IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 2, Issue 2 (Mar. – Apr. 2013), PP 05-09
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Image Smoothening Gradient Magnitude and Hystersis Calculation for Canny Edge Detector Using FPGA for Area Optimization

Chandrashekar N.S.[1], Dr. K.R. Nataraj [2]
[1,]*Department of ECE, Don Bosco Institute of Technology, Bangalore, Karnataka, India.*
[2,]*Department of ECE, SJB Institute of Technology, Bangalore, Karnataka, India.*

Abstract: *In this paper, image smoothening, gradient magnitude, hystersis used in Canny edge detection algorithm is presented. The new algorithm used has a low-complexity 8-bin non-uniform gradient magnitude histogram to compute block-based hysteresis thresholds that are used by the Canny edge detector. This also achieves less memory requirements, decreased latency and increased throughput with no loss in edge detection. Furthermore, the hardware architecture of the proposed algorithm is presented in this paper and the architecture is synthesized on the Xilinx Virtex 5 FPGA. The design development is done in VHDL and simulated results are obtained using modelsim 6.3 with Xilinx 12.2.*

Keywords: *- Canny Edge detector, Distributed Processing, FPGA, Non-uniform quantization.*

I. Introduction

Edge detection, as a basic operation in image processing, has been researched extensively. A lot of edge detection algorithms, such as Robert detector, Prewitt detector, Kirsch detector, Gauss-Laplace detector and canny detector have been proposed. Among these algorithms, canny algorithm has been used widely in the field of image processing because of its good performance [1]. The edge detection process serves to simplify the analysis of images by drastically reducing the amount of data to be processed, while at the same time preserving useful structural information about object boundaries. There is certainly a great deal of diversity in the applications of edge detection, but it is felt that many applications share a common set of requirements [2]. 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. Unfortunately, the canny edge detection algorithm contains extensive pre-processing and post-processing steps and is more computationally complex than other edge detection algorithms. Furthermore, it performs hysteresis thresholding which requires computing high and low thresholds based on the entire image statistics. This places heavy requirements on memory and results in large latency hindering real-time implementation of the Canny edge detection algorithm [3]. In [1, 2], a recursively implementable edge detection algorithm is suggested and optimized using retiming techniques. But its performance is quite poor in images with low SNRs. The approach of [5, 6] combines the derivative and smoothening operations of the Canny algorithm into a single mask to reduce computations. The pipelined implementation is a block-based approach with a block-size of 2 rows of pixels. It overcomes the dependencies between the blocks by fixing high and low thresholds to a constant value. In both of these approaches gradient thresholds are not adapted to the image characteristics, and their performance is not guaranteed for blurred images and images with low SNRs [3].

In [7], an improvised approach is done for image edge detection and its efficiency with a set of images. Results confirm that the DOW-canny edge detector, besides being insensitive to noise, is able to fabricate superior and accurate edges in regions of fine graining, geometrical figures and alpha-numerics as compared to the current edge detection techniques. The original Canny algorithm computes the higher and lower thresholds for edge detection based on the entire image statistics, which prevents the processing of blocks independent of each other. In order to reduce memory requirements, decrease latency and increase throughput, a distributed canny edge detection algorithm is proposed in [4]. The hysteresis threshold calculation is a key element that greatly affects the edge detection results. However, the original canny algorithm computes the high and low thresholds for edge detection based on the entire image statistics, which prevents the processing of individual blocks independently. Quality control has been performed in hard work conditions and made easier using automated visual systems quality control. It performs faults detection and final quality control. Edges and faults on ceramic tiles were detected using Canny edge detector. Problem of defining hysteresis thresholds was resolved with a histogram subtraction method [5]. In this paper it is presented the method for faults detection based on edge detection techniques. A method based on non-uniform and coarse quantization of the gradient magnitude histogram was proposed. In addition, the proposed algorithm is mapped onto reconfigurable

hardware architecture. The threshold is calculated using the data of the histogram of gradient magnitude rather than is set manually in a failure-and-try fashion and can give quite good edge detection results without the intervening of an operator.

This paper is organized as follows. Section 2 gives a Brief analysis of the Canny edge detector algorithm. Section 3 presents the implementation of proposed Canny edge detection algorithm. Simulation results are presented in Section 4. A conclusion is given in Section 5.

II. ANALYSIS OF CANNY EDGE DETECTOR

The block diagram of the Canny edge detection algorithm is demonstrated in "Fig." 1. The original canny algorithm [2] consists of the following steps:

- Smoothing the input image by Gaussian mask. The output smoothed image is denoted as I(x, y).
- Calculating the horizontal gradient $G_x(x, y)$ and vertical gradient $G_y(x, y)$ at each pixel location by convolving the image I(x, y) with partial derivatives of a 2D Gaussian function.
- Computing the gradient magnitude G(x, y) and direction $\theta_G(x, y)$ at each pixel location.

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (1)$$

$$\theta_G = \arctan(G_y/G_x) \quad (2)$$

- Applying non-maximum suppression (NMS) to thin Edges
- Computing the hysteresis high and low thresholds based on the histogram of the magnitudes of the gradients of the entire image.
- Performing hysteresis thresholding to determine the edge map.

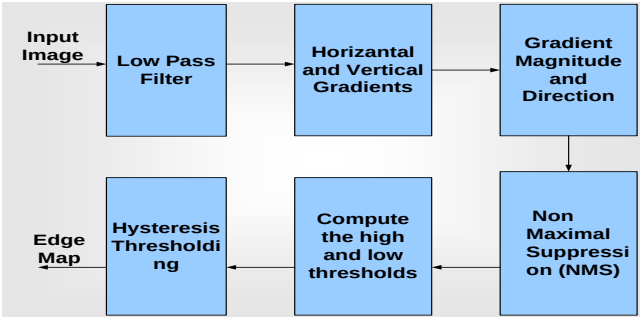

Fig 1.Block diagram of the Canny edge detection algorithm.

II. IMPLEMENTATION OF THE PROPOSED CANNY EDGE DETECTOR

In this section, the hardware implementation of proposed canny edge detection algorithm on the Xilinx FPGA is described.

In the proposed architecture, it consists of the following 5 units in "Fig." 2.

- Smoothening unit using Gaussian filter.
- Vertical and horizontal gradient Magnitude calculation unit.
- Directional non-maximum suppression unit.
- High and low threshold Calculation unit.
- Thresholding with hysteresis unit.

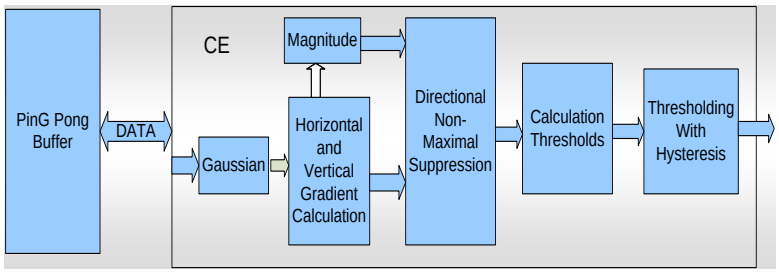

Fig. 2: Block diagram of the compute engine.

2.1 Image Smoothening:

The input image was smoothened using a 3×3 Gaussian mask. The Gaussian filter is separable and, thus, the implementation of the 2-D convolution with the 3×3 Gaussian mask was achieved using row and column 1- D convolutions.

2.2 Gradients and Gradient Magnitude Calculation:

Calculation of the vertical and horizontal gradients using convolution kernels was done in this stage. The kernels vary in size from 3 × 3 to 9 × 9, depending on the sharpness of the image. The whole design was pipelined, and thus the output was generated every clock cycle. This was an input to the magnitude calculation unit which computes, at each pixel location, the gradient magnitude from the pixel's horizontal and vertical gradients.

2.3 Directional Non Maximum Suppression:

In order to access all the pixels' gradient magnitudes in the 3 × 3 window at the same time, two FIFO buffers were employed. The horizontal gradient Gx and the vertical gradient Gy control the selector which delivers the gradient magnitude of neighbors along the direction of the gradient, into the arithmetic unit. This arithmetic unit consisted of one divider, two multipliers and one adder, which were implemented using the Xilinx Math Functions IP cores. The output of the arithmetic unit was compared with the gradient magnitude of the center pixel, and the pixel that has no local maximum gradient magnitude was eliminated.

2.4 Calculation of the hysteresis thresholds:

Since the low and high thresholds were calculated based on the gradient histogram, the histogram of the image was computed after it had undergone directional non-maximum suppression. As discussed in Section 2, an 8-step non-uniform quantizer was employed to obtain the discrete histogram for each processed block. The block-based hysteresis thresholds (high threshold ThH and low threshold ThL) were computed.

2.5 Thresholding with hysteresis:

Since, the output of the non maximum suppression unit contains some spurious edges, the method of thresholding with hysteresis was used. Two thresholds, high threshold ThH and low threshold ThL, which were obtained from the threshold calculation unit, were employed. Let f(x, y) be the image obtained from the non maximum suppression stage, f1(x, y) be the strong edge image and f2(x, y) be the weak edge image.

2 SIMULATION RESULTS AND ANALYSIS

Proposed system is implementation in FPGA using Device Virtex 5 XC5VTX240T and Package FF1759. "Fig." 3(a) shows design utilization summary of Image smoothening (thin blurring) and "Fig." 3(b) and "Fig." 3(c) shows RTL top module schematic and simulation results of Non smoothening (thin blurring) respectively.

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	8353	149760	5%
Numbers of Slice LUTs	19968	149760	13%
Number of fully used LUT-FF pairs	3242	25079	12%
Number of bonded IOBs	22	680	3%
Number of BUFG/BUFGCTRLs	3	32	9%

Fig.3 (a). Design Summary of Image smoothening (thin bluring) Unit

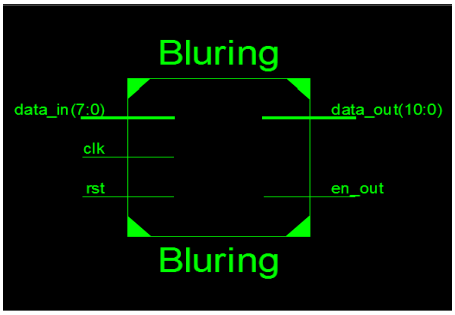

Fig.3 (b). RTL Top module Schematic of Image smoothening (thin bluring) Unit

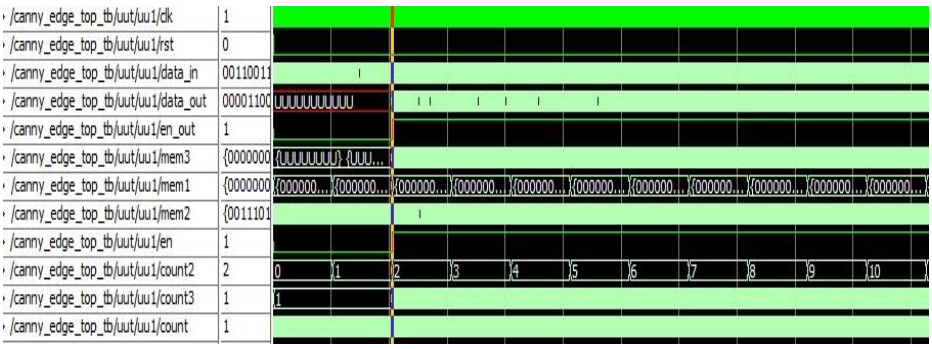

Fig. 3(c). Simulation Results of Image smoothening (thin bluring) Unit

"Fig." 4(a) shows design utilization summary of Gradient phase Unit and "Fig." 4 (b) and "Fig."4(c) shows RTL top module schematic and simulation results of Gradient phase Unit respectively.

Logic Utilization	*Used*	*Available*	*Utilization*
Number of Slice Registers	*11434*	*149760*	*7%*
Numbers of Slice LUTs	*24093*	*149760*	*16%*
Number of fully used LUT-FF pairs	*5120*	*30407*	*16%*
Number of bonded IOBs	*26*	*680*	*3%*
Number of BUFG/BUFGCTRLs	*3*	*32*	*9%*
Number of DSP48Es	*2*	*96*	*2%*

Fig.4 (a). Design Summary of Gradient phase Unit.

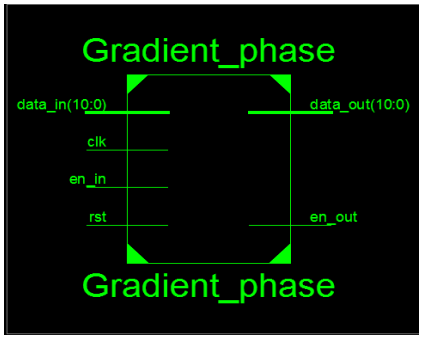

Fig.4 (b) RTL Top module Schematic of Gradient phase Unit

Fig. 4(c). Simulation Results of Gradient phase Unit

"Fig." 5(a) shows design utilization summary of Hysteresis Unit and "Fig." 5(b) and "Fig." 5(c) shows RTL top module schematic and simulation results of Hysteresis Unit respectively.

Device Utilization Summary(estimated values)			
Logic Utilization	Used	Available	Utilization
Numbers of Slice LUTs	10	149760	0%
Number of fully used LUT-FF pairs	0	10	0%
Number of bonded IOBs	36	680	5%
Number of BUFG/BUFGCTRLs	1	32	3%

Fig.5 (a). Design Summary of Hysteresis Unit.

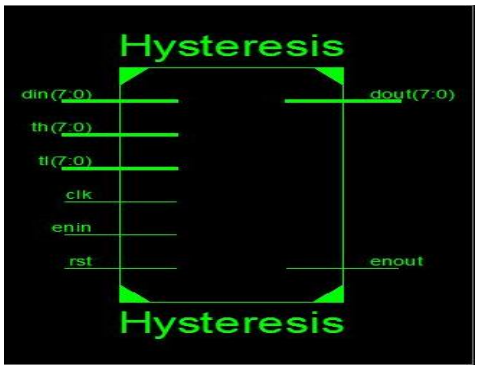

Fig.5 (b) RTL Top module Schematic of Hysteresis Unit.

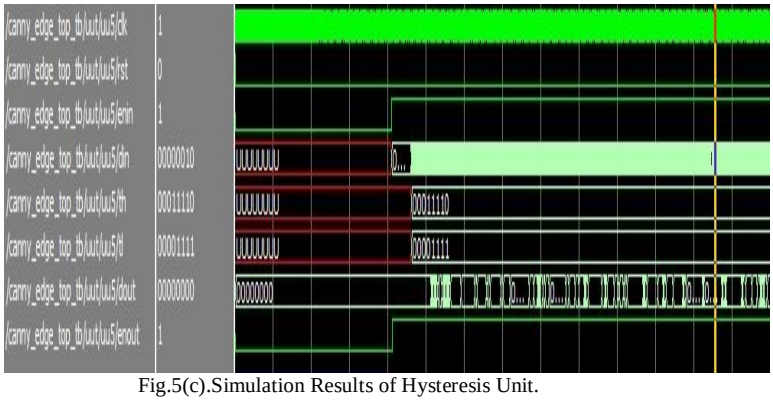

Fig.5(c).Simulation Results of Hysteresis Unit.

V. CONCLUSION

A new canny edge detector algorithm using FPGA implementation for Image smoothening and gradient magnitude was proposed in this work. A novel non-uniform quantized histogram calculation method was used in this work in order to reduce the computational cost of the hysteresis threshold selection. As a result, the computational cost of the proposed algorithm was found to be very low compared to the original Canny edge detection algorithm. The algorithm is mapped to onto a Xilinx Virtex-5 FPGA platform and tested using ModelSim. It is capable of supporting fast real-time edge detection for images and videos with various spatial and temporal resolutions including full-HD content.

REFERENCES

[1] L. Torres, M. Robert, E. Bourennane, and M. Paindavoine, "Implementation of a recursive real time edge detector using Retiming techniques," *VLSI, Aug. 1995,* pp.811 –816.

[2] J. Canny, "A computational approach to edge detection,"*IEEE Trans. PAMI, vol. 8, no. 6, Nov. 1986*, pp. 679 –698.

[3] Qian Xu, Chaitali Chakrabarti and Lina J. Karam*,"*A Distributed Canny Edge Detector and Its Implementation On FPGA", *Arizona State University, Tempe, AZ.* 2011.

[4] S. Varadarajan, C. Chakrabarti, L. J. Karam, and J. M.Bauza, "A Distributed psycho-visually motivated canny edge detector," *IEEE ICASSP, Mar. 2010,* pp. 822 –825,.

[5] Željko Hocenski, Suzana Vasilić*,* "Improved Canny Edge Detector in Ceramic Tiles Defect Detection", Osijek, *Croatia 6-10 Nov2006.*

[6] W. He and K. Yuan, "An improved Canny edge detector and its Realization on FPGA," *WCICA, Jun. 2008*, pp. 6561 –6564.

[7] Anand Gupta , Ravi Kumar Dalal,Rahul Gupta,Pulkit Wadhwa "DOW-Canny: An Improvised Version Of Canny Edge Detector". *(ISPACS) December 2011,* pp. 7-9.