The Design of Sobel Edge Extraction System on FPGA

Yu ZHENG[1,*]

[1]*School of software, Beijing University of technology, Beijing 100124, China;*

Abstract. Edge is a basic feature of an image, the purpose of edge detection is to reduce the amount of data in an image while preserving the shape information of objects .Field programmable gate arrays (FPGA) are best suited to implement image processing algorithms with the reason that it provides infinite reprogram ability and high parallelism.The aim of this paper is to design a real-time sobel extract system based on HLS method and the implement it on ZYNQ FPGA. The result shows the software processing time is almost 80 times than the hardware processing. The hardware acceleration has remarkable effect on sobel filtering.

1 Introduction

Edge is a basic feature of an image, the purpose of edge detection is to reduce the amount of data in an image while preserving the shape information of objects in an image. Edge detection is a preprocessing step for further image processing such as feature extraction and object segmentation. With the continuous research on edge detection algorithm, the noise immunity and quality of edge detection has been greatly improved, but the complexity of the algorithm is also greatly increased simultaneously. Furthermore, image processing systems are recognized as computational and data intensive systems. Such systems need high performance capabilities to meet the real time applications requirements. Those applications include real-time face/fingerprint recognition, video surveillance, military aerial systems. Designing special circuits for special applications is a thought of speeding up. Field programmable gate arrays (FPGA) are best suited to implement image processing algorithms with the reason that it provides infinite reprogram ability, high parallelism, and costs advantages when compared to ASIC with similar level of performance in most cases. Historically, the programming model of an FPGA was centered on register-transfer level descriptions and it is a great obstacle for software designers. High-Level Synthesis makes it possible that software engineer set up an SOC quickly when lacking of the knowledge of Hardware Description Language. The aim of this paper is to design and optimize a sobel filter system on Zynq based on HLS method.

2 The Sobel Edge Detection Algorithm

Sobel edge detection is a classical algorithm in the field of image and video processing for the extraction of object edges. Edge detection using sobel operators works on the premise of computing an estimate of the first derivative of an image to extract edge information. It takes advantage of the gradient

[a] Corresponding author: zhengyu@mails.bjut.edu.cn

differences of adjacent pixels between horizontal direction and vertical direction. And it judges whether the point is boundary point or not by comparing the gradient differences and the threshold.

Assuming that the function F(x,y) shows level of grey level of pixel in the original image, the function G(x,y) shows the grey level in the gradient image and θ is degree of elevation that shows the direction of the edge.The gradient can be expressed as :

$$G(x,y)=\frac{\partial F(x,y)}{\partial m}\cos\theta+\frac{\partial F(x,y)}{\partial n}\sin\theta \tag{1}$$

The magnitude form can be predefined as the sum of the magnitudes of vertical and horizontal gradient. In the partial derivatives of vertical and horizontal can be represented as Gx and Gy:

$$G(x,y)=G_x(x,y)\cos\theta+G_y(x,y)\sin\theta \tag{2}$$

The amplitude of gradient is:

$$G(x,y)=\sqrt{[G_x(x,y)]^2+[G_y(x,y)]^2} \tag{3}$$

Gx and Gy can be obtained by the convolution between regional template and image.

Figure.1 shows the 3*3 window of neighborhood of point(x,y), the horizontal gradient operator and vertical gradient operator.

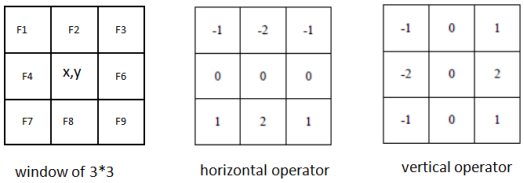

Figure 1. Sobel operator

The horizontal gradient and vertical gradient of F(x,y) is shown respectively:

$$G_x=(F_7+2F_8+F_9)-(F_1+2F_2+F_3) \tag{4}$$

$$G_y=(F_3+2F_6+F_9)-(F_1+2F_4+F_7) \tag{5}$$

After calculation of gradient for each pixel, threshold value t is selected. G(x,y) > t, is chosen to be the edge point.

3 The design of sobel IP core

The core of the system is sobel filtering, It's necessary to made sobel filtering IP core for the reason of reuse for other design and simplification hierarchical structure.

3.1 Memory architecture

When calculate the gradient of one pixel, the nine pixel points around it need to be accessible concurrently.

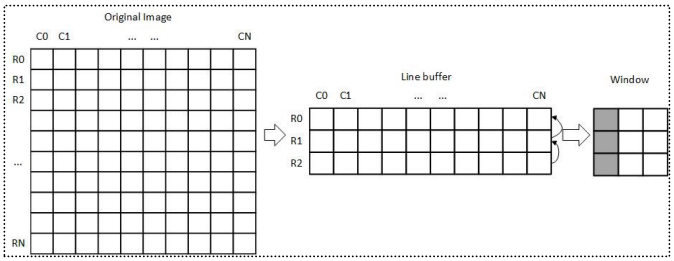

Figure 2. Original image pixels

As shown in Figure.2, when computing the output pixel of F, all the pixels around it need to be accessible.The line buffer is selected to meet concurrent multi-access requirement. To meet the 9-pixel-per-clock-cycle requirement, it is needed to add a memory window to the algorithm source code in addition to the line buffer. A memory window is a storage element implemented using the FF resources from the FPGA fabric. The purpose of this block is to store only the minimum number of pixels required for functional correctness and not to store the entire image. The overall data movement from input to computation through a tiered memory.The memory architecture is shown in Figure.3. At each clock cycle, the contents of the window shift left to make room for a new column from the line buffer. As Figure.4 shows the data reuse and distributed implementation of the window memory provides the nine memory operations required by the algorithm. No additional latency is introduced into the design by this memory. The window data movement operations occur concurrently with those of the line buffer. These memory structures can be implemented in C/C++ code as class window and class line buffer by using HLS library.

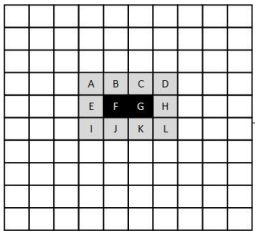

Figure 3. Memory architecture

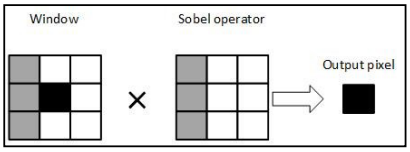

Figure.4. Calculation with sobel operator

3.2 Port-protocol

The design of the port protocol defines how the module starts and stops working and how to communicate with other modules. Xilinx provides the port protocol library and packed in ap_interfaces.h. This project uses ap_ctrl_hs block-level protocol and AXI4-Stream port-level protocol. The design is implemented in C/C++ and synthesized into RTL-level block in Vivado HLS. The IP core package diagram is shown in Figure.5

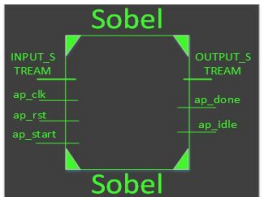

Figure 5. Sobel IP block

4 System design

4.1 Hardware system architecture

The processing result of sobel extraction is verified by a real-time video system. The original image is captured by the USB camera, after processing, the image is shown on the display device through the HDMI interface. The entire project is implemented on Zedboard, which is one of the Zynq family FPGA .It integrates a dual-core ARM Cortex based processing system (PS) and programmable logic (PL) in a single device. The hardware architecture is shown as Figure.6. An embedded linux runs on the PS and the camera acquisition control and memory data mapping control is responsible for the PS. PL is in charger of the implementation of sobel processing and control of display device. The block diagram in Figure.6 is implemented as IP core in the hardware design. Pixel date is first captured by a USB camera and is transmitted to from DDR to VDMA through memory map by AXI interconnect. Then the origin frame image is processed to extract the edge and the processed pixel date is stored in a VDMA again. A video display controller is then required to output the processed video. This project uses AXI4 Streaming protocol to communicate pixel data.

4.2 Software development

After finishing the hardware design, software can be developed based on it. The software design is divided into two main parts, embedded linux setup and the application software design.

In the context of a Zynq Soc runnning under Linux, applications based on Linux don't have access to the sobel IP in the FPGA logic. The kernel application program interfaces or system calls is used to make it possible to access to the hardware. These APIs or system calls interact with the Linux device driver to enable access to the hardware accelerator from the user application. Some files are needed to start an embedded linux on PS: u-boot.elf(Linux boot loader), bitstream (hardware configuration file), fsbl.elf (first stage boot loader), zImage(.linux kernal Image), devicetree. Dtb (device tree), embedded linux file systemAfter finishing the operating system transplantation, software of edge extraction can be developed based on the embedded linux. OpenCV library should be transplanted to control the USB camera. The flow chart of software is shown as Figure.7.

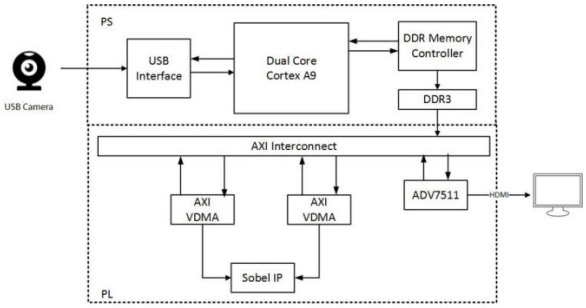

Figure 6. Hardware architecture

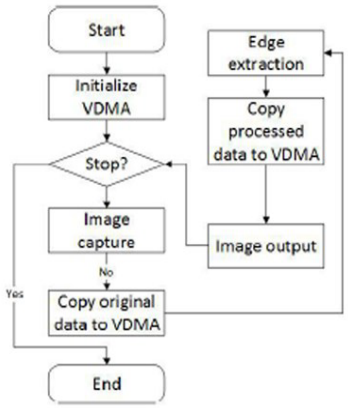

Figure 7. Software flow

5 Testing and results

A SD card with two partitions is needed to bring linux desktop. A FAT32 partition for Zynq Boot Images and an ext4 partition for root file system. Copy the generated Boot. Bin, zimage devicetree.dtb files to FAT32 partition and copy the file system files to ext4 partition. Start the linux and compile the software application and then test the system.

Different input images and their edge detected images are shown the Figure.8. The figures A, B are the input images and figures C,D are their edge detected results respectively.

Figure 8. Sobel edge extraction results

A comparative test is done to verify the hardware acceleration with software filtering on ARM. Two parameters are selected to measure the efficiency of the program. Latency represents the number of iterations of the image processing to one frame. Timing represents the processing time of one frame. The results are shown in Table 1.

Table 1 comparatinve test resulte

	Latency	Timing(s)
Software process	899	0.089134
Hardware process	129	0.001141

It's obvious that hardware process is more efficient than software process. Software processing time is almost 80 times than the hardware processing. The hardware acceleration has remarkable effect on sobel filtering. The reason of the reduction of latency is FPGA logic provides with parallel pipeline processing while CPU provides with serial processing.

Summary

The design of FPGA based sobel edge detection system is presented in this paper. High level synthesize method is used in the design of sobel IP core. The project is realized on ZYNQ FPGA kit. The original video is captured by a USB camera and processed video is shown on a HDMI display device. The FPGA supports high levels of parallel processing data flow structures that are important for efficient implementation of image processing algorithms. Through the software and hardware comparison test, it can be seen that the hardware processing time is reduced by nearly 80 times compared with the software, and it's necessary for a real-time system.

References

1. G. x. Yao, "Design of edge detection algorithm for image sobel based on FPGA," 2015 4th International Conference on Computer Science and Network Technology (ICCSNT), Harbin, 851-853.(2015)
2. J. Tian, J. Wu and G. Wang, "Realization of real-time sobel adaptive threshold edge detection system based on FPGA," Information and Automation, 2015 IEEE International Conference on, Lijiang, 2740-2743.(2015)
3. İ Koyuncu, Ö Çetin, F. Katırcıoğlu and M. Tuna, "Edge dedection application with FPGA based Sobel operator," 2015 23nd Signal Processing and Communications Applications Conference (SIU), Malatya, 1829-1832.(2015)
4. G. Chaple and R. D. Daruwala, "Design of Sobel operator based image edge detection algorithm on FPGA," Communications and Signal Processing (ICCSP), International Conference on, Melmaruvathur, 2014, 788-792 (2014)
5. Vanishree and K. V. Ramana Reddy, "Implementation of pipelined sobel edge detection algorithm on FPGA for High speed applications," Emerging Trends in Communication, Control, Signal Processing & Computing Applications (C2SPCA), 2013 International Conference on, Bangalore,1-5 (2013)