https://doi.org/10.2352/ISSN.2470-1173.2017.13.IPAS-203

ICA-based background subtraction method for an FPGA-SoC

***Fernando Carrizosa-Corral*[1], *Alberto Vázquez-Cervantes*[1], *Josué-Rafael Montes*[1], *Teresa Hernández-Díaz*[1], *Leonardo Barriga-Rodríguez*[1], *Jorge Alberto Soto-Cajiga*[1], *and Hugo Jiménez-Hernández*[1]**

[1] ***Centro de Ingeniería y Desarrollo Industrial, Laboratorio de Modelado, Simulado y Visión por Computadora. 76125 Querétaro, Qro., Mexico.***

Abstract

This work presents a background subtraction method based on Independent Components Analysis (ICA) implemented on a Field-Programmable Gate Array (FPGA) System-on-a-Chip (SoC) with an embedded processor. A previous work showed background subtraction utilizing ICA achieves better results than Mixture Of Gaussians (MOG) when the background is dynamic. However, that approach was developed for a computer and its proposed mean of using ICA is too complex, requiring a high-end computer for implementation. With the purpose of extending this approach to current embedded vision systems, a method is developed for an FPGA-SoC with embedded processor. This recent technology complements the parallelism of FPGAs with the general purpose computing of processors, maintaining a small footprint and a low power consumption. In this work, background is continuously updated by estimating the mean with Expectation-Maximization (EM), foreground is extracted via FastICA, and movement is determined by a threshold based on standard deviation. The herein presented approach to these algorithms allows exploiting the architecture of FPGA-SoCs, however, there are alternatives which could replace these algorithms. The developed method was tested on two image sequences and the accuracy of movement detection was measured by the precision and the recall.

Introduction

Movement detection is a critical step for several computer vision tasks which seek to understand scene dynamics. Some example applications are vigilance, human movement analysis, anomaly detection, robot trajectory planning, event detection, and traffic analysis. Nevertheless, there are various challenging issues in a real scenario such as periodic background movements, lighting changes, shadows, camouflage, and camera noise [1]. As vision systems need to endure more of these issues the movement detection methods become more computationally demanding and require expensive hardware. Recent technologies may offer means to avoid this by exploiting the benefits of both parallel and serial computing.

In order to select a movement detection method, a developer must consider memory requirements, frame rate, and the accuracy required by the application [2]. The FPGA is one of the devices most frequently used in literature, since it can achieve high pixel-level parallelism while requiring low power consumption. However, commonly these applications also benefit from high-level programming schemes, which are more appropriate for a processor. Similar needs have impulsed the commercialization of Systems-on-a-Chip (SoCs) which integrate one or multiple embedded processors and an Field-Programmable Gate Array (FPGA), referred simply as FPGA-SoCs in the rest of this paper.

Rodríguez-Andina et al. in [3] presented a review on the latest advances in FPGA technology that are expected to greatly impact industrial digital systems. Among them is using processors embedded within FPGA integrated circuits, particularly those FPGA-SoC that include processors from the Cortex-A family. Furthermore, they highlighted there are still just a few research works using these new technologies, although many applications could benefit from it. Also regarding FPGA-SoCs, Eberli [4] analyzed some of their applications and concluded they will aid in extending the usage of embedded systems to tasks which were unfeasible some years ago. According to the analysis, the main advantage of these technologies is the developer being able to use the processor as a control mechanism, while utilizing FPGA logic to accelerate numerical processing. A method completely implemented on hardware (FPGA) can achieve a higher processing speed in comparison to one implemented only using software (processor). However, some operations and data types increase the complexity of developing a solution utilizing only hardware. Thus, usually a balance must be met between hardware and software [5]. Therefore, using these FPGA-SoCs may ease developing embedded vision systems by employing methods which properly combine processors and the parallelism of FPGAs.

In the work presented by Schmid et al. [6], various FPGAs and processors were integrated into a system for estimating the orientation of a mobile robot by combining stereo vision and an Inertial Measurement Unit (IMU). The weight of the system was suitable for some UAVs, however, it could benefit from using FPGA-SoCs. Also combining stereo vision and an IMU, an odometer for a micro-UAV was implemented on an FPGA-SoC in [7]. Han and Oruklu in [5] employed an FPGA-SoC for detecting traffic signs and sending them to a computer, accomplishing real-time processing. An object detection scheme implemented on an FPGA-SoC was presented by Heo et al. in [8]. A robotic-oriented Simultaneous Localization And Mapping (SLAM) sensor was designed by Nikolic et al. in [9]. The sensor included an FPGA-SoC. The FPGA was employed for calculating Harris and Features from Accelerated Segment Test (FAST), while the processor executed the SLAM algorithms. In [10], background subtraction via Mixture Of Gaussians (MOG) was implemented on an FPGA-SoC. The results indicate the power consumption was 600 times lower than a solution completely based on ARM processors. These works highlight some of the benefits of including FPGA-SoCs within embedded vision systems, however, these technologies may also enable utilizing some higher accuracy methods that currently are only suitable for high-end computers.

Jiménez-Hernández in [11] presented a background subtrac-

tion approach based on Independent Component Analysis (ICA). This approach correctly detected movement both indoor and outdoor. Moreover, it was robust against lighting changes and shadows. Also, it achieved a better accuracy than MOG, which is one of the most commonly used background subtraction approaches. Nevertheless, it is not suitable for an embedded system, did not explore parallelism, and the resulting frame rate is too slow for most applications. Thus, in our present work we exploit the architecture of FPGA-SoCs to develop and implement an ICA-based background subtraction method suitable for an embedded system. This is mainly accomplished by examining algorithms to find opportunities for parallelism and distributing the required operations among the FPGA and the processor, approaching the processor as a flow-control unit and the FPGA as a hardware accelerator. The accuracy of the results obtained from the FPGA-SoC is measured and implementation details are discussed, which could be easily extended to devise other methods for embedded vision systems utilizing FPGA-SoCs.

This paper is structured as follows. Firstly, a general description of the main electronic devices and algorithms used within this work is presented. Next, these algorithms are integrated into the background subtraction method and implementation details are discussed. Then, experimental results are exhibited and evaluated. Finally, conclusions and future work are presented.

Main electronic devices and algorithms

This section describes the main features of FPGAs and processors to provide a general understanding of their different advantages. Also, some algorithms that will serve as components of the background subtraction method are described.

Processor

This device is responsible for most of the processing in Personal Computers (PC). Developing software for this technology is relatively fast and its popularity has lead to a wide variety of readily available operating systems, programming languages, libraries, and development kits. Multi-core processors enable some limited spatial parallelism and high clock rates are possible, however these features are usually restrained within embedded devices in exchange for a lower power consumption and a smaller footprint.

FPGA

FPGAs are reconfigurable logic which can be programmed to perform as a digital circuit. FPGAs usually include hardware resources such as lookup tables, flip-flops, DPS slices, RAM blocks, among others. These devices are highly versatile enabling developers to select different levels of spatial and temporal parallelism. One of the major challenges when designing with FPGAs is that complex projects demand a substantial software development effort and might eventually exhaust the hardware resources of the FPGA.

Mean estimation based on Expectation-Maximization

Expectation-Maximization (EM) is a resource-efficient solution for maintaining an updated mean as it does not require storing previous inputs within a buffer. A simple mean estimation is given by

$$O^+ = (1-\alpha)O + \alpha I, \tag{1}$$

where O^+ is the updated mean output, O is the previous mean output, I is the most recent input, and α is a learning rate parameter adjustable in the set $(0,1)$. This mean estimation might adapt to either slow or sudden changes depending on the choice of α.

FastICA

ICA is a method employed for solving the classical cocktail party problem, a blind source separation where several components have been mixed and must be recovered. The basic ICA model [12] supposes there are N linear and statistically independent components $\mathbf{s}_i$ $(i = i,...,N)$ and these components cannot be directly observed. Instead by utilizing N sensors, N signals $\mathbf{x}_i$ $(i = 1,,N)$ are observed which are different linear combinations of the source components $\mathbf{s}_i$. This model is expressed as follows:

$$X = AS, \tag{2}$$

where A is the unknown mixing matrix, X and S are the matrix representations of the vectors $\mathbf{x}_i$ and $\mathbf{s}_i$, respectively. In this model, it is possible to calculate an estimate, $\tilde{\mathbf{s}}_i$, of the source components with only the mixed —signals $\mathbf{x}_i$ by calculating an unmixing matrix W, where $W = A^{-1}$. This estimate is given by

$$\tilde{S} = WX, \tag{3}$$

where $\tilde{S}$ is the matrix representation of the estimated independent components $\tilde{\mathbf{s}}_i$.

FastICA is a Newthon-Raphson based iterative method which calculates such an unmixing matrix [13]. The main FastICA expression is as follows:

$$\mathbf{w}_i^+ = E\{Z(g(\mathbf{w}_i^T Z))^T\} - E\{\dot{g}(\mathbf{w}_i^T Z)\}\mathbf{w}_i, \tag{4}$$

where $\mathbf{w}_i$ is a vector of the unmixing matrix which estimates one of the source components, Z is obtained by whitening the vectors $\mathbf{x}_i$, and g and $\dot{g}$ are the first and second derivatives of a nonlinear nonquadratic function (such as $g = tanh(\theta)$ and $\dot{g} = 1 - tanh(\theta)^2$), respectively. Each $\mathbf{w}_i$ is randomly selected at first and Equation (4) must be repeated until the direction of $\mathbf{w}_i$ converges, normalizing $\mathbf{w}_i$ before each iteration.

Development and implementation of the background subtraction method

In this section, the background subtraction method is described and implementation details concerning both the FPGA and the processor are discussed. The method consists of three main blocks: background modeling, foreground extraction, and movement detection. These blocks are described in the following subsections.

Background modeling

Each pixel of the background is estimated by using the EM-based mean estimation described in Equation (1). When adjusting the value for the parameter α, a useful initial value is $\frac{1}{k}$. k being the number of frames a moving object is expected to remain within the same area, this depends both on the size and speed of

the object. This algorithm is adequate for FPGA-SoCs as every pixel is calculated independently, it does not require storing previous images on Block RAM, and it only employs multipliers and adders, which are easily implemented on the FPGA.

Foreground extraction

This process is based on the approach proposed by Jiménez-Hernández [11], where the complete sequence of images, each one considered a vector $\mathbf{x}_i$, is used as the input of ICA and the output consists of components representing background, foreground, and noise. Each component is categorized in accordance with the amount of information it provides, with most information being provided by the background and the least by the noise. This comparison is made by examining the eigenvalues obtained by the Singular Value Decomposition (SVD) of W^{-1}, i.e., a larger eigenvalue corresponds to a larger amount of information. Suppressing the eigenvalues associated with the background and noise components allows recovering only the foreground information, completing the foreground extraction step. This approach requires storing the complete sequence of images which is unfeasible for an FPGA-SoC since memory-wise it would either demand an FPGA with a massive number of RAM Blocks or transferring the complete sequence from the processor to the FPGA on each new frame.

Hence, this work proposes using only two images as the input for ICA, the first input ($\mathbf{x}_1$) being the most recent frame and the second one ($\mathbf{x}_2$) representing an estimate of the background. Moreover, in order to facilitate an implementation on an FPGA-SoC, the FastICA main expression is reformulated:

$$w_{i,k}^{+} = \sum_{j=1}^{M} z_{k,j} g(w_{i,1} z_{1,j} + w_{i,2} z_{2,j}) - \dot{g}(w_{i,1} z_{1,j} + w_{i,2} z_{2,j}) w_{i,k}, \quad (5)$$

where M is the number of pixels, $w_{i,k}$ is an element of the matrix W, $z_{k,j}$ is an element of the matrix obtained by whitening the vectors $\mathbf{x}_i$, and g and $\dot{g}$ are defined as in Equation (4). Note that $w_{i,1}$ and $w_{i,2}$ should be updated simultaneously, as they conform $(w)_i$ from Equation (4).

With this reformulation the contribution of each pixel j can be calculated independently and a simple addition may be performed at a latter stage, these characteristics are suitable for parallelism. Pixels can be divided into a predefined number of bins, in accordance with the resources available on the FPGA. Furthermore, each bin can be processed in parallel employing a pipeline, achieving both spatial and temporal parallelism.

After calculating W, the independent components are estimated by Equation (3) and the mixing matrix A is obtained by W^{-1}. Since there are only two input vectors, A will comprise two eigenvalues. The larger one being associated with the background and the second one merging both foreground and noise information. Thus, the foreground can be estimated by

$$X^* = A^* \tilde{S}, \quad (6)$$

where the first row of X^* is the foreground (represented as a vector) and A^* is obtained by suppressing the larger eigenvalue of A.

Movement detection

After the foreground image is obtained, one must determine which pixels correspond to movement. Restraining from restoring the mean after whitening the vectors $\mathbf{x}_i$ will result in a foreground image where the pixels that correspond to the background have values close to zero. Assuming the probability density function (pdf) of these pixels is Gaussian, the problem of classifying a pixel as movement is reduced to determining whether the pixel belongs to this Gaussian distribution.

An immediate approach to this problem is using a standard deviation based threshold, where the mean μ is zero (since mean was not restored) and the standard deviation σ is estimated by utilizing the entire foreground image. Hence, movement is determined by

$$M(p) = \begin{cases} 0 & \text{if } abs(p) \leq \beta\sigma \\ 1 & otherwise, \end{cases} \quad (7)$$

where p is the intensity of the pixel in the foreground image and β is an adjustable parameter. This algorithm is divided in two steps, the standard deviation must be calculated first and afterward the inequality is carried out. Both steps offer opportunities for parallelism. Furthermore, since the mean μ is zero, obtaining the standard deviation is reduced to

$$\sigma = \sqrt{\sum_{j=1}^{M} (x_{1,j}^*)^2}, \quad (8)$$

where $x_{1,j}^*$ is an element of the first row of X^*, i.e., a pixel of the foreground image. At this stage, movement detection may be improved by various techniques, such as a morphological filter. However, the results presented in this work are simply the output of Equation (7).

Implementation details

While it would be faster to perform every operation on an FPGA, it could require a higher developing effort and an FPGA with a massive amount of hardware resources. By properly combining both the FPGA and the processor, implementing the method is faster and a low-cost FPGA can be employed. Therefore, increasing the feasibility of integrating the parallelism of FPGAs into an embedded vision system. The approach herein presented uses the processor to control the flow of the procedure and performs most of the operations that are either difficult to implement on the FPGA or require a large amount of hardware resources. On the other hand, the FPGA is treated as a hardware accelerator. It is used in steps where a large block of information needs to be processed (such as a complete image) and some parallelism available.

However, steps that require exchanging large inputs or outputs need further considerations, since the communication between the processor and the FPGA might slow down the process and additional hardware resources are expended. Large inputs can be avoided by storing the current image x_1 and the estimate of the background x_2 on the FPGA and when a new large input is required the processor can send only enough data to calculate the required input. E.g., when implementing Equation (5) instead of sending the 2-by-M matrix Z, the FPGA only needs to receive the 2-by-2 whitening matrix Wh in order to calculate Z as follows:

$$Z = Wh[\mathbf{x}_1 - E\{\mathbf{x}_1\} \quad \mathbf{x}_2 - E\{\mathbf{x}_2\}].^T, \quad (9)$$

where $E\{\mathbf{x}_1\}$ and $E\{\mathbf{x}_2\}$ are the average intensity values of $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, which can be easily calculated during a previous step. Equation (9) implies some additional multiplications and additions, nevertheless, these can be implemented within the FastICA pipeline. Thus, the difference in processing time is negligible.

In this work, the method is implemented on a Xilinx's Zynq7000-based ZedBoard [14] and communication between the processor and the FPGA is achieved through Xillybus [15].

Although the scheme for exploiting parallelism on the movement detection block has already been devised, it is yet to be implemented on the FPGA. Therefore, no measurements of the implementation's frame rate are currently available.

Experimental results and evaluation

The proposed background subtraction method was tested using two sequences extracted from the PETS database [16], each one consisting on 310 images from a train station with an image size of 320 by 200. Figures 1 and 2 show some of the results from sequences 1 and 2, respectively. These results were obtained using only gray scale images, however, the method could be extended to different color spaces.

The evaluation of these results is limited to the movement detection, for which the ground truth is available. A pixel-based approach was used [17], obtaining the precision and the recall. First, we obtained true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) for each frame. Then, the precision (P) of each frame was obtained by

$$P = \frac{TP}{TP+FP}, \tag{10}$$

whereas the recall (R) is calculated as follows:

$$R = \frac{TP}{TP+FN}. \tag{11}$$

Tables 1 and 2 present the precision and recall values for the results displayed in Figures 1 and 2, respectively. For both sequences α and β (see Equations (1) and (7)) were selected as 0.02 and 2.50, respectively, for both sequences. The average precision and recall values of sequence 1 were 0.88 and 0.72, respectively. On the other hand, the calculated precision and recall averages for sequence 2 were 0.81 and 0.64, respectively. When calculating these averages, frames with little or no movement were discarded as these resulted in extreme values. Thus, 240 frames were considered from each sequence.

Results show both foreground extraction and movement detection are correctly obtained. Examining the results of Figure 2, it is observed that the mean estimation (b) retains too much information of the previous movement. This would indicate a lower value of α is required for sequence 2, as it contains a larger moving area. Alternatively, different background models may be explored in order to increase robustness or a morphological filter could be used.

Table 1. Precision and recall calculated from the results of sequence 1.

	Precision	Recall
(1)	0.94	0.86
(2)	0.94	0.47
(3)	0.89	0.67
(4)	0.71	0.81
(5)	0.74	0.82
Sequence Average	0.88	0.72

Table 2. Precision and recall calculated from the results of sequence 2.

	Precision	Recall
(1)	0.92	0.81
(2)	0.85	0.93
(3)	0.82	0.62
(4)	0.72	0.63
(5)	0.78	0.52
Sequence Average	0.81	0.64

Conclusions

This work presented an ICA-based background subtraction method developed for and implemented on an FPGA-SoC with embedded processor. The approach proposed for designing the method allows exploiting the architecture of the FPGA-SoC, aiming to increase the feasibility of using these technology in embedded vision systems. Furthermore, considerations regarding implementation are discussed.

The method was divided into three main blocks: updating the background model via an EM-based mean estimation, extracting the foreground using ICA, and detecting the movement with a threshold based on standard deviation. Opportunities of parallelism are discussed for each block. In the results obtained by implementing the method on the FPGA-SoC, both foreground extraction and movement detection are correctly determined on two different image sequences.

The evaluation of these results was restricted to the movement detection, by using a ground truth to calculate precision and recall. Using the first image sequence, the average precision and recall values were 0.88 and 0.72, respectively. For the second image sequence, the average precision and recall values were 0.81 and 0.64, respectively. The lower values obtained for the second sequence could be explained by a poor background estimation, originated by a larger moving area which would require a lower learning rate of the mean estimation.

Future work includes: exploring alternative background models that could also benefit from paralelism, measuring the processing speed of the proposed implementation against an execution employing only the embedded processor without the paralelism of the FPGA, and evaluating the robustness of the method against particular disturbances, such as lighting changes.

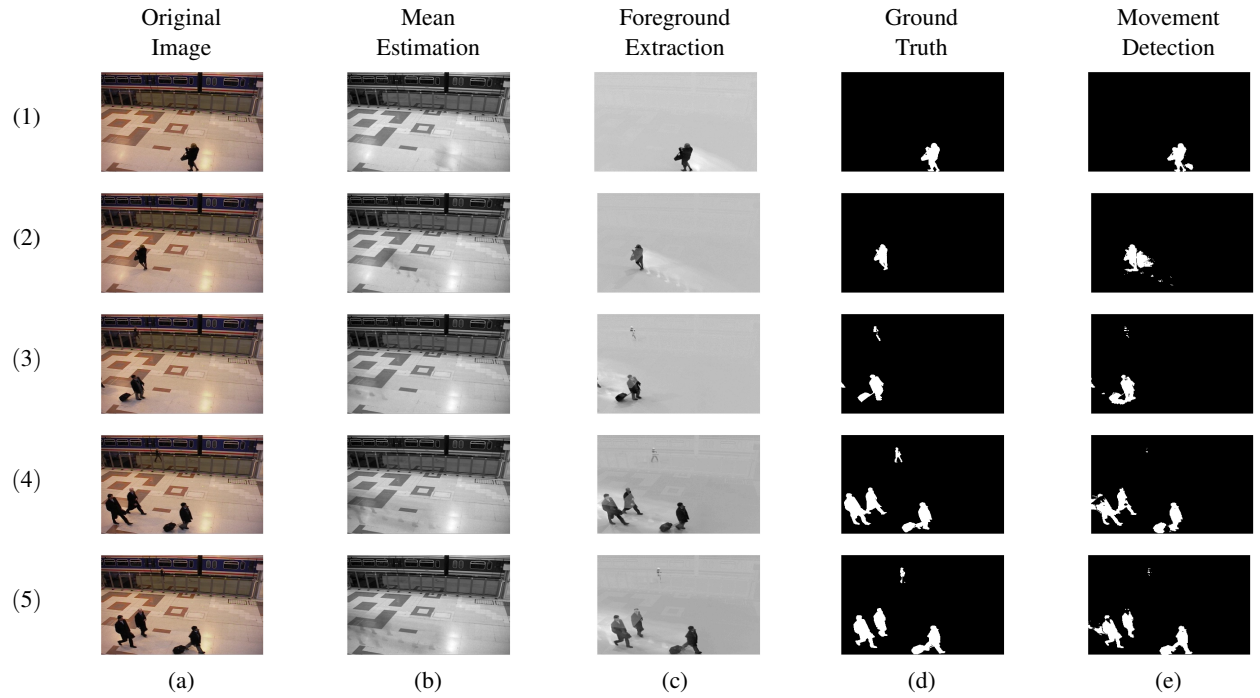

Figure 1. *Results obtained from sequence 1*

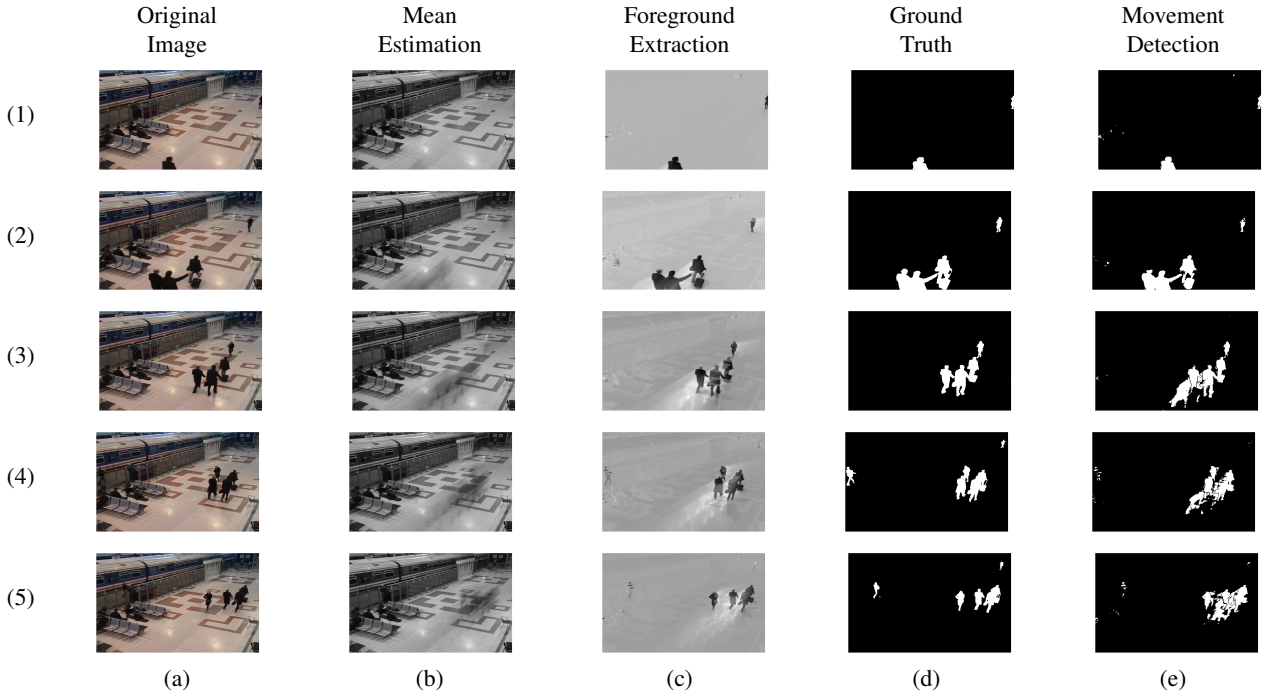

Figure 2. *Results obtained from sequence 2.*

Acknowledgments

The work of Fernando Carrizosa-Corral, Alberto Vázquez-Cervantes, Josué-Rafael Montes, and Teresa Hernández-Díaz was supported by a CONACYT scholarship. Leonardo Barriga-Rodríguez, Jorge Alberto Soto-Cajiga, and Hugo Jiménez-Hernández acknowledge financial support from SNI-México. We also acknowledge PETS2006 workshop for granting free download of the sequences database.

References

[1] J. S. Kulchandani and K. J. Dangarwala, "Moving object detection: Review of recent research trends," in *2015 International Conference on Pervasive Computing (ICPC)*, pp. 1–5, Pune, India, January 2015. doi: 10.1109/PERVASIVE.2015.7087138

[2] M. Piccardi, "Background subtraction techniques: a review," in *2004 IEEE International Conference on Systems, Man and Cybernetics*, pp. 3099–3104, La Haya, Países Bajos, October 2004. doi: 10.1109/ICSMC.2004.1400815

[3] J. J. Rodríguez-Andina, M. D. Valdés-Pea, and M. J. Moure, "Advanced Features and Industrial Applications of FPGAs A Review," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 853–864, 2015. doi: 10.1109/TII.2015.2431223

[4] F. Eberli, "Next Generation FPGAs and SOCs - How Embedded Systems Can Profit," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 610–613, Portland, Oregon, USA, June 2013. doi: 10.1109/CVPRW.2013.92

[5] Y. Han and E. Oruklu, "Real-Time Traffic Sign Recognition Based on Zynq FPGA and ARM SoCs," in *IEEE International Conference on Electro/Information Technology*, pp. 373–376, Milwaukee, USA, June 2014. doi: 10.1109/EIT.2014.6871793

[6] K. Schmid and H. Hirschmuller, "Stereo vision and IMU based real-time ego-motion and depth image computation on a handheld device," in *2013 IEEE International Conference on Robotics and Automation*, pp. 4671–4678, Karlsruhe, Germany, May 2013. doi: 10.1109/ICRA.2013.6631242

[7] G. Zhou, J. Ye, W. Ren, T. Wang, et al., "On-board inertial-assisted visual odometer on an embedded system," in *2014 IEEE International Conference on Robotics and Automation*, pp. 2602–2608, Hong Kong, China, June 2014. doi: 10.1109/ICRA.2014.6907232

[8] H. Heo, J. Lee, K. Lee, et al., "FPGA based implementation of FAST and BRIEF algorithm for object recognition," in *2013 IEEE Region 10 Conference (TENCON 2013)*, pp. 1–4, Xi'an, China, October 2013. doi: 10.1109/TENCON.2013.6718842

[9] J. Nikolic, J. Rehder, M. Burri, et al., "A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM," in *2014 IEEE International Conference on Robotics and Automation*, pp. 431–437, Hong Kong, China, June 2014. doi: 10.1109/ICRA.2014.6906892

[10] H. Tabkhi, M. Sabbagh, and G. Schirner, "A Power-efficient FPGA-based Mixture-of-Gaussian (MoG) Background Subtraction for Full-HD Resolution," in *2014 IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 241–241, Boston, USA, May 2014. doi: 10.1109/FCCM.2014.76

[11] H. Jiménez-Hernández, "Background Subtraction Approach Based on Independent Component Analysis," *Sensors*, vol. 10, no. 6, pp. 6092–6114, 2010. doi: 10.3390/s100606092

[12] G. R. Naik, "Introduction: Indepent Component Analysis," in *Independent Component Analysis for Audio and Biosignal Applications*, G. R. Naik, Ed., pp. 3–22, InTech, 2012. doi: 10.5772/52324

[13] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, Wiley-Interscience, June 2001.

[14] *Zedboard*. Internet, http://zedboard.org/, 2016.

[15] *Xillybus*. Internet, http://xillybus.com/, 2016.

[16] D. Thirde, J. Ferryman, and J. L. Crowley. *Performance Evaluation of Tracking and Surveillance (PETS)*. Internet, http://pets2006.net/ (accessed on 5 February 2010).

[17] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art," *Recent Patents on Computer Science*, vol. 1, no. 1, pp. 32–54, 2008. doi: 10.2174/1874479610801010032

Author Biography

Fernando Carrizosa-Corral was born in Mexico in 1990. He received the B.E. degree in mechatronics engineering from the Instituto Tecnologico de Culiacan, Sinaloa, Mexico, in 2013. In 2015, he enrolled as a Master student at Centro de Ingenieria y Desarrollo Industrial (CIDESI). He is interested in embedded systems, computer vision, robotics, and automatic control. His current research examines FPGA with embedded processor-based vision systems.

Alberto Vázquez-Cervantes is a PhD student at the CIDESI. He received his BS in Electronic Engineering, in Institute Technological of Querétaro, México, MS degrees in Science and Technology in Mechatronic, at CIDESI. His current research interests include Computer Vision, Embedded system, and Artificial Intelligent.

Josué Rafael Montes received his BSc. in Mechatronics Engineering from the Universidad del Valle de México (UVM), México, 2009. Aspiring to a PhD. degree in Mechatronics at CIDESI, in Querétaro, México. His work has focused on Automatic Computational Modelling of Human Behaviour and Visual Surveillance through Multi-camera Vision System aided by Augmented Reality.

Teresa Hernández Díaz received the BSc. degree in Mechatronics Engineering at Universidad Politécnica de Sinaloa (UpSin), México, 2009. Currently a candidate to a PhD. degree in Mechatronics at CIDESI, in Querétaro, México. Nowadays, her recent researches focus on automatic environment modelling through a laser array system, detecting dynamic objects interaction, automatic model generation and associative memories.

Leonardo Barriga Rodríguez, PhD in Mechatronics, in CIDESI, Mexico; with 10 years of experience in software development, computer vision, image processing and inspection robotics. He has participated in the development of more than 20 research projects in industry. He is currently a research professor of CIDESI (Center for Engineering and Industrial Development)

J.A. Soto-Cajiga (M11) received the BEng Degree in Control and Instrumentation Engineering from the Technological Institute of Queretaro, Mexico, in 2003, the MEng Degree in Mechatronics from the PICYT (interinstitutional graduate in science and technology), Mexico, in 2006 and the Ph.D. Degree also from PICYT in 2012. He works for CIDESI (Engineer Center and Industrial Development), as researcher in the applied research area, designing embedded systems and real time signal processing algorithms. He is also professor of graduate level in PICYT, and his research interests include real-time signal processing implemented in hardware, algorithms parallelizing for signals and video processing and electronics for ultrasonic PIGs.

Hugo Jiménez Hernández, PhD in Computer Vision. in CICATA-IPN, Mexico, MS in Computer Sciences in CIC-IPN,Mexico, Bachelor in Computer Sciences in ITQ-Mexico. His current interest includes Automatic Learning, Computer Science Theory and Automatic Visual Inferences. Nowadays he is the head of Computer Vision Lab at CIDESI-Mexico. He has around 50 works in several international journals and congress.