Best-Shot Selection for Video Face Recognition Using FPGA

Leonardo Chang[1], Ivis Rodés[1], Heydi Méndez[1], and Ernesto del Toro[2]

[1] Advanced Technologies Application Center, Cuba
{lchang,irodes,hmendez}@cenatav.co.cu
[2] Microelectronics Research Center, CUJAE, Cuba
ernesto@electrica.cujae.edu.cu

Abstract. Video face recognition is widely used for security surveillance and other applications in which the information about faces is extracted and processed. One of the problems usually present in video face recognition is to determine in real time the suitable images for the good performance of the algorithms, taking into account that although computers keep getting faster, the amount of information to process is higher than the capacity of image processing algorithms available. In this work we propose a method that allows to obtain in real time the best image of each person present in the scene from a sequence of images, considering both image and face characteristics, and using FPGA technology to accelerate the image processing. With the proposed implementation of the method we managed to process 37 times more images per second, and 97% of the selected images proved to be adequate for face recognition.

Keywords: FPGA, best-shot selection, video face recognition.

1 Introduction

Face recognition is one of the most used biometric techniques. Among its greater advantages as opposed to other biometrics methods, is that it is a noninvasive technique, reason why it is widely used in concealed applications, like for example monitoring and security, in which people do not know that they are being subject of an automatic process of identification and a fast answer of the systems is needed.

According to the test report FRVT 2006 [1] a significant increase in the recognition rate with respect to the reports of the 2002 was detected [2]. This increase was achieved by face recognition algorithms that has as input high-resolution still and 3D images, however face recognition in videos remains one of the most difficult challenges in computer vision. In face video applications an important problem is the low quality of the input images due to the small video resolution and the pose and expression variations of the subjects; other key factor that leads to undesired images is the blur, generally caused by motion of the subject when the exposure time of the camera is not short. In face recognition, the better the input image the better the performance of the algorithms. Pose variations, is one of the elements that affects the most the

J. Ruiz-Shulcloper and W.G. Kropatsch (Eds.): CIARP 2008, LNCS 5197, pp. 543-550, 2008.

accuracy of face recognition algorithms, demonstrating better results when images have faces with rotation angles smaller than 10° [2]. On the other hand, one of the aspects pointed as unacceptable in an image for face recognition is when the eyes are not open [3].

In many face applications like video surveillance systems, the rising number of subjects to control and the quantity of tasks that are necessary to realize in real time can cause that the traditional solutions are not computationally viable. A solution that a large number of research has successfully applied to accelerate computing applications is the use of field-programmable gate arrays (FPGAs). An FPGA is an array of bit-processing units whose function and interconnection can be programmed after fabrication [4]. In the last decade they have experienced a growing interest because in less than 15 years they have become 40 times faster, have increased their capacity (number of available logic cells) 200 times and expend 50 times less power, together with a 500 times reduction of cost [5]. This technology presents potentially a second order-of-magnitude advantage in computational density over conventional processors; it also allows controlling operations at bit level and one of its principal advantages is the capability to realize parallel processing hardware [4].

In this paper we propose a method that allows obtaining the best face image from a set of images of a person, taking into account important properties in an image for face recognition. We also present an efficient hardware implementation in FPGA that supports higher computational density and video frame rate, giving a fast answer in real time. This paper is organized as follows: Section 2 describes the proposed method. The implementation in FPGA is explained in Section 3. In Section 4 experimental results and performance analysis of the FPGA implementation are presented. Finally, Section 5 concludes the paper.

2 Best-Shot Selection

Many face recognition systems that have as input a sequence of images, make the comparison against all the input images, or need to select manually the best image [6] [7]. However, the need to process a lot of images has motivated the creation of methods that obtain the best image or a set of images from the sequence that better represents the persons.

In [8] and [9] the authors present methods to select the best-shot in a video sequence, both of them are based on the face pose estimation and do not think over the quality of the image as such. In [10] a method that analyzes objective image characteristics and subjective characteristics of faces to select the most appropriate face images is proposed; they consider important elements for the image such as contrast and introduce a facial expression recognition algorithm but forget an important aspect for face recognition: pose.

We propose a method that allows to obtain in real time the best image of every person present in the scene from a sequence of images, considering important characteristics so much for the image as well as for the face. For the image quality we measure the sharpness as a significant property, while for face we take into account the face pose, occlusions and the openness of the eyes.

The proposed method can be seen in Fig. 1. By means of a camera, a sequence of images of a scene is captured. Each person in the scene is tracked, their faces and eyes locations are retrieved in each frame. In real time, using FPGA technology, for each subject the face image with the best combination of frontal position, no occlusions and open eyes is determined, taking into account that the images are not blurred.

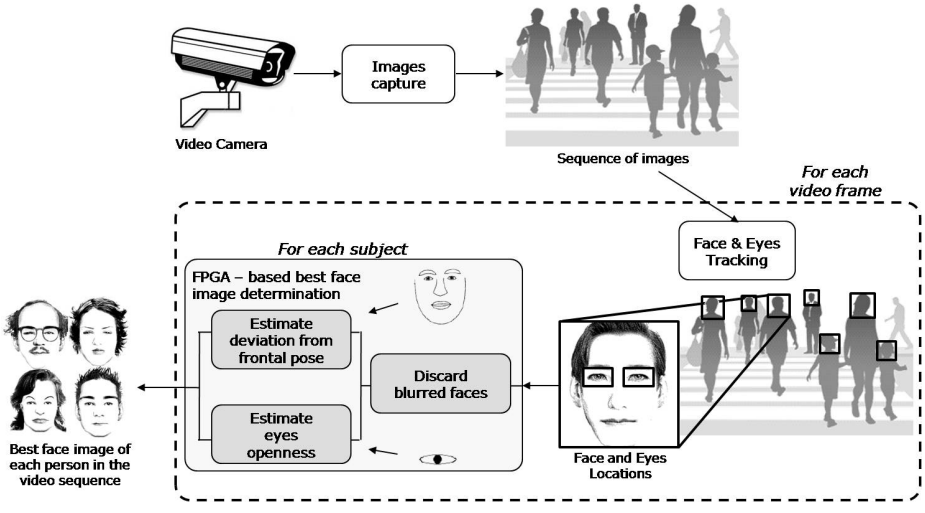

Fig. 1. Proposed method for obtain the best image for face recognition

As can be seen in the figure above, for each face image of every person, three processes are executed to measure how suitable are for face recognition and to determine the best of them.

The first process is to discard blurred faces. The other two are based on template matching, one against a frontal face template and the other against an open eyes template. In the following we will see these processes in more details.

2.1 Discard Blurred Faces

When motion blur occurs, the edges of the image lose their sharpness. In a face image the edges are among the most important features [11], they are principally present in eyes, eyebrows, nostrils and mouth, then a blurred image is not adequate for face recognition.

The mean filter is the simplest type of low-pass filter which is defined by the selected kernel width, height and shape. It is often used to smooth images by averaging surrounding pixel values with a simple $w \times w$ mean rectangular filter. An image with fine details and edges has a lot of pixels in which their neighbors are very different from them; therefore if we replace each pixel with the mean value of the intensities of the surrounding pixels, the smoothed image is quite different from the original. On the other hand, if the image is blurred the pixel intensities in a little neighborhood are very similar and the filtered image is very similar to the original.

We can then determine if an image is blurred or not using the 3×3 mean filter: where f is the original image and f' the filtered image, we subtract the filtered image to the original one, being $D = \text{abs}(f - f')$ the difference image; the higher the average pixel value over image *D*, the better focused is the original image. Then we discard those face images with lowest values over *D*.

2.2 Template Matching

With the aim of obtaining an estimation of the pose of the face and the state of the eyes a global approach was used. One principal advantage of this kind of method is that only the face and eyes need to be located, no facial landmarks location, training or models are required. Besides, global approaches can manage very low resolution images. Template matching is a popular global method to estimate head pose and eyes state, we specifically employ normalized cross-correlation, given by:

$$\gamma = \frac{\sum_{x,y}[f(x,y) \cdot t(x,y)]}{\sqrt{\sum_{x,y} f(x,y)^2 \sum_{x,y} t(x,y)^2}} , \quad (1)$$

where t is the template, f is the input image, *x* and *y* are the coordinates of each pixel of the images, and $0 \leq \gamma \leq 1, \gamma \in \mathcal{R}$ the correlation value. As higher γ be, stronger will be the relation between the input image and the template.

One of the main drawbacks of this method as template matching method is that it does not present good results when partial or total occlusions are presented in the objects. However in the proposed method this is used in a favorable way, to discard face images with occlusions, since the correlation value between face images with occlusions against a frontal face template will be low.

Once the correlation of all the input faces images against the frontal face template and the correlation of all the eyes within these faces against the open eyes template are obtained, the best face image selected will be the one whose combination of both correlations is the greatest.

3 FPGA Implementation

We propose the use of FPGA technology to improve image analysis and be able to process a higher number of frames. Some hardware architectures have been presented in literature aimed at accelerating image processing methods [12] [13] [14] but no one intended to perform best-shot selection. In this section we expose the proposed memory organization and addressing schema and also some details about the hardware architecture for our real time best-shot selection method. Our proposal leads to achieve a higher computational density with a higher frame rate, being this a contribution to real time video face recognition. Since there is a big difference between processors speed and memory access time, the last is a crucial issue to deal with. The presented memory schema aims at reducing the number of memory accesses and encouraging data parallelism, which is exploited by the proposed pipelined hardware architecture.

3.1 Memory Organization and Addressing Mode

Since the process of discarding face images blurred by motion is the most memory intensive task we will concentrate the explanation of our memory access and organization based on this method.

In a traditionally pure sequential implementation of the mean filter, the $w \times w$ mask is slide horizontally across a given row until its end is reached, then it is moved to the next row and this process is repeated until the last row is processed. Using this implementation over an image with a bitmap representation in memory, each pixel is read several times where the total amount of memory accesses to treat the entire image is roughly given by:

$$s_{MA} = w^2 \times M \times N\,, \tag{2}$$

where $M \times N$ are the image dimensions. Under this memory organization and according to eq. 2, the number of memory accesses, in an $M \times N$ image, will increase quadratically in relation to the mask width.

Considering that each pixel value is used several times due to windows overlapping, in order to reduce the number of memory accesses and taking advantage of the intrinsically data parallelism involved in these processes, we proposed the following image organization in memory.

The image is divided and addressed in windows of $w \times w$ pixels. Each window occupies one memory location of $w \times w \times 8\ bits$ width, Fig. 2 (a) shows an example. The memory is sequentially ordered from left to right, along each row, and continuing in the following one. If there are remaining pixels rows or columns from this division, extra $w \times w$ windows rows or columns are used, tagging the complementary pixels. With this representation all the information needed to obtain one pixel mean filter value is stored in a fixed maximum number of four memory locations, as can be seen in Fig. 2 (c), against w^2 locations with the bitmap representation, which is one of the contributions of this work.

An efficient memory addressing schema is needed to allow the execution of several operations in parallel, taking into account that memory reading is a sequential process. Under the proposed memory organization the addressing mode used is shown in Fig. 2 (b) and is given by the following indexes sequence, where the indexes are taken from the traditional matrix representation of an image:

$$1{,}1 \rightarrow 1{,}2 \rightarrow 2{,}1 \rightarrow 2{,}2 \rightarrow 2{,}1 \rightarrow 3{,}2 \rightarrow \cdots \rightarrow m, 1 \rightarrow m, 2 \rightarrow 1{,}2 \rightarrow 1{,}3 \rightarrow 2{,}2 \rightarrow 2{,}3 \rightarrow 3{,}2 \rightarrow 3{,}3 \rightarrow \cdots \rightarrow m, 2 \rightarrow m, 3 \rightarrow 1{,}3 \rightarrow 1{,}4 \rightarrow \cdots \rightarrow 1, (n-1) \rightarrow 1, n \rightarrow 2, (n-1) \rightarrow 2, n \rightarrow 3, (n-1) \rightarrow 3, n \rightarrow \cdots \rightarrow m, (n-1) \rightarrow m, n$$

where m and n are the M'^{-th} and N'^{-th} pixels of the new $M' \times N'$ representation.

In terms of memory accesses, this means that each pixel is read only twice, excepting the pixels of the first and last row, which are read once. So, according to our proposed memory addressing schema, the total number of memory accesses needed to process the entire image is given by:

$$n_{MA} = 2 \times M' \times (N' - 1)\,. \tag{3}$$

According to eq. 3, the number of memory accesses to process an $M \times N$ image will decrease as the size of the mask is increased.

An important contribution of the proposed memory organization and addressing mode and according to eq. 2 and eq. 3, is that the number of memory accesses is reduced by a factor of $\frac{w^4}{2}$ in relation to the traditional sequential implementation.

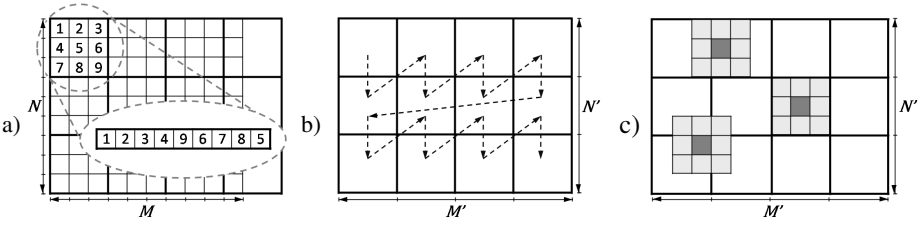

Fig. 2. Example of (a) memory organization for a 3 × 3 mask, (b) consequential memory addressing sequence and (c) amount of memory addresses needed for different positions

3.2 Pipelined Architecture

The proposed architecture uses a $w \times w$ matrix addressable memory to give one pixel sharpness result per clock cycle. For explanation purposes, the designed architecture will be described assuming a 3 × 3 mask for the mean filter, which is the one used by our sharpness calculation method, but this architecture could be generalized for other masks and image sizes.

For sharpness calculation, four unsigned adders perform the first eight pixels operations and the two respective results are registered for the next stage. A second addition is performed in the next clock cycle and the result added with the remaining pixel completing the sum of nine pixels in four clock cycles. We use a pipelined divider so it will output the result of one division operation per clock cycle after an initial latency of fourteen cycles returning mean filter result. Next, a subtraction module is connected. An accumulation of all the mean filter results and a final division according to the image height and width, match the sharpness calculation for the processed image. The logic diagram is shown in Fig. 3.

We could make throughput estimation for sharpness calculation in order to evaluate the architecture performance in terms of architectural parameters. Due to the pipelined architecture, the result for one pixel mean filter operation is delivered to the accumulator every clock cycle after an initial latency given by $t_l = t_{ps} + t_{pd}$ where

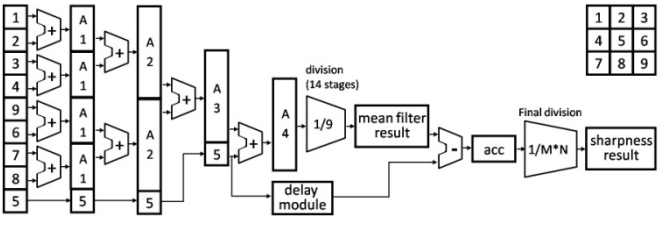

Fig. 3. Logic diagram of the proposed pipelined architecture

t_{ps} and t_{pd} are the adders and divider pipeline delay respectively. After this, the accumulator delay t_a is one clock cycle period and the final divider latency t_d is a function of the bit width d_w of the dividend and is defined as $t_d = (d_w + 2) \times t_{clk}$ where t_{clk} is the clock period. The overall time to get sharpness result of an entire image could be expressed as:

$$\tau = t_l + (\rho \times t_{clk}) + t_a + t_d \,, \tag{4}$$

where ρ is the total number of one pixel mean filter operations and is given by $\rho = M \times N$.

4 Experiments and Results

The performance of the blur detection method was tested with a database created by us with images captured under the motion of the subject. This database was composed of 410 face images of 10 different subjects where 232 were blurred. A false negative rate of 88,2% and a false positive rate of 81,9% was achieved.

To prove the effectiveness of the proposed best-shot selection method, we used it in real scenarios, capturing 500 sequences of about 100 frames of face images of different subjects. We could verify that less than the 2% of the obtained images presented head rotations over 10 degrees. However in those images the variations were smaller than 20 degrees and the eyes were perfectly open. On the other hand, the 99% of the captured faces were with perfectly open eyes and none of them was with completely closed eyes. None of the selected images were blurred. In summary, more than the 97% of the captured images fulfills the requirements for face recognition.

The simulation of the proposed architecture design was performed using Xilinx's ISE software with ModelSim. The FPGA chip targeted is a Virtex-II Pro XC2VP30-4FF1152. In all modules, we use relationally placed macro mapping and placement technology for maximum and predictable performance and future floorplaning. Using a clock constrain of 4.5 ns the design occupies 891 Slices and 1383 Flip Flops. This represents 6% of available resources.

Under this implementation, with a 50 MHz clock rate it is possible to process 780 images of 320×200 size in a second, which is equivalent to say that the best face image of 39 subjects, even from different cameras sources, with a frame rate of 20 fps could be determined. This result was compared with a traditional sequential software implementation in C++. Our FPGA implementation resulted 37 times faster. The efficiency of this FPGA implementation is related to the targeted hardware and the synthesis and place and route tool used.

5 Conclusions

This paper has proposed a method that allows obtaining in real time the best image of every person present in a sequence of images of a scene, considering important characteristics so much for the image as well as for the face, proving an accuracy of 97%. We also introduced the utilization of FPGAs as a novel technique in the solution

of real time video face recognition problems; for this purpose we presented an efficient and flexible hardware implementation for best-shot selection, which demonstrates to handle higher computational density in shorter times than conventional processors. The FPGA implementation exhibited that it could handle 39 subjects with 320×200 size faces at a frame rate of 20 fps.

References

1. Phillips, P.J., Scruggs, W.T., O'Toole, A.J., Flynn, P.J., Bowyer, K.W., Schott, C.L., Sharpe, M.: FRVT 2006 and ICE 2006: Large-Scale Results. NISTIR (2007)
2. Phillips, P.J., Grother, P.J., Micheals, R.J., Blackburn, D.M., TabassiI, E., Bone, J.M.: Face recognition vendor test 2002: Evaluation report. NISTIR (2003)
3. International Committee for Information Technologies and Standards: Face Recognition Format for Data Interchange. INCITS Secretariat, Information Technology Industry Council (2006)
4. DeHon, A.: The density advantage of configurable computing. IEE Computer 33(4), 41–49 (2000)
5. Mühlbauer, F., Bobda, C.: A Dynamic Reconfigurable Hardware/Software Architecture for Object Tracking in Video Streams. EURASIP Journal on Embedded Systems 2006, 55–62 (2006)
6. Gordon, G., Lewis, M.: Face Recognition Using Video Clips and Mug Shots. In: Proceedings of the Office of National Drug Control Policy (ONDCP), International Technical Symposium, Nashua, NH (1995)
7. Zhang, Y., Martinez, A.M.: From Stills to Video: Face Recognition Using a Probabilistic Approach. In: Computer Vision and Pattern Recognition Workshop, vol. 27(02), p. 78 (2004)
8. Yang, Z., Al, H., Wu, B., Lao, S., Cai, L.: Face Pose Estimation and its Application in Video Shot Selection. In: Proceedings of the 17th International Conference on Pattern Recognition, pp. 322–325 (2004)
9. Takuya, O., Lao, S.: Best-shot Selection from a Video Sequence of Face Images: An Implementation using Boosting Regression Algorithm. Omron Tech. 44(1), 17–20 (2004)
10. Shotaro, M., Kosuke, H., Shintaro, W., Hiroshi, K.: A Facial Expression-Oriented Best-Shot Face Recording System. Eizo Joho Media Gakkai Gijutsu Hokoku 29(74), 45–48 (2005)
11. Takacs, B.: Comparing face images using the modified Hausdorff distance. Pattern Recognition 31, 1873–1881 (1998)
12. Draper, B.A., Beveridge, R., Willem, Böhm, A.P., Ross, C., Chawathe, M.: Accelerated image processing on fpgas. IEEE Transactions on Image Processing 12(12), 1543–1551 (2003)
13. Torres, C., Arias, M.: FPGA-based Configurable Systolic Architecture for Window-based Image Processing. EURASIP Journal on Applied Signal Processing, Special Issue on Machine Perception on a Chip 2005(7), 1024–1034 (2005)
14. Chang, C., Hsiao, P., Huang, Z.: Integrated Operation of Image Capturing and Processing in FPGA. IJCSNS International Journal of Computer Science and Network Security 6(1), 173–180 (2006)