3rd International Conference on Automation, Control, Engineering and Computer Science (ACECS'16)
Proceedings of Engineering & Technology (PET)
pp. 41-45

Implementation of real time edge detection system for HD video on Zynq

Haythem AMEUR[1], Abdelhamid HELALI[1], Hassen MAAREF[1], Anis YOUSSEF[2]
[1] *Laboratory of Micro-Optoelectronic and Nanostructure, University of Monastir Tunisia, Monastir*
[2] *TELNET Innovation Labs Tunisia, Tunis*
ameur_haythem@yahoo.com

***Abstract*— Several image processing applications (edge detection, feature extraction, tracking, pattern recognition …) are so computationally and software solutions cannot meet the real-time requirements [1, 2]. In this work, we highlight a new and simple process to implement a real time edge detection enhancement algorithm on heterogeneous platforms. The implemented design extracts discontinuities in intensity from an HD video (1080x1920 pixels) at 60 fps. The architecture was designed in Matlab Simulink. The hardware IP core was generated with HDL coder, then packaged and implemented in an embedded platform (Zc702) with Xilinx Vivado environment. The experimental results show an accurate edge detection that satisfy the requirement of the real-time constraints.**

***Keywords*—Real time; Edge detection; HDL coder; Zynq**

I. INTRODUCTION

Edge detection is one of the most important elements in image analysis. It has a significant role in many applications and gives information as a precursor step particularly for segmentation, feature extraction and object recognition. Usually edge detection operators are Roberts, Sobel, Laplacian, Prewitt, Canny, etc [3]. Such operators detect image edge by calculating the gradient in several directions and they require high computation power. Therefore, to solve this problem, hardware implementation is essential which offers much greater speed than software implementation [4]. With new high-resolution standards, edge detection operators must receive input data at a rate up to 3 Giga samples per second. For example, the HD video standard (1080x1920 p) requires 60 fps, with approximately 2.1 megapixel per frame, and 126 megapixel per second resulting a data rate of 3 Gigabit per second using uncompressed RGB encoding [5].

On the other hand, Field Programmable Gate Array (FPGA) are becoming more and more attractive for image processing applications, due to its potential to have parallel and high computational density as compared to a general purpose microprocessor. In our application, a single FPGA with an embedded soft processor (ARM Cortex-A9) can deliver the requisite level of computing power. It can be made with larger flexibility and simplified board complexity. In fact, the generated IP core is designed to be connected to the ARM processor on a Zynq device through the AXI ports (see Fig 1).

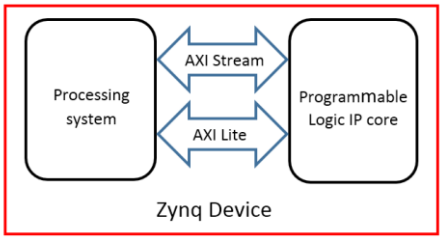

Fig. 1 Zynq architecture

It allows the user to change the threshold value, or to enable and disable the edge detection operator in real time.

The rest of the paper is organized as follows. In Section II we present the architecture of the edge detection unit. Experimental results and hardware implementation are addressed in section III. Section V Concludes this work.

II. EDGE DETECTION UNIT ARCHITECTURES

1. *Derivative mask*

According to the experimentation in [6, 7] the use of a simple derivative mask centered [-1, 0, 1] turns out to be the best results for the histograms of oriented gradients (HOG) descriptor. Since, so far, we are interested in implementing the HOG descriptor, we will present in this paper a kind of parallel processing of a simple derivative masks in the horizontal and vertical directions. Equations are presented below:

$$dx(x,y) = [-1,\ 0,\ 1] * I(x,y) \quad (1)$$

$$dy(x,y) = [-1,\ 0,\ 1]^{T} * I(x,y) \quad (2)$$

$$M(x,y) = |dx(x,y)| + |dy(x,y)| \quad (3)$$

Where, dx(x,y), dy(x,y) and M(x,y) represent respectively the horizontal gradient, the vertical gradient and the magnitude of pixel.

This process can quickly get the result of one pixel in only one clock period. In addition, it does not only meet the real-time requirements, but also gives the programmer the possibility to implement some other serial images processing like the magnitudes computation, gradients orientations computation, histograms distributions…

The edge detector IP core was established in Matlab Simulink environment. This tool allows the user to drive the

ISSN: 2356-5608

Zynq programmable logic (PL) part at a high level without having to deal with low level hardware details. The model is designed to detect edges by using 1-D image filters in the vertical and horizontal orientations. The proposed architecture is shown in *Fig 2*.

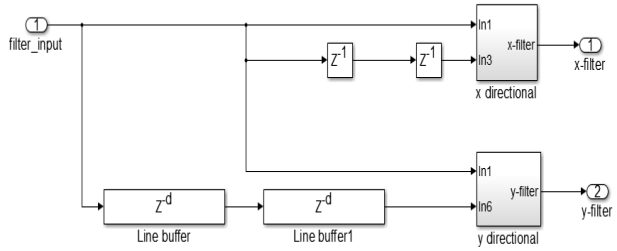

Fig. 2 1-D image derivative masks

The outputs x-filter and y-filter represent respectively the horizontal and vertical gradients. The two filters are separable and can be decomposed, which makes its hardware implementation efficient. The incoming pixels are shifted through two line buffers (1920 pixels each line) for vertical direction and three sample buffers for the horizontal direction. This mechanism creates several delays that feed the filter array simultaneously with the relevant pixels from the video stream.

2. *Local Minimum search*

Known the high resolution used in HD standard, one edge presented in the image can be coded on successive dozen of pixels (see Fig. 3). Consequently, classical edge detector will result continuities in strong edges that will dominate the visual appearance. To resolve this problem, we proposed to search a local minimum gradient for each four successive pixels in both x and y directions just before the magnitude's calculation step. *Fig. 3* presents the architecture of our added block.

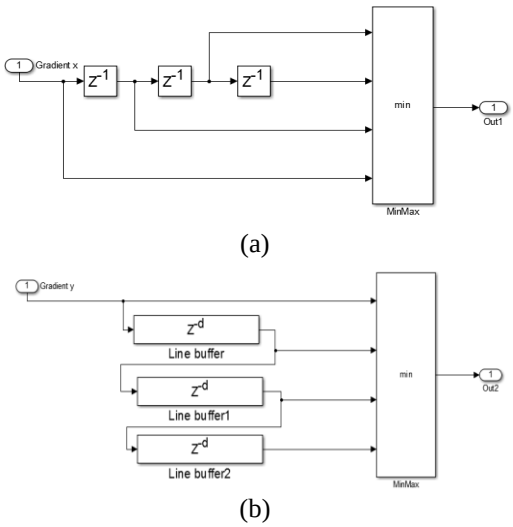

Fig. 3 Local minimum search: (a) Local minimum search in horizontal direction, (b) Local minimum search in vertical direction

This process alleviates problems associated with edge continuities by identifying strong edges, and preserving the relevant weak edges. It can be also delete the higher noise distributed in random positions in mage. Results are showed in *Fig 5*. Performances being compared the traditional edge detection algorithm and it was observed that the outputs of this algorithm provide much more distinct marked edges and reduced noise, thus it have better visual appearance.

3. *Overall system*

The overall system of the edge detector is composed of three steps; Gradients calculation, magnitude calculation & local minimum search and binary segmentation based on flexible threshold that can be defined by the user. The structure of the overall system is presented in the following figure (*Fig* 4).

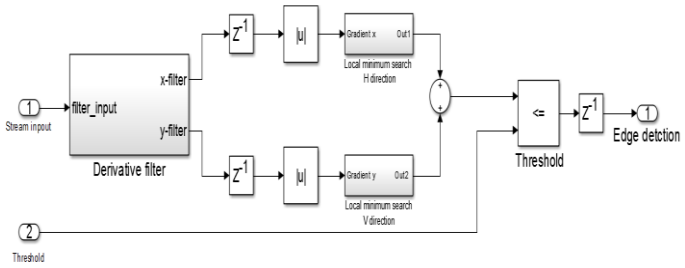

Fig. 4 Overall system of the edge detector

Once vertical and horizontal gradient values are determined, each pixel gradient is compared to the value of their four neighboring gradients. Pixel with gradient intensity less than their neighboring will automatically replace all others, in order to keep the edges thin and suppress the higher noise if it is present. The final stage consists in thresholding the addition between the two local minimum search stages and create a binary image.

III. HARDWARE IMPLEMENTATION

In this section, we describe the hardware implementation of our proposed distributed edge detection algorithm on the Xilinx zynq device. Based on HDL work flow advisor, we have chosen the AXI interfaces connecting the IP core to the embedded processor ARM 9. Then we have generated HDL codes from the Simulink blocks. Tow interfaces are used: the AXI stream interface to handle the input output stream pixels and AXI lite interface to adjust the parameters of the edge detector IP core.

Therefore, with Vivado IP integrator [8], we have packaged and integrated the generated IP core. To guarantee the acquisition and the display of pixels from and to the external word, two IP cores are added. They are respectively; FMC IMAGEON HDMI input module and FMC IMAGEON HDMI output module [9].

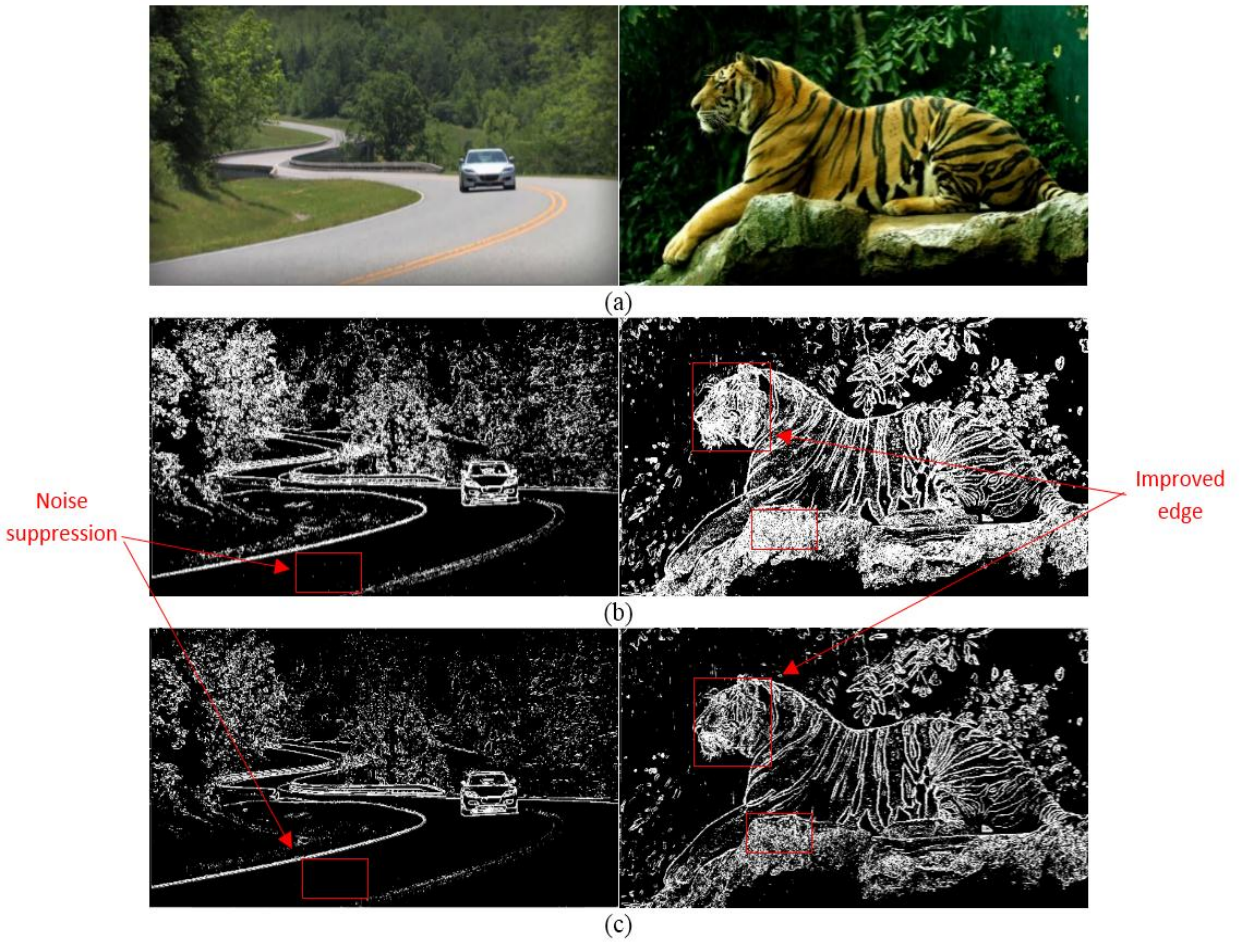

Fig. 5 Experimental results: (a) Original image, (b) Edge detection by traditional algorithm, (c) Edge detection by improved algorithm

In Fig. 6 we show the experimental result of our real time edge detection system. An HD video delivered through the integrated HDMI port from laptop to the Zynq Zc702 platform. After all processing steps, the output video is displayed to the LCD monitor.

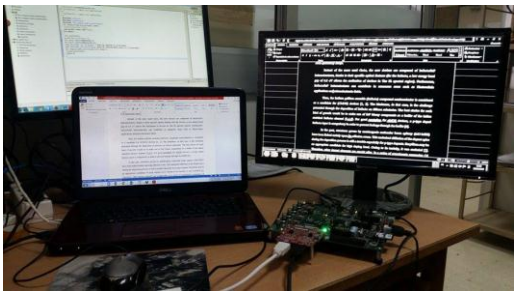

Fig. 6 Experimental result

The device utilization summary of the whole system and the edge detector IP core are given respectively in Table. I and table II. Small resource is taken up, so there is a possibility to implement more processes with this architecture on the same device. The system clock frequency is 150 MHz. The whole system of the edge detection can be extracted in less than 5 ns for a single pixel. Therefore, the implemented system requires (1920*1280*5) 12.288 ms to process one single HD frame and (12.288*60) 0.73728 s to process 60 frames. The result indicates that our implemented system can handle 80 frame/s if a camera with higher frame rate is available.

TABLE I . RESOURCE UTILIZATION OF THE WHOLE SYSTEM

Resource	***Utilization***	***Available***	***Utilization%***
FF	6075	106400	6
LUT	4878	53200	9
Memory LUT	388	17400	2
I/O	39	200	20
BRAM	5	140	4
DSP48	18	220	8
BUFG	3	32	9

TABLE II. RESOURCE UTILIZATION OF THE EDGE DETECTOR IP CORE

Resource	***Utilization***	***Available***	***Utilization%***
FF	764	106400	1
LUT	579	53200	1
Memory LUT	32	17400	2
BRAM	3	140	3

Resource	***Utilization***	***Available***	***Utilization%***
DSP48	13	220	1
BUFG	0	32	0

IV. CONCLUSION

This paper discussed the implementation of the edge-detection algorithm on a heterogeneous platform. Performance analysis indicate a significant speedup and show the capability of supporting adaptive edge detection for real-time image processing. Future work will involve more complex image processing algorithms into the Zynq device.

ACKNOWLEDGMENT

This work was supported by the Project of Support System for Research and Innovation (PASRI), Tunisia.

REFERENCES

[1] Donald. L. Hung, Heng-Da Cheng, and Savang Sengkhamyong, "Design of a Hardware Accelerator for Real-Time Moment Computation: A Wavefront Array Approach ", IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 46, NO. 1, pp. 207 - 218, 1999.

[2] Xin Yuanfang and Sun Xia, "System Design for Real-time Image Processing Based on Multi-core DSP", JOURNAL OF NETWORKS, VOL. 9, NO. 11, pp 3143 – 3149, 2014.

[3] "Effects of noise on various edge detection techniques", International Conference on Computing for Sustainable Global Development (INDIACom), pp 827 - 830, 2015.

[4] Mahesh Prasanna, Shantharama Rai, "Image Processing Algorithms - A Comprehensive Study" International Journal of Advanced Computer Research Volume - 2 Issue 15 June –pp 2277 – 7970, 2014.

[5] https://en.wikipedia.org/wiki/1080p

[6] N. Dalal and B. Triggs "Histograms of Oriented Gradients for Human Detection," Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. II, pp.886-893, 2005.

[7] Haythem AMEUR, Abdelhamid HELALI, Mohsen NASRI, Hassen MAAREF, "Improved feature extraction method based on Histogram of Oriented Gradients for pedestrian detection", Computer & Information Technology (GSCIT), 1 – 5, 2014.

[8] http://www.xilinx.com/products/design-tools/vivado/integration.html

[9] http://www.em.avnet.com/en-us/design/drc/Pages/HDMI-Input-Output-FMC-module.aspx