2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science

Image Edge Detection Based on FPGA

Zhengyang Guo，Wenbo Xu，Zhilei Chai
School of Information Technology
Jiangnan University
Wuxi, P.R.China
e-mail:guozhengyang@yahoo.com.cn

***Abstract*—Field Programmable Gate Array (FPGA) is an effective device to realize real-time parallel processing of vast amounts of video data because of the fine-grain reconfigurable structures. This paper presents a kind of parallel processing construction of Sobel edge detection enhancement algorithm, which can quickly get the result of one pixel in only one clock periods. The algorithm is designed with a FPGA chip called XC3S200- 5ft256, and it can process 1024×1024×8 Gray Scale Image successfully. The design can locate the edge of the gray image quickly and efficiently.**

Keywords- Edge detection;Sobel operator;FPGA;Real-time; Parallel processing

I. INTRODUCTION

The edges of image are considered to be most important image attributes that provide valuable information for human image perception [1-3]. The edge detection is a terminology in image processing, particularly in the areas of feature extraction, to refer to algorithms which aim at identifying points in a digital image at which the image brightness changes sharply [4-6].The data of edge detection is very large, so the speed of image processing is a difficult problem. FPGA can overcome it [7]. Sobel operator is commonly used in edge detection. Sobel operator has been researched for parallelism [8], but Sobel operator locating complex edges are not accurate; it has been researched for the Sobel enhancement operator in order to locate the edge more accurate and less sensitive to noise, but the software can not meet the real-time requirements [9]. For this reason, a FPGA implementation of edge detection enhancement algorithm has proposed in this paper, it not only meets the real-time requirements, but also accurately locate the image edges.

II. SOBEL EDGE DETECTION ENHANCEMENT ALOGRITHM

In edge detection, the Sobel operator is used commonly. The Sobel operator is a classic first order edge detection operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is the corresponding norm of this gradient vector. The Sobel operator only considers the two orientations which are 0° and 90° convolution kernels. The operator uses the two kernels which are convolved with the original image to calculate approximations of the gradient.

The two convolution kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point. The gradient magnitude is given by:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y|$$

This is much faster to compute.

The Sobel operator has the advantage of simplicity in calculation. But the accuracy is relatively low because it only used two convolution kernels to detect the edge of image. Therefore, the orientation of the convolution kernels is increased from 2 to 4 in order to increase the accuracy of edge detection. The four convolution kernels are shown in Fig.1.

III. FPGA HARDWARE IMPLEMENTATION

This design uses 3×3 convolution kernels, processing 1024×1024×8 Gray Scale Image. The architecture is shown in Fig.2.The system is divided into four modules: 3×3 pixel generation module, Sobel enhancement operator module edges control module and binary segmentation [10,11].

In this system, Clk is the clock signal, Reset is the reset signal and EN is data control signal, Data input is the pixel signal of Gray Scale Image, Result is the result of edge detection operator signal, Generation data and Data are the middle signal. The function and structure of each module are as follows

A. 3×3 Pixel Generation Module

The structure of 3×3 pixel generation module is shown in Fig.3.

−1	−2	−1
0	0	0
1	2	1

0°

−2	−1	0
−1	0	1
0	1	2

45°

−1	0	1
−2	0	2
−1	0	1

90°

0	1	2
−1	0	1
−2	−1	0

135°

Figure 1. Convolution kernel.

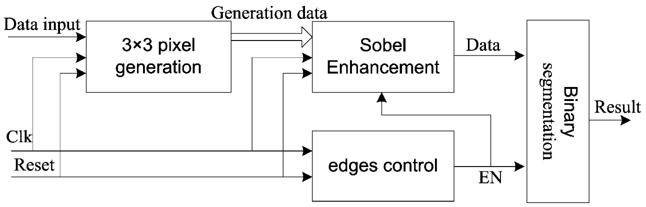


Figure 2. Architecture

978-0-7695-4110-5/10 $26.00 © 2010 IEEE
DOI 10.1109/DCABES.2010.39

This module consists of 3 shift register groups and two FIFO. The FIFO is used to cache a line of image data. The image data input according to the clock signal, so P1, P2, •••, P9 is the 3 × 3 image data template. When the data is continuously input, 3×3 image data template change. It can contain all pixels of an image. The FIFO is generated by dual-port RAM instead of FIFO IP core [12].

B. *Sobel Enhancement Operator Module*

The structure of Sobel enhancement operator module is shown in Fig.4

The parallel processing construction is used in orientation convolution kernel. The orientation convolution result is compared each other, and then, the maximum value is the output. The pipeline structure is used to calculate each orientation convolution kernel. It is six corresponding input data because three coefficients of each convolution kernel are zero; Multiplied by 2 is instead by one left. The structure is shown in Fig.5.

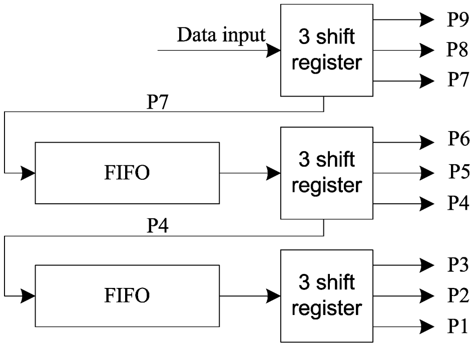


Figure 3. 3×3 pixel generation

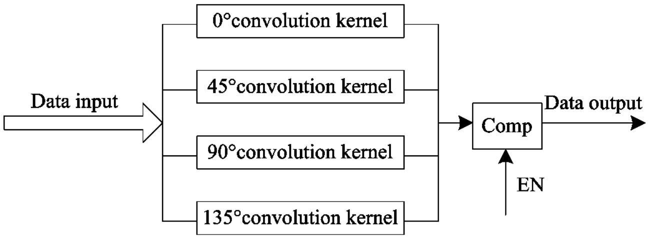


Figure 4. Sobel operator parallel structure

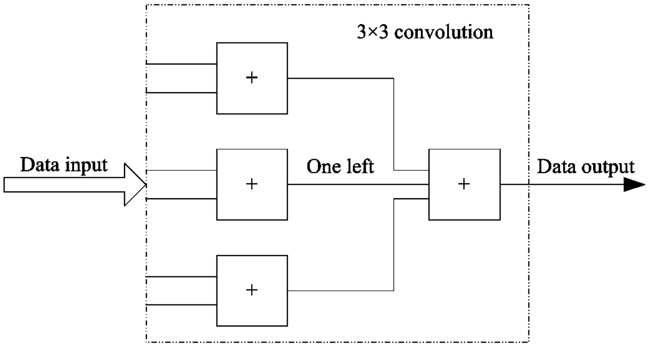


Figure 5. Convolution structure

C. *Edges Control Module*

The structure of edges control module is shown in Fig.6. Clk is the clock signal and Reset is the reset signal; Turn is enable signal, when the Turn valid, the module work. EN is the output data control signal. This module can know where the current pixel location and whether it is the edges of the pixel image. Sobel edge detection enhancement operator can not deal with the left edge, right edge, the up edge and down edge. In this design, the result of the edge pixels is set to zero, otherwise, call Sobel enhancement operator module.

D. *Binary Segmentation Module*

The structure of binary segmentation module is shown in Fig.7. EN is the output data control signal. Data is the result of the Sobel enhancement operator module. Result is 0 or 255. In this module, the final result is the binary image of edge detection having only two pixel values according to the given threshold value, i.e., 0 and 255.

IV. Experimental Results

The design was implemented in the XILINX Spartan3 XC3S200 FPGA by ISE9.2. The device utilisation summary is given in Table. I. Small resource is taken up, so there is possibility of implementing some more parallel processes with this architecture on the same FPGA.

The system clock frequency is 50MHz, The edges of 1024×1024×8 pixel gray image can be found out in only

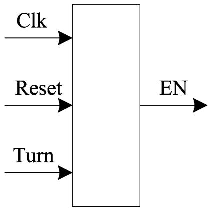


Figure 6. Edges control

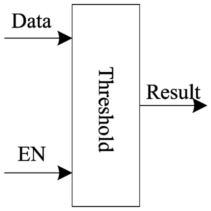
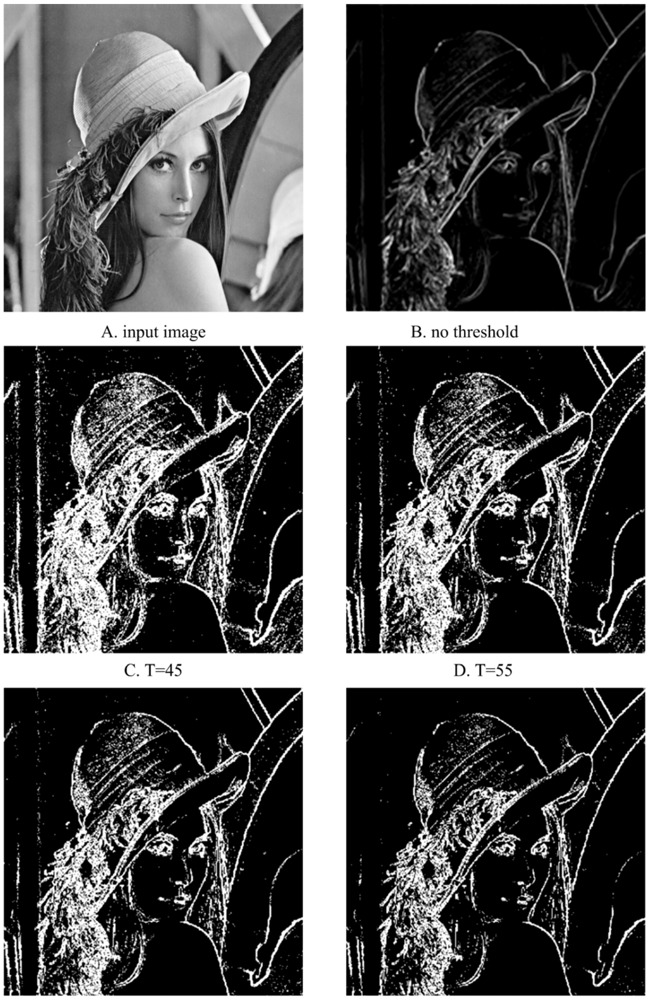


Figure 7. Binary segmentation

TABLE I. Utilisation Summary

Logic Utilization	*Used*	*Available*	*Utilization*
Number of Slices	216	1920	11%
Number of Slice Flip Flops	289	3840	7%
Number of 4 input LUTs	346	3840	9%
Number of bonded IOBs	18	173	10%
Number of BRAMs	2	12	16%
Number of GCLKs	1	8	12%

21ms, It meets the real-time requirements. The input image and the results are shown in Fig.7. As is shown in Fig.7, 7(A) is the input image. 7(B) is the result of the Sobel edge detection enhancement algorithm without threshold, as is shown in 7(B), the boundary lines are located accurately, it is thin and it is not sensitive to noise. The image from 7(C) to 7(F) is the result of the Sobel edge detection enhancement algorithm with different given threshold.



A. input image

B. no threshold

C. T=45

D. T=55

E. T=65

F. T=80

Figure 8. Result

V. CONCLUSION

The Sobel operator adding the orientation of the convolution kernels can locate accurately the edge, thin the boundary lines, and not be sensitive to noise. The FPGA implementation of it meets the real-time requirements. This architecture based on FPGA is much better than processing images on software platform using high level programming languages like C or C++ [13].

REFERENCES

[1] Jain, Anil K. (1989). Fundamentals of Digital Image Processing, Prentice-Hall, Inc.

[2] Chanda, B. and Dutta, D. Majumdar. (2001). Digital Image Processing and Analysis,Prentice-Hall of India.

[3] Gonzalez, Rafael C. and Woods, Richard E. (2002). Digital Image Processing, Pearson Education, Inc.

[4] Pratt, W. K. (2004). Digital Image Processing, John Wiley & Sons, Inc.

[5] Bose, Tamal (2004). Digital Signal and Image Processing, John Wiley & Sons, Inc

[6] Rafael C. Gonzalez, Richard E. Woods . Digital Image Processing (2nd Edition). Prentice Hall, 2nd edition (January 15, 2002)

[7] D. T.Saegusa, T.Maruyama, Y.Yamaguchi, "How fast is an FPGA in image processing?", IEICE Technical Report, Vol.108. No.48, 2008, pp.83–88

[8] Yangli ,Yangbing. "Study of FPGA based Parallel Processing of Sobel Operator" AI Modern Electronics Technique 2005.J.

[9] SHEN fengting WEI hong "An Improved Thread Edge Detection Method Based On Sobel Algorithm". CONTROL&AUTOMATION 2008

[10] Steve Kilts, Advanced FPGA Design: Arichitecture , Implementation , and Optimization , John Tiley & Sons

[11] Arrigo Benedetti, Andrea Prati, Nello Scarabottolo. "Image convolution on FPGAs: the implementation of a multi-FPGA FIFO structure". Euromicro Conference, 1998.

[12] Spartan FPGA Complete Data Sheet Xilinx Inc.

[13] P. Athanas and A. Abbott. Real-time image processing on acustom computing platform. In IEEE Computer, Feb. 1995