Laplacian Edge Detection Algorithm for Road Signal Images and FPGA Implementation

Issam Bouganssa, Mohamed Sbihi, and Mounia Zaim

Abstract—**The applications of image processing for road safety, detecting panels and roadway have attracted considerable attention in literature and research, especially in the field of information processing on embedded systems. However, the demanding nature of image processing algorithms conveys a substantial burden for any conventional real-time implementation. Meanwhile, the emergence of reconfigurable architectures, especially FPGA chips, which have been given many facilities for rapid prototyping, where an image processing algorithm, can be designed, tested, and synthesized in a relatively short period of time compared to conventional or traditional approaches. This paper studies a hardware and software combination to obtain an optimal solution for the edge detection, this step is considered essential for the detection of panels and roadways, the dedicated algorithm is based on the Laplacian calculation for Edges detecting and implemented in a Xilinx Spartan 6 FPGA, and the results are displayed by standing in a VGA monitor, with a sync and display controller.**

Index Terms—**Edge detection, Laplacian, real time, FPGA.**

I. Introduction

An outline is materialized by a sudden change in intensity, and the goal of edge detection is to produce something like a line drawing of an image. In the application of edge detection algorithms we will look for places in the image where the intensity changes rapidly. In general, object boundaries tend to produce sudden changes in the intensity of the image.

For example, different objects are usually different colors or hues and this causes the image intensity to change as we move from one object to another. In addition, different surfaces of an object receive different amounts of light, which again produces intensity changes. Thus, much of the geometric information that would be conveyed in a line drawing is captured by the intensity changes in an image. Unfortunately, there are also a large number of intensity changes that are not due to geometry, such as surface markings, texture, and specular reflections. Moreover there are sometimes surface boundaries that do not produce very strong intensity changes. Therefore the intensity boundary information that we extract from an image will tend to indicate object boundaries, but not always. The detected edges are materialized by the intensity break down in the image in a given direction. Several methods exist to detect the break. Some of them are more or less complex.

In this work, for the safety and processing of road images, the contour detection step, which is essential for characterizing road signs, proposes methods based on the variation of the intensity in the time domain by calculating the Laplacian, which will be followed by a judicious thresholding, the latter allows to isolate the contours of the rest of the image [1].

The proposed algorithms are implemented in real-time on a Xilinx Spartan-6 FPGA programmable circuit, which provides the performance needed for processing real-time image sequences, while maintaining the flexibility of the system to support an adaptive algorithm.

We start our work with the characteristics and the principle of Edge detection, then we present the algorithms based on the gradient and Laplacian computation for contour extraction in the road images, after we sit the different hardware software solutions for implementation of these algorithms on reconfigurable FPGA circuits. Finally, we expose the results and discussions of the implementation and a conclusion at the end of our work.

II. Characteristics of Contours in Road Signal Images

As well as the visual characteristics of color and texture, contours also have their peculiarities, to predict which detector will be the most effective [2]. In order to estimate the efficiency of the detectors, we will refer to some errors encountered when detecting contours (Perfect contour Fig. 1a):

- Omission of certain pixels on the contour to be detected. It is measured by counting the number of forgotten pixels with respect to the total number of pixels of the ideal contour (Fig. 1b).
- Multiple responses by detecting multiple contours. It is measured by counting the number of ambiguous pixels compared to those that are not ambiguous (Fig. 1d).
- Location: this error occurs when a pixel of an unambiguous ideal outline is not in the right place. It is measured by counting the total distance between the detected contour and the ideal contour (Fig. 1c).
- Sensitivity: This error is often related to noise and corresponds to false contours detected near the ideal contour. It is measured by counting the number of false contours and the total number of contours detected (Fig. 1e).

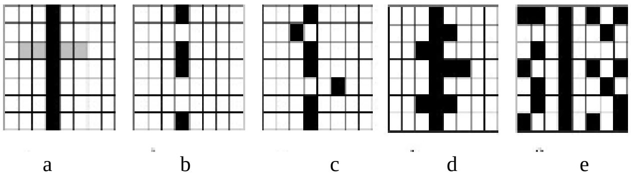

Fig. 1. Examples of contour errors; a: ideal contour; b: omission; c: relocation; d: multiple responses; e: sensitivity.

Manuscript received May 24, 2018; revised October 29, 2018.

The authors are with Laboratory LASTIMI, High School of Technology SALE Mohammed V University, Rabat, Morocco (e-mail: Issam.bouganssa@gmail.com, Mohamed.sbihi@yahoo.fr, Zaim.mounia@yahoo.fr).

doi: 10.18178/ijmlc.2019.9.1.765

In the image, a contour can be considered in different ways. Here we describe three main ways to consider a contour:

First, a contour can be seen as an abrupt change in the intensity of the image particularly for images in gray level [3]. Secondly, a way very similar to that mentioned above, to consider the contour as a difference on color. Thirdly, if we consider the image as a 2D signal, we can go in the frequency domain (Fourier transform or wavelet for example) [4]. In this case, a contour can be represented as the high frequency signal.

III. Edge Detection Principle with the Algorithms of Gradient and Laplacian

A local variation of intensity is a primary source of information in image processing. It's measured by the gradient [3] vector function of the pixel [i, j].

A. *The Gradient of an Image*

In an orthogonal coordinate system (Oxy) where (Ox) is the horizontal axis and (Oy) the vertical axis, the image gradient (or rather the luminance f) at any point or pixel coordinates (x, y) [3], [4] is denoted by Equation 1:

$$Grad\, f = \boldsymbol{\nabla} f = (\begin{matrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{matrix}) \qquad (1)$$

The module of the gradient quantifies the importance of the contour highlighted, that is to say the magnitude of the jump intensity observed in the image:

$$\|\nabla f\| = \sqrt{(\frac{\partial f^2}{\partial x}) + (\frac{\partial f^2}{\partial y})} \qquad (2)$$

The direction α_o of the gradient determines the present edge in the image. Indeed, the gradient direction is orthogonal to that of the outline:

$$\alpha_o = \arctan(\frac{\partial f/\partial y}{\partial f/\partial x}) \qquad (3)$$

The principle of edge detection by the use of the gradient is to calculate, in the first time, the gradient of the image in two orthogonal directions, then the gradient module. The next step is to make a selection of the most marked contours, that is to say the points of stronger contrast with adequate thresholding.

B. *Laplacian Mask with Second Derivative*

The gradient operators seen above exploit the fact that a contour in an image corresponds to the maximum of the gradient in the direction orthogonal to the contour.

However, the zero crossing of the second derivative of an intensity break also makes it possible to highlight the contour [5].

The second derivative is therefore determined by the Laplacian calculation:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial y}\right) \qquad (4)$$

The equation can be written:

$$\nabla^2 f = \nabla_x\big(f(x+1,y) - f(x,y)\big) + \nabla_y\big(f(x,y+1) - f(x,y)\big) \qquad (5)$$

Now we can define the Laplacian (Ox) by:

$$\nabla_x\big(f(x+1,y) - f(x,y)\big) = f(x+1,y) - f(x,y) - \big(f(x,y) - f(x-1,y)\big) \qquad (6)$$

$$\nabla_x\big(f(x+1,y) - f(x,y)\big) = f(x+1,y) + f(x-1,y) - 2f(x,y) \qquad (7)$$

And we define the Laplacian (Oy) by:

$$\nabla_y\big(f(x,y+1) - f(x,y)\big) = f(x,y+1) - f(x,y) - \big(f(x,y) - f(x,y-1)\big) \qquad (8)$$

$$\nabla_y\big(f(x,y+1) - f(x,y)\big) = f(x,y+1) + f(x,y-1) - 2f(x,y) \qquad (9)$$

So the Laplacian (Oxy) can be written:

$$\nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y) \qquad (10)$$

This Laplacian calculation operation can then be applied to an image via filtering with the following mask 3 * 3:

0	1	0
1	-4	1
0	1	0

Other masks can be used for the application of the Laplacian algorithm:

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	-8	-1
-1	-1	-1

1	-2	1
-2	4	-2
1	-2	1

After filtering the image by means of one of these filters, it is necessary to detect the zero crossings while keeping only the most marked passages. Indeed, the technique is particularly sensitive to noise due to the double derivation. It is therefore a question of not considering the noise, which can very well result in oscillations around zero, like an edge.

It is the role of the threshold S which will be used in this approach to take into account only the relatively high amplitude zero crossings corresponding to true Edge of the image.

The difference between the methods presented in this article and the classical edge detection such as Sobel, Prewitt and Robert, is that the classical masks are codes in several directions (vertical, horizontal, diagonal) and a module calculus of these directions is obligator, to have a single image that presents the edges.

The mask present in this document is applied only one liver and presents the different edge of the image.

C. Thresholding

The previous methods of Laplacian algorithms were used to determine the double derivation of the image that allows effectively highlighting these contours. At this level, the resulting image is expressed in gray levels, indicating here the importance of each fracture intensity [6].

To isolate the edges of the rest of the image, a new step is needed to obtain more accurate information that can test the presence of edges. The resultant image IB (i, j) of this treatment is in black and white. White pixels (value 1) indicate the presence of an edge, the absence of black pixels (value 0).

The conversion of a gray-scale image to a black-and-white 'binarization' image after the Laplacian filter IM (i, j) image, requires the setting of the thresholding and the determining of a threshold value to represent the edges more signified and avoids the edges of the noisy.

This parameter, noted S, is chosen to present the most significant contours found from a cumulative histogram of the image at the gray level. If the value of the image pixel exceeds the threshold, the resulting pixel value is 1. Otherwise, the pixel value is set to 0:

$$\boldsymbol{I}_{\mathrm{B}}(i, j) = 1 \text{ if } \boldsymbol{I}_{\mathrm{M}}(i, j) \geq S$$
$$\text{Else } \boldsymbol{I}_{\mathrm{B}}(i, j) = 0 \qquad (11)$$

D. Simulation of the Laplacian Algorithm on Different Road Images

We present in this part, the application of mask based on the calculation of Laplacian on different road images, the goal being to verify the effectiveness of the mask to give optimal results for the detection of the edges.

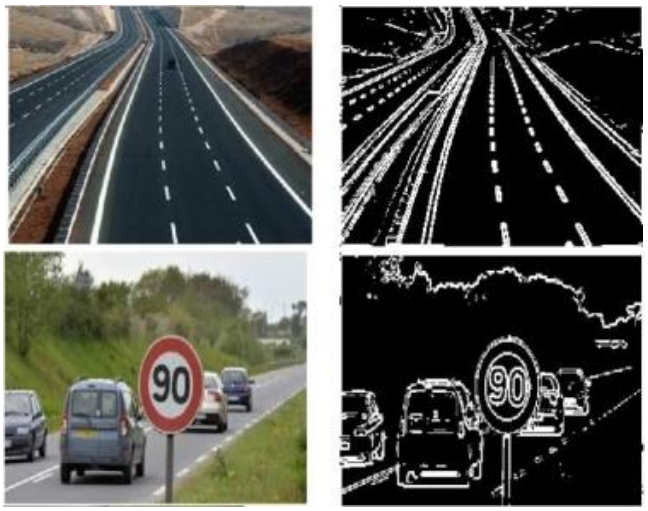

Fig. 2. Application of Laplacian algorithm to characterize different images

In the following figure (Fig. 2), we present the application of the Laplacian mask on four road images, with the aim of detecting the line in the middle of the road or detecting the signs of the Highway Code.

The objective of this mask is the ease of implementation and the speed of processing [5], [6]. On the other hand, its disadvantage is its sensitivity to the acoustic noise of the second derivative. However, the results are often quite broad. The filters based on the calculation of the variation of the intensity, which it is by the first or second derivative are the most used in the industrial applications requiring constraints for the implementation in real time on an embedded system.

IV. Hardware Software Tools Needed for Road Images Processing

Real-time image processing requires high computing power. For example, the standard image.jpg is about 0.8 megapixels per frame, with a calculated power of 30 frames per second for a 25 MHz processor.

The size of the image may be larger [7], the amount of processing required per pixel also depends on the processing algorithm used.

High resolution images become more common, processing requirements will increase. High resolution images of standards typically have ten times more pixels per image. The computing load is about ten times higher. They require more DSP or a very expensive high-end DSP. In this scenario, FPGAs provide real-time alternative image processing. FPGA effectively supports high levels of parallel processing data stream structures, which are important for the efficient implementation of image processing algorithms.

A. Xilinx ISE

ISE is an integrated digital system development environment aimed at FPGA hardware synthesis / implementation. Designs can be described in three main forms: as diagrams, as HDL or as state diagrams (Fig. 3).

For this, ISE integrates various tools to pass through the entire design flow of a digital system. It includes editors and tools for system input, as well as a set of tools, synthesis and implementation, grouped into a single design flow [7]. The details of these tools are given as follows:

- Design Entry: HDL Editor, Finite State Machine Editor (State CAD), Engineering Capture System (ECS), and IP Generator (CORE Generator).
- Synthesis: XST (Xilinx Synthesis Technology), Leonardo Spectrum by Mentor Graphics and Synplify by Cadence.
- Simulation: Test Bench HDL Generator and Model Technology ModelSim simulator integration.
- Implementation: Translate MAP, placement and routing (PAR), floor planner, FPGA editor, timing analyzer XPower, Fit (only CPLD) and Chip viewer (only CPLD).
- Programming the component and formatting the Bit stream file: Bit Gen and IMPACT.

B. DDR2 Memory Management Used on FPGA Spartan-6 from Xilinx Nexys-3

The Nexys-3 board contains three external memories, all from Micron: a 128 Mbit cellular RAM (pseudo-static DRAM), a parallel non-volatile PCM 128Mbit (phase change memory); and a 128Mbit PC serial device (Fig. 4). The cellular RAM and the parallel PCM device share a common bus, and the serial PCM is on a dedicated quad-mode (x4) SPI bus. Nonvolatile PCM memories are editable bytes and bits without the need for block erasure, so they are faster and more versatile than conventional flash in most applications [8].

The 16Mbyte cellular RAM (Micron M45W8MW16 circuit number) has a 16-bit bus that supports 8-bit or 16-bit data access. It can operate as a typical asynchronous SRAM with 70ns read and write cycle times, or as a synchronous memory with an 80 MHz bus. When used as an asynchronous SRAM, the cellular RAM automatically refreshes its internal DRAM arrays, allowing a simplified memory controller (similar to any SRAM controller) [9]. When operating in synchronous mode, continuous transfers up to 80MHz are possible.

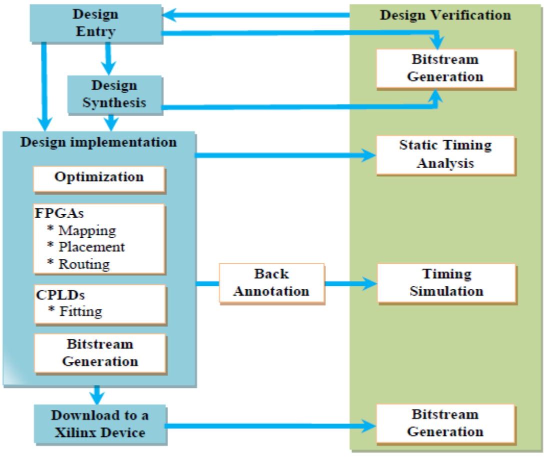

Fig. 3. Generic Xilinx design flow [7].

V. Real-Time Implementation of Algorithm on FPGA Xilinx Spartan-6

We used Spartan-6 FPGA from the Xilinx Nexys-3 platform for the implementation of our algorithm and a VGA monitor to display the results. The algorithm was developed on the Xilinx ISE interface and the different blocks are programmed in VHDL and presented in the following Table I:

TABLE I: List of Different Blocks

Block program	objective of the block in the project
Memory	The "Memory block" that exists in the party IP (intellectual property) to the Xilinx software is configured according to the size of the image that will store, each memory cell contains 8 bits (8 bits is the value which is coded each RGB pixel)
Memory Playback	Since the image is not necessarily the same size as the screen, the program is necessary for the correct positioning of the image on the screen
VGA Screen	The program's role to define. The VGA monitor is controlled by five 10-bit coded signals: red (3 bits), green (3 bits), blue (2 bits), horizontal sync and vertical sync 1 bit 1 bit. The three color signals, collectively known as the RGB signal are used to control the color of a pixel at a location on the screen. In order to produce other colors, each color analog signal is to be supplied with a voltage between 0.7 and 1.0 volts for varying the intensities of the colors
Synchronization	The program lets you synchronize the scanning of pixels on the screen of a horizontally and vertically. Signals are used to control the synchronization of the scanning speed. The horizontal synchronizing signal determines the time to scan one line, while the vertical synchronizing signal determines the time to scan the entire screen. By manipulating these signals, the images are formed on the monitor screen
Algorithm	In this part of the program we use the principle defined previously in part 3 for the detection of contours. The pixels after the conversion are always smaller than the total number of pixels [10]. The upper limit guarantees that the operations depending on the FIFO list can be completed during the real-time processing period of a frame

Fig. 4 shows a diagram of the different blocks presented in Table I, programmed in VHDL and synthetized by Xilinx ISE software [7], [8].

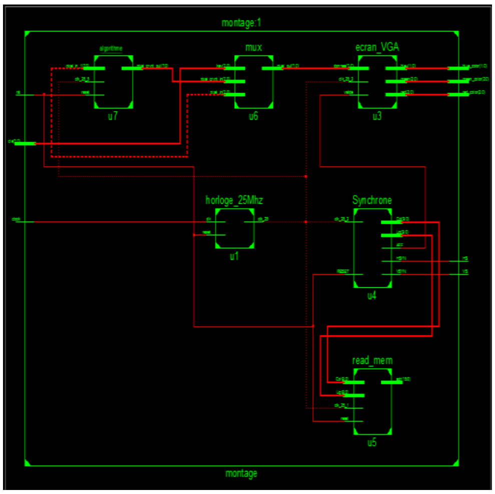

Fig. 4. The blocks implemented on Xilinx ISE.

VI. Results of Implementation and Discussion

To display the processed images we used a VGA screen connected directly to a Xilinx Spartan-6 FPGA circuit. Fig. 5 shows the pre-processing images and Fig. 6shows the same images after processing.

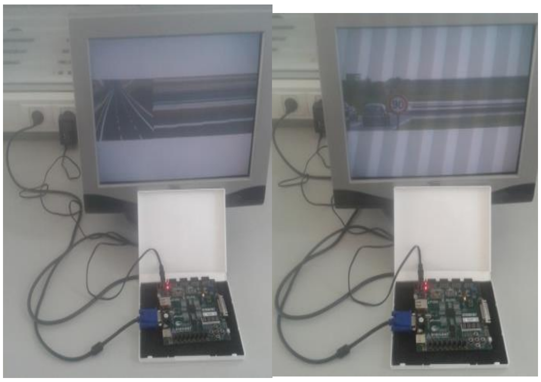

Fig. 5. The images results before treatment.

We used a road images with a resolution of 200x200. The module is fully pipelined where a resulting pixel is calculated at each clock cycle. With this rate and the clock frequency, it is possible to treat more than 400 image frames per second at 200x200 resolutions.

In addition, the entire chain has been developed following the appropriate methodology, Algorithm Architecture. With the regular calculations, the necessary reduced memories and the important intrinsic parallelism, it forms an ideal candidate for hardware implementation on embedded systems [10], [11].

The results obtained show in Fig. 6 the clear presence of all the edges that exist in the images, these results can be used as a tool for detecting road signs to differentiate zones of the image in order to characterize the lines of the road and to extract information reduced, often relevant to characterize the road images.

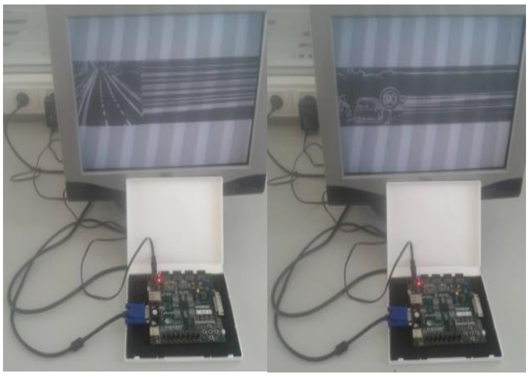

Fig. 6. The images results after treatment.

In terms of performance and power consumption, FPGAs are commonly the lowest in embedded systems, and can compete by being available quickly, reprogrammable and manufactured with the latest technology [12]. They provide an attractive alternative when the resources (cost, time) required by ASIC development are not available. The ability to parallelize operations and perform customizable functions also makes them competitive with the sequential microprocessor.

VII. Conclusion

In this paper, we presented an implementation of edge detection images by Laplacian algorithms on an FPGA for the applications for road safety, detecting panels and roadway. The method is based on the detection of zeros in the change of intensity, the masks of convolutions obtained by the computation of the second derivative (for the variation of the intensity) and a thresholding to select the strongest contours in the road imagery.

In this context, the Field Programmable Gate Array (FPGA) with its large integration and reconfiguration capabilities make it a key component for rapidly developing prototypes [10], [11]. In order to encourage the widespread diffusion of such circuits, it is necessary to improve the development environments to make them more accessible to non-experts in electronics.

To increase the storage capacity on the Spartan-6 circuit, three external memories available on the Xilinx Nexys-3 platform can be configured: a 128 Mbit cellular RAM (pseudo-static DRAM), a parallel non-volatile PCM 128Mbit (memory of phase change); and a 128Mbit PC serial device.

To access these external memories, whatever the type, a memory controller is used which serves as an intermediary between the user program and the physical memory. The controller is also used to perform the operations of commands; physical address calculations from logical addresses, data logging and addresses in FIFOs for Read / Write, automatic refresh [12]. The user program deals only with the logical address generation part as well as the Read / Write commands.

References

[1] D. Maar and E. Hildreth, “Théorie de la détection de bord,” *Actes royaleSoc. Londres,* vol. 207, pp. 187-217, 1980.

[2] T. Lindeberg, “Edge detection et détection Ridge avec la sélection echelle automatique,” *International Journal of Computer Vision*, vol. 30, no. 2, pp. 117-154, 1998.

[3] D. Marimont and Y. Rubner, “Un probabiliste Frameword pour Edge Détection et Échelle de sélection,” *Actes de la Conférence internationale de l'IEEE sur lavision par ordinateur*, pp. 207-214, Janvier 1998.

[4] J. F. Canny, “A computation approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 8, no. 6, pp. 769-798, November 1986.

[5] Pankiewicz, W. Powiertowski, and G. Rozszak, “Implementation of the Lane Detection Algorithm,” presented at 15th International Conference IEEE (2008), Poland.

[6] J. C. Mc.Call and M. Trived, “Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation,” *IEEE Transactions on Intelligent Transportation Systems,* vol. **7**, no. 1, March 2006.

[7] Design-tools. [Online]. Available: http://www.xilinx.com/products/design-tools/ise-design-suite.html

[8] fpga-programmable-log. [Online]. Available: http://store.digilentinc.com/fpga-programmable-log

[9] S. Nazari *et al*., “Multiplier-less digital implementation of neuron–astrocyte signalling on FPGA,” *Neurocomputing,* 2015.

[10] I. Bouganssa, M. Sbihi, and M. Zaim, “Implementation of Edge Detection Digital Image Algorithm on a FPGA,” *MATEC Web of Conferences,* vol. 75, no. 03003, Sept 2016.

[11] I. Bouganssa, M. Sbihi, and M. Zaim, “Implementation in an FPGA circuit of Edge detection algorithm based on the discrete wavelet transforms,” *Journal of Physics: Conf. Series,* vol. 870**,** no. 012016, 2017.

[12] M. Strzelecki, P. Brylski, and H. Kim, ‘FPGA-based system for fast image segmentation inspired by the network of synchronized oscillators,” in *Proc. International Conference on Artificial Intelligence and Soft Computing,* pp. 580-590, 2017.

Issam Bouganssa was born on February 3, 1987. He got the university diploma of technology specialized in industrial maintenance at the Higher School of Technology Sale 2009, professional license mechatronics at the faculty of Sciences Rabat 2011, specialized microelectronic master's degree at the Faculty of Sciences Kenitra 2013, PhD student in science and techniques for engineers in the Laboratory of System Analysis, Information Processing and Industrial Management at Mohammed-V University in Rabat. Search domain, data processing on embedded systems and microelectronic.

Mohamed Sbihi was born on November 22, 1962. He is a full professor of higher education. He has received a graduate degree in electronics and signal processing from the Faculty of Science at the Mohammed V University in Rabat 1995, and the PhD in automatic and information processing from the Faculty of Science in IbnTofail University in Kenitra 2006. He was employed since 1996 and member of Laboratory of System Analysis, Information Processing and Industrial Management at the Higher School of Technology-SALE in Mohammed V University in Rabat, Morocco. He works on Image processing, Data analysis, University teaching and embedded systems and robotics.

Mounia Zaim was born on July 19, 1973. She received the B.S. in eng science from University Mohammed V, Rabat, in 1996, and the M.S. in telecommunications & microwaves from University Chouaib Doukkali, El-Jadida, in 1997 and the Ph.D. degree in computer vision from University Sidi Mohammed Ben Abdellah, Fes, in 2002. She is currently a Habilited professor at the Higher School of Technology-SALE in Mohammed V University, Rabat in Morocco, since 2003 and a member of the Research Lab LASTIMI (Laboratory of System Analysis, Information Processing and Industrial Management). Her fields of interest are in computer vision, artificial intelligence, data analysis and e-learning.