International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

RTL Design and FPGA Implementation of Canny Edge Detector with Real Time Threshold Adjustment Capability

Lakshmamma K M[1], Chandana B.R[2]

[1]Department Electronics and Communication Engineering (M. Tech VLS)

[2]Guide/Assistant Professor, Department ECE, PA College of Engineering, Mangalore, India

Abstract: ***The Canny edge detector is an edge detection operator detect a wide range of edges in images. Canny edge detector is read both column and rows pixels in images this is extra feature we are added in our article. Canny Edge detector is very fast and easy detector compared other detector like sobel detector. Edge detection is a very important area in the field of Computer Vision. Edges define the boundaries between regions in an image, which helps with segmentation and object recognition. Compared with the implementation in a PC based system, pipelined implementation on FPGA is easy way and it takes much less implementation time and also more efficiency. Then also we are improving Thresholding.***

Keywords: Non maxima suppression(NMAx),Vector sobel operator,Realtime logic(RTL),Field Programmable Gate Array(FPGA).PC(personal computer),DSP(Digital Signal Processing).

1. Introduction

Edge detection is the basic operation in image processing and has wide application in research area. Many edge detection algorithms have been proposed such as Robert detector, Prewitt detector, Kirsch detector, Gauss-Laplace detector and Canny edge detector. Due to its good performance Canny algorithm has been widely used in the field of image processing . Edge detection is a very important area in the field of Computer Vision. 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. 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 fo**r** the mobile robot vision system which is very strict for the real-time performance of its vision system. Generally image processing algorithms are implemented on DSP kits. Image processing algorithms are repetitive in nature and require more computation. The alternative choice is to implement algorithm on the very expensive application specific integrated circuits (ASIC) or Field Programmable Gate Array (FPGA). Since FPGAs offer the features like reprogram ability flexibility parallelism short development time and computational power these FPGAs are best suited to implement image processing algorithms.

2. Methodology

1. Further literature survey has to carried out to extract more ideas
2. A design plan should be created with an application
3. RTL Design of the proposed design should made
4. RTL verification should be done to make sure the design is working as decided
5. Visual validation should be done using MATLAB
6. First level FPGA implementation should be done. This will complete one full cycle.
7. Programmable threshold should be tested.
8. Next, speed, area, power and other parameters should be observed and tabulated.
9. Comparison of the parameters should be done
10. A demonstration should be setup to prove how the edge detector helpful. This can be done using the application decided.
11. Thesis/dissertation should be written.

3. System Design

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.

The block diagram is divided into following modules:
1) cache system and thresholding
2) Gradient and Magnitude calculation unit
3) 3x3 cache for both H and V
4) Directional Non-Maximal suppression unit

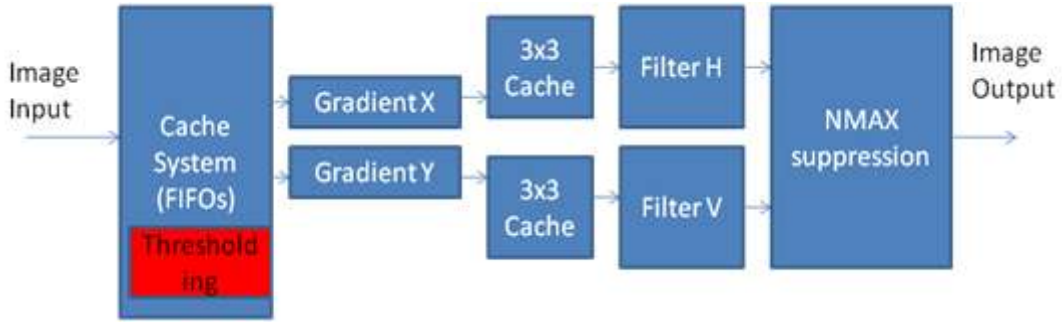

Figure 3.1:.Block diagram of canny edge detector

i. Cache system and Thresholding

The image is first passed through low pass filter (LPF) to reduce the noise, i.e. all high frequency components are eliminated. A 3x3 moving window Gaussian operator is used two FIFO buffers with the depth of one row pixels of image are employed to access all the pixels in the 3x3 window at the same time.Consider a two dimensional image of size 128x128 as input and it is represented as I(x,y) in spatial domain. The pixel values are obtained as array of values and these values are stored in a text file. This text file data is used as input during the smoothing process.

ii .Gradient and magnitude calculation unit

The horizontal and vertical gradient values obtained in the gradient unit are the inputs to the magnitude block; these gradient values are used for finding the magnitude of the input image. Magnitude of the gradient is calculated using the formula Gmag = $\sqrt{Gx^2 + Gy^2}$.

iii.3x3 cache for both H and V

Cache system stored image in 3x3 format and also stored both horizantle and vertical format.Then filter any glitches in cache using filter H and V.

iv.Directional Non-Maximal suppression unit

The architecture of NMS block is shown in figure 3.4. It consists of window 3x3 unit, selector unit, arithmetic and comparator unit. Window 3×3 unit holds the 8 pixel values at a time and feeds the input to the selector unit and the middle value is given to the comparator unit.In the selector block angle is calculated at each pixel location simultaneously after the calculation of gradient of an image. Edge direction is taken by considering the inverse tangent function of vertical gradient to horizontal gradient. Θ = arctan (Gy/Gx). The angle Θg(x, y) is calculated using the gradients i.e. arc tan (gy/gx).finally we got edge image output.

4. Result

Different input images and their edge detected images are shown in Figure 4.1 The Figures A is the input images and Figures B is the respective edge detected result.

Figure 4.1:.A B

5. Conclusion

In this paper we implemented Canny Edge Detection algorithm on Spartan 6 FPGA. 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 synthesised using Xilinx and Spartan 6 FPGA board .

References

[1] Divya.d, sushma p.s " Fpga implementation of a distributed canny edgedetector"*IEEE 2013.*

[2] Chaithra.N.M., K.V. Ramana Reddy "Implementation of Canny Edge Detection Algorithm on FPGA and displaying Image through VGA Interface *" IEEE 2013*

[3] *Qian Xu, Chaitali Chakrabarti and Lina J. Karam "A Distributed Canny Edge Detector and Its Implementation On FPGA" School of Electrical, Computer and Energy Engineering, Arizona State University, IEEE, 2011*

[4] Parvinder Singh Sandhu, Mamata Juneja and Ekta Walia "Comparative Analysis of Edge Detectin Techniques for extracting Refined Boundaries" 2009 International Conference on Machine Learning and Computing ,IPCSIT vol 3, 2011.

[5] Wenhao He and Kui Yuan "An Improved Canny Edge Detector and its Realization on FPGA" IEEE Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China, June 25 - 27, 2008, pp. 6561-6564.

[6] Osman Z.E.M; Hussin ;Ali, N.B.Z "Optimization of Processor Architecture for Image Edge Detection Filter" IEEE transaction on Computer Modelling and Simulation, 2010, pp 648-652.

[7] Alasdair Mc Andrew. "Introduction to Digital Image Processing with MATLAB".

[8] Gao Jie and Liu Ning "An improved adaptive threshold canny edge detection algorithm", IEEE International Conference on Computer Science and Electronics Engineering, 2012, pp. 164-168.

[9] Muralikrishna, B.; Gnana Deepika ,K.; Raghu Kanth, B.; Swaroop Vemana, V.G.; "Image Processing using IP Core Generator through FPGA", International Journal of Computer Applications, vol 46-No.23, May 2012,pp. 48-52.
[10] Enoch Hwang, "Build a VGA Monitor Controller", Circuit Cellar, Issue 172 , November 2004,pp. 12-17.