Efficient Real Time Face Tracking Operator Study and Implementation Within Virtex FPGA Technology

Messaoud Mostefai[1], Mechta Djamila[1], and Youssef Chahir[2]
[1]Computer Science Department, University of Bordj Bou Arreridj, Algeria
[2]Computer Science Department, University of Caen, France

Abstract: *In this paper, we present the development of a new real time face-tracking operator. This last is based on motion detection techniques, less complex than techniques based on skin color, which are sensitive to illumination or those based on geometric standards which are efficient but expensive in computation tasks. The developed operator allows a robust face tracking with a considerable reduction in hardware resources, more particularly memory resources, which remains a powerful criterion in the construction of real time embedded systems. Moreover, the operator maps efficiently into a highly pipelined architecture, well suited for an implementation in reconfigurable technology. The developed operator is implemented on a Virtex FPGA architecture. The obtained experimental results show the effectiveness and correctness of our approach.*

Keywords: *Face tracking, motion detection, real time, reconfigurable technology.*

Received June 11, 2005 ; accepted September 3, 2005

1. Introduction

Face detection and tracking in a natural scene is an essential and important step before any facial recognition operation. This last finds its application in various fields such as security and surveillance teleconferencing, facial gesture recognition, as well as human-computer interactions [7, 14, 15].

The different methods proposed in the literature to solve this problem use pre established rules elaborated from our knowledge of human faces to detect them or seek to localize invariant features of faces, such as face skin color for detection or at least employ templates that are not pre-defined and that require statistical analysis or machine learning from examples to find the relevant face characteristics. However, these methods are generally very expensive in terms of computational times and require additional constraints to perform this operation. The approach we follow uses motion to perform a face detection and tracking from a real world scene. Moving edges are extracted with simple and efficient image processing techniques, well suited for a real time implementation with low cost hardware.

To perform a moving edges analysis, we use an operator which, applied to an image sequence, allows to extract the edges of elements in movement from a dynamic scene.

The system we propose here has been implemented on a low-cost reconfigurable architecture based on Virtex Field Programmable Gate Arrays (FPGAs). These architectures are sufficiently flexible to permit the implementation of new algorithms on existing hardware, and their performance is adequate for real time operation of many image processing problems.

After a brief introduction to moving detection techniques, we present our operator and show its efficiency to perform an efficient face detection and tracking. Software results are presented, as well as the details of the implementation within a reconfigurable Virtex FPGA architecture for a real time image processing.

2. Moving Objects Detection Techniques Overview

Two categories of methods are used for the detection of moving objects in a sequence of images: Those that make a comparison of the current image with a reference image that represents the static model of the filmed scene [3, 9]; those that compare the current image with the neighboring images, by concentrating directly on zones where changes are observed [6, 11].

The construction of a reference image is often a difficult task, because variations of lighting within the analyzed sequence can make some images incompatible with the reference image. Haynes and Jain [3] suggest an operator that performs the product of the absolute difference of two successive images with the gradient of the current image I_n. This operator does not perform a coincidence between edges but rather between the gradient of the current image and the absolute value. Another operator proposed by Stelmaszyk [9], consists in a multiplication between the gradient of the difference of two successive images and the gradient of the current image, allowing thus to have a true coincidence between edges. In general, operators that operate only on two consecutive images induce false edges. Indeed, in this case, any visible object on the current image which was not found on the one before is considered as mobile, even if it concerns an element of the static bottom that was hidden, and that comes to appear. In order to avoid this problem, the processing must operates on three consecutive images. Vieren [11] has proposed an operator which makes a product of the absolute differences between neighboring couples (I_{n-1}, I_n) and (I_n, I_{n+1}), with I_{n-1}, I_n, I_{n+1} three consecutive images. Another operator using three consecutive images in its processing was also proposed by Orkisz [6]. This last, allows mobile frontiers in the current image to be located by comparison of frontiers of the current image to those of next and preceding images so as to distinguish mobile elements of static ones. The expression of this operator is as follows:

$$\forall x, y\ R_n(x, y) = \max[G_n(x, y), G_{n-1}(x, y), G_{n+1}(x, y)] - \max[G_{n-1}(x, y), G_{n+1}(x, y)] \qquad (1)$$

with G_{n-1}, G_n and G_{n+1} respective gradients of three consecutive images I_{n-1}, I_n, I_{n+1}, and R_n the result of the operation which corresponds to the moving edges.

From all known operators that use varying elements between neighbor images, the Orkisz operator represents the best compromise between efficiency and simplicity. Indeed, compared to other operators (e. g., Vieren operator) which consists in performing products of gradients, the Orkisz operator requires only comparison and subtraction operations, easy to implement and not affecting the dynamic as with multiplication operation.

3. Limitation of the Orkisz Operator

Tests made with the Orkisz operator on real world scenes shows that it allows to eliminate effectively false edges (case of occulted objects), but does not perform efficient face tracking, more particularly in case of rapid motion. We note that only eyes and mouth motion are useful to locate the face. As these two elements are parts of face, the occultation problem concerns only the face borders and thus can be ignored. This important feature has encouraged us to propose our own operator, which does not take into account occultation problem, and allows to have an efficient face location and tracking even for fast movements.

4. Proposed Operator

The proposed operator uses three consecutive images and performs a motion amplification which consists in retaining the maximum gradient difference between neighboring couples (I_{n-1}, I_n) and (I_n, I_{n+1}). The expression of this operator is as follows:

$$\forall x, y\ R_n(x, y) = \max[(|G_n(x, y) - G_{n-1}(x, y)|), (|G_n(x, y) - G_{n+1}(x, y)|)] \qquad (2)$$

Our operator allows to detect the smallest motion between successive images and gives a null result only in case of static scenes. On the other hand, the Orkisz operator gives a null result in case of static scenes, but also in case where the gradient of the preceding or the following image (I_{n-1} or I_{n+1}) is higher than the gradient of the image in progress (I_n,), thus making this operator less sensitive to motion. Figure 1 illustrates the difference between the Orkisz operator and the proposed operator.

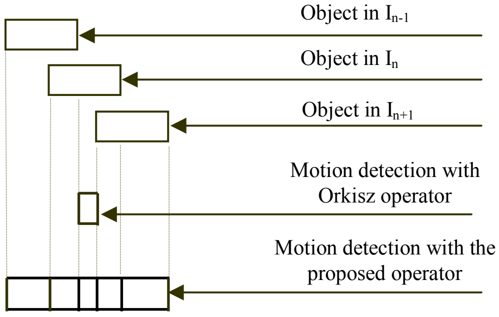

Object in I_{n-1}
Object in I_n
Object in I_{n+1}
Motion detection with Orkisz operator
Motion detection with the proposed operator

Figure 1. Motion detection using both the Orkisz and the proposed operator.

In Figure 2, we show an example of results obtained with the two operators using a real world sequence. In order to keep only the areas of interesting with strong motion intensity, we associate to the motion detection operator an intra image filter based on the search of moving segments instead of moving pixels as shown in Figure 3. We define an edge segment as a whole of 3 adjacent edge pixels.

The procedure to extract a moving segment is as follows A candidate segment is consider ed as an edge segment if it only exists in the current image and not in the preceding and following images.

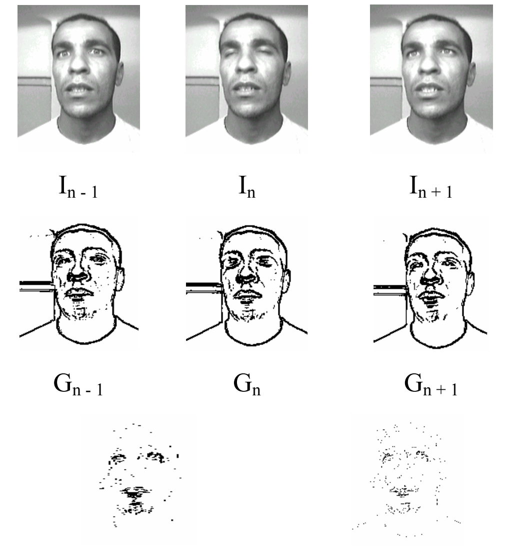
5. Simulation and Tests

5.1. Edges Detection

Various comparative tests were carried out on video sequences to choose the best suited edge detector for our operator in terms of:

1. Smoothness and continuity of contours.
2. Number of moving points in the areas of interest.
3. Computation and hardware costs.

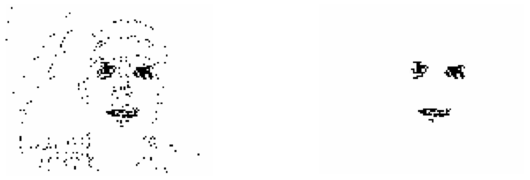
Figure 4 presents the evolution of the number of moving points (N) in the areas of eyes and mouth obtained by the application of our operator with various edge detectors (Robert, Prewitt and Sobel) on a sequence of 100 images during which a filmed person says his/her name and blinks his/her eyes. Based on mathematical formulation of the three operators [2] and obtained simulation results as shown in Figure 4, Prewitt operator has been chosen because it presents the best compromise between efficiency and computation simplicity.



(a) Motion detection with the proposed operator.

(b) Motion detection with the Orkisz operator.

Figure 2. Application to a real world sequence.



(a) Moving pixels detection.

(b) Moving segments detection.

Figure 3. Moving pixels and segments detection.

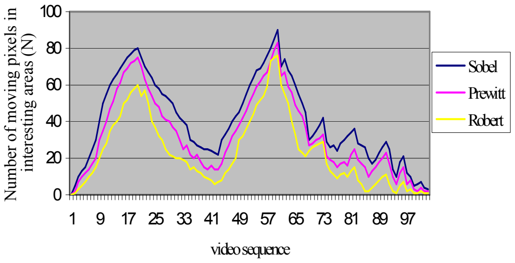


Figure 4. Motion detection using different edge detectors.

5.2. Face Model Drawing

Based on the inter-distance between the two eye candidates (L), we can set up the corresponding windows to cover the eyes and mouth regions, as shown in Figure 5.

We assume that the rotation of the human face is less than 45°. The two eye of the face should have similar orientation and be of similar size. Moreover, the inter distance between the two eye candidates should not be less than a threshold. If the face size is too small compared to the image size (Figures 6-a, 6-b), face location becomes impossible.

Experiments carried out with a great number of persons at different positions from the camera enabled us to determine a threshold of the face/image surface ratio (F_s / I_s) below which the face location becomes difficult as shown in Figure 7. In cases were the ratio $F_s / I_s \geq 1/15$, face location is nearly perfect as shown in Figures 8-a, 8-b.

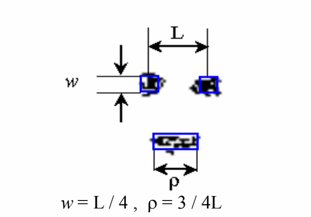
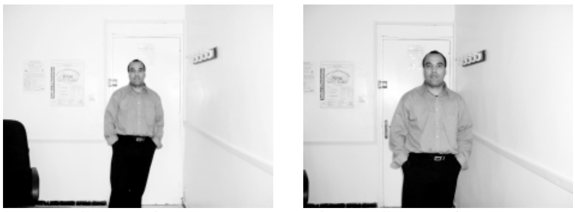


Figure 5. Face model drawing.



(a) 3m distance from camera.

(b) 2m distance from camera.

Figure 6. Failed face location.

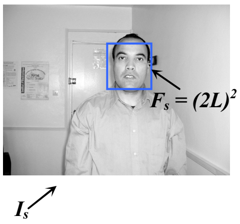


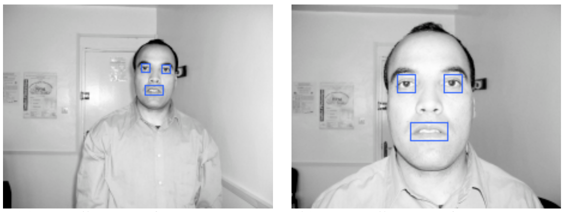
Figure 7. Face/image surface ratio.

Even when respecting the face/image surface ratio threshold, the face location could be erroneous if the lighting of the scene is not uniform. Indeed, a same scene induced different results in natural light and in artificial light; shade of elements induce wrong contours notably for objects in movement. A uniform lighting of the scene as well a smoothing operation of acquired images are therefore necessary if we want to have a robust and precise face location.

6. Hardware Implementation

A system designer is constantly faced with a tradeoff between performance and generality: A digital system designed to handle different types of tasks is usually slower than a system tailored to a specific application. This is the reason why computationally intensive tasks are handled by dedicated system architectures usually implemented in Application Specific Integrated Circuits (ASIC's). Although they offer more than adequate performance, ASIC's often lack the ability to adapt themselves to a changing environment. Moreover, ASIC design has high non-recurring costs and requires large engineering effort. A new methodology for the implementation of digital logic circuits, based on Field Programmable Gate Arrays (FPGA), emerged during the mid 80's.

The basic architecture of a FPGA consists of a large number of Configurable Logic Blocks (CLB) and a programmable mesh of interconnections. Both the function performed by the logic blocks and the interconnection pattern can be specified by the circuit designer. FPGA allow large-scale parallel processing and pipelining of data flow. Latest FPGA provide enormous processing resources, significant on-chip RAM and support very high clock speeds [1, 5, 10]. FPGA are therefore suitable for implementing real time image processing algorithms.



(a) 1m distance from camera. (b) 0.5m distance from camera.

Figure 8. Successful face location.

6.1. Implementation Details

The complexity of the system dictated the use of modern logic synthesis tools throughout all design phases. Synopsys FPGA compiler has been used for most part of the logic design starting from descriptions specified in the VHDL hardware description language.

Design of FPGA based computing machines is carried out through the following steps:

1. Definition of data processing resources;
2. Definition of memory resources;
3. Design of the control logic.

The result of the first step depends on a tradeoff between the complexity of the algorithm and the available logic and communication resources, e.g. the number of FPGA devices, their capacity, and the number of interconnection lines between programmable devices. The processing resources required by this application (adders, subtractors, comparators…) are automatically inferred by the synthesis tool from VHDL code and mapped to elements of the Xilinx XBLOX library. The basic memory resource required by any FPGA based image processing system operating on pixel neighborhoods is the pixel line delay, that is usually implemented either with external RAM memory devices or with FPGA internal memory. The Xilinx Virtex FPGA architecture, in fact, allows to configure every CLB as a 32x1 bit shift register, and several registers can be cascaded in order to build FIFO memories of various widths and depths. The integration of the FIFO on the same piece of silicon does not only mean increased performance but also a decrease in the number of external components [8, 12, 13]. The proposed operator has been validated on a real time image development system as shown in Figure 9 based on series 150/151 boards, compatible with the VME bus [4]. FPGAs which are cabled on our specific applications card use various initialization and framing signals in order to operate in the field of validity of the video signal. The general structure of the moving segments detector is presented in Figure 10. Three memory plans (F_1, F_2 and F_3) of the Frame Buffer card (FB) are used in cascade to store the binary edges images G_{n-1}, G_n and G_{n+1} required by our face location operator.

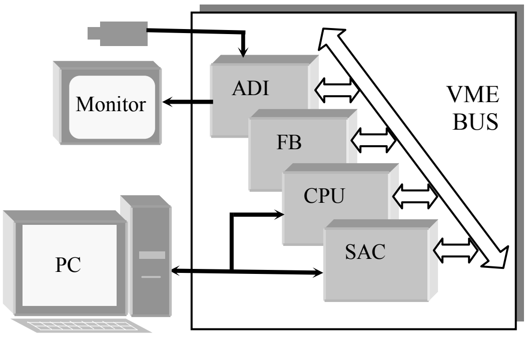


Figure 9. Development system.

The edges detector module allows to extract binary edges. The obtained three consecutive binary edges pictures (G_{n-1}, G_n, G_{n+1}) are used to perform the moving segment detection operation. In order to perform a real time processing on 3x3 size masks, two delay lines and six data registers are necessary for each output memory plan, allowing thus a simultaneous access to the nine binary edges data of the processed masks necessary for a moving segment detection.

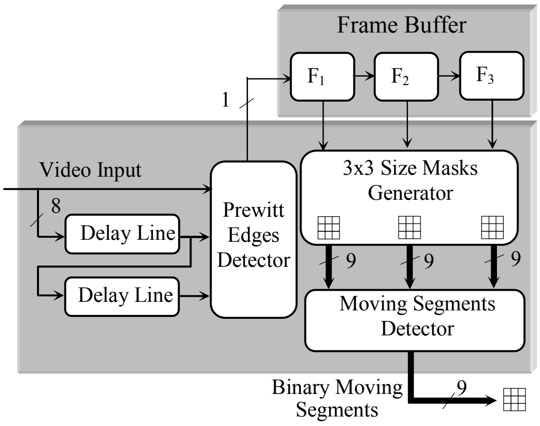


Figure 10. Moving segments detector.

The face location detector has been implemented within FPGA Virtex XCV400 [8]. The cost in surface evaluated by the number of CLBs used depends directly on the input pixels format as shown in Table 1.

Table 1. Implementation costs.

Pixels Format	Gray Levels	Surface Cost (Clbs)
6 bits	64	350
7 bits	128	512
8 bits	256	814

6.2. Real Time Tests

Some real time results obtained with our system are presented in Figure 11. It is important to note that the magnitude of the additive noise is largely linked to the type of chosen lighting, artificial lighting Figures (11-a, 11-b) or daylight Figures(11-c, 11-d). As mentioned before, shade of objects induce wrong contours notably for objects in movement. Presently the system allows a real time moving segments extraction at areas of interest (mouth and eyes). Face model drawing is done by the 68030 CPU card.

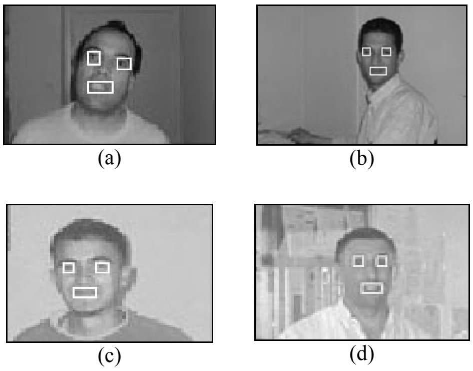

(a) (b) (c) (d)

Figure 11. Real time snapshot results.

7. Conclusion and Perspectives

In this paper, we have presented the design and implementation of a real time moving face tracker within a Virtex FPGA. This operator is based on the integrated use of both spatial and temporal information of the video sequence. Experimental results indicate that the proposed operator and refinement based on moving segments detection significantly reduce the noise effect and allow a robust real time face location. The reconfigurable aspect of FPGA circuits has enabled us to obtain a great flexibility in the adjustment of the architecture and the algorithm to implement. Future areas of development are the incorporation of the developed operator into a multimodal face recognition system and the migration towards ASICs for a final system implementation.

Acknowledgment

The authors would like to thank the referees for providing constructive comments.

References

[1] Dehon A., "The Density Advantage of Configurable Computing," *IEEE Transactions on Computers*, vol. 33, no. 4, pp. 41- 49, 2000.

[2] Gonzalez R. C. and Woods R. E., *Digital Image Processing*, Addison-Wesley, 1992.

[3] Haynes S. and Jain R., "Detection of Moving Edges," *Computer Graphics Image Processing*, vol. 21, pp. 345-367, 1983.

[4] Imaging Technology Series 150/151 Boards, 1990.

[5] Nozal L., Aranguren G., Martin J., and Ezquerra J., "Moving Images Time Gradient Implementation Using RAM Based FPGA," *in Proceedings of SPIE*, San José, California, vol. 3028, pp. 108-116, 1997.

[6] Orkisz M., "Localisation d'Objets Mobiles dans des Scènes Filmées par une Caméra Fixe," *Revue Traitement du Signal*, vol. 9, no. 4, pp. 325-346, 1992.

[7] Rogerio S. F., de Campos T. E., and Cesar R. M., "Detection and Tracking of Facial Features in Video Sequences," *Lecture Notes in Artificial Intelligence*, vol. 1793, pp. 197-206, April 2000.

[8] Sawyer N., *Self Addressing Fifo*, XAPP 291, v1.1, 2002.

[9] Stelmaszyk P., Bonnet P., and Postaire J. G., " Analyse de Scènes Dynamiques Par Recherche des Contours en Mouvement," *Congrès Reconnaissance de Formes et Intelligence Artificielle AFCET / ADI / INRIA*, Grenoble, France, pp. 1181-1189, 1985.

[10] Valls J., Kuhlmann M., and Parhi K. K., "Efficient Mapping of CORDIC Algorithms on FPGA," *in Proceedings of the 2000'IEEE Workshop on Signal Processing Systems (SiPS) Design and Implementation*, Lafayette, LA, pp. 336-345, 2000.

[11] Vieren C., Postaire J. G., Bonnet P., and Deparis P., "Détection du Contour Extérieur d'Objets en Mouvement sur Fond non Uniforme," *in Proceedings of Colloque GRETSI Traitement du Signal et ses Applications*, Juan les pains, pp. 621-624, June 1989.

[12] Xilinx, *The Programmable Logic Data Book*, Xilinx Book Inc, 2001.

[13] Xilinx, *Virtex Series Configuration Architecture User Guide*, available at: http://www.xilinx.com/ xapp/xapp151.pdf, v1.5, September 2000.

[14] Yang J. and Waibel A., "A Real Time Face Tracker," *in Proceedings of the 3rd Workshop on Applications of Computer Vision*, pp. 142-147, 1996.

[15] Yang J., Stiefelhagen R., Meier U., and Waibel A., "Visual Tracking for Multimodal Human Computer Interaction," *in Proceedings of SIGCHI'98*, pp. 140-147, 1998.

Messaoud Mostefai received his PhD in automatic and digital signal processing from the University of Reims, France, in 1995. His research interests include flexible and real time embedded systems, digital signal processing, and video coding algorithms.

Mechta Djamila received her Master in software engineering from UFA Setif, Algeria, in 2004. Her research interests include face recognition and multimodal identification. Currently, she is working toward a doctorate degree in biomedical image processing.

Youssef Chahir received his PhD degree in computer science from the University of Lyon, France, in 2000. His research interests include motion detection, automatic video segmentation and indexing, and image splitting and clustering.