Implementation of Canny Edge Detection Algorithm on FPGA and displaying Image through VGA Interface

Azra Tabassum[1], Harshitha P[2], Sunitha R[3]

[1-2] 8th sem Student, Dept of ECE, RRCE , Bangalore, Karnataka, India

[3] Associate Professor, Dept of ECE, RRCE , Bangalore, Karnataka, India

ABSTRACT

The edge detection is one of the key techniques in most image processing applications. The canny edge detection is proven to be able to significantly outperform existing edge detection techniques due to its superior performance. Unfortunately, the implementation of the systems in real-time is computationally complex, high hardware cost with increased latency. The proposed canny edge detection algorithm uses approximation methods to replace the complex operations; the pipelining is employed to reduce the latency.In this paper we present canny edge detection algorithm implemented on Spartan 3E FPGA and developed VGA interfacing for displaying images on the screen. We have taken 128×128 Image and displayed same on the monitor through FPGA.

Keyword: - *Pipelining, Canny Edge Detector, FPGA , Block Memory, VGA Interface.*

. 1. INTRODUCTION

In computer vision, machine vision and many imaging applications, edge detection plays a vital role as a preprocessing step. Edge defines the object boundaries within the image and occurs when discontinuities present in the pixel values. The implementation of an edge detection system in hardware is a challenging task because of the presence of interference in an image due to noise, various lighting conditions, high processing speed, and high accuracy. These challenges make edge detection becomes unreliable and more difficult.

Many researchers have proposed diverse techniques for edge detection [1]-[5]. There are various edge detection techniques, such as Robert detector [1], Prewitt detector [2], Sobel detector [3], Laplace of Gaussian detector [4] and canny edge detector [5]. Among several edge detection techniques. canny based edge detection provides better performance by addressing the limitations of all other edge detectors and thereby achieving greater improvement in terms of sharp edges and noise immunity [6]. Edges define the boundaries between regions in an image, which helps with segmentation and object recognition. They can show where shadows fall in an image or any other distinct change in the intensity of an image [2]. However, the canny edge detection algorithm is computationally complicated and high latency for real-time applications. For real-time applications, high-speed canny edge detection is necessary. So far, several papers have dealt with FPGA based canny edge detection algorithms for real-time applications [7-9].

2. RELATED WORK

In this paper, it mainly concentrate on edge detection. The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.In this paper algorithm have been developed to identify edges . The recent studies on Canny edge detection algorithm shows that the traditional Canny edge detector has two shortcomings. The threshold of the algorithm needs to be set by manual. Secondly, the algorithm is very time consuming and cannot be implemented in real time. A new self-adapt threshold Canny algorithm is proposed and a pipelined implementation on FPGA is designed to overcome the above disadvantages.

Compared with the implementation in a PC based system, pipelined implementation on FPGA takes much less implementation time and can therefore be used for the mobile robot vision system which is very strict for the real-time performance of its vision system.

3. PROPOSED BLOCK DIAGRAM

The block diagram of the project is as shown in the Figure 3.1. The input image is gray scale image of size 128x128 each pixel of 8 bits. The Coefficient (coe) file of the input image is generated using MATLAB. This coe file is stored in Block ROM which is processed using the canny edge detection algorithm. The resulting image is displayed on monitor using VGA interfacing.

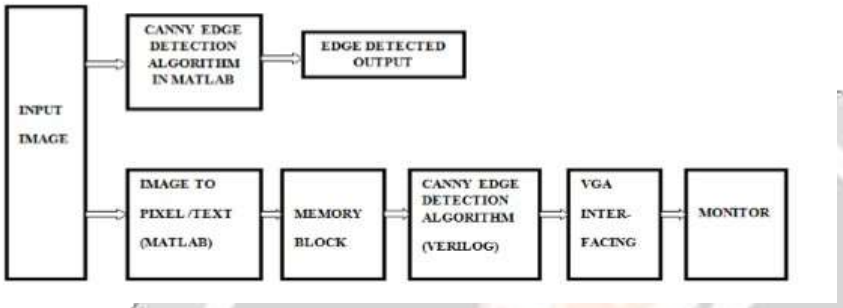

Fig-1: Block Diagram of the Proposed Work

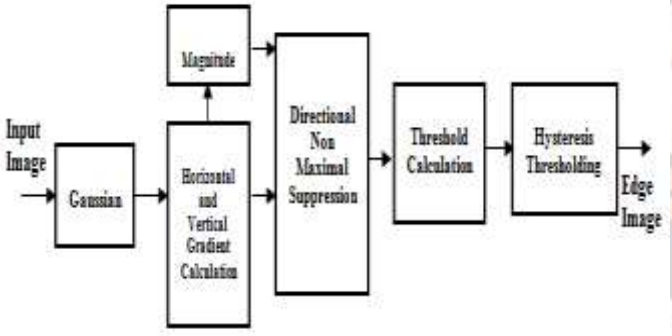

Fig -2: Block diagram of canny edge detection algorithm

The block diagram of Canny edge detection algorithm is shown in figure.
It is divided into following modules

i. Guassian smoothing
ii. Gradient and Magnitude calculation unit
iii. Directional Non-Maximal suppression unit
iv. Threshold calculation
v. Hysteresis thresholding

Step 1 :Gaussian Smoothing

In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods.. Image smoothing is the first stage of the canny edge detection. The pixel values of the input image are convolved with predefined operators

to create an intermediate image. This process is used to reduce the noise within an image or to produce a less pixilated image. Image smoothing is performed by convolving the input image with a Gaussian filter.

Step2 :Gradient Calculation

After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. The operator performs a 2-D spatial gradient measurement on an image. In this stage, the blurred image obtained from the image smoothing stage is convolved with a 3x3 operator. The operator is a discrete differential operator that generates a gradient image .The operators used to calculate the horizontal and vertical gradients. To obtain the gradient image, a smoothened image from the first stage is convolved with the horizontal and vertical operators.

Step3: Gradient Magnitude and Direction

Gx and Gy are images with pixel values that are the magnitude of the horizontal and vertical gradient, respectively. The magnitude, or edge strength, of the gradient is then approximated using the formula, The direction of the edge is computed using the gradient in the x and y directions. Edge direction is defined as the direction of the tangent to the contour that the edge defines in 2- dimensions. The edge direction of each pixel in an edge direction image is determined using the arctangent.

Step4 :Non Maximum Suppression

The edge strength for each pixel in an image obtained from equation is used in non maximum suppression stage. The edge directions obtained from equation are rounded off to one of four angles 0 degree, 45 degree, 90 degree or 135 degree before using it in non-maximum suppression. After the edge directions are known, non maximum suppression now has to be applied. Non maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image. Non-maximum suppression (NMS) is used normally in edge detection algorithms. It is a process in which all pixels whose edge strength is not maximal are marked as zero within a certain local neighborhood. This local neighborhood can be a linear window at different directions of length 5 pixels.

Step5: Thresholding and Hysterisis

Thresholding with hysteresis is the last stage in canny edge detection, which is used to eliminate spurious points and non-edge pixels from the results of non-maximum suppression. The input image for thresholding with hysteresis has gone through Image smoothing, calculating edge strength and edge pixel, and the Non-maximum suppression stage to obtain thin edges in the image. Results of this stage should give us only the valid edges in the given image, which is performed by using the two threshold values,

4. EXPERIMENTAL RESULTS

The design of the canny edge detection algorithm is done in verilog. Design and testing of individual module has been carried out. The simulation result of window 3×3 unit is shown in figure-1. The inputs to this block are the pixel values of an image.

Fig-3: simulation result of 3x3 window generator.

Fig-4: Simulation result of gradient calculation unit

The simulation result of gradient unit is shown in figure-3. The input to this block is from smoothing unit. These values are convolved with horizontal and vertical convolution masks.

The simulation result for Magnitude calculation block is shown in Figure -4. The input to this unit is from gradient unit. Gradient is calculated at each point both in horizontal and vertical direction these values are taken as input to the magnitude unit.

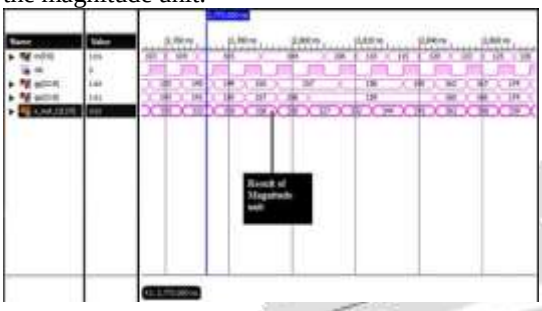

Fig-5: Simulation result of magnitude unit

Fig -6: simulation result of NMS block.

The simulation result for Magnitude calculation block is shown in Figure-5. The input to this block is from gradient unit and also from magnitude unit. The obtained angle plays major role in the selection of magnitude from the magnitude unit. The simulation result is shown below.

The simulation result for Magnitude calculation block is shown in Figure-6. The output from NMS block is fed as input to this unit and is compared with high and low thresholds to get true and strong edges. The final result obtained after integration of entire block is as shown in below diagram.

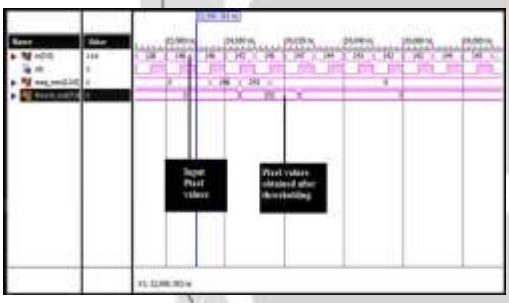

Fig-7: Simulation result of thresholding unit

Different input images and their edge detected images are shown in Figure-7. The Figures A, C are the input images and Figures B, D are their respective edge detected results.

Fig-8: Simulation result

Initially we loaded an image of size 128×128 in block memory. Canny edge detection algorithm has been implemented on FPGA and then VGA interfacing is developed. The interfacing of Spartan 3E FPGA with VGA monitor for displaying obtained edge detected image is shown in figure-9.

Fig-9: Edge detected image after VGA interface

5. CONCLUSION

The proposed canny edge detector uses the pipelining technique and handles the multiple blocks at the same time. Ithelps to achieve fastest working speed. In this paper we implemented Canny Edge Detection algorithm on Spartan 3E FPGA. VGA interfacing is developed for displaying the images on the monitor. An image of size 128×128 is first stored in block Rom on FPGA and then processed through Canny edge detection algorithm and displayed on VGA monitor. The entire system is developed simulated and synthesized using Spartan 3E FPGA board. In future videos can be stored in the memory and video edge detection can be performed camera interfacing can be done to take real time images and the system can be used for security purposes.

6. REFERENCES

[1] R. Ponneela Vignesh, R. Rajendran, "Performance and Analysis of Edge Detection Using FPGA Implementation". International Journal of Modern Engineering Research (IJMER) Vol.2, Issue.2, Mar-Apr 2012 pp-552-554 ISSN: 2249-6645.

[2] Chandrashekar N.S, Dr. K.R. Nataraj, "A Distributed Canny Edge Detection and Its Implementation on FPGA" InternationalJournal of Computational Engineering Research (ijceronline.com) Vol. 2 Issue.7. Issn 2250-3005(online) November 2012.

[3] Tejaswini H.R, Vidhya N, Swathi R Varma,Santhosh B, "An Implementation of Real Time Optimal Edge Detection and VLSI Architecture". International conference on electronics and communication engineering, 28th april-2013, bengaluru, isbn: 978-93- 83060-04-7.

[4] Samina Jafar, Anupsingh Ramprakashsingh Rajput, "Improved Distributed Canny Edge Detection In VHDL". VSRD International Journal of Electrical, Electronics & Communication Engineering, Vol. 3 No. 6 June 2013.

[5] T. Rupalatha, Mr. C. Leelamohan, Mrs. M. Sreelakshmi, "Implementation of Distributed Canny Edge Detection On FPGA". International Journal of Innovative Research in Science, Engineering and Technology, Vol. 2, Issue7, July 2013.

[6] Divya. D, Sushma P. S, "FPGA Implementation of a Distributed Canny Edge Detection". International Journal of Advanced Computational Engineering and Networking, ISSN: 2320-2106 Volume- 1, Issue- 5.

[7] Mallavarapu. Ramu, T. V. S. Adinarayana, "A Hardware/Software Co-Design Architecture Implementation of Canny Edge Detection Using FPGA and MATLAB". International Journal of software& hardware research in engineering. [8] Raman Maini, Dr. Himanshu Aggarwal "Study and Comparison of Various Image Edge Detection Techniques" IJIP, Volume (3): Issue (1).

[8] Xiaoyang Li,Jie Jiang, Qiaoyun Fan "An Improved Real-time Hardware Architecture for Canny Edge Detection Based on FPGA" Third International Conference on Intelligent Control and Information Processing July 15-17, 2012 - Dalian, China.

[9] Chandrashekar N.S., Dr. K.R. Nataraj "Design and Implementation Of A Modified Canny Edge Detector Based On FPGA" International Journal of Advanced Electrical and Electronics Engineering, (IJAEEE), ISSN (Print): 2278-8948, Volume-2, Issue-1, 2013