Performance and Evaluation Sobel Edge Detection on Various Methodologies

Hong Nguyen.T.K.[1, 2], Cecile. Belleudy[1], and Tuan.V.Pham[2]
[1]Laboratory of Electronic, Antenna Telecomunication, University of Nice Sophia Antipolis, Nice, France
[2]Electronic & Telecommunication Engineering Dept, Danang University of Technology, Danang, VietNam
Email: tkhnguyen@unice.fr, pvtuan@dut.udn.vn, Cecile.Belleudy@unice.fr

***Abstract*—This paper compares various methodologies for the design of Sobel Edge Detection Algorithm on Field Programmable Gate Arrays (FPGAs). We show some characteristics to design a computer vision algorithm to suitable hardware platforms. We evaluate hardware resources and power consumption of Sobel Edge Detection on two studies: Xilinx system generator (XSG) and Vivado_HLS tools which both are very useful tools for developing computer vision algorithms. The comparison the hardware resources and power consumption among FPGA platforms (Zynq-7000 AP SoC, Spartan 3A DSP) are analyzed. The hardware resources by using Vivado_HLS on both platforms are used less 9 times with BRAM_18K, 7 times with DSP48E, 2 times with FFs, and approximately with LUTs comparing with XSG. In addition, the power consumption on Zynq-7000 AP SoC spends more 30% by using Vivado_HLS than by using XSG tool and for Spartan 3A DSP consumes a half of power comparing with by using XSG tool. In the study by using Vivado_HLS shows that power consumption depends on frequency.**

***Index Terms*—Sobel Edge Detection, Xilinx System Generator, Vivado_HLS, hardware resources, power consumption**

I. Introduction

Currently, computer vision system has been hailed in many aspects of our lives. They use image data from cameras to determine objects, features, or activities in some special areas like that geometric, human faces, human fall detection, vehicles, finger printer or handwritten characters. One of the ways to extract features from image data in computer vision system is edge detection technique. There are many methods to perform edge detection such as Sobel Method, Prewit Method, Roberts Method and Canny Method [1][2]. These methods have been proposed for detecting transitions in images.

The challenge today is how to implement these computer vision algorithms to meet rigorous demands of the market such as real-time processing, strict power consumption and hardware resources. There are various ways to execute these algorithms on hardware by using Xilinx System Generator Tool based on Matlab/Simulation, C/C++; Vivado_HLS based on C/C++/SystemC or programming directly in Verilog, VHDL.

We will show evaluation of Sobel Edge Detection on variable methodologies such as Xilinx System Generator (XSG) based on Matlab/Simulink, Vivado_HLS based on C++ about performance, resource utilizations, and power consumption. Moreover, we also design independent modal of this algorithm by Simulink for the reference image output. The overview of our frame work is showed Fig. 1. The reason for using Sobel Edge Detection is that not only its mask is smaller than other such as Robert, Laplacian kernels and others [3] but also gives correct results.

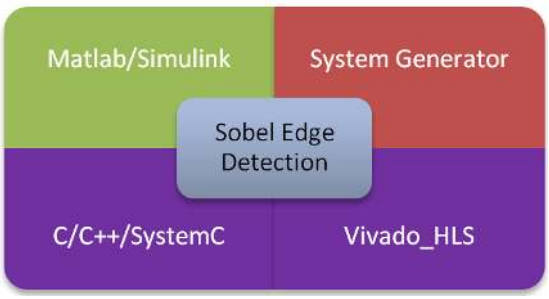


Figure 1. The overview of evaluation Sobel Edge Detection algorithm on various methodologies.

II. Related Works

Traditionally, Matlab, C/C++ and C/C++ with OpenCV are chosen for simulation and test the accuracy of computer vision algorithms. Matlab supports Computer Vision Toolbox, Image Processing Toolbox, and Signal Processing Toolbox… which facilitate to simulate and test these algorithms for designers [4][5]. Samta Gupta and Susmita Ghosh Mazumda [6] have described the different kernels (3x3; 5x5) of Sobel edge detection algorithms. In Ref.[7] C.S. Panda compared the derivative filters of various edge detection methods like that Sobel, Prewit, Roberts, Laplacian, Proposed algorithms based on root mean square error and root mean square of signal to noise ratio. Some authors use more edge detection algorithms together to increase the quality of their works, such as in [3] Q. Ying-Dong, C. Cheng-Song, C. San-Ben, and L. Jin-Quan proposed a new edge detection approach combining Zernike moments operator with Sobel operator with high sub

Manuscript received November 1, 2013; revised January 10, 2014.

doi: 10.12720/ijeee.2.1.15-20

pixel precision and lower runtime of image processing compared to use alone Zernike moments operator. M. Ouamri, A. Keche, M. Mnouar, Oran Boumediene applied the Sobel Edge Detection for Vehicle Detection Approach. The authors have defined the search area by the Hough Transform to meet real time computation and the contour area is extracted by a Sobel filter or a Canny filter. This algorithm was executed 11 frames/s on desktop computer equipped with an Intel E7300 processor at 2.6 GHz [8].

Most of researchers face great challenges of implementation edge detection on hardware, because it requires large and complex hardware. Manually, the Hardware Description Languages (HDL) uses for providing hardware model of computer vision algorithms. In Ref. [9] Arash Nosrat and Yousef S. Kavian have showed hardware architecture of Sobel Edge Detection algorithm for implementing on Field Programmable Gate Array (FPGA) with gray scale 1024×1024 images. Their project was simulated and synthesized on Xilinx Spartan3 XC3S200. I.Yasri, N.H.Hamid, and V.V.Yap have explained the performance of Sobel Edge Detection by using 12 states of Finite State Machine (FSM) [10]. This design with 720 x 720 images was modeled using HDL compiled, synthesized and downloaded to Cyclone II Altera development board by Quartus II 7.2 SP3 web edition at 27 MHz clock frequency. The proposed FPGA based architecture for Sobel edge detection algorithm is operated at 204.750 MHz. This architecture uses the test images with size 512 × 512 pixels with 256 gray levels implemented some more parallel processes on Xilinx Spartan 3 XC3S50-5PQ 208 FPGA. The details of Edge Detection design was show in [11]. Abdulsattar M. Khidhir, Nawal Younis Abdullah implemented this algorithm on Xilinx Virtex5-XC5VSX50T using Xilinx ML506 development board in which using one Xilinx Microblaze Soft core processor running at clock rate of 125 MHz.

In addition, Y. Said, T. Saidani, F. Smach, and M. Atri [12] have shown a methodology using for implementation real-time DSP algorithm by XSG tool. The architecture for Sobel Edge Detection implemented on both target platforms Spartan3A DSP 3400(XC3SD3400A-4FGG676C) and Virtex5 (xc5vlx50- 1ff676). In Ref.[13], Alba M. Sánchez G., Ricardo Alvarez G., Sully Sánchez G. have provided filter image processing using Sobel, Prewit and Roberts algorithms applied shine, contrast, threshold modifications. They designed the filter architecture based on System Generator tool. In the other hand, H.Amano and J.Hiraiwa have recently proposed a video processing architecture based on FPGA for real-time embedded vision systems by using both Verilog/VHDL and Vivado_HLS for its evaluation [14].

III. Contribution

Our main contribution is the proposal of a various selection for the computer vision design based on FPGA. We show some ways of designing methodology these algorithms that reflect the goal of running on different FPGA platform, exploiting the evaluation of hardware and power consumption. We provide detailed analyses of two case methods to justify our approach. In the first case method, we present a model of Sobel Edge Detection designed by using XSG Tool. In this case, we examine aspects in which FPGA implementations are different platforms like that Zynq-7000 AP SoC, Spartan 3A DSP, Virtex5. We create the model by using Xilinx blocksets in Matlab/Simulink. The load image and display image will be hold by Matlab. The XSG is a tool which helps the designers can generate this high level language (Matlab/simulink) into HDL (Verilog or VHDL). Especially, some technique parameters will be showed in which reflect the different requirements of hardware and power utilizations. These depend on technique and architecture of each platform. In the second case method, we use the other high level language (C, C++) that could be transmitted into Verilog/VHDL by using Vivadol_HLS.

Input image: lena.jpg, resolution (512x512).

Target FPGA platforms :

Zynq-7000 AP SoC 7z020clg484-1

Spartan 3A DSP xc3sd3400a-4fg676

Virtex5 xc5vfx70t-1ff1136

The motivation of this design has had the designers choose the best way for their defined proposal.

IV. Sobel Edge Detection Algorithm

Sobel edge detection algorithms are the most commonly used techniques in image processing for edge detection [6]. In this paper 2 types of Sobel operators were used (horizontal, vertical). The operator calculates the gradient of the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. The Sobel kernels are given by

$$\boldsymbol{G_x} = \begin{bmatrix} \mathbf{-1} & \mathbf{0} & \mathbf{1} \\ \mathbf{-2} & \mathbf{0} & \mathbf{2} \\ \mathbf{-1} & \mathbf{0} & \mathbf{1} \end{bmatrix}, \quad \boldsymbol{G_y} = \begin{bmatrix} \mathbf{-1} & \mathbf{-2} & \mathbf{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{-2} & \mathbf{1} \end{bmatrix} \tag{1}$$

Here the kernel G_x is sensitive to changes in the x direction, i.e., edges that run vertically, or have a vertical component. Similarly, the kernel G_y is sensitive to changes in y direction, i.e., edges that run horizontally, or have a horizontal component. The two gradients computed at each pixel (G_x and G_y) by convolving with above two kernels can be regarded as the x and y components of gradient vector. This vector is oriented along the direction of change, normal to the direction in which the edge runs. Gradient magnitude and direction are given by:

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

An approximate magnitude is computed using:

$$|\boldsymbol{G}| = |\boldsymbol{Gx}| + |\boldsymbol{Gy}| \tag{3}$$

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = a\tan\left(\frac{G_y}{G_x}\right) \tag{4}$$

V. The First Method: Sobel Edge Detection Using Xilinx System Generator.

A. Introduction

With Matlab/Simulink we can explore an algorithm in a functional domain. Using XSG blocks (Xilinx blocksets) we are also able to design in the functional domain, but towards an FPGA implementation (physical domain). Xilinx system generator is a tool, which uses to convert a Simulink model (DSP or logical digital) into hardware. It provides high-level abstractions that can be automatically compiled for target FPGA platforms. Models constructed from Xilinx blocks behave in exactly the same way in Simulink as they do in hardware. As part of the Simulink environment, these models can readily be combined with non-Xilinx blocks to model parts of a system not bound to the FPGA. Setting parameters from the Matlab workspace can customize Xilinx blocks, like most Simulink blocks [15].

With XSG, there are three steps called software simulation (design the model within Simulink environment), hardware co-simulation (convert the simulation into hardware code or release bitstream file), and hardware implementation (download bitstream file into target FPGA platform). For that we can develop and test the Sobel edge detection algorithm by FPGA platform [16].

B. Sobel Edge Detection Based on Xilinx System Generator.

The design flow of Sobel Edge Detection using XSG is illustrated in Fig. 2. The Input image and Output Image are simulink blocksets in which the design can read/write image from/into workspace of Matlab. The Pre-processing and Display Controller Unit that transmit the image into the suitable standard of image processing for next unit are also given by simulink blocksets. The Sobel Edge Detection is designed by Xilinx blocksets. The Fig. 3 elaborates of this unit.

Our top level design was built by using XSG (Fig. 3). Inside the Edge Filter block is the Sobel Edge Detection designed with the steps are described in the section IV by Simulink blocksets. We show the comparison hardware resources and power consumption in three types of platforms of FPGA (Table I). The power parameters are obviously rather different. The quiescent power of implement on Virtex 5 is 1441mW in total. On the other hand, these are only 181mW (Spartan3A) and 79mW (Zynq AP SoC) compare with the total power. We can see that the number of power which implements on Virtex 5 is greater ten times than Zynq-7000 AP SoC. Thus, the technology and architecture of the platform prove these evidences [20].

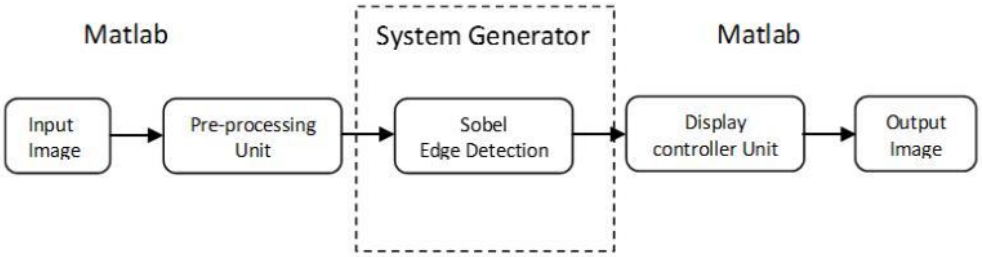


Figure 2. Sobel Edge Detection design flow based on Xilinx system generator

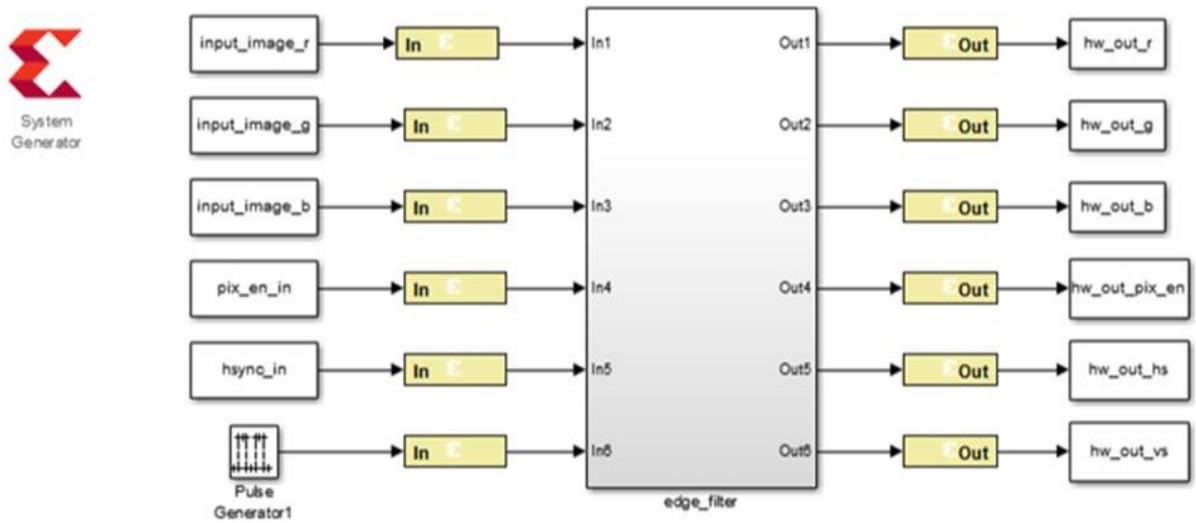


Figure 3. Block diagram of Sobel Edge Detection project using XSG tool

TABLE I. FPGA RESOURCES AND POWER CONSUMPTION INFORMATION

	Zynq-7000 AP SoC 7z020clg484-1 (197.161 MHz)	**Spartan 3A DSP xc3sd3400a-4fg676** (94.832MHz)	**Virtex5 xc5vfx70t-1ff1136** (180.999MHz)
BRAM_18K	27	27	21
DSP48E	15	15	15
FF	1716	1560	-
LUTs	1254	823	798
Slices	506	-	1881
Power (mW)	134	323	1529

The Table II provides the hardware data of our design compare with the reference design in [12] on the same platform Spartan 3A DSP. Our design takes a half of FFs and LUTs.

TABLE II. PERFORMANCE COMPARISON

	Our Design Spartan 3A DSP xc3sd3400a-4fg676	**Design**[12] **Spartan 3A DSP 3400**
BRAM_18K	27	-
DSP48E	15	3
FFs	1560	3023
LUTs	823	1755
Slice	-	2302
Power(mW)	323	-

VI. THE SECOND METHOD: SOBEL EDGE DETECTION BASED VIVADO_HLS

A. *Introduction.*

High Level Synthesis Tool (HLS) is given various efficient in implementing RTL from behavioral/functional models [17]. The designers often start with high level languages liked C/C++ or SystemC, then using software program to verify or test the algorithms. After that they can concentrate on the synthesis parts and generate the versions of RTL code to explore the different performance, area, and power constraints. Moreover, the automatic co-simulation aspect has defined the correctness of the final RTL of this framework. Recently, Vivado_HLS is known a modern HLS tools based on C/C++ or SystemC [18]. Vivado_HLS helps us rapidly implement algorithms on FPGAs without releasing RTL based on hardware description languages such as Verilog and VHDL (See Fig. 4).

B. *Pseudo-Codes for Sobel Edge Detection Algorithm*

Input : An input Image.
Output : An output edge image.
Step 1 : Converse RGB pixel to Y.
Step 2 : Apply Gx,Gy kernel to the input image.
Step3 : Compute approximation of the gradients.
Step 4 :Apply Edge thresholding.
Step5 : Main function for Sobel Edge Detection Algorithm which including line buffer, allocation memory for a streaming implementation.
Step 6 : The output image is edged the positions.

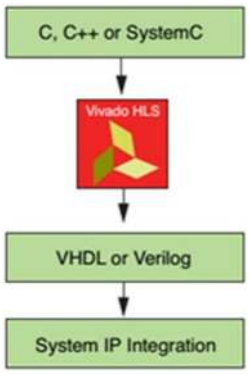


Figure 4. High level synthesis flow [18]

C. *Sobel Edge Detection Design Based Vivado_HLS .*

The FPGA hardware resources are estimated by using Vivado_HLS on two platforms showed on Table III. In this paper, we synthesize the Sobel Edge Detection Algorithm [19] on Vivado_HLS 2012.4 and ISE 14.4.

TABLE III. FPGA HARDWARE RESOURCES COMPARISON OF TWO PLATFORMS

	Zynq-7000 AP SoC 7z020clg484-1		Spartan 3A DSP xc3sd3400a-4fg676	
Frequency (MHz)	94.832	197.161	94.832	197.161
BRAM _18K	3	3	3	3
DSP48E	2	2	2	2
FFs	512	797	736	983
LUTs	1207	1247	925	948
SLICES	0	0	0	0
Power (mW)	162	402	155	378

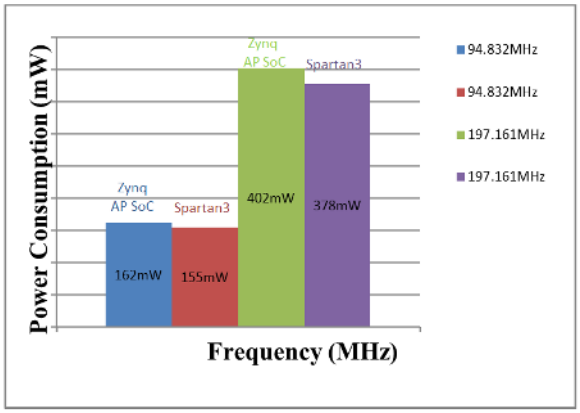


Figure 5. Performance power consumption on Zynq-7000 AP SoC and Spartan3 DSP executing in frequencies.

The Fig. 5 shows a proportion relation between frequency and power consumption. There are not too much differences of power consumption value, around 5-7 mW at the same frequency. At 94.832 MHz the Zynq-7000 AP SoC uses less FFs (512) than Spartan 3A DSP (736). In contrary, Spartan 3A DSP takes 925 LUTs comparing with 1207 LUTs on Zynq-7000 AP SoC.

VII. Discussion

In the first method, we provide a proper performance evaluation, the implemented Sobel Edge Detection project using low cost available Spartan 3 and Zynq-7000 AP SoC, the development system of Xilinx. The properties of other designs along with ours are listed in Table II. As seen from Table II, the design of the Sobel Edge Detection proposed by [12] requires double of FFs and LUTs on the clock frequency of 59.552MHz compare with our project working frequency up to 94.832MHz.

With the second ones, the estimated results of Sobel Edge Detection on two FPGA platforms are showed in Table III. However the power values of both consume approximately, the hardware resources (DSP48E, BRAM18K and FFs) of Zynq-7000 AP SoC are spent a half less than of Spartan 3. The graph (Fig. 5) describes the less frequency uses for this design, the less power consumes with the synthesis report in Vivado_ HLS.

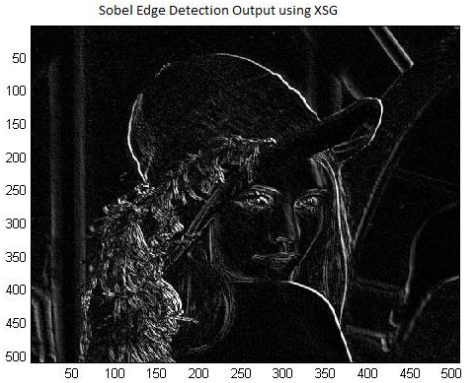
XSG is a very useful tool for developing computer vision algorithms. Besides, Vivado_HLS is recently powerful tool of Xilinx because of supporting the computer vision library OpenCv, which has more than 47,000 users. In addition, it is given accelerating solution of OpenCv function by using HLS. We can create the best possible implementation for particular algorithm in power, area and performance by using directives.

VIII. Conclusion

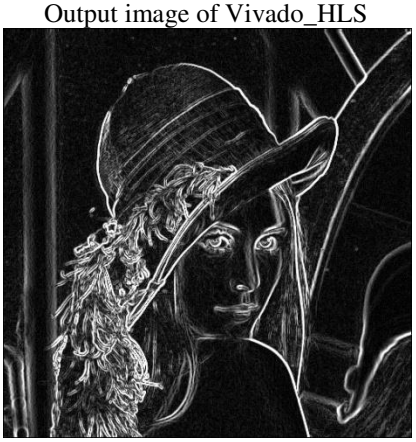
In this paper, the various methodologies of synthesis the Sobel Edge Detection are given. We can find that Vivado_HLS tool gives the better output image by vision, and at the same frequency both platforms spend less hardware resources than XSG tool excepting the power consumption value. Besides, traditional method based computer vision applications is like that system generator, Vivado_HLS is provided with a new methodology for design, synthesize and design exploration for performance, area, and power. Our methodologies would like to give the users some solution for implementing the computer vision algorithms. It also provides the comparison among the platforms toward the trading off between hardware resources and power consumption. In theory, the power consumption of Zynq-7000 AP Soc by using Vivado_HLS tool has to less than the other method, but in this experiment this aspect has really been the challenge for the authors.

Appendix Input/Output image

Output image of XSG



Output image of Vivado_HLS



References

[1] S. Islam and M. Ahmed, "A study on edge detection techniques for natural image segmentation," no. 3, pp. 80–83, 2013.

[2] M. D. More and G. K. Andurkar, "Edge detection techniques : A comparative approach," vol. 2, no. 4, pp. 142–145, 2012.

[3] Q. Ying-Dong, C. Cheng-Song, C. San-Ben, and L. Jin-Quan, "A fast subpixel edge detection method using Sobel–Zernike moments operator," *Image and Vision Computing*, vol. 23, no. 1, pp. 11–17, Jan. 2005.

[4] S. Giannarou and T. Stathaki, "Optimal edge detection using multiple operators for image understanding," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, p. 28, 2011.

[5] S. Wang, F. Ge, and T. Liu, "Evaluating edge detection through boundary detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, pp. 1–16, 2006.

[6] S. Gupta and S. G. Mazumdar, "Sobel edge detection algorithm," vol. 2, no. 2, pp. 1578–1583, 2013.

[7] C. S. Panda, "Filtering corrupted image and edge detection in restored grayscale image using derivative filters," no. 3, pp. 105–119.
[8] M. Boumediene, A. Ouamri, M. Keche, and O. Mnouar, "Vehicle detection algorithm based on horizonal /vertical edges signal and image," no. 1, pp. 396–399, 2011.
[9] A. Nosrat, "Hardware description of multi-directional fast Sobel Edge Detection processor by VHDL for implementing on FPGA," vol. 47, no. 25, pp. 1–7, 2012.
[10] I. Yasri, N. H. Hamid, and V. V. Yap, "Performance analysis of FPGA based Sobel Edge Detection operator," in *Proc. International Conference on Electronic Design*, Dec. 2008， pp. 1–4.
[11] A. M. Khidhir and N. Y. Abdullah, "FPGA based edge detection using modified Sobel Filter," vol. 2, no. 1, pp. 22–32, 2013.
[12] Y. Said, T. Saidani, F. Smach, and M. Atri, "Detection for video processing system," pp. 852–855, 2012.
[13] A. M. S. G, R. A. G, S. S. G, and F. C. E. Buap, "Architecture for filtering images using Xilinx system generator," vol. 1, no. 4, 2007.
[14] J. Hiraiwa and H. Amano, "An FPGA implementation of reconfigurable real-time vision architecture," in *Proc. 27th International Conference on Advanced Information Networking and Applications Workshops*, Mar. 2013, pp. 150–155.
[15] A. T. Moreo, P. N. Lorente, F. S. Valles, J. S. Muro, and C. F. Andrés, "Experiences on developing computer vision hardware algorithms using Xilinx system generator," *Microprocessors and Microsystems,* vol. 29, no. 8–9, pp. 411–419, Nov. 2005.
[16] V. Elamaran, A. Praveen, M. S. Reddy, L. V. Aditya, and K. Suman, "FPGA implementation of spatial image filters using Xilinx system generator," *Procedia Engineering*, vol. 38, pp. 2244–2249, Jan. 2012.
[17] J. Cong, S. Neuendorffer, J. Noguera, and K. Vissers, "High-level synthesis for FPGAs: From prototyping to deployment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 4, pp. 473–491, Apr. 2011.
[18] S. Vision, "All eyes on Zynq SoC for smarter vision," no. 83, 2013.
[19] M. V. Fernando, C. Kohn, and P. Joshi, "Zynq all programmable SoC Sobel filter implementation using the Vivado HLS tool," vol. 890, pp. 1–16, 2012.

Hong. Nguyen. T. K was born in Da Nang, Vietnam in September 1979. She received the engineering degree of Electronic and Telecommunication at the Polytechnic University of Da Nang in 2002 and a Masters degree of Electronics at The University of Da Nang in 2007. Now, she is a PhD student in the Electronics, Antennas, Telecommunications Laboratory (LEAT), the University of Nice Sophia Antipolis, France. Her research has been mainly focussed in designing and developing fall detection algorithms. She also proposed the optimization method for the hardware resources and power consumption of for FPGAs. She also interested in image and video processing, multi-core processor architectures. She has a publication related with these areas in 2013 RUNSUD conferences

Cécile Belleudy joined the Electronics, Antennas, Telecommunications Laboratory (LEAT) in the University of Nice Sophia Antipolis as an assistant professor. Her current work focuses on studying and developing low power strategies (off-line, on-line) for embedded system. More especially, she addresses the problem of low power RTOS for monoprocessor and multiprocessor platforms and energy aware memory allocation. She has also developed new approach for managing energy in wireless sensor network at different level (PHY, MAC, and Routing) and for autonomous node (Energy harvesting). In the field of low power SoC design methodologies, she participates in national and European collaborative projects.

Tuan. V. Pham was born in November 24th 1976 in Hoi An ancient town, Vietnam. He received Engineering and MSc. degrees in Electronics & Telecommunications and Dr. techn. degree in Electrical Engineering from Danang Uni. of Technology (1999), Hanoi University of Science and Technology (2001) and Graz University of Technology (2007). Tuan was granted with a post-doctoral scholarship at the SPSC Lab of TUG from 2007 to 2008 and promoted to Associate Professor at ETE Department, DUT since 2012. Tuan's major field of study is signal processing for speech and image, video processing. He has been a lecturer at DUT since 1999, Vice-Chair of Electronic & Telecommunication Engineering Dept., DUT since 2010 and Vice-Director, Center of Excellence, DUT since 2011. He has published a reference book and about 30 conference, journal papers. His research interests are as follows: Detection, estimation, information theory; Speech/ Image/ Video processing and applications; Machine learning for signal processing; Real-time implementation and optimization of DSP algorithms. Assoc.Prof.Dr. Pham is IEEE member, IEEE SSCS Chapter member, member of Radio Electronics Association of Vietnam (REV). He functions in Editorial Board, REV Journal on Electronics and Communications, Vietnam. He joins TPC of many international conferences as IEEE-RIVF 2009, IEEE-ICCE 2010, IEEE-ATC 2011, GIIS 2011, UK-VN ASSW 2012, UK-VN ICBSBE 2012.

