An FPGA based Embedded Vision System for Real-Time Motion Segmentation

Jorge Hiraiwa, Enrique Vargas
Department of Electronics and Informatics Engineering
Asuncion Catholic University
Asuncion, Paraguay
{jorge_hiraiwa, evargas}@uca.edu.py

Sergio Toral
Department of Electronics Engineering
University of Seville
Seville, Spain
toral@esi.us.es

Abstract—This paper proposed an FPGA based real-time video processing platform for motion detection. The main advantage of FPGA based systems respect to processor based systems is its high performance when processing large amount of data flow, as video streams. This work details the hardware implementation of real-time motion segmentation algorithms on a FPGA including the capture, processing and display stages. The proposed platform offers the flexibility of changing the processing modules, allowing the implementation of different motion segmentation algorithms and the online observation of the processing results. The system uses a minimum resource of cells allowing its implementation on low cost FPGAs.

Keywords-field programmable gate array; embedded vision system; motion segmentation; real-time video processing.

I. INTRODUCTION

Motion detection is a very important part for many computer vision applications, like human or vehicle tracking. These applications require segmenting the motion region out of the scene. Various motion detection methods have been extensively investigated in the literature. However, background subtraction methods are typically used when dealing with fixed cameras 363. Although computer vision algorithms are usually implemented in video processors 363, there are some previous works related to FPGA-based platforms. In 363 a platform for interfacing CMOS image sensor and VGA monitor for image processing is detailed. In 363 an image processing system for remote robot control is developed. They implemented several image processing algorithms like convolution filters, edge detection, binary morphology and image statistics. A parallel architecture for image pre-processing such as filtering and correlation for an image of 256x256 at high frame rate is designed in 363. These previous works performs only spatial processing. However, a spatio-temporal processing is also possible. For instance, an object segmentation system based on frame differencing and edge detection is developed in 363. Finally, an embedded human action recognition system is illustrated in 363. It recognized a small number of hand gestures at 12 fps with an 80% average recognition rate. In this paper we propose the design and implementation of a flexible platform for the real-time segmentation of moving objects in a video sequence using background modeling techniques and morphological post-processing.

The paper is organized as follows: Section II describes the motion segmentation techniques commonly used for stationary cameras. The design of modules for FPGA based video processing platform is presented in Section III. Experimental results using the proposed system are described in Section IV. Section V concludes this paper and discusses future work.

II. VIDEO PROCESSING TECHNIQUES

The detection of moving objects in video sequences is the first step for relevant information extraction in many computer vision applications, including video surveillance, and the detection and tracking of people and vehicles. The technique often used for the detection of moving objects with a fixed camera is the background subtraction 363 363. The idea consists of extracting moving objects as the foreground elements obtained from the difference between the current frame and the background model of the scene, which is a representation of the scene after removing all non-stationary objects. Equation (1) shows the detection mask *D* obtained by background subtraction:

$$D_t = \begin{cases} 1, & \text{if } (|I_t - B_t| > T) \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where *I* is the current frame, *B* is the current background model, and *T* is the threshold that determines whether the pixel belongs to the foreground or the background.

The most common methods used for background modeling for real-time applications are based on *running average filter* 363 363 and *approximated median filter* 363. The following equation, called *selective running average filter*, updates the background model using a Infinite Impulse Response (IIR) filter:

$$B_t = \begin{cases} \alpha I_t + (1-\alpha) B_{t-1}, & \text{if } (D_{t-1} = 0) \\ B_{t-1}, & \text{otherwise} \end{cases}, \quad (2)$$

where α is the learning factor or velocity update, usually *0.05*. The *conditional approximated median filter* is defined by the following equation:

$$B_t = \begin{cases} B_{t-1} + 1, & \text{if } ((I_t > B_{t-1}) \wedge (D_{t-1} = 0)) \\ B_{t-1} - 1, & \text{if } ((I_t < B_{t-1}) \wedge (D_{t-1} = 0)) \\ B_{t-1}, & \text{otherwise} \end{cases}. \quad (3)$$

For both techniques, the detection mask D is obtained using (1) with a predetermined global threshold T. Adaptive thresholds have also been proposed in the literature for these algorithms. The *selective running Gaussian filter* 363 is based on Gaussian distribution to model the background. Each pixel of the background is modeled with its mean (background model), as defined by (2), and standard deviation, defined in the following equation:

$$\sigma_t = \alpha|I_t - B_t| + (1-\alpha)\sigma_{t-1}, \quad (4)$$

where the standard deviation σ is used as the local adaptive threshold for each pixel. In this case, the detection mask is defined as:

$$D_t = \begin{cases} 1, & \text{if } (|I_t - B_t| > k\sigma) \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

where k is an arbitrary parameter, usually *2*. The *conditional* Σ-Δ *background estimation* 363 is based on approximated median filter. The background model is updated as defined by (3) and the variance is defined as follows:

$$V_t = \begin{cases} V_{t-1} + 1, & \text{if } (V_{t-1} < N \times |I_t - B_t|) \\ V_{t-1} - 1, & \text{if } (V_{t-1} > N \times |I_t - B_t|) \\ V_{t-1}, & \text{otherwise} \end{cases}, \quad (6)$$

where V is defined as Σ-Δ variance and N is an arbitrary parameter, usually *4*. The detection mask is defined as:

$$D_t = \begin{cases} 1, & \text{if } (|I_t - B_t| > V_t) \\ 0, & \text{otherwise} \end{cases}. \quad (7)$$

In the proposed platform, the *selective running Gaussian filter* and *conditional* Σ-Δ *background estimation* have been implemented, and also pre-processing modules, such as convolution filter and edge detection, and post-processing modules, which performs binary morphology like erosion and dilation, and connected component labeling. Table I details the list of implemented processing algorithms.

III. FPGA BASED VIDEO PROCESSING PLATFORM

In this section, the system architecture and design is described.

SYSTEM ARCHITECTURE

A Xilinx Evaluation Platform ML403 which contains a Virtex-4 and various peripherals has been used. The board includes a ZBT Synchronous SRAM of 1MB and a Video DAC with an interface to the VGA output. For the image sensor, a color CMOS camera module with digital output that uses an OmniVision's OV7620 image sensor 363 has been selected. This module provides a 16 bits interface of parallel data and some synchronization signals. The default video output format is the ITU-R BT601, and can be configured through an I2C interface.

TABLE I. IMPLEMENTED PROCESSING ALGORITHMS

Filtering
Convolution (Gaussian Blur / Sharpness)
Edge Detection (Prewitt / Sobel)
Binary Morphology
Erosion / Dilation / Opening / Closing
Connected Component Labeling
Background Subtraction
Selective Running Gaussian Filter
Conditional Σ-Δ Background Estimation

The camera module is connected to the evaluation board through the expansion connector, and a VGA monitor is connected to the VGA output connector, as shown in Fig.1. We have also used the SRAM as frame buffer. In this work we perform gray scale processing. Therefore just the luminance information is used. In addition, the resolution has been reduced to QVGA (320x240@30fps) to keep as many frames as possible in the frame buffer. Thus a frame buffer with 13 banks in 1MB SRAM is achieved.

DESIGN OF MODULES

Fig.2 shows the modules of the system. The FPGA is connected to a CMOS camera, a VGA monitor and a external SRAM memory through the *camera interface module*, *memory interface module* and *display interface module* respectively. In addition there are four modules that perform the video processing task. The *filter module* performs the pre-processing to the pixels acquired in the camera interface module. The *background subtraction module* applies the segmentation and background modeling. The *binary morphology module* applies morphological operations like erosion and dilation to the detecting mask. And finally, the *connected component labeling module* assigns a unique label to the detected blobs. These modules work in parallel in a pipelined architecture as shown in Fig.3. The acquisition, pre-filtering and background subtraction is performed in a single stage. The pipelined architecture is working at 30Hz that is generated by FODD signal provided from the camera module, so each stage is synchronized with this signal. Each stage reads the previous stage results from a double buffer, and writes the processed result to another double buffer. Thus a 3 double buffer has implemented on SRAM. Each stage is clocked at 25MHz and the SRAM is working at 100MHz.

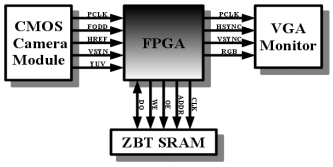

Figure 1. Main components of the proposed platform.

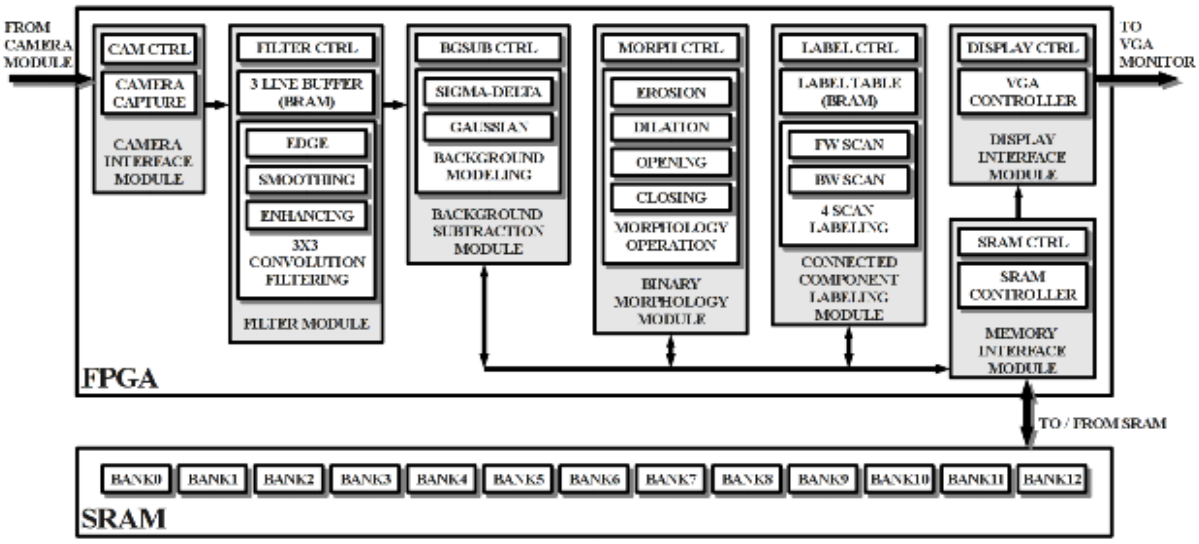

Figure 2. Modules of the video processing system on the FPGA.

The operation details of each module are next detailed.

A. Camera Interface Module

The image sensor by default is set to 640x480@30fps with YCrCb 4:2:2 interlaced output. The data is sampled every 2 pixel clock cycles only at the odd frames, reducing the resolution to 320x240@30fps. Each acquired pixel is stored in a 4 pixels local buffer and sent to the filter module.

B. Filter Module

Each acquired pixel from the previous stage is stored in a 3 line buffers. Each line buffer has 320 pixels that belong to each frame line. Using these buffers, convolution filtering is performed using a 3x3 mask. It is able to process 4 pixels in parallel at a single clock cycle. Results are sent to the background subtraction module.

C. Background Subtraction Module

Two algorithms have been separately implemented in this module: selective running Gaussian filter, which uses (2), (4) and (5), and conditional Σ-Δ background estimation, which uses (3), (6) and (7). First, it is necessary to read the 4 pixels belonging to the background model B, and the variance V or standard deviation σ, and the detection mask D from the SRAM. For each 4 acquired pixels from the previous module (the current image pixels I), the background subtraction algorithms are computed and the updated model rewritten to SRAM. The resulting detection mask is written in the double buffer.

D. Binary Morphology Module

A binary morphological operation is applied to the detection mask. The procedure for applying the filter is identical to the one applied in the filter module. It reads the pixels of detection mask from the SRAM and applies the 8-neighborhood morphological operation.

E. Connected Component Labeling Module

The four scans connected component labeling algorithm 363 has been used. This algorithm requires a one dimensional table to maintain the provisional labels. This table is implemented using internal RAM synthesized on the FPGA.

F. Memory Interface Module

The external SRAM is shared by 4 general modules. Consequently, memory accesses must be multiplexed in time. As the memory can be accessed simultaneously by several modules, the memory interface module is acting as bus arbiter to synchronize memory writing and reading

G. Display Interface Module

This module is constantly accessing to memory and displaying on a VGA monitor the stored frames in memory. As the selected resolution is 320x240, 4 frames are simultaneously displayed on the 640x480 VGA monitor. This module also provides an interface with buttons to change the frames that will be displayed on the monitor.

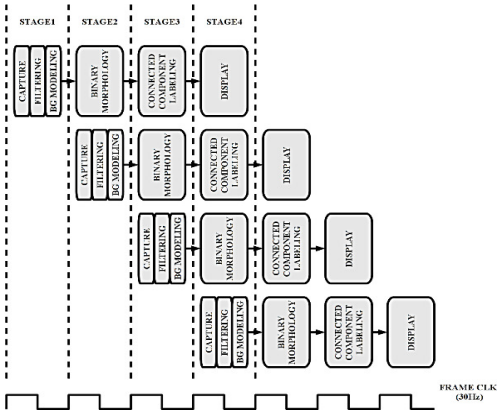

Figure 3. Processing pipeline diagram.

IV. Results

The video processing modules are implemented in a Xilinx Virtex-4 FPGA (XC4VFX12) which has been designed using VHDL and synthesized using XST of Xilinx. A minimal amount of resources are required for

both, Σ-Δ segmentation and Gaussian segmentation, as shown in Table II. Consequently, a low-cost FPGA such as the Spartan-3 family could be used instead. The background subtraction module can be performed at 186fps in the case of the Σ-Δ segmentation and 144fps in the case of the Gaussian segmentation. The implemented system also allows the online evaluation and comparison of the described algorithms using a VGA monitor (see Fig.4). Fig.5 illustrates the segmentation algorithms applied to a data set of ITEA CANDELA project 363. The first column of the Figure shows the current image at frame 320. The second, third and fourth columns represent the current background model, the normalized variance image, and the detection mask, respectively. The first and second row corresponds to the Gaussian segmentation and Σ-Δ segmentation, respectively.

TABLE II. FPGA CONSUMED RESOURCES

		Σ-Δ		*Gaussian*	
	Avail.	***Used***	***Util.***	***Used***	***Util.***
Number of Slices	5472	3824	69%	3770	68%
Number of Slice Flip Flops	10944	1131	10%	1132	10%
Number of 4 input LUTs	10944	6583	60%	6451	58%
Number of bonded IOBs	320	105	32%	105	32%
Max. Frequency (MHz)		205.297		196.16	

Figure 4. System illustration.

Figure 5. Results of the implemented algorithms.

V. CONCLUSIONS

In this paper, an FPGA based video processing platform has been presented. The system employs parallel processing and pipelined architecture to achieve real-time motion segmentation and a morphological post-processing with component connected labeling (30fps). An important advantage of using FPGA is that the processing element can be easily improved to increment the general performance. For instance, more pipeline stages can be added if desired. As future work, more complex algorithms for motion segmentation and further stages of processing such as the feature extraction and object tracking can be considered.

ACKNOWLEDGMENT

This work was supported by the Integrated Action D/023993/09 funded by the Spanish Agency for International Development and Cooperation (AECID).

REFERENCES

[1] M. Piccardi, "Background subtraction techniques: a review", IEEE Inter. Conf. on Systems, Man and Cybernetics 2004, pp. 3099-3104, Oct 2004.

[2] S. L. Toral, M. Vargas and F. Barrero, "Embedded Multimedia Processors for Road Traffic Parameter Estimation", Computer, Vol 40, Iss. 12, pp. 61-68, 2009.

[3] M. K. Birla, "FPGA Based Reconfigurable Platform for Complex Image Processing", IEEE Inter. Conf. on Electro/information Technology 2006, pp. 204–209, May 2006.

[4] S. Jin, D. Kim, X. D. Pham and J. W. Jeon, "FPGA-based image processing system for remote robot control", IEEE Inter. Conf. on Robotics and Biomedics 2008, pp. 120-124, Feb 2009.

[5] S. McBader and P. Lee, "An FPGA implementation of a flexible, parallel image processing architecture suitable for embedded vision systems", Proc. Inter. Parallel and Distributed Processing Symposium 2003, pp. 5, April 2003.

[6] K. Ratnayake and A. Amer, "An FPGA-based implementation of spatio-temporal object segmentation", IEEE Inter. Conf. on Image Processing 2006, pp 3265-3268, 2006.

[7] H. Meng, M. Freeman, N. Pears and C. Bailey, "Real-time human action recognition on an embedded, reconfigurable video processing architecture", Journal of Real-Time Image Processing, Vol. 3, Iss. 3, Sep 2008.

[8] S. L. Toral, M. Vargas, F. Barrero and M.G. Ortega, "Improved Sigma-Delta Background Estimation for Vehicle Detection", Electronics Letters, Vol. 45, Iss. 1, pp. 32-34, 2009.

[9] J. Heikkila and O. Silven, "A real-time system for monitoring of cyclists and pedestrians", Proc. of Second IEEE Workshop on Visual Surveillance, pp. 74-81, 1999.

[10] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao and S. Russell, "Towards Robust Automatic Traffic Scene Analysis in Real-time", Proc. ICPR'94, pp. 126-131, Nov. 1994.

[11] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images", Machine Vision and Applications, Springer-Verlag 1995, pp. 187-193, 1995.

[12] C. Wren, A. Azarbayejani, T. Darrell and A. P. Pentland, "Pfinder: real-time tracking of the human body", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 780-785, 1997.

[13] A. Manzanera and J. C. Richefeu, "A new motion detection algorithm based on Σ-Δ background estimation", Pattern Recognition Letters, vol. 28, no 3, pp. 320-328, Feb 2007.

[14] OV7620, "Product Specifications - Rev.1.3", Omnivision Technologies Inc., 2000.

[15] K. Suzuki, I. Horiba and N. Sugie, "Linear-Time connected-component labeling based on sequential local operations", Computer Vision and Image Understanding, 89(1):1-23, 2003.

[16] ITEA CANDELA Project, "Road intersection scenario data sets", http://www.multitel.be/~va/candela/.