Sobel Edge Detector Implementation using NIOS II Soft-Processor and Cyclone III FPGA

Derra Mourad[1], Amghar Abdellah[2], Moutakki Zakaria[3], Ouloul Mohamed Imad[4]

[1,2,3,4]Laboratory of Metrology and Information Treatment, College of Science, Ibn Zohr University, BP 8106, 80000, Agadir, Morocco
mourad.derra@edu.uiz.ac.ma

ABSTRACT

This paper presents the implementation of Sobel edge detector which is based on the NIOS II soft-core embedded processor. This image processing system is implemented on the Cyclone III EP3C120F780 FPGA (Field Programmable Gate Array) using the features of Quartus II SoPC (System on a Programmable Chip) builder. The results are displayed in a graphic LCD interface.

KEYWORDS

Cyclone III EP3C120F780 FPGA, Image Processing, Sobel Edge Detection, NIOS II Soft-Processor, SoPC Builder.

1. INTRODUCTION

Digital image processing, in general, is the realization of various analysis methods and processes aimed at using pictorial information. Today, there are many field studies about digital image processing applications. These work areas include medical sciences [1-2], measurement and instrumentation applications [3-4], cryptography sciences [5], earth sciences [6], control [7], food industry, optimization, and artificial neural networks.

Image processing is the process of creating a processed digital image as a result of analysis and use of everyday analog images. The preliminary stage, which aims to make a picture more understandable; discrimination by visual subdivision, identification of subdivisions in the used property direction, labeling of defined objects, and finally decision making and interpretation by looking at the objects to be tagged.

Edge detection techniques are used at the stage of image segmentation. The boundaries of the objects in the image are the edges of the reflection and illumination changes within the objects. Pixel neighbors have an important place at the edges. It is possible to combine under four headers, including edge detection methods implemented for edge detection, slope edge detectors, zero crossing (secondary derivative detector), Gaussian edge detectors (Canny) and vector sequencing statistics.

In the Sobel operator, which is one of the slope edge detectors, the horizontal and vertical derivatives of the view are calculated with 3×3 kernels. With these results, the slope values of each point in the image are collected [8].

In the secondary derivative detectors, edge information about the image can be obtained by performing zero crossing in the second derivative after the first derivative peak is determined [9].

The Canny application is a method of using step edges separated by Gaussian. This application is an edge detector and noise-reducing [10].

Field Programmable Gate Array (FPGAs) are a programmable logic circuit designed to be programmed or more precisely to be configured to emulate any circuit or digital system. FPGAs provide a cheaper solution and faster system design compared to Application Specific Integrated Circuits (ASICs) that require significant resources in terms of both charges and time. The configuration of FPGAs takes little time, and their prices are around a few hundred to a few thousand dollars. The major advantage of FPGAs is their flexibility, speed, and high consumption compared to ASICs. As a result, FPGA chips are preferred for image processing applications where high

processing power and performance are required due to their fast and low cost; reprogram ability and high-frequency operation [11].

2. RELATED WORK

Various studies have been done with FPGA-based Sobel operator in the literature. Anusha et al (2012), implemented 3D Sobel operator on FPGA [12]. In another work done by Mehra et al (2012), a 2-D FPGA-based Sobel operator application was presented and the maximum operating frequency of the system was obtained at 148 MHz [13]. In Kumar et al. work (2013), 2-dimensional Sobel operator implementation is implemented on FPGA-based systems [14]. Chaple et al (2014), presented a design of a Sobel edge detection algorithm to find edge pixels in gray scale image. Xilinx ISE Design Suite-14 software platform is used to design an algorithm using VHDL language [15]. Koyuncu et al (2015), proposed an edge detection algorithm using Sobel operator based on FPGA architecture. This system is designed using IEEE 754-1985 floating-point standard and VHDL hardware description language. The design is synthesized for Xilinx Virtex-6 FPGA chip with 160 MHz operating frequency [16]. Ben Amara et al (2016), presented an HD video streaming architecture and a Sobel edge detection IP core design and implementation using a high-level synthesis workflow [17]. The aim of the Yu Zheng (2017) work is to design a real-time Sobel extract system based on HLS method and to implement it on ZYNQ FPGA [18]. It is in this same niche that the focus of our research work is. It is a continuation of the research work mentioned above. In this work, we implemented the Sobel edge detector using NIOS II soft-core processor and Cyclone III FPGA.
The rest of the paper is organized as follows: the second section will present the Sobel edge detection operator. The third section will describe the design of the edge-detection process using the FPGA-based Sobel operator. The last section will discuss and interpret the obtained results.

3. THE SOBEL EDGE DETECTION ALGORITHM

One of the applications used for the analysis and use of digital image data is edge detection, which can be performed using many different operators. The Sobel operator, a basic edge detection operator, reveals the rate of color change in the image and is highly preferred because of its less distortion at high noise levels. In other words, this operator specifies the slopes of each point in the image. Mathematically, the gradient of a bivariate function are the derivatives of horizontal and vertical directions of each image point shown in two dimensions. Here, two alternating functions are shown as the density function of the view. The density function of the image $f(x, y)$ is defined as (x, y) coordinates in slope (1), it is defined as two-dimensional vectors.

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad \textbf{(1)}$$

The size of this vector is calculated using equations (2) and (3):

$$\nabla f = mag(\nabla f) \quad \textbf{(2)}$$

$$|\nabla f| = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right] \quad \textbf{(3)}$$

While the components of the slope vector are linear operators, the value of this vector is not linear since it is the process of taking the square root. In general, it is expressed as the sum of the absolute values of slope magnitudes, such as in Equation (4).

$$\nabla f \approx |G_x| + |G_y| \quad \textbf{(4)}$$

In Fig. 1, the image points are given a notation shown in a 3×3 mask to calculate the slope of the pixel value given as Z_5.

Z_1	**Z_2**	**Z_3**
Z_4	**Z_5**	**Z_6**
Z_7	**Z_8**	**Z_9**

Figure 1: 3x3 mask display

Moreover, the first derivative of a one-dimensional function $f(x)$ is expressed as in (5):

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \quad \textbf{(5)}$$

The gradient components G_x and G_y are given by equations (6) and (7):

$$G_x = (Z_8 - Z_5) \quad \textbf{(6)}$$

$$G_y = (Z_6 - Z_5) \quad \textbf{(7)}$$

The 3×3 masks in the operator are used to approximate the horizontal and vertical derivatives of the image. Equation (8) and (9) are obtained by considering A as the source image, and the horizontal and vertical derivatives at each point of the image are defined by G_x and G_y.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \quad \textbf{(8)}$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad \textbf{(9)}$$

In Fig. 2 horizontal and vertical weights of the Sobel mask are given.

-1	0	+1
-2	0	+2
-1	0	+1

(a)

+1	+2	+1
0	0	0
-1	-2	-1

(b)

Figure 2: Sobel mask (a) horizontal and (b) vertical weight display

Here, the magnitude G of the slope value of each point in the image is calculated by using the vertical derivative values G_x and G_y given in equation (10):

$$G = \sqrt{\left(G_x^2 + G_y^2\right)} \qquad \textbf{(10)}$$

4. CYCLONE III DEVELOPMENT BOARD

The Cyclone III development board [19] provides a hardware platform for developing and prototyping high-performance digital circuits that consume less space and consume minimal power. The card offers a large number of peripherals and memory interfaces as shown in the following figure (Fig. 3).

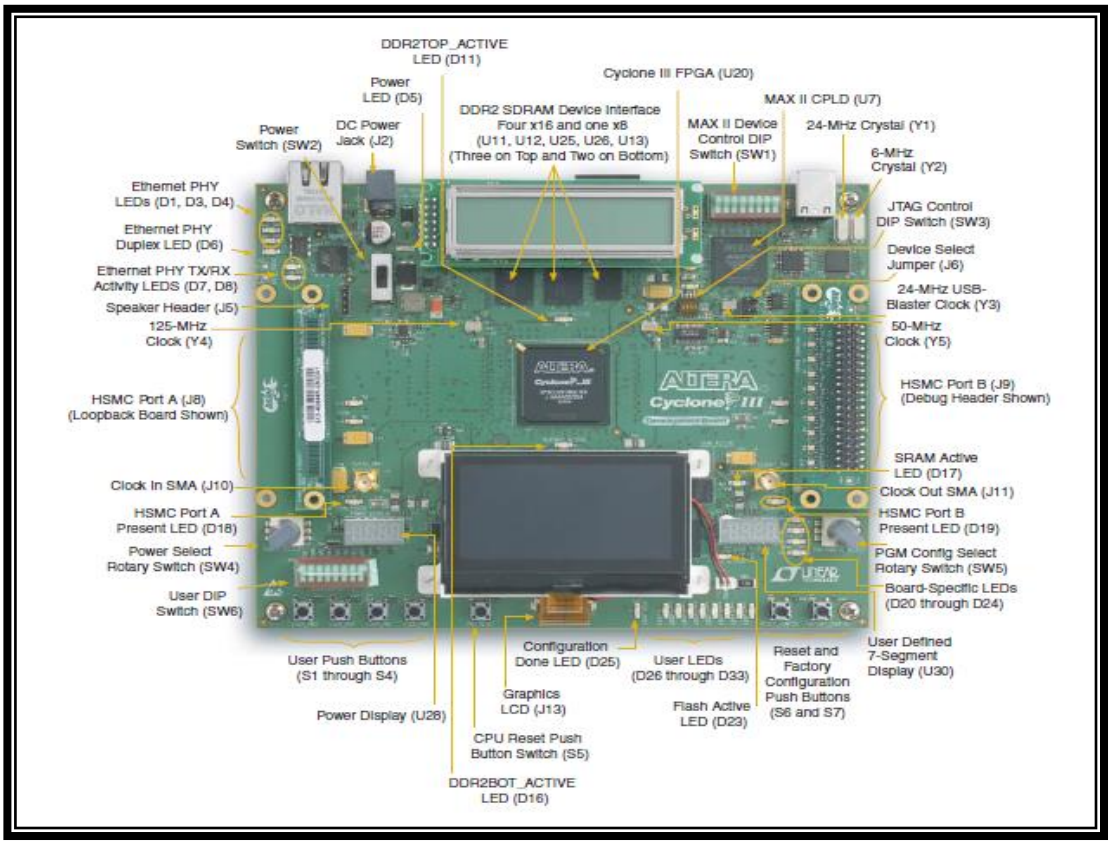


Figure 3: Structure of the card development Cyclone III

4.1. Characteristics of the development board

A. Cyclone III development board

- Cyclone III EP3C120F780 FPGA
- Embedded USB-Blaster™ circuitry (includes an MAX® II CPLD) allowing download of FPGA configuration files via the flash device or the host computer.

B. Memory

- 256 MB of dual-channel DDR2 SDRAM with error correction code
- 8 MB of synchronous SRAM
- 64 MB of flash memory

C. Communication ports

- 10/100/1000 Ethernet
- USB 2.0

D. Power and analog devices from Linear Technology

- Switching power supply LTM4601
- Switching and step-down regulators LT1931, LT3481, and LTC3418
- Analog-to-digital converter LTC2402
- LDO regulators LT1963 and LT1761

E. Clocking

- 50-MHz and 125-MHz on-board oscillators
- SMA inputs/outputs
- I/Os for the two HSMC connectors
- Various buttons, switches, and indicators

F. Display

- 128 x 64 graphics LCD
- 2-line x 16-character LCD

G. Connectors

- Two HSMCs
- USB type B

H. Debug tools

- Three HSMC debug cards (two loop-backs and a debug header)

I. Cables and power/analog

- 14-V to 20-V DC input
- On-board power measurement circuitry
- 19.8 W per HSMC interface

- Power cord with plug adapters (U.S., UK, and EU)

5. CYCLONE III FPGA IMPLEMENTATION OF THE NIOS II PROCESSOR-BASED SYSTEM

5.1. Proposed system

To better optimize our system we choose this architecture in which the NIOS II [20] processor communicates with SDRAM and graphics LCD via the IP of PIO: parallel input-output (Fig. 4).

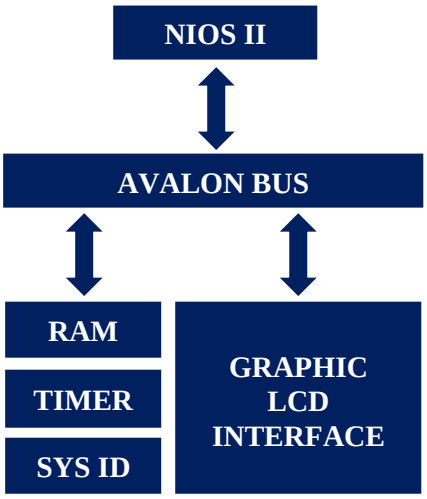


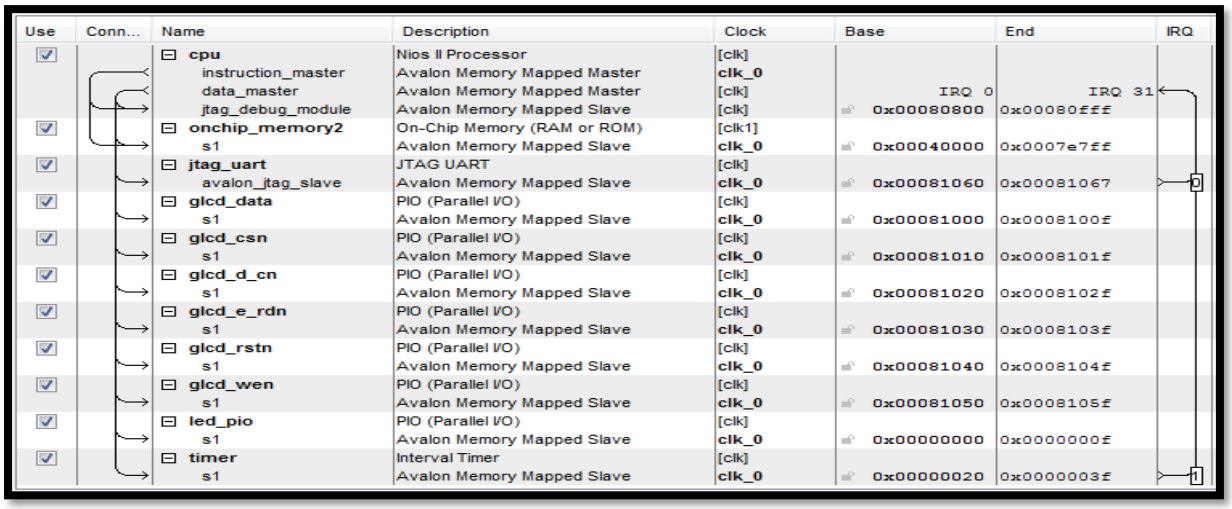
Figure 4: NIOS system

The image information is saved in the 8MB SRAM by the CMOS entities and the output of our system provides the processed image to the graphic LCD interface.

5.2 System creation on Quartus II and SoPC builder

In order to satisfy our vision, we use SoPC builder to create the NIOS processor and their peripheral as shown in Fig. 5.

After generating the IPs, we are interested in establishing the connection between the different components of the system and the real PINs of pus cyclone III using the map guide (Fig. 6).



Use	Conn...	Name	Description	Clock	Base	End	IRQ
☑		⊟ cpu	Nios II Processor	[clk]			
		instruction_master	Avalon Memory Mapped Master	clk_0			
		data_master	Avalon Memory Mapped Master	[clk]	IRQ 0	IRQ 31	
		jtag_debug_module	Avalon Memory Mapped Slave	[clk]	0x00080800	0x00080fff	
☑		⊟ onchip_memory2	On-Chip Memory (RAM or ROM)	[clk1]			
		s1	Avalon Memory Mapped Slave	clk_0	0x00040000	0x0007e7ff	
☑		⊟ jtag_uart	JTAG UART	[clk]			
		avalon_jtag_slave	Avalon Memory Mapped Slave	clk_0	0x00081060	0x00081067	0
☑		⊟ glcd_data	PIO (Parallel I/O)	[clk]			
		s1	Avalon Memory Mapped Slave	clk_0	0x00081000	0x0008100f	
☑		⊟ glcd_csn	PIO (Parallel I/O)	[clk]			
		s1	Avalon Memory Mapped Slave	clk_0	0x00081010	0x0008101f	
☑		⊟ glcd_d_cn	PIO (Parallel I/O)	[clk]			
		s1	Avalon Memory Mapped Slave	clk_0	0x00081020	0x0008102f	
☑		⊟ glcd_e_rdn	PIO (Parallel I/O)	[clk]			
		s1	Avalon Memory Mapped Slave	clk_0	0x00081030	0x0008103f	
☑		⊟ glcd_rstn	PIO (Parallel I/O)	[clk]			
		s1	Avalon Memory Mapped Slave	clk_0	0x00081040	0x0008104f	
☑		⊟ glcd_wen	PIO (Parallel I/O)	[clk]			
		s1	Avalon Memory Mapped Slave	clk_0	0x00081050	0x0008105f	
☑		⊟ led_pio	PIO (Parallel I/O)	[clk]			
		s1	Avalon Memory Mapped Slave	clk_0	0x00000000	0x0000000f	
☑		⊟ timer	Interval Timer	[clk]			
		s1	Avalon Memory Mapped Slave	clk_0	0x00000020	0x0000003f	1

Figure 5: NIOS processor and their peripherals

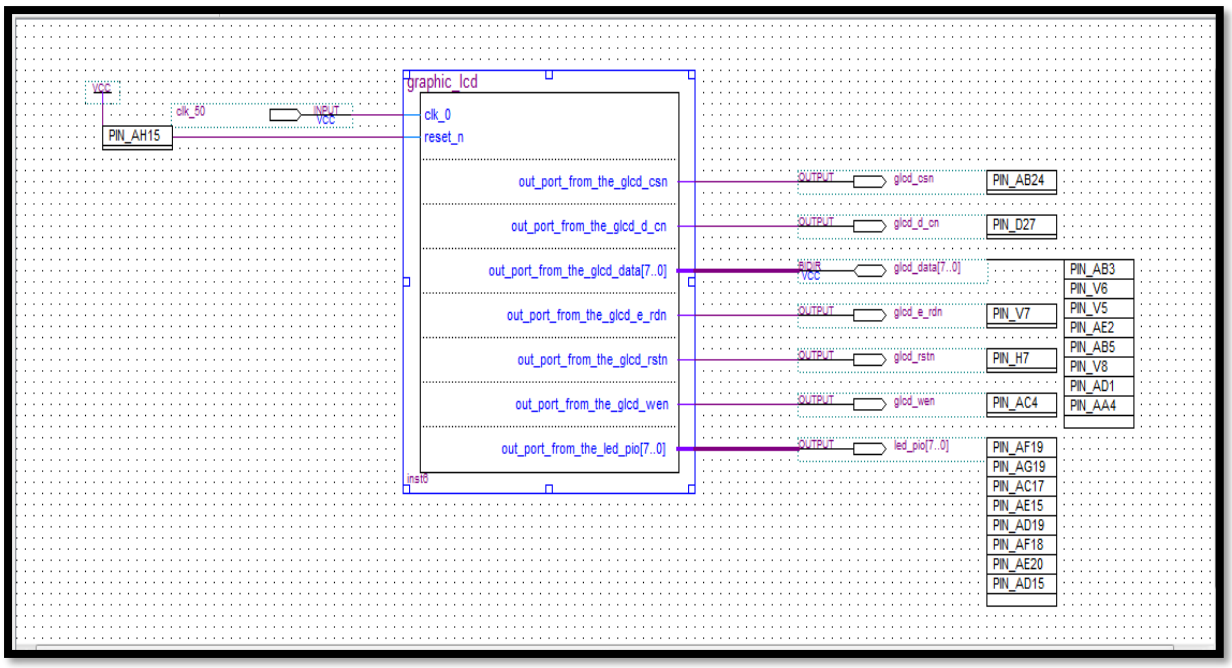


Figure 6: Block diagram of our system

5.3 Results and discussions

As it was mentioned above the total operation of the proposed system is controlled by the NIOS II soft-processor. The program which is executed by the NIOS II soft-processor is developed in C/C++ using the features of the NIOS II SBT (Software Build Tools) for Eclipse IDE.
The results of image and Sobel edge detection are displayed in the graphics LCD as it is shown in the Fig. 7.

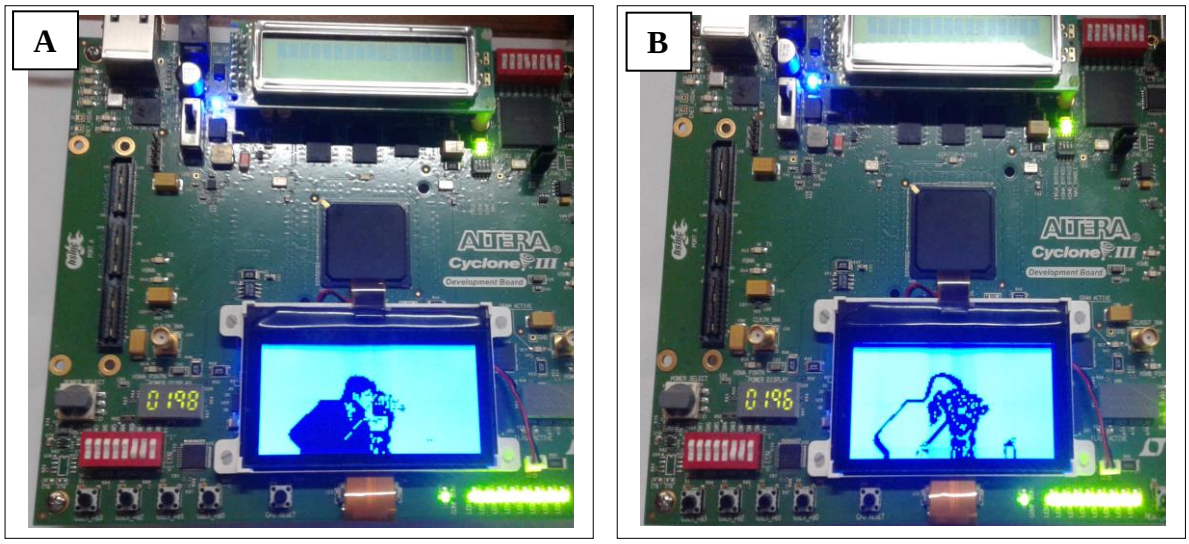


Figure 7:(A) The original image; (B) The image edges detected using Sobel detector

We use 5 images obtained from the standard database to test our proposed work, these images are of size (128×64) presented in Fig. 8.

Figure 8: Standard Dataset

The results of this standard dataset are displayed in the graphics LCD as it is shown in Fig. 9.

Figure 9: The first column represents the original images, the second column represents the original images displayed in the graphic LCD, the third column represents the image edges detected using Sobel detector displayed in the graphic LCD based on our system

6. CONCLUSIONS

This article presents an image processing system based on the NIOS II soft-core processor which is implemented on the Cyclone III FPGA. The user is able to observe the processed image system through the graphic LCD interface. The main feature of the work presented in this paper is to implement OpenCv library on Nios II through μClinux.

REFERENCES

[1] Prakash, J., Dehghani, H., Pogue, B. W., and Yalavarthy, P. K., "Model-Resolution-Based Basis Pursuit Deconvolution Improves Diffuse Optical Tomographic Imaging", IEEE Trans. on Medical Imaging, vol. 33, pp. 891–901, April 2014.

[2] Hao, F., Guoping, Q., Jie, S., and Ilyas, M., "A Novel Polar Space Random Field Model for the Detection of Glandular Structures", IEEE Trans. on Medical Imaging, vol. 33, no. 3, pp. 764–776, 2014.

[3] Rana, K. B., Agrawal, G. D., Mathur, J., and Puli, U., "Measurement of void fraction in flow boiling of ZnO water nanofluids using image processing technique", J. of Nuclear Eng. and Design, vol. 270, pp. 217–226, 2014.

[4] Li, L., Gong, M., Chui, Y. H., and Schneider, M., "A MATLAB-based image processing algorithm for analyzing cupping profiles of two-layer laminated wood products", Measurement, vol. 53, pp. 234–239, 2014.

[5] Barakat, M. L., Mansingka, A. S., Radwan, A. G., and Salama, K. N., "Hardware stream cipher with controllable chaos generator for color image encryption", Image Proc., vol. 8, pp. 33–43, 2014.

[6] Lu, W., Lifan Z., Guoan B., Chunru, W., and Lei, Y., "Enhanced ISAR Imaging by Exploiting the Continuity of the Target Scene", IEEE Trans. on Geoscience and Remote Sensing, vol. 52, no. 9, pp. 5736–5750, September 2014.

[7] Xiao, Y., and Dong, S., "Multilevel-Based Topology Design and Cell Patterning With Robotically Controlled Optical Tweezers", IEEE Trans. on Control Syst. Techn., vol. 23, pp. 176–185, 2015.

[8] Yuan-Hui, Y., and Chang, C. C., "A new edge detection approach based on image context analysis", Image and Vision Computing, vol. 24, pp. 1090–1102, 2006.

[9] Gonzales, R. C., and Woods, R. E., Digital Image Processing. New Jersey: Prentice-Hall, 2002.

[10] Canny, J. F., "A computational approach to edge detection", IEEE Trans. on Pattern Anal. Mach. Intell., vol. 6, pp. 679–698, 1986.

[11] Koyuncu, I., Özcerit, A. T., and Pehlivan, I., "Implementation of FPGA-based Real Time Novel Chaotic Oscillator", Nonlinear Dyn., vol.75, no. 1–2, pp. 49–59, 2014.

[12] Anusha, G., Prasad, T. J., and Narayana, D. S., "Implementation of Sobel edge detection on FPGA." Int. J. of Computer Trends and Techn., vol. 3, pp. 472-475, 2012.

[13] Mehra, R., and Rupinder, V., "Area Efficient FPGA Implementation of Sobel Edge Detector for Image Processing Applications", Int. J. of Comp. Appl., vol. 5, pp. 7-11, 2012.

[14] Kumar, S., and Prabat, P., "FPGA Implementation of Image Segmentation By Using Edge Detection Based On Sobel Edge Operator", Int. J. of Research in Eng. and Tech., vol. 2, pp. 198-203, 2013.

[15] Chaple, G., and Daruwala, R. D., "Design of Sobel operator based image edge detection algorithm on FPGA", International Conference on Communications and Signal Processing (ICCSP), Melmaruvathur, pp. 788-792, 2014.

[16] Koyuncu, İ., Çetin, Ö., Katırcıoğlu, F., and Tuna, M., "Edge dedection application with FPGA based Sobel operator ", 23nd Signal Processing and Communications Applications Conference (SIU), Malatya, pp. 1829-1832, 2015.

[17] Ben Amara, A., Pissaloux, E., and Atri, M., "Sobel edge detection system design and integration on an FPGA based HD video streaming architecture", 2016 11th International Design & Test Symposium (IDT), pp. 160-164, 2016.

[18] Zheng, Y., "The Design of Sobel Edge Extraction System on FPGA", ITM Web of Conferences, DOI: 10.1051/itmconf/2017110, pp. 1-6, 2017.

[19] Cyclone III Development Board Reference Manual - Altera, *www.altera.com*

[20] Nios II Processor Reference Handbook, *www.altera.co.jp*

AUTHORS BIOGRAPHY

Derra Mourad is a Professor of secondary education of physics in Agadir, Morocco. He was born in El Jadida, Morocco, in 09/04/1988, he got the master thesis Instrumentation and Telecommunication in 2012 from the University of Ibn Zohr, Agadir, Morocco. In November 2017, he has Ph.D degree in Instrumentation, Metrology and Information processing from Department of Physics, Faculty of Science, University Ibn Zohr ,Morocco. His areas of research interests include Ultrasonic and embedded systems.

Amghar Abdellah is a Professor in the Physics Department, Faculty of Science, University Ibn Zohr ,Morocco. He received his DEA and DES degree in 1994 from Department of Physics , Faculty of Science, University Hassan II , Morocco. In January 2002, he has Ph.D degree in microelectronic from Department of Physics, Faculty of Science, University Ibn Zohr ,Morocco. His areas of research interests include Cryptography, DNT, embedded systems and microelectronic.

Moutakki Zakaria was born in casablanca, Morocco, in 25/07/1988, he got the master thesis Instrumentation and Telecommunication in 2011 from the University of Ibn Zohr, Agadir, Morocco. Since October 2011, he prepared his Ph. D thesis on computer vision and embedded systems. His main interests are the video surveillance systems applied on road safety, traffic management and the embedded systems using FPGA boards.

Imad, Ouloul Mohamed was born in Taza, Morocco, in 30/07/1989, he got the Master thesis Instrumentation and Telecommunication in 2013 from the University of Ibn Zohr, Agadir, Morocco. Since November 2013, he prepared his Ph.D thesis on computer vision and embedded systems. His main scientific interests are face detection/recognition and complex systems based on FPGA board.