

A HARDWARE ARCHITECTURE OF PREWITT EDGE DETECTION

ARAMESH SEIF

UNIVERSITI TEKNOLOGI MALAYSIA

ABSTRACT

The objective of this project is to develop a real-time hardware architecture for Prewitt edge detection algorithm. Prewitt edge detection provides differencing operation in the single kernel. Verilog hardware description language was used as the hardware programming language for a real-time edge detection system. The architecture is capable of operating with a clock frequency of 145 MHz at 550 frames per second. Computation error analysis performed shows that the proposed architecture produces outputs similar to that obtained by software simulation using Matlab.

ABSTRAK

Objektif projek ini adalah untuk menghasilkan senibina peranti perkakasan masa-sebenar untuk algoritma Prewitt pengesanan tepi. Pengesan tepi Prewitt menyediakan operasi pembezaan dalam satu kernel. Bahasa keterangan perkakasan Verilog digunakan sebagai bahasa pengaturcaraan peranti perkakasan untuk sistem masa-sebenar pengesanan tepi. Senibina ini mampu beroperasi pada 550 rangka sesaat dengan frekuensi jam 145 MHz. Analisa hingar perhitungan yang dilakukan menunjukkan bahawa senibina yang dicadangkan menghasilkan keluaran yang sama dengan yang diperolehi melalui simulasi menggunakan perisian Matlab.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	viii
	LIST OF TABLES	ix
	LIST OF FIGURES	x
	LIST OF ABBREVIATIONS	xi
	LIST OF SYMBOLS	xii
	LIST OF APPENDICES	xiii
1	INTRODUCTION	1
	1.1 Edge Detection	1
	1.2 Problem Statement	2
	1.3 Objectives	3
	1.4 Scope of work	3
	1.5 Report Outline	3
2	LITERATURE REVIEW	4
	2.1 Prewitt Edge Detection	4
	2.2 Implementation and Technology Issues	6
	2.3 Hardware Optimization Techniques	8
	2.4 Chapter Summary	9
3	PROPOSED HARDWARE ARCHITECTURE OF PREWITT EDGE DETECTOR	10
	3.1 Related Work	10
	3.2 Proposed Architecture of Edge Detection	11

3.3	Functional Block Diagram of Edge Detection	11
3.4	Memory pointer unit with external memory and delay line	12
3.5	The Internal Structure of the Moving Window and AU	13
3.6	Arithmetic Unit	15
3.7	Control Unit	16
3.8	Chapter Summary	16
4	EXPERIMENTAL WORK	17
4.1	Experimental Setup	17
4.2	Matlab Implementation Edge Detection	17
4.3	Computational Result of Edge Detection	18
4.4	Experimental Result and Analysis	20
5	CONCLUSIONS	21
5.1	Significance of Findings	21
5.2	Future Works	21
	REFERENCES	23
	Appendices A – B	25 – 27

LIST OF TABLES

TABLE NO.	TITLE	PAGE
4.1	Synthesis Result.	20
4.2	The comparison between previous works.	20

CHAPTER 1

INTRODUCTION

Real-time video and image processing is used in a wide variety of applications from video surveillance and traffic management to medical imaging applications. Image processing typically require very high computational power. To meet demands for high throughput image processing capability, parallel processing make them an attractive implementation option offered by hardware architecture. Instead of using multiple digital signal processors, a single field programmable gate array (FPGA) can deliver the essential level of computing power at a lower cost [7]. The combined MATLAB and Verilog provide an easy interface to import and export data to the designed hardware implementation to read and display images.

This project report proposes a hardware architecture of Prewitt edge detection. The hardware architecture provides the necessary performance for real-time image and video processing, while retaining the system flexibility to support an adaptive algorithm.

1.1 Edge Detection

Edge detection is a fundamental tool used in most image processing applications to obtain information from the frames before feature extraction and object segmentation. This process detects outlines of an object and boundaries between objects and the background in the image. An edge detection filter is also used to improve the appearance of blurred or low-pass filtered video streams [7].

Technically, edge detection is the process of locating the edge pixels, in order to increase the contrast between the edges and the background so that the edges become more visible [7]. The basic edge-detection operator is calculated by forming a matrix

centred on a pixel chosen as the center of the matrix area. Some of the edge detection techniques are Prewitt [7], Sobel [7] and Canny [19] operators that are described below. All are gradient-based algorithms that have kernel operators to calculate the strength of the slope in directions that are orthogonal to each other, generally horizontal and vertical.

The Prewitt operator uses a pair of 3×3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). $|G_x| + |G_y|$ gives an indication of the intensity of the gradient in the pixel.

The Sobel operator is similar to the Prewitt operator. The difference is that the Sobel operator assigns a higher weight to pixels located at shorter distances from the middle pixel.

The Canny algorithm uses an optimal edge detector based on a set of criteria which include finding the most edges by minimizing the error rate, marking edges as closely as possible to the actual edges to maximize localization, and marking edges only once when a single edge exists for minimal response. The first stage involves smoothing the image by convolving with a Gaussian filter. This is followed by finding the gradient of the image by feeding the smoothed image through a convolution operation with the derivative of the Gaussian in both the vertical and horizontal directions [18].

1.2 Problem Statement

Image processing algorithms used to be implemented only on software. But increasing demands higher throughput solutions. Due to the advances in the very large scale integration (VLSI) technology, hardware implementation has become an attractive alternative. This project proposes a hardware architecture for implementing a suitable edge detection for high speeds image processing.

1.3 Objectives

The aim of this project is to propose and develop a hardware architecture of Prewitt edge detection on FPGA. To achieve this aim, it is necessary to know the mathematical properties of edge detection. Therefore, the first objective of the project is to analyse the mathematical properties of Prewitt edge detection. The second objective is to map the mathematical operations as a hardware architecture models, architecture targeted for Altera FPGA using Quartus II. The last objective is to integrate and verify the hardware architecture with a Matlab implementation .

1.4 Scope of work

This project is limited by the following: Input image is limited to 8-bit grayscale and a frame size of 256×256 pixels. Moving pixel window is limited to 3×3 pixels. The proposed architecture is targeted for Altera FPGA using Quartus II. Verification through synthesis only with parameters obtained from simulation on Matlab.

1.5 Report Outline

This rest of the report is organized as follows. Chapter 2 introduces issues of edge detection architecture. Chapter 3 explains methodology of this project. Chapter 4 describes implementation details of Prewitt edge detection. Finally, Chapter 5 summarizes this project and proposes directions for future work.

CHAPTER 5

CONCLUSIONS

This chapter summarizes the findings and suggests potential future work.

5.1 Significance of Findings

The hardware architecture for Prewitt edge detection operator is proposed. This architecture can be reconfigured for any resolution image by just changing the memory pointer unit. The architecture is capable to operate a speed of 145 MHz with 553 fps which is much better than processing images on software platform using traditional high level programming languages like C or C++. This architecture also is capable of operating at a clock frequency of 145 MHz. Hence, the edges of 256×256 pixel image can be calculated in just 1.8 ms, faster than previous works reported elsewhere.

The work done demonstrates the ability of the proposed architecture to implement real-time image processing algorithms. This work also demonstrates some limitation in implementing image processing algorithms on hardware architecture which requires more computing elements and more memory elements for implementing real-time image processing algorithms.

5.2 Future Works

This project can be extended to the following, the increasing demand for high resolution image processing may require image sizes up to 512×512 or larger. This would imply changes of the MPU and using larger memory for storing image. Another possible future work for this project could include hardware prototyping on

FPGA. Image processing algorithms are excellent choices for FPGA implementation provided, they are optimized to exploit hardware parallelism and simplified operations. Reprogramability of FPGAs allows faster and cheaper design cycle of DSP systems compared to ASIC.

REFERENCES

1. R. McCready, "Real-time face detection on a configurable hardware platform," M.S. thesis, Dept. Elect.Comput. Eng., Univ. Toronto, Toronto, ON, Canada, October 2000.
2. P.-Y. Hsiao, C.-H. Chen, H. Wen, and S.-J. Chen, "Real-time realisation of noise-immune gradient-based edge detector" IEE Proc.-Comput. Digit. Tech, Vol. 153, No. 4, July 2006, pp. 261-269.
3. D. Nguyen, D. Halupka, P. Aarabi, A. Sheikholeslami, "Real-Time Face Detection and Lip Feature Extraction Using Field-Programmable Gate Arrays", Systems, Man, and Cybernetics, Part B, IEEE Transactions on Vol 36, no 4, Aug 2006. pp. 902 - 912.
4. D. Maar, E. Hildreth, "Theory of edge detection", Proceedings Royal Soc. London, vol. 207, May 1980 detect.pdf. pp. 187-217.
5. M. Venkatesan, D. Venkateshwar, "Hardware Acceleration of Edge Detection Algorithm on FPGAs" Rao Department of Electrical and Computer Engineering University of Nevada Las Vegas. research papers, June 2004.
6. J.-R. Parker, "Algorithm for image processing and computer vision" John Wiley and Sons, April 1996.
7. AN333: Developing Peripherals for SOPC Builder: <http://www.altera.com/literature/an/an333.pdf>, IEEE Transactions on Consume Electronics. Vol. 43, No.4, November 1997.
8. M.-B. Gokala, J.-M. Stone, M. Kalinowski, "Stream -Oriented FPGA Computing in the Streams -C High Level Language", IEEE symposium on Field Programmable custom computing machines, April 2000, pp.49-56.
9. N. Banerjee, et. al., "MATCH: A MATLAB Compiler for Configurable Computing Systems," Technical Report, Center for Parallel and Distributed Computing, Northwestern University, August 1999.
10. Ng. Man, "High Level Design for High Speed FPGA Devices". Masters Thesis, Dept of computing. Imperial College. June 2002.