented as follows:

- Converting the image into a vector in a text file extension *.Coe using the tool MATLAB.
- Loading the file *. Coe in Rom .
- Detecting image contours with different operators of gradient Sobel, Robert, Prewitt in VHDL.
- Display output image on LCD by VGA interface.

In the proposed work i have designed sobel and prewitt algorithm in MATLAB and then converted in vhdl code by using HDL Coder. Simulation result for sobel and prewitt algorithm is presented. Simulation is done in modelsim 6.4.result for hand coded code of sobel and prewitt is also presented. In that we created .do file in MATLAB which is input in vhdl program of sobel and prewitt. Convolution and other process is done in vhdl and output text file is generated withi edge information which is again shown in MATLAB.

V. RESULT

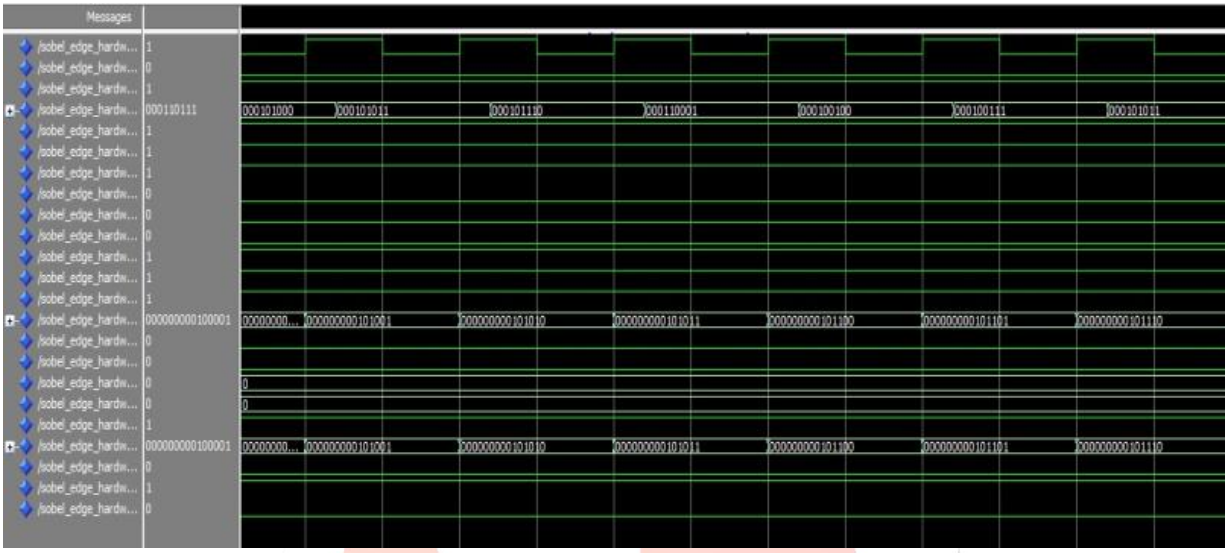

Figure 5:simulation result of sobel

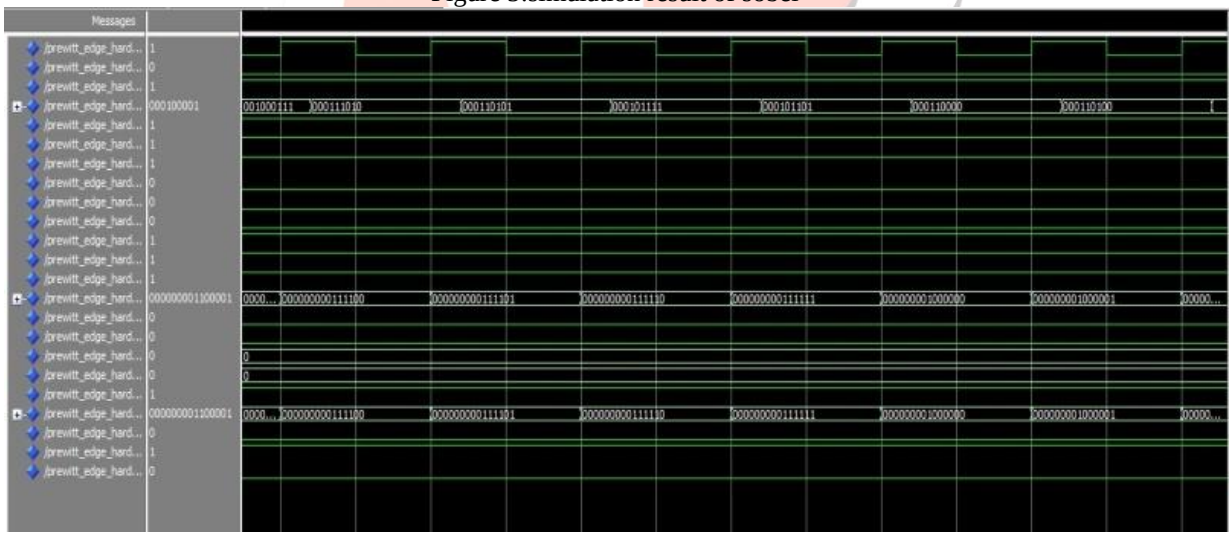

Figure 6:simulation result of prewitt

Figure 7:original image

Figure 8:sobel hdl coder result

Figure 9:prewitt hdl coder result

Figure 10:original image

Figure 11:sobel hand doded result

Figure 12:prewitt hand doded result

VI. CONCLUSIONS

In this paper comparative study of edge detection algorithm and detail of canny algorithm is presented. Vhdl structure of sobel and prewitt is generated by using MATLAB HDL Coder simulation result, edge detected result and hand coded result is presented. Hand coded result shows good result compared to HDL coder result.

REFERENCE

[1] Yarsi and N.H.Hamid, "Performance analysis of FPGA based Sobel edge detection operator", IEEE, 2008.

[2] Dr.D.Selvathi and J.Dharni, "Realization of Beamlet Transform edge detection algorithm using FPGA", IEEE, 2013.

[3] Erke Shang and Jian Li, "Lane Detection using steerable filters and FPGA based implementation", IEEE, 2011.

[4] Rajesh Mehra and Rupinder Verma, "Area efficient FPGA implementation of sobel edge detector for image processing application", international journal of computer application, volume 56-No.16, 2012.

[5] Tejaswini H and Vidhya N, "An implementation of real time optimal edge detector and VLSI architecture", international conference on electronics and communication engineering, 2013.

[6] Iulian Nita and Vasile Lazarescu, "Multiprocessor real time edge detection using FPGA IP core", IEEE, 2011.

[7] Rajesh Mehra , Rachna B. Sardana and Rupinder Verma, "FPGA based area efficient edge detection filter for image processing application", international journal of VLSI and Signal processing application, Vol.1, Issue 2, 2011.

[8] Varun Sanduja and Rajeev Patil, "sobel edge detection using parallel architecture based on FPGA", International journal of Applied information system (IJAIS), New York, Volume 3-No.4, 2012.

[9] Christos Kyrkou, Christos Ttofis and Theocharis Therocharides, "FPGA Accelerated Object Detection using edge information", IEEE, 2011.

[10] T.Mahalakshmi and R.Muthaiah, "VLSI Implementation of Edge Detection for Images, Research Journal of Applied Science", Engineering and Technology 4(24):5474-5479, 2012.

[11] Sudeep K C and Dr. Jharna Majumdar, "A Novel Architecture for Real Time Implementation of Edge detector on FPGA", IJCSI, Vol 8, Issue 1, 2011.

[12] L.Padma Latha, "Design of edge detection Technique using FPGA and DSP", VSRD international journal of Electrical, Electronics & Communication Engineering, Vol 2(6), 2012.

[13] Arash Nosrat and Yousef S.Kavin, "Hardware Description of Multi-Directional Fast Sobel Edge detection Processor by VHDL for implementing on FPGA", International Journal of Computer Applications, Volume 47-No.25, 2012.

[14] B.Muralikrishna, P.Sujitha and Habibulla Khan, "Real time edge detection modeling with FPGA", International Journal of VLSI and Embedded System-IJVES, Vol 4, Article 06103 , 2013.

[15] R.Ponnela Vignes and R.Rajendran, "Performance and Analysis of Edge detection using FPGA Implementation", International Journal of Modern Engineering Research (IJMER), Vol. 2, Issue 2, 2012.

[16] Ms.Priyanka S. Chikkali and K. Prabhushetty, "Fpga based Image edge detection and segmentation", International Journal of Advance Engineering Science and Technologies, Vol No. 9, Issue No. 2, 2011.