P. Y. Hsiao, L. T. Li, C. H. Chen, S. W. Chen and S. J. Chen, "An FPGA architecture design of parameter-adaptive real-time image processing system for edge detection", IEEE *Emerging Information Technology Conference*, 15-16 Aug. 2005, pp.1-3.

An FPGA Architecture Design of Parameter-Adaptive Real-Time Image Processing System for Edge Detection

Pei-Yung Hsiao[1], Le-Tien Li[1], Chia-Hsiung Chen[2], Szi-Wen Chen[1], and Sao-Jie Chen[2]
[1]Department of Electronic Engineering, Chang Gung University, Tao Yuan, Taiwan, ROC
[2]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, ROC

***Abstract*-In this paper we present an FPGA architecture design of parameter-adaptive real-time image processing system for edge detection. The system contains two edge detection algorithms which are suitable for hardware realization and insensitive to noise. The two adopted algorithms are able to produce different outputs suitable for different applications. A controller which integrates parameter setting, continuous edge detection task processing and output selection is proposed. The proposed modified LGT algorithm not only preserves the original edge detection performance, but also greatly reduces the use of hardware resource. We adopt FPGA design flow as our early-stage verification platform, and result in a maximum working frequency of 54MHz, which is able to process 205 512x512 grayscale images, and is 90 times faster than the software execution.**

I. INTRODUCTION

Over the years, edge detection techniques have played an important role in digital image processing and object recognition [1]. To this day, edge detection has been widely used in medical imaging, license plate recognition, and intelligent surveillance system. More recent literatures show the significance of edge detection in the advanced intelligence vehicle system applications, such as lane marker detection [7], autonomous driving system, and adaptive cruise control system. In spite of all these wide-spread applications, the result of an edge detector would still deeply affect the overall performance of a high-end application, and whether the edge detection task can be done in real-time has become an importance issue, which shows that the availability of hardware reference design of properly selected edge detection algorithm is highly desirable, and would be highly valuable in the near future.

To evaluate the performance of an edge detector, and assist us choose the suitable algorithms, there are three important factors [2]: (1) the ability to resist to noise, (2) the ability to mark edge points as close to the center of true edge, and (3) the connectivity and completeness of detected edges. When designing hardware architecture for a computationally-complex software algorithm, we need to additionally consider whether it has the characteristic of hardware regularity and whether it can perform edge detection in real time.

According to the aforementioned criterions, we adopted the non-gradient based LGT (Local and Global Threshold) algorithm [3] and gradient-based ADM (Absolute Difference Mask) algorithm [4] in our implementation. As for hardware implementation and verification platform, we select the FPGA design flow as our early-stage design platform. FPGA design flow has the merits of rapid prototyping and easy design verification. It is also an important step we have to go through before the ASIC flow tape out.

The remaining part of this paper is organized as follows: Section II reviews the LGT and ADM algorithm, while Section III describes the architecture design of our parameterized dual edge detection system, and the PA controller which make all things work. Section IV gives the simulation and experimental results and finally, we conclude this paper in Section V.

II. ALGORITHM OVERVIEW

In this section, we will briefly overview the modified non-gradient based LGT and gradient-based ADM algorithm. Comparisons between the two algorithms will be given at the end of this section.

A. Non-gradient Based Modified LGT Algorithm

LGT algorithm is a threshold-based algorithm. It locates edge points in images using statistical local and global threshold values. In local flow, the local threshold value T_L is first obtained by the local mean value of the current 3x3 mask subtracted by a user inputted constant C, and used to help generate another 3x3 data mask W_L by comparing T_L with the current mask. The W_L mask is then compared with 16 pre-defined templates and decides whether the center pixel is an edge pixel. On the other hand, the global flow determines the center pixel to be a global edge point if the absolute average deviation (AAD) of the current 3x3 mask is larger than the global threshold value Tg. We use AAD to approximate the variance σ^2 used in the original algorithm to simplify the calculation. The final edge out is obtained by ANDing the result of local and global edge decision. Fig. 1(a) shows the flow diagram of the LGT algorithm.

B. Gradient-based ADM Algorithm

The ADM algorithm is a noise-immune three-stage algorithm. The inputted image I is first passed through the semi-Gaussian smoothing unit to generate a blurrier but cleaner image I'. Image I' is then used to detect the edge

0-7803-9329-5/05/$20.00 ©2005 IEEE

strength and edge detection, and finally, the edge points of the original image is obtained using those edge strength and direction information. Fig. 1(b) shows the processing flow of the ADM algorithm.

C. Algorithm Comparison

We will now demonstrate the noise-immune property of the adopted algorithms. Refer to Fig. 2, the original input image has been infected by Gaussian noise with σ = 0.004. We can find that both algorithms perform quite well even when the original image is noisy. It is also not hard to find that with different values of C and Tg employed in the original and modified LGT algorithm, we can have similar performance. However, from the result shown in Figs. 2 (b)(c) and (d), the LGT algorithm is more sensitive to the fore-scene edges while the ADM algorithm provides more complete edge map of the original images. In other words, the result of LGT algorithm shows objects of different scene level while the ADM algorithm presents all edges on the same level. The availability of both algorithms provides us different choices for different applications.

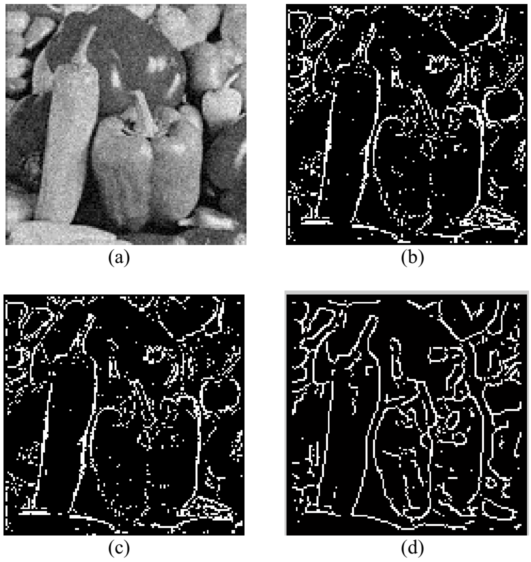

Fig. 2. Performance evaluation. (a) Pepper image with σ=0.004 Gaussian noise, (b) result of the original LGT algorithm with C=10, Tg=340, (c) result of the modified LGT algorithm with C=7, Tg=134, and (d) reslult of ADM algorithm.

III. PARAMETERIZED DUAL EDGE DETECTOR

The proposed parameterized dual edge detection system contains three main parts: (1) LGT edge detection unit (LGTEDU), (2) ADM edge detection unit (ADMEDU), and (3) parameter controller (PA Controller). With different input parameters the system is able to output LGT/ADM edge detection result, or Gaussian/average filtered image. The block diagram of the parameterized dual edge-detection system is shown in Fig. 3, and is described in detail in the following subsections.

A. LGTEDU

LGTEDU is composed of the mean unit and the LGT processing unit. The mean unit calculates the mean of a current data window, and the result mean value is used in both local and global flow of the LGT algorithm. Here we approximate the divide-by-9 operation with a shift-based operation to simplify the hardware. The maximum error caused by this simplification is 1.6%, which is minor and acceptable for the overall operation. The LGT processing unit is a 3-stage pipelined processing unit which contains a local operation unit (LOU), a global operation unit (GOU) and an AND unit. The number of pipeline stages in the LOU and GOU units is the same so as to maintain the data dependency constraint.

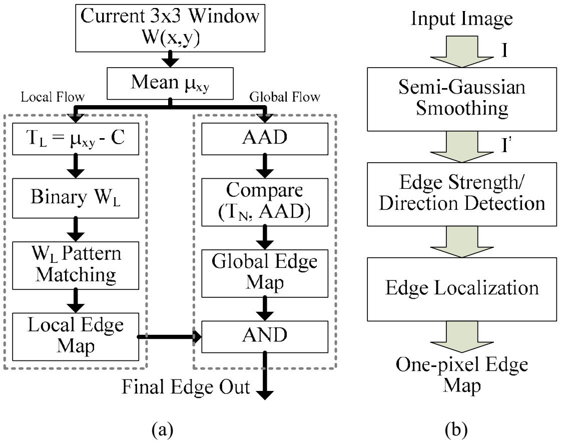

Fig. 1. (a) Flow diagram of the LGT algorithm, and (b) flow of the ADM algorithm.

B. ADMEDU

ADMEDU is comprised of a smoothing unit, an edge strength unit and an edge localization unit [3]. Each unit

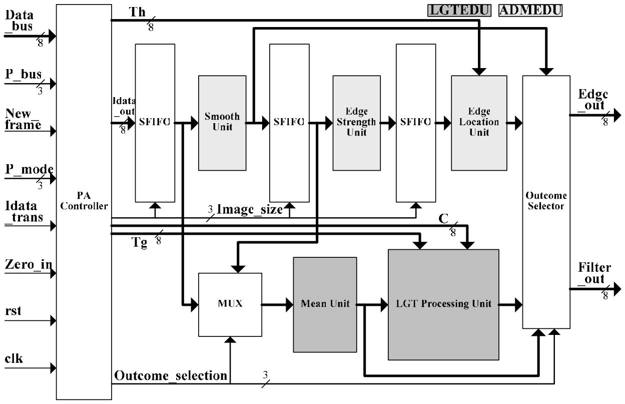

Fig. 3. Architecture block diagram of the proposed dual edge detection system.

corresponds to the three main blocks of the original algorithm. The smoothing unit is basically a convolution operation. It can be easily achieved by incorporating a 5x5 systolic array. The edge strength unit and edge localization unit can be implemented using simple operations, and is properly pipelined to shorten the latency of each stage.

C. PA Controller

To achieve the goal of designing a complete control flow for our image processing system, a controller which connects all parts together is needed. The PA controller has the following basic operations: (1) control of the processing flow, (2) parameter setting, and (3) result output selection. To handle all input data and control signals, direct signals to the correct processing block and accomplish the basic operations, we designed an 8-state state transition diagram for the controller, as shown in Fig. 4.

IV. Experimental Results

To shorten the design time of the proposed dual edge detection system, we adopted Xilinx Foundation 4.2i as the synthesis, simulation and verification software tool, and select the Xilinx Virtex XCV1000EHQ240 FPGA as our target platform. The maximum working frequency for the system is 54MHz, which is able to process 512x512 grayscale images at 205 frames/s. To further characterize the value of hardware edge detection system, we also compare the performance of the proposed architecture with software execution, and show the results in Table I. The software simulation is performed on a Pentium4 1.5GHz CPU with 256MB RAM using C programming language. We can see that the processing speed gained from the hardware implementation can be ranged from 65 – 90 times faster than that of the software execution.

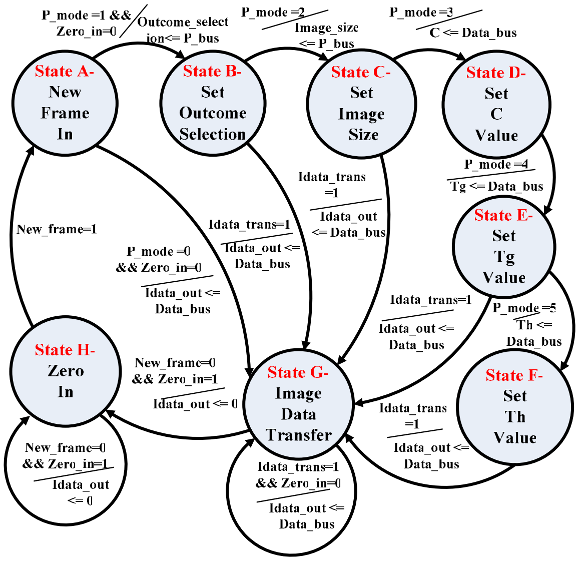

Fig. 4. State transition diagram for PA controller.

TABLE I
Hardware utilization and performance matrix.

Function	HW time	SW time	Speed diff.
Mean filter	4.854 ms	318 ms	65 times
Gaussian filter	4.854 ms	324 ms	66 times
ADM	4.854 ms	439 ms	90 times
LGT	4.854 ms	427 ms	87 times

V. Conclusion

In this paper, we presented a real-time parameter-adaptive dual edge detection system. The system contains the hardware implementation of two noise-immune edge detection algorithms, and a PA controller to provide a unified parameter setting interface, output selection and continuous image processing. The system is also able to output mean or Gaussian filtered images. All functional units in the system have been carefully pipelined and substituted with suitable low-complexity operations to increase the throughput.

This design has been verified on the Xilinx Virtex FPGA at a maximum working frequency of 54MHz, which is able to process grayscale images of size 512x512 at 205 frames/s, and is about 90 times faster than software execution. Its high frame-rate and regular hardware architecture make it suitable for high-demanding real-time image processing applications. In the future, we will further realize the idea on ASIC or DIP to show the value of this work.

Acknowledgement

This work was supported in part by National Chip Implementation Center and National Science Council of the Republic of China under Contract NSC 93-2215-E-182-005.

References

[1] R.C. Gonzalez and R.E. *Woods, Digital Image Processing*. Addison-Wesley Pub Co, 2nd Edition, 2002.

[2] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 679-714, 1986.

[3] F.M. Alzahrani and T. Chen, "A Real-Time Edge Detector: Algorithm and VLSI Architecture," *Real-Time Imaging*, vol. 3, issue 5, pp. 363-378, October 1997.

[4] M.B. Ahmad and T.S. Choi, "Local Threshold and Boolean Function Based Edge Detection", *IEEE Trans. Consumer Electron.*, vol. 45, No. 3, pp. 674-679, August 1999.

[5] P.Y. Hsiao, H. Wen, Y.P. Chen, and S.J. Chen, "Real-Time Implementation of Noise-Immune Gradient-Based Edge Detection," *IEEE International Symposium on Signals, Circuits and Systems, Iasi, Romania*, July 14-15, 2005.

[6] R.R. Rakesh, P. Chaudhuri, and C.A. Murthy, "Thresholding in Edge Detection: A Statistical Approach," *IEEE Trans. Image Processing*, vol.13, No.7, pp.927-936, 2004.

[7] S.S. Huang, C.J. Chen, P.Y. Hsiao, L.C. Fu, "On-Board Vision System for Lane Recognition and Front-Vehicle Detection to Enhance Driver's Awareness," *IEEE International Conference on Robotics and Automation, New Orleans, LA, USA*, pp.2456-2461, April 26 – May 1, 2004.