Optical Engineering 50(2), 000000 (February 2011)

Palmprint and face score level fusion: hardware implementation of a contactless small sample biometric system

Audrey Poinsot
Fan Yang
Vincent Brost
Q1 University of Burgundy
Le2i Laboratory
Batiment Mirande, Aile de l'Ingenieur, 9
BP 400, Dijon, 21000 France
E-mail: audrey.poinsot@u-bourgogne.fr

Abstract. Including multiple sources of information in personal identity recognition and verification gives the opportunity to greatly improve performance. We propose a contactless biometric system that combines two modalities: palmprint and face. Hardware implementations are proposed on the Texas Instrument Digital Signal Processor and Xilinx Field-Programmable Gate Array (FPGA) platforms. The algorithmic chain consists of a preprocessing (which includes palm extraction from hand images), Gabor feature extraction, comparison by Hamming distance, and score fusion. Fusion possibilities are discussed and tested first using a bimodal database of 130 subjects that we designed (uB database), and then two common public biometric databases (AR for face and PolyU for palmprint). High performance has been obtained for recognition and verification purpose: a recognition rate of 97.49% with AR-PolyU database and an equal error rate of 1.10% on the uB database using only two training samples per subject have been obtained. Hardware results demonstrate that preprocessing can easily be performed during the acquisition phase, and multimodal biometric recognition can be treated almost instantly (0.4 ms on FPGA). We show the feasibility of a robust and efficient multimodal hardware biometric system that offers several advantages, such as user-friendliness and flexibility. [DOI: 10.1117/1.3534199]

Subject terms: multimodal biometrics; face recognition; contactless palmprint recognition; small-number sample sets; palm codes; score fusion; hardware implementation.

Paper 090932RRR received Nov. 24, 2009; revised manuscript received Nov. 16, 2010; accepted for publication Dec. 7, 2010; published online Feb. 00, 2011.

1 Introduction

Biometrics has drawn extensive attention during the past 30 years for its huge potential in many applications, such as building/store access control, suspect identification, surveillance, and human computer interfacing. The key issue of these applications is the identification of individuals by their physiological or behavioral characteristics (e.g., face, fingerprint, iris, signature, or gait). Each biometric characteristic has its own strengths and weaknesses: unimodal biometric systems have to contend with a variety of problems, such as noisy data, nonuniversality, spoof attacks, and unacceptable error rates. In the past few years, researchers have more and more focused on the possibility of including multiple sources of information. Such systems, known as multimodal biometric systems, are more reliable.[1]

In many real-world applications, the number of available training samples is small, especially in the case of large-scale biometric systems. Typically, for the face recognition problem in identity documents, the number of images from each class is considerably limited: only one or two faces can be acquired from each person. Moreover, systems using less training samples have a shorter enrollment stage and are more pleasant for users. A small number sample sizes allows us to use little memory. Nevertheless, in the small-number sample context, many statistical methods show poor generalization ability and degrade the classification performance.[2] In this paper, a reliable and contactless general-public multimodal biometric system is presented. It respects the small-number sample constraint and tries to be user-friendly.

Palmprint can be used as a reliable human identifier because the pattern of ridges is unique and their details are permanent. Compared to other physical biometric characteristics, palmprint biometrics have several advantages: low-intrusiveness, stable line features, and low-cost capturing device.[3] Although palmprint is traditionally a contacting biometric, we use it without contact, which allows us to keep a pleasant and hygienic system. For that matter, an increasing number of works have interest in the use of contactless sensors.[3–5]

Face is one of the most studied and commercialized biometrics. It is well accepted because humans routinely use facial information to recognize each other. But it suffers from some weaknesses: it is particularly affected by pose, expression, or illumination. In the past decades, a lot of face recognition algorithms have been proposed: statistical analysis as principal component analysis (PCA), independent component analysis (ICA), or linear discriminant analysis (LDA);[6] neural networks;[7] graph matching;[8] etc.

Fusion of face and palmprint is studied because it allows are to greatly improve performance while keeping a user-friendly and well-accepted system. Kumar and Zhang[9] proposed a personal verification method combining

0091-3286/2011/$25.00 © 2011 SPIE

palmprint, face, and claimed user identity to increase authentication performance: a feed-forward neural network is used to integrate individual matching scores and generate a combined decision score. Jing et al.[6] use face and palmprint for small-number sample recognition: the fusion occurred at the pixel level on feature images is obtained due to a Gabor filter bank. Zhang et al.[10] present a geometry preserving projection (GPP) approach to preserve the interactions between the different modalities during the subspace selection procedure: with GPP, all raw biometric data (face, palmprint obtained with contact, and gait) from the different identities and modalities are projected onto a unified subspace, on which classification is performed.

However, none of those methods are adapted to the calculation cost or memory constraints of embedded systems. Biometric algorithms work on raw and uncompressed images, whose processing requires a large number of operations. However, most of these operations are independent and can be performed on different parts of the image at the same time. Because of this possibility of reaching a high parallelism degree, biometric algorithms are the right candidates for hardware implementation. For example, some research has been conducted in order to reduce the calculation time of monomodal biometric systems: Yang and Paindavoine[11] have implemented a face-detection and recognition algorithm—based on radial basis function (RBF) neural network—on field-programmable gate array (FPGA), digital signal processor (DSP), and zero instruction set computer (ZISC) chips in order to compare the execution times. Lopez-Ongil et al.[12] present the FPGA implementation of an authentication system based on hand geometry, which uses the continuous hamming distance to compare hand dimension vectors. Other works explore multimodal biometrics: Yoo et al.[13] have developed two DSP systems for iris-fingerprint and face-fingerprint recognition. In their system, the most consuming tasks are implemented on FGPA in order to increase the system speed.

The aim of our project is to build a reliable general-public biometric system, that respects multiple constraints: hygienic, low-cost, straightforwardness, user-friendliness, real-time processing, limited memory, small sample set, etc. The developed system could be used in businesses, hospitals, or schools to control door opening, record hours worked by employees, restrict access to sensitive areas, control access to school canteens, etc. Therefore, we present the hardware architecture of a multimodal biometric recognition system with massive exploitation of the inherent parallelism. Implementations are simulated on a Texas Instrument Digital Signal Processor (DSP) and Xilinx Field Programmable Gate Array (FPGA) platforms. DSPs are widespread processors that are optimized to signal processing, whereas FPGAs are inexpensive devices adapted to parallel calculation that give the ability to quickly create a rapid and fully functional prototype that can emulate and verify solutions or even be embedded into the final system. That is why we chose to implement our algorithm on these two devices. The remainder of the paper is organized as follows: Section 2 provides details of the algorithm model from image acquisition to the steps of fusion and decision, while Sec. 3 presents designed architectures and their hardware implementations. Performance of the system is presented in Sec. 4 and discussed in Sec. 5. This is followed by the conclusion and presentation of the perspectives in Sec. 6.

2 Algorithm Model

This section introduces the complete face and hand processing chain, which includes four principal steps: acquisition of images, hand preprocessing, palmprint and face feature extraction, and score fusion. A brief algorithm-oriented presentation of all the modules is available in Ref. 14.

2.1 *Acquisition of Images*

Traditional hand-based biometrics use contact with a surface and sometimes rigid placement guides. These have the advantage of having a fixed focal field, and if they use pegs, can rely on a standard placement. On the contrary, face is a typical contactless biometric. We have designed a user-friendly system to acquire real-time hand and face images that is totally contactless. Two low-cost Logitech QuickCam Pro 9000 USB cameras are used with a maximum resolution of 1600×1200 to capture images under typical office lighting and daylight conditions.

Subjects enroll themselves thanks to an easily usable software. For the hand, they are only asked to place it horizontally and ensure that their fingers do not touch each other. Each subject could place his hand anywhere from a few dozen inches to a few inches from the sensor: the upper limit is defined by the position of a green background [see Fig. 1(a)]. Subjects must furthermore place their face in an enclosing frame of 360×480 pixels drawn on the webcam preview [see Fig. 1(b)]. Expression, accessories, and background are not controlled: expression can vary from neutral to broad grin, and subjects choose to wear their eyeglasses or not.

2.2 *Image Preprocessing*

Working on palmprint in a contactless context requires some preprocessing. The region of interest (ROI) must indeed be extracted from the hand image. Palm extraction requires hand localization, followed by palm localization in the hand, and

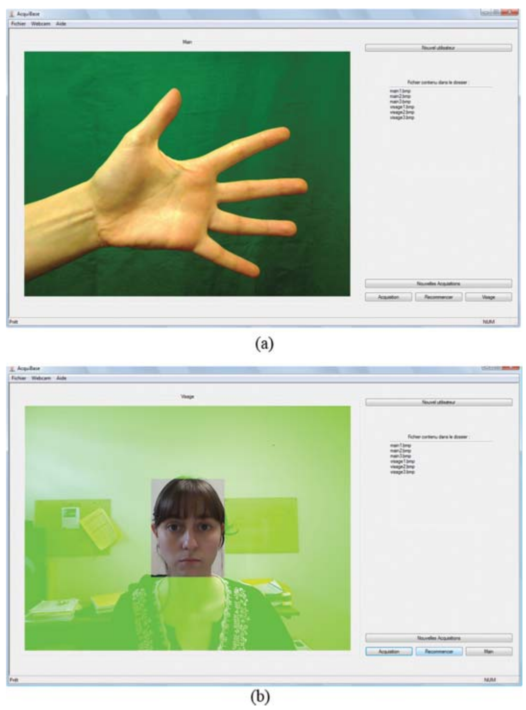


Fig. 1 Acquisition software: (a) palm interface and (b) face interface.

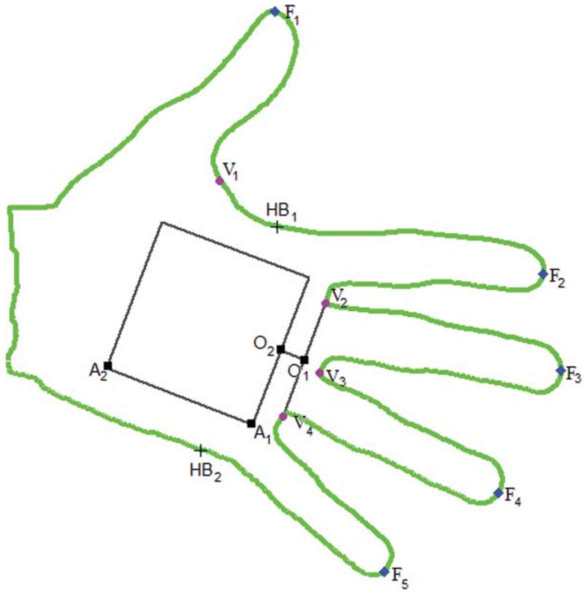


Fig. 2 Palm window definition.

then normalization because of the rotation and scale variation induced by the free placement. Hand segmentation consists of a thresholding on the red component of the RGB space: because a green background has been chosen, the redder pixels belong to the hand. Some morphological operations are also used in order to enhance the hand edges. After this step, multiple reference points are defined; they correspond to the fingertips and valleys between fingers. This localization of the hand extremities is achieved in two steps.

First, a contour extraction is performed using an eight-neighborhood-borders tracking algorithm known as the Freeman algorithm. Second, hand extremities's locations are found. As subject fingers are located on the right of the image, local minima and maxima of the hand contour abscissa can be considered as fingertips and valleys. Because these initialized locations are not accurate, we applied a refining algorithm inspired by the method described in Ref. 5, which minimizes the euclidean distance between the considered point and its two neighbors among the reference points.

Doublet et al.[4] propose a simple and efficient method to extract the palm from the location of such characteristic points. Our adaptation of this process consists of two steps: First, adding two characteristic points in order to calculate the hand width, and second localizing the palm window corners. Location of the new fiducial points is deduced from the length of the index and little fingers. Figure 2 shows the square window, which corresponds to the ROI. The distances $\|O_1O_2\|$ and $\|A_1A_2\|$ depend on the distance between the hand and the camera. Therefore, they are taken proportional to the hand width ($\|HB_1HB_2\|$).

Because the palmprint images are of different sizes and orientations, we normalize them. First, they are rotated around the vertical axis. Then, they are resized to a standard image size of 64×64 pixels and converted into a gray-level image.

Because of the experimental setup, the pose of the face varies only slightly. Moreover, as we work on low-resolution images, it is not necessary to extract ROI. That is why the face preprocessing only takes up the last palm preprocessing steps: resizing to 64×64 pixels and conversion into a gray-level image.

2.3 *Gabor Feature Extraction*

Palmprints exhibit a rich pattern of striations that enable discriminating between people. Therefore, most of the studies in palmprint recognition treat palmprints as textured images and apply well-known pattern recognition techniques, such as wavelets,[15] PCA or ICA,[16] and many others. Because of its good performance and specific qualities of luminosity robustness and frequency location, the Gabor filter is the most efficient and popular tool.[1,6,17]

Face recognition is a mature biometric for which many recognition approaches exist. Nevertheless, classical methods such as Eigenface or Fisherface are not adapted to the small sample set problem, as explained in Ref. 2 or 18. Therefore, many variants of these algorithms have been proposed in order to improve recognition performance in this situation.[19,20] Other methods, which combine image filtering by a Gabor filter bank and PCA (Ref. 6) or LDA (Ref. 21) have also been studied to solve the small-number sample set problem. However, all these methods based on statistical analysis require too high calculation complexity and too much memory to be used in embedded systems. However, some studies look into the use of one or more pertinent Gabor filters,[22,23] which is the same principle as our palmprint recognition algorithm.

Here, this filter is used to extract palmprint and face features: a coding-based method is employed, that is founded on the works of Refs. 4 and 24. This choice is also consistent with the electronic embedded system context: regular calculations, such as convolution operation, are easily implemented on hardware systems and reduce power consumption. Moreover, applying the same method on both palmprint and face will facilitate hardware implementations.

A variety of implementations of this filter exists. Considering its performance and the need to reduce computation time and memory consumption, we use the ellipsoidal filter in the real domain proposed in Ref. 4,

$$G(x, y) = \exp\left[-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right] \cos\left(2\pi \frac{0.56x'}{\sigma}\right), \qquad (1)$$

where

$$\begin{cases} x' = (x - x_0)\cos(\Theta) - (y - y_0)\sin(\Theta) \\ y' = (x - x_0)\sin(\Theta) + (y - y_0)\cos(\Theta) \end{cases}. \qquad (2)$$

The couple (x_0, y_0) defines the function center, Θ controls the orientation, σ is the standard deviation of the Gaussian factor, and γ is the spatial aspect ratio of this ellipsoidal function fixed at 0.5. For more luminosity robustness, the filter is normalized by the subtraction of the coefficient average from each coefficient.

Gabor palmprint features are obtained by the convolution of the image with a single Gabor filter (whose coefficients are empirically chosen, see Sec. 5), followed by a thresholding operation with a threshold equal to 0. This binarization limits the characteristic size and the computation time in the comparison phase. The feature extraction step is illustrated in Fig. 3. For identity classification and verification, a similarity measurement must be created in order to compare the extracted parameters. For this matching process, we use the traditional comparison method of binary matrices: the

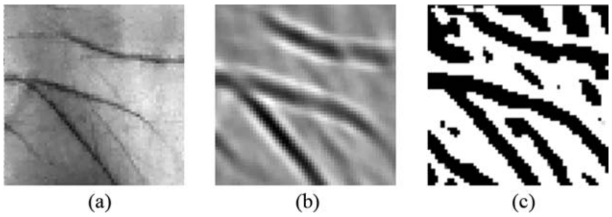


Fig. 3 Feature extraction of the palm: (a) corresponding images, (b) gabor features ($\Theta = \pi/4$, $\sigma = 4.6$), and (c) final feature matrix.

Hamming distance, which is a pixel-by-pixel comparison using the Boolean operator $\oplus$ (XOR).

Because palmprint and face localizations are not necessarily ideal, we introduce a tolerance in translation by calculating the distance for multiple shifts and taking the minimum. The final matching measurement for two feature matrices A and B of size $N \times N$ is

$$D(A, B) = \min_{|x|,|y|<2} \left[\sum_{i=0}^{N} \sum_{j=0}^{N} T\{A(i, j), x, y\} \oplus B(i, j) \right], \quad (3)$$

where $T\{A, x, y\}$ is the translation of image A horizontally by x and vertically by y.

2.4 *Fusion Scheme*

Combining one or more biometric traits provides new independent information that gives the opportunity to greatly improve recognition performance. Furthermore, it increases the probability that one of the traits suits the user, which gives a larger population coverage and complicates spoof attacks by requiring more kinds of information.

A generic biometric system includes four principal steps: data acquisition, feature extraction, matching to the template database, and decision. Information fusion can occur at any of the aforementioned steps. Most studies agree on the fact that integrating information at an early stage of processing is more effective than performing integration at a later stage.[1] Earlier stages contain richer information about the input biometric data than later stages. However, fusing pixels or feature vectors implies a high compatibility between fused data and does not allow modality-adapted processing, as in our case.

We use fusion at score level because there is sufficient information content at this step and it is easy to access and combine the matching scores. Savic and Pavesic[25] have demonstrated that the combination approach performs better in biometric systems. Therefore, tree combination rules have been tested. Let P_i be the score obtained thanks to the matching between the current palmprint feature and the ith template of the palmprint matching base, let F_i be the score obtained thanks to the matching between the current face feature and the ith face template, the corresponding final score Fus_i can be calculated from the minimum [Eq. (4)], the sum [Eq. (5)], and the multiplication Eq. (6) rules as follows:

$$\text{Fus}_i = \min(P_i, F_i), \quad (4)$$

$$\text{Fus}_i = P_i + F_i, \quad (5)$$

$$\text{Fus}_i = P_i \times F_i. \quad (6)$$

The final decision of the classifier is then given by choosing the class that minimizes the fused matching measures between the sample and all templates of the matching base.

If at least one of the two scores is low enough to success in the recognition task, the fused score (obtained by minimum, sum, or multiplication rules) would also allow one to succeed in this task. That is why multimodal systems outperform unimodal systems and increase the population coverage: If one modality is vulnerable to certain conditions, then the others take over. The way we designed the system (see Fig. 4) allows us, moreover, to use palmprint only, face only, or fusion of the two. Using this architecture makes it possible to add other textured modalities, such as knuckleprint or ear.

3 Hardware Implementations

Each of the proposed algorithms respects the embedded system constraints. They work in particular with a low calculation cost and low memory, which makes them particularly suitable for DSP implementation. Moreover the coding scheme proposes a high potential of parallelization, which could be fully exploited by application-specific integrated

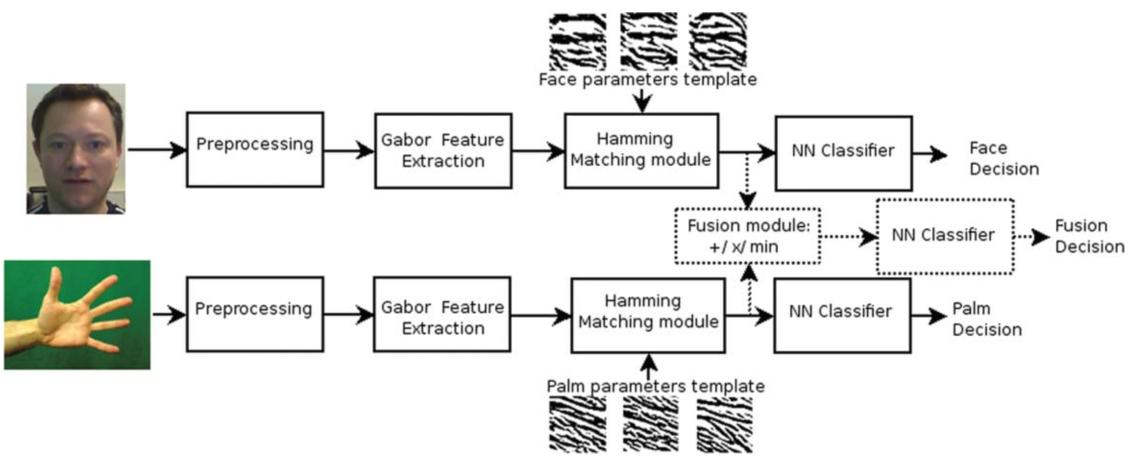


Fig. 4 Entire processing chain with possible score fusion using nearest-neighbor (NN) classifier.

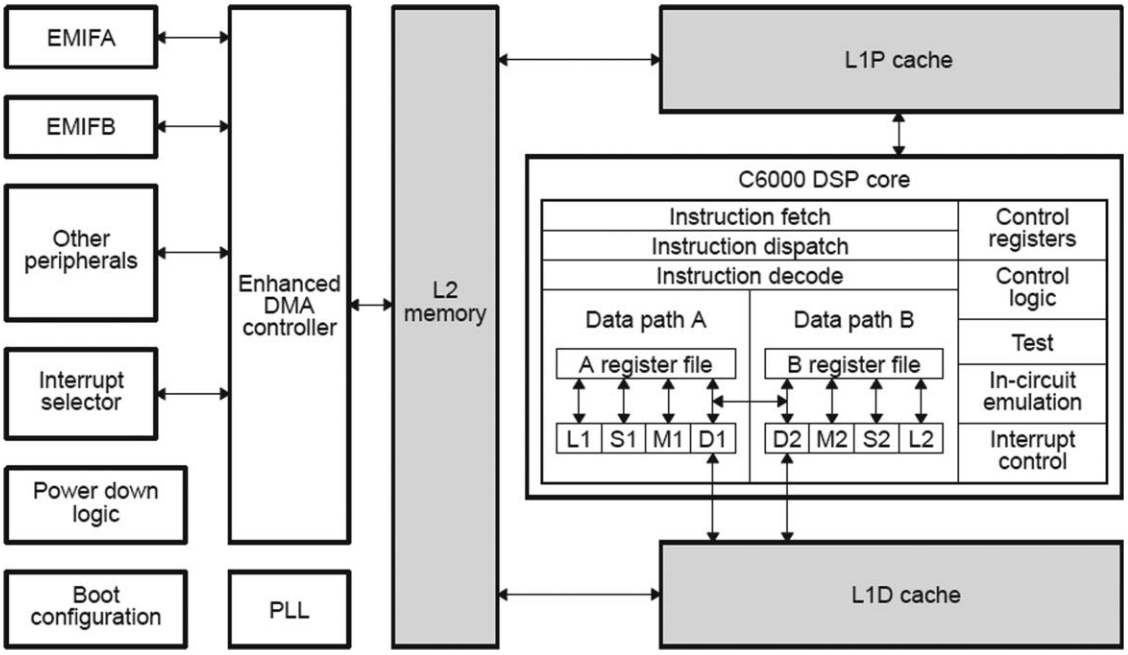


Fig. 5 TMS320C64x DSP block diagram.

circuit (ASIC) or FPGA. We propose the implementation of the entire system on a Texas Instrument DSP platform, and the implementation of the multimodal recognition step on a Xilinx FPGA platform. Implementing the proposed multimodal recognition chain in FPGA efficiently is the key step of an ASIC solution design.

3.1 *DSP Implementation*

The implementation of the processing chain has been simulated on a TMS320C64xx DSP platform of Texas Instruments[26] thanks to the Code Composer Studio (CCStudio) tool. Such platforms are particularly well adapted to classical image processing algorithms, and allow one, at the same time, to easily implement more sophisticated processing. The C64x central processing unit (CPU), as shown in Fig. 5, consists of eight functional units, two register files, and two data paths. Devices of the c64x family can execute, for example, four 16-bit$\times$16-bit multiplies every cycle, or eight 8-bit$\times$8-bit multiplies. They have a two-level memory architecture for program and data. The first-level program cache is designated L1P on Fig. 5, and the first-level data cache is designated L1D. Both the program and data memory share the second-level memory, designated as L2, which is configurable and can provide up to 1024 KB of on-chip SRAM. A DSP implementation description has been made in C language. After an optimization step, we let the compiler of the CCStudio environment decide the possibilities of parallelization.

The number of CPU cycles required for the palmprint extraction depends of the hand shape. Our simulations empirically show that it is between 350×10^6 and 390×10^6, which corresponds to 350 and 390 ms at a frequency of 1 GHz. With such a short execution time, palmprint extraction can be performed during face image acquisition. The face preprocessing always uses the same number of CPU cycles, which is lower than 6×10^6 and corresponds to an execution time of 6 ms. Face preprocessing could run in real time while storing pixels. For these first hardware implementations, we have chosen to work on a database of 25 people with two samples per individual in the matching base. Guided by the results of our algorithmic model (see Sec. 4 or Ref. 14), we have chosen to perform the fusion thanks to the sum rule. As feature samples are 52×52 binary matrices, the total size of the base is only of 16 KB. The coding scheme and the recognition step requires about 7×10^6 CPU cycles, which corresponds to 7 ms.

Although parallelization possibilities are high for this kind of device, parallelism potential of the face and palmprint recognition algorithms is only lightly exploited on a DSP. That is why, we have also simulated the hardware implementation of the last steps of the processing chain (feature extraction, matching, fusion, and decision) on an FPGA platform.

3.2 *FPGA Implementation*

We work on a Virtex-5-XC5VFX70T FPGA of the Xilinx society.[27] It has been chosen for its configuration: It contains, in particular, 128 DSP slices (with 25×18 multipliers and 48-bit adder/subtracter/accumulator), which support massively parallel digital signal processing algorithms, and 22,400 configurable logic blocks (CLBs). Slices of the CLBs can be used to provide logic, arithmetic, and ROM functions; a part of them can also be used as distributing RAM or 32-bit data registers.

FPGA implementations have been simulated with the Very High-Speed Integrated Circuit, Hardware Description Language (VHDL) description using the Xilinx ISE tool. Results of the FPGA implementations will be presented in terms of used resources and processing speed. As for the DSP implementation, we have worked on a database of 25 people with two samples per individual in the matching base and we use the sum rule.

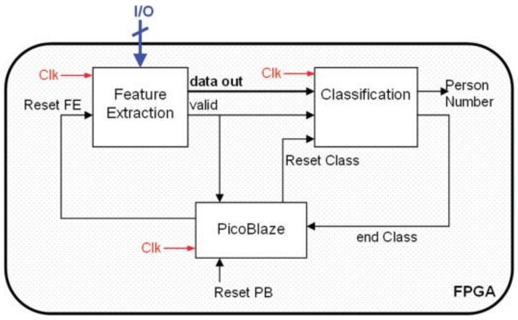


Fig. 6 Recognition chain hardware realization on FPGA: the software microcontroller core PicoBlaze (PB) controls the feature extraction (FE) and classification (Class) blocks.

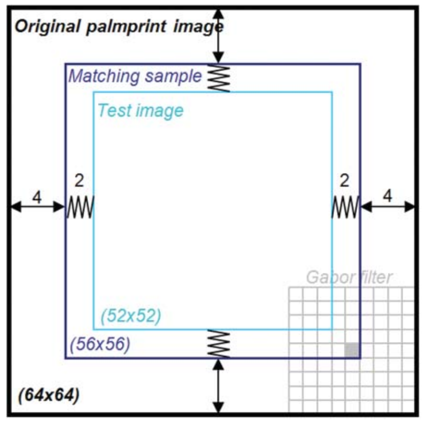


Fig. 8 Image sizes during processing: original images = 64×64 pixels, training sample images = 56×56 pixels because of the convolution with a 9×9 Gabor filter, and test images = 52×52 pixels because of the 2×2 pixels margin introduced by the elastic matching.

Figure 6 displays the entire recognition chain. We use a PicoBlaze (PB) microcontroller core implemented on the FPGA in order to synchronize the two stages of the palmprint recognition chain [i.e., the feature extraction (FE) and the classification (Class)]. PB is intellectual property of the ISE software;[27] this softcore microcontroller is programmed in assembly. It triggers the FE block when a palmprint image arrives in the FPGA, triggers it again when a face image arrives, and starts the Class block when the FE block processing is finished. When the Class block provides the template number, which corresponds to the person's identity, the complete system is ready for the next recognition.

Figure 7 displays the proposed design of the feature extraction block. We can see that data parallelism is fully exploited using the pipeline technique. In agreement with Fig. 8, each final test image size is 52×52 pixels. Because original images are larger than needed, border pixels are not used and we work on the 60×60 central pixels. In this module, the original image is stored in a Block RAM and processed by windows of 60×9 pixels. The process ends after 52 shifts of the vertical window. The convolution operation is realized using a structure of 81 DSP slices. Each of these slices multiplies a received pixel value with a filter coefficient and accumulates the previous result.

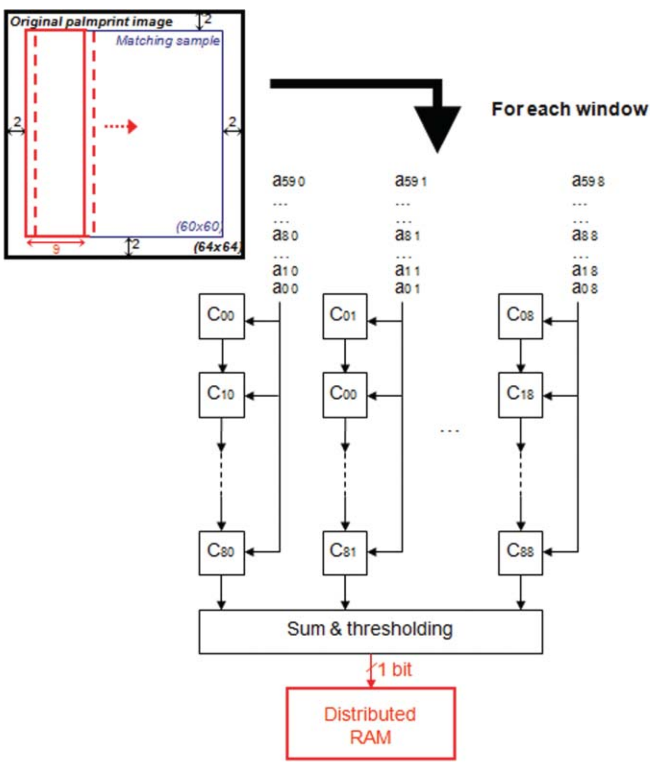


Fig. 7 Parallel structure for feature extraction stage: (a) the palmprint image is distributed in successive windows of 60×9 pixels, (b) feature extraction is realized using an architecture composed of 9 lines×9 columns of DSP slices that perform operations simultaneously.

A total of 61×52 = 3172 clock cycles are necessary in order to run this feature extraction block; 89 DSP slices, 1 Block RAM and 187 slices are used. The corresponding operating frequency is equal to 175 MHz.

The classification module consists of the calculation of 100 elastic Hamming distances (25 templates×2 samples ×2 biometrics), followed by 50 score fusions (each palm score is fused with the corresponding face score), and a comparison between each of these fused scores (NN classification). The 100 templates have been performed offline and loaded in distributed RAM during the hardware-configuration phase. Moreover, each elastic Hamming distance is performed by the calculation of 25 Hamming distances.

Figure 9 illustrates hardware realization of the elastic matching stage. We have chosen to carry out this step in 25 iterations corresponding to the 25 shifts of the elastic distance. We have designed a logic block in order to perform horizontal and vertical shift control. At each iteration, 100 Hamming distances are calculated in parallel. An inner loop provides in parallel 100 XOR operation results to 100 accumulators. When this loop is completed after 2704 (= 52×52) cycles, each accumulator provides a Hamming distance value, which can be compared to precedent values. The 100 minima are stored in registers of the FPGA. At the end of the 25 iterations, the fusion occurs by summing scores two by two. A final comparison step then finds the minimal value among the 50 fused minima. The person's identity is given by the corresponding template number. A total of (2704 + 1)×25(XOR) + 1(sum) + 55(final comparison) = 67681 clock cycles are necessary in order to perform this elastic matching stage.

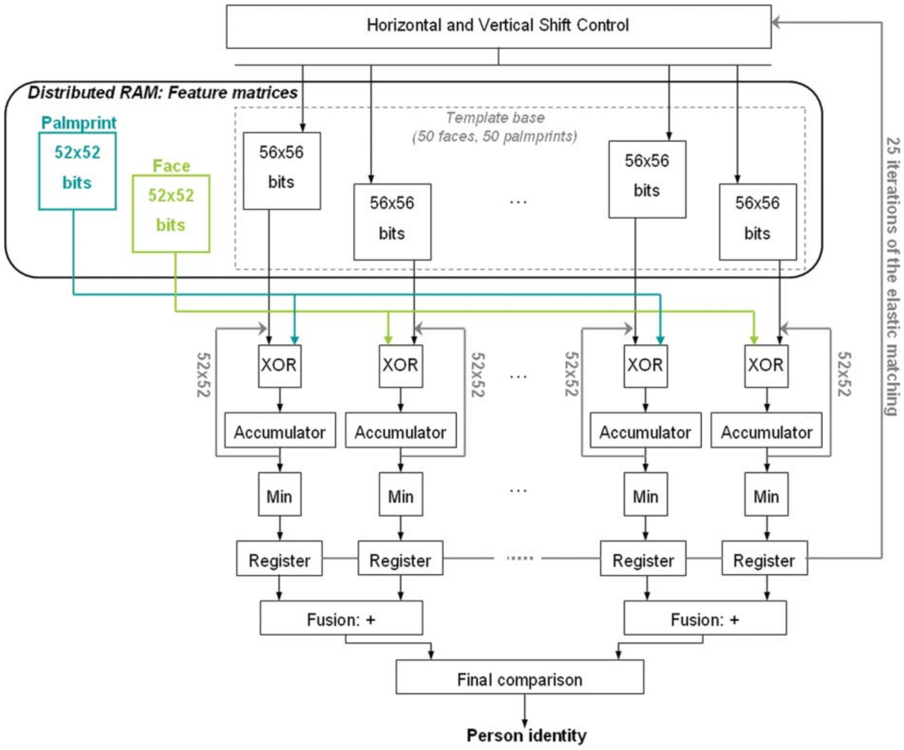


Fig. 9 Implementation of the classification block: for each of the 25 iterations, 100 Hamming distances are performed in parallel.

The elastic matching step does not use DSP slice or Block RAM but only CLB resources: a total of 8035 slices are used. The obtained operating frequency is equal to 175 MHz.

The general operating frequency is equal to 175 MHz, it corresponds to the frequency of both EM and Class modules. Thus, because our processing needs about $3172\times2 + 67681$ clock cycles, the entire operating time is on the order of 423 μs.

Chosen algorithms respect the constraints of simplicity, low-cost, regularity, and low-memory use. Thanks to the parallelization work, the entire processing is performed in only 0.4 ms. Moreover, implementations have been achieved using only a portion of the available resources of the Virtex-5-XC5VFX70T FPGA (see Table 1). In particular, we use very few logical resources (total ratio of 19.1%): because the Class block does not use DSP slice but only register slices and LUT slices, the number of recognizable people could be increased and reach 100.

Table 1 Hardware implementation results of the recognition chain on a Virtex-XC5VFX70T FPGA.

Logic element	Used number	Total number	Used ratio (%)
DSP48 slices	89	128	69.5
Block RAMs	2	148	1.4
Slices	8566	44800	19.1

4 Extensive Experimental Results

4.1 *Presentation of Experiments*

For this feasibility study, we built a database called the uB (University of Burgundy) database. It consists of images from 130 people, with nine face images and nine hand images per person. Pairs of images were recorded in three sessions of three images. The period of time between each session is spread from one day to a few weeks in order to take into account luminosity variation and possible variation in positioning or appearance. The acquisition environment is totally contactless and very user friendly (see Sec. 2.1).

In order to verify our approach, we also tested the processing on a multimodal database, which consists in the fusion of two public databases: the Hong Kong Polytechnic University (PolyU) palmprint database[28] and the AR face database.[29] The PolyU palmprint database contains 7752 gray-scale images from 386 different palms. Twenty samples from each of these palms were collected in two sessions (of 10 samples). The average interval between the first and second collection was two months. The size of every original image is 384×284 pixels. Fig. 10 shows some original palm images of the PolyU database. They have been obtained with contact and pegs in controlled lighting conditions.[30]

Our palm extraction method has been adapted to the processing proposed in Ref. 30: the fixed focal has been taken

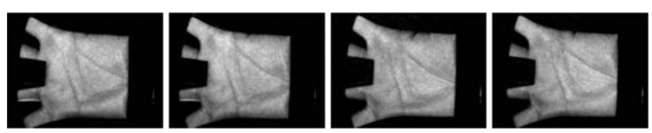


Fig. 10 Four images of the same palm from the PolyU database.

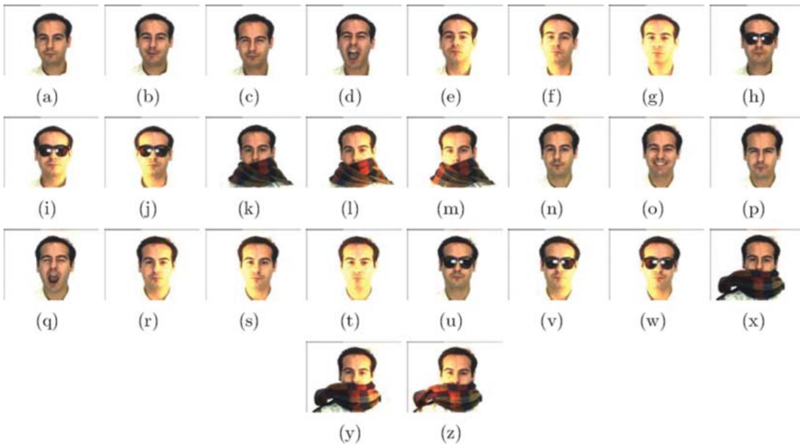


Fig. 11 Demonstration images of one subject from the AR database: (a–m) are from Session 1 and (n–z) are from session 2.

into account by using a fixed size window when defining the ROI. Algorithms have also been adapted to the number of visible fingers, which is no longer 5. Palmprints are still rotated and scaled to the size of 64×64 pixels.

The AR face database is composed of ∼4000 color face images of 126 people (70 men and 56 women), including frontal views of faces with different facial expressions, under different lighting conditions, and with various occlusions[29] (see Fig. 11). Face images were acquired in two sessions separated by two weeks. Each session captured 13 color images. The two sessions are available for 119 individuals. The preprocessing is the same as that of the uB database: all color images are transformed into gray-level images and each image (of 768×576 pixels) is scaled down to 64×64 pixels.

We take sample subsets of the same size from these two databases in order to create the multimodal database. As Jing et al.,[6] we use the first 119 palmprint classes with each class containing all 20 samples and all 119 face classes with each class including the first 20 samples.

The two Gabor filters have been chosen empirically on the uB database and applied to both uB and AR-PolyU databases. The way the Gabor filter coefficients have been chosen is explained in Ref. 14. Actually, the chosen filter is the same for the two modalities: its coefficients are set as $\lambda = 8.20$ and $\Theta = 2\pi/8$.

In this paper, all the results take into account the constraints of the hardware implementation. The preprocessing algorithms have been adapted to fixed-point calculation (for their DSP implementation), and the remaining processing has been quantified (for their FPGA implementation). In this way, results of the hardware system can be compared to those of the algorithmic system presented in Ref. 14.

4.2 *Recognition Performance*

The uB database contains $130 \times 9 = 1170$ images of each modality. For the recognition tests, it is divided in two parts: the training sample set and test sample set. As we respect the small sample set constraint, the number of samples per person in the matching base varies from 1 to 3. We defined two different protocols to conduct our experiments. Protocol 1: samples of the matching base are picked up randomly among the nine available ones, and all the remaining samples are used for tests. Protocol 2: samples of the matching base are picked up randomly among the three available ones of a unique session, and only the samples of the two other sessions are used for tests. Thus, when the matching base contains n samples per person ($n \in \{1, 2, 3\}$), $(9 - n)\times 130$ tests are performed according to the protocol 1 and $6\times 130\,(= 780)$ according to the protocol 2. Protocol 1 is the most used in studies because it allows one to take into consideration all the information contained in the database. Protocol 2 is used to verify the robustness of the algorithm in more realistic conditions: in the real world, all the matching samples are acquired during the enrollment phase, so the captured variability is reduced.

Results are qualified by the *recognition rate*, which is the ratio between the number of correct classification results and the total number of tests. Because it depends on the selected samples, nine tests with nine different matching bases are performed (for a matching base built according to protocol 2 in the three samples cases, only three tests are performed, since it is only possible to build three different bases). They are then averaged to constitute a final result [the for averaged recognition rate (ARR)], which objectively describes the performance of the system.

Results obtained thanks to the protocol 1 are given in Table 2. They are very similar to those of our former algorithmic study:[14] quantification of the Gabor filtering and transition to fixed-point do not introduce any performance degradation. As with the algorithmic model, the palmprint

Table 2 Average recognition rate obtained according to the protocol 1 on the uB database.

Method	ARR (%) one sample	ARR (%) two samples	ARR (%) three samples
Face recognition	79.10 ± 2.71	90.93 ± 4.60	95.25 ± 6.93
Palm recognition	91.05 ± 1.22	96.82 ± 1.52	98.27 ± 1.84
Minimum score	92.43 ± 1.70	97.40 ± 2.12	98.79 ± 2.63
Summed score	96.02 ± 0.95	98.96 ± 0.71	99.59 ± 0.76
Multiplied score	96.38 ± 0.94	99.07 ± 0.65	99.61 ± 0.79

Table 3 Average recognition rate obtained according to the protocol 2 on uB the database.

Method	ARR (%) one sample	ARR (%) two samples	ARR (%) three samples
Face recognition	73.50 ± 2.35	81.85 ± 2.19	84.96 ± 2.52
Palm recognition	89.32 ± 1.54	94.06 ± 1.26	95.56 ± 0.85
Minimum score	90.16 ± 1.51	93.69 ± 1.04	94.74 ± 1.22
Summed score	94.89 ± 0.82	97.68 ± 0.66	98.46 ± 0.56
Multiplied score	95.31 ± 0.77	97.89 ± 0.62	98.42 ± 0.52

Table 4 Average recognition rate obtained with 20 random tests for each method using the AR-PolyU database.

Method	ARR (%) one sample	ARR (%) two samples	ARR (%) three samples
Face recognition	68.22 ± 3.36	83.69 ± 10.4	86.21 ± 9.97
Palm recognition	85.46 ± 1.29	93.90 ± 0.77	96.03 ± 0.58
Minimum score	71.95 ± 2.78	86.15 ± 9.66	88.27 ± 9.13
Summed score	92.04 ± 1.18	97.49 ± 0.95	98.48 ± 0.63
Multiplied score	92.99 ± 1.11	97.92 ± 1.70	98.66 ± 1.25

recognition chain achieves, alone, a high-performance level. Face recognition does not perform as well as palm recognition, but results are rather high for such a low-computational-cost method in natural illumination conditions. Fusion always performs better than unimodality, and the difference between fusion methods is low: averaged recognition rates differ only by a few tenths. Considering the computational cost and the results of each method, the addition is very interesting in our case. Minimum has a low complexity but does not give good results, and the small performance increase induced by the multiplication does not compensate the difference of cost. There is a high similarity between the sum and multiplication rules. Very good results are obtained in the two-samples case: error is $\sim$1%. It can be noted that performance grows substantially between the one- and two-sample cases, while the increase between the two- and three-samples cases is minor.

Results obtained thanks to protocol 2 are presented in Table 3. As expected, they are generally not as good as those obtained according to protocol 1. However, they are still high: ARR after fusion is between 94.9% in the one-sample case and 98.5% in the two-sample case. All the comments made for the Table 2 are applicable to Table 3: fusion allows one to substantially increase the performance, there is only a small difference between addition and multiplication, and the gap between the one- and two-sample cases is significant. The only difference lies in the results of the minimum fusion, which does not bring a performance increase to the palmprint recognition. Moreover, we can see that fusion is more robust than monomodality: when the variability captured in the sample base decreases, the standard deviation of the face and palm results are greatly reduced, whereas it keeps similar values for the fusion.

Table 4 illustrates the average results of 20 random tests conducted on AR-PolyU database according to the protocol described in Ref. 6. We can see that all trends revealed by the tests conducted on the uB database are confirmed on these public databases.

For the face, errors are typically caused by the occasional wear of accessories (such as glasses) and by changes in expression or pose. For the palm, they are often due to a lack of image quality (bad focus, inhomogeneous illumination, etc.). These criteria are not correlated. That is why, most of the time, only one modality fails when a pair of images is tested. The fusion of the two often brings enough information to override the confusion: for example, the sum of two small distances (calculated on the samples of the same user) can be smaller than the sum between a very small distance (calculated on the samples, which are confused) and a large one (calculated on the samples of the other modality, which are not confused). Sometimes, both modalities are mistaken, but the overall system succeeds, as in Fig. 12. This is

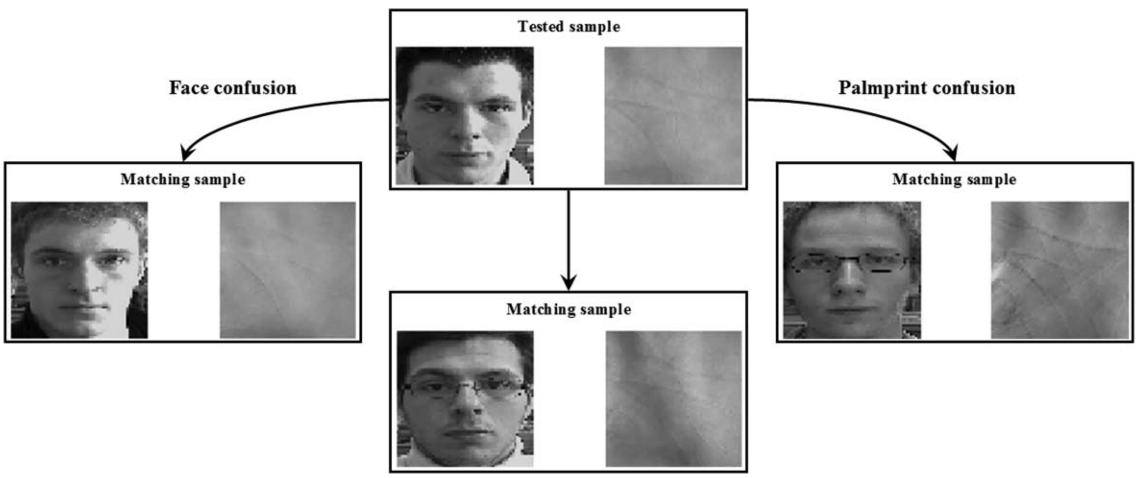


Fig. 12 Example of overall system success despite failure of the monomodal systems.

Table 5 Small sample biometric recognition performance comparison using the AR face database and PolyU palmprint database. Average recognition rate in the two- and three-sample cases.

	ARR (%) two-samples case			ARR (%) three-samples case		
Method	Face	Palm	Fusion	Face	Palm	Fusion
Jing et al.	65.67	63.33	92.66	74.88	64.29	96.14
Proposed method	83.69	93.90	97.49	86.21	96.03	98.48

[a]Reference 6.

probably because the two modalities are confused with samples of two different users, which cannot occur when they are fused because they are considered simultaneously.

4.3 *Verification Performance*

Performance of biometric verification systems is measured in terms of false rejection rate (FRR), which consists in the error rate in the intraclass comparisons, and false acceptance rate (FAR), which is computed from the interclass comparisons. A given FRR is achieved at a fixed FAR, and vice versa. By varying the FRR (or the FAR), the receiver operating characteristic (ROC) curve is obtained. In order to judge the performance of a verification algorithm, it is usual to use the operating point where the FAR and FRR are equal. It corresponds to the so-called equal error rate (EER).

In biometric verification systems, the test person is compared to a single reference person and a decision is made whether the two are identical or not. That is why biometric verification usually needs more images per individual for training in order to capture intraclass variability. Therefore, biometric verification often suffers more from the small sample size problem than biometric recognition.[19]

As for the biometric recognition, we arbitrarily take 1–3 samples of each of the 130 individuals in order to build the training set. The remainder is used as test set. We compared each test sample to all training samples: for a given test sample of the uB database, we perform genuine tests with the samples of its own class, and impostor tests with the samples of the other 129 classes. For example, in the three-samples case, the system tests 780 (130×6) genuine users and 100,620 ($130\times129\times6$) impostors.

Table 6 gathers EERs calculated in the one-, two-, and three-sample cases on the uB and AR-PolyU databases, and Fig. 13 displays the ROC curves in the one-sample case. It has to be noted that all results correspond to average verification rates obtained by averaging the verifications rates of 9 or 20 random tests. We can see that verification follows the same trends as recognition: palm achieves good performance alone and fusion allows one to greatly improve the results. Figure 13 shows that the curve behavior is the same on the two multimodal databases and that fusion by addition and multiplication is very similar.

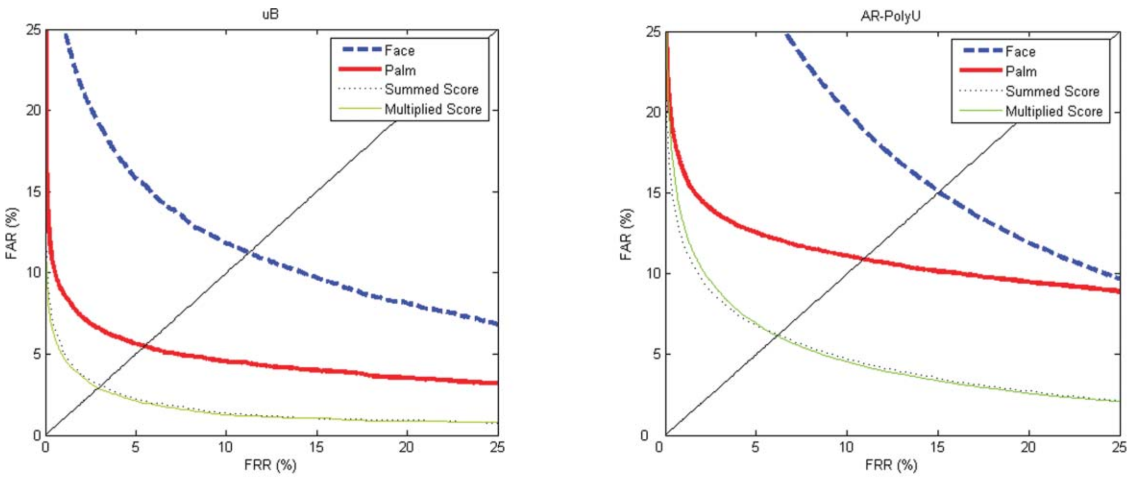
5 Discussion

Proposed system not only reach good performance in terms of hardware implementation, but also in terms of experimental results: it obtains similar results to those we can find in the literature. In the same conditions of biometric recognition on the AR-PolyU database, Jing et al.[6] obtain slightly lower performance, which keeps the same trends (see Table 5). For this, they use a Gabor feature

Table 6 Average equal error rate comparison for biometric verification. The bottom two rows correspond to the results obtained by Kumar et al.[9] without using (A) and using (B) subject-claimed identity.

			AEER (%)		
Method	Database	Sample Size	Face	Palm	Fusion
Proposed method	AR-PolyU	1	15.1	10.9	6.21
Proposed method	AR-PolyU	2	7.07	4.62	2.38
Proposed method	uB	1	11.2	5.43	3.12
Proposed method	uB	2	5.22	2.23	1.10
Proposed method	uB	3	3.54	1.53	0.79
Kumar et al. (A)[a]		4	5.48	5.24	2.21
Kumar et al. (B)[a]		4	4.28	4.45	0.72

[a]Reference 9.



(a) (b)

Fig. 13 ROC curves of biometric verification in the one-sample case calculated on (a) the uB database and (b) the AR-PolyU multimodal database.

fusion followed by a feature compression using a Kernel Discriminative Common Vectors (KDCVs) approach and a classification by the radial basis function (RBF) network.

Kumar et al.[9] propose a score fusion using a feed-forward neural network trained on a base of four samples per person. Face features are extracted by the Eigenface method, and palmprint by the combination of four directional filters. The proposed method is tested on a multimodal database designed by the authors that contains 70 subjects and is acquired in more controlled conditions (for example, illumination, distance between hand and sensor). Table 6 tries to make some biometric verification performance comparisons between the two bimodal systems. Kumar et al.[9] show that combining subject claimed identity allows one to reduce verification error (EER from 2.21 to 0.72%). On the uB database with only three samples, we also obtain better performance than this reference method, which does not use the claimed identity, and our EER is very similar to the one obtained using claimed subject identity. Results obtained with AR-PolyU in the two-sample case are also comparable to those of the reference method.

It must be noted that performance calculated on the AR-PolyU database is not as good as that calculated on the uB database because hand images are of lower quality and do not show the entire hand, which makes palm extraction less accurate. Moreover, faces are not acquired in the same conditions and show a large white background.

In terms of hardware implementation, as Yang and Paindavoine[11] or Lopez-Ongil et al.[12] (who work on face and hand geometry recognition, respectively), we prove that FPGA implementation is highly better than DSP (or general purpose processor) implementation. For Ref. 11 and 12 execution time is multiplied by 3, and in our case it is multiplied by 17. This high coefficient is achieved because we work on very short time (0.4 ms for FPGA and 7 ms for DSP) and use massive parallelism on FPGA. If we compare the work of Yoo et al.[13] on multimodal recognition to ours, then we can see that with comparable EER (1.5% for the iris, for example), execution times are better (total execution time of $<$500 ms for us and $\sim$1 s for the iris in Ref. 13).

We observe that with a sequential architecture the execution time of the last steps of processing depends on the number of subjects in the comparison base. However, thanks to the parallel architecture of the FPGA implementation, recognition of 50 or more individuals could be realized using the same chip (FPGA Virtex-XC5VFX70T) with the same processing speed. On the other hand, authentication would be even faster on DSP because the comparison base would contain the samples of a single user.

6 Conclusion and Perspectives

In this paper, we have presented a contactless biometric system that combines two modalities: palmprint and face. A complete processing chain has been developed from the acquisition of hand and face images to classification decision, and a hardware architecture has been implemented on DSP and FPGA. Face and palmprint are two decorrelated modalities, that can be acquired easily with minimal equipment (a webcam) and without contact. Multimodal systems have many advantages over monomodal systems, such as better robustness or greater universality. Therefore, using these two biometrics in a multimodal system ensures one to create an efficient general public system.

As we work on palmprint in a contactless context, a hand preprocessing (which consists of a palm extraction) has been developed and simulated on a DSP platform. Hardware implementation of the rest of the multimodal recognition chain has been simulated on the DSP and on a FPGA Virtex-5 device. Hardware results demonstrate that preprocessing can easily be performed during the acquisition phase, and multimodal biometric recognition can be treated almost instantly. Only 0.4 ms are necessary using 50 training samples recorded on 25 persons with low-resource consumption on FPGA, while no more than 7 ms are needed on DSP.

A database of 2340 images (130 subjects$\times$2 modalities $\times$9 views) was built in real-world conditions (user-friendly interface and natural illumination, for example). Experimental results show that multimodal fusion always reaches better performance than monomodality. The proposed algorithm, which is based on low-complexity operations, such as Gabor filtering and similarity measurement by binary comparison, fits palmprint recognition particularly well. The fusion of palmprint and face at score level allows us to achieve high recognition rates (98.96% using uB database and 97.49% using AR-PolyU database with only two training samples per person and per modality). In the same manner, the used fusion strategy provides good performance for the biometric verification task (EER = 1.10% in the two sample case). We can note that the adaptation of the algorithms to hardware implementation do not introduce performance degradation. Our experiments demonstrate that the proposed approach is an effective solution for the small sample biometric problem and can outperform memory-consuming methods, such as the ones that use Gabor filter banks. Moreover, using the same algorithm, performance may be increased with other modalities having an oriented texture such as knuckleprint or ear.

This soft- and hardware study shows the feasibility of a robust and efficient embedded multimodal biometric system that offers several advantages; for example, flexibility, user-friendliness, and real-time processing. Besides, the proposed system is able to work with real-world application challenges, such as lighting changes and variations in hand position and orientation. Our final objective is to implement the complete biometric application on a hardware system. Our next step consists of doing new processing optimizations and complexity analysis of the palmprint image extraction task (hand localization, palmprint extraction and normalization), before achieving FPGA implantations. The chosen FPGA contains a PowerPC processor core that could be used to perform some calculations.

References

1. A. K. Jain, A. Ross, and S. Pankanti, "Biometrics: a tool for information security," *IEEE Trans. Inf. Forensics Secur.* **1**(2), 125–143 (2006).
2. D. Masip and J. Vitria, "Shared feature extraction for nearest neighbor face recognition," *IEEE Trans. Neural Netw.* **19**(4), 586–595 (April 2008).
3. G. K. O. Michael, T. Connie, and A. B. J. Teoh, "Touch-less palm print biometrics: novel design and implementation," *Image Vis. Comput.* **26**(12), 1551–1560 (2008).
4. J. Doublet, O. Lepetit, and M. Revenu, "Contact less hand recognition using shape and texture features," in *Proc. of 8th Int. Conf. on Signal Processing (ICSP'06)*, Vol. 3, Guilin, China (November 2006).
5. X. Jiang, W. Xu, L. Sweeney, Y. Li, and R. Gross and D. Yurovsky, "New directions in contact free hand recognition," in *Proc. IEEE* **2**, 389–392 (September 2007).
6. X. Y. Jing, Y. F. Yao, D. Zhang, J. Y. Yang, and M. Li, "Face and palmprint pixel level fusion and kernel DCV-RBF classifier for small sample biometric recognition," *Pattern Recogn.* **40**(11), 3209–3224 (2007).
7. X. Geng, Z. H. Zhou, and K. Smith-Miles, "Individual stable space: an approach to face recognition under uncontrolled conditions," *IEEE Trans. Neural Netw.* **19**(8), 1354–1368 (2008).
8. S. Zafeiriou, A. Tefas, and I. Pitas, "The discriminant elastic graph matching algorithm applied to frontal face verification," *Pattern Recogn.* **40**(10), 2798–2810 (2007).
9. A. Kumar and D. Zhang, "User authentication using fusion of face and palmprint," *Int. J. Image Graph.* **9**(2), 251–270 (2009).
10. T. Zhang, X. Li, D. Tao, and J. Yang, "Multimodal biometrics using geometry preserving projections," *Pattern Recogn.* **41**(3), 805–813 (2008).
11. F. Yang and M. Paindavoine, "Implementation of an rbf neural network on embedded systems: real-time face tracking and identity verification," *IEEE Trans. Neural Netw.* **14**(5), 1162–1175 (September 2003). Q2
12. C. Lopez-Ongil, R. Sanchez-Reillo, J. Liu-Jimenez, F. Casado, L. Sanchez, and L. Entrena, "FPGA implementation of biometric authentication system based on hand geometry," in *Proc. of Field Programmable Logic and Application (FPL) Conf.* Vol. 3203, p.43–53. (august 2004).
13. J.-H. Yoo, J.-G. Ko, Y.-S. Chung, S.-U. Jung, K.-H. Kim, K.-Y. Moon, and K. Chung, "Design of embedded multimodal biometric systems," in *SITIS '07: Proc. of 2007 3 Int. IEEE Conf. on Signal-Image Technologies and Internet-Based System*, Washington, DC, p. 1058–1062 IEEE Computer Society, (2007).
14. A. Poinsot, F. Yang, and M. Paindavoine, "Small sample biometric recognition based on palmprint and face fusion," in *ICCGI '09: Proc. of 2009 4 Int. Multi-Conf. on Computing in the Global Information Technology*, p. 118–122 (2009).
15. T. Connie, A. T. B. Jin, M. G. K. Ong, and D. N. C. Ling, "An automated palmprint recognition system," *Image Vis. Comput.* **23**(5), 501–515 (2005). Q3
16. H. Dutagaci, B. Sankur, and E. Yoruk, "Comparative analysis of global hand appearance-based person recognition," *J. Electron. Imaging* **17**(1), (2008).
17. A. Kong, D. Zhang, and M. Kamel, "Palmprint identification using feature-level fusion," *Pattern Recogn.* **39**(3), 478–487 (2006).
18. X. Tan, S. Chen, Z. H. Zhou, and Fuyan Zhang, "Face recognition from a single image per person: a survey," *Pattern Recogn.* **39**(9), 1725–1745 (2006).
19. M. Kyperountas, A. Tefas, and I. Pitas, "Weighted piecewise LDA for solving the small sample size problem in face verification," *IEEE Trans. Neural Netw.* **18**(2), 506–519 (March 2007).
20. D. Xu, S. Yan, L. Zhang, S. Lin, H. J. Zhang, and T. S. Huang, "Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis," *IEEE Trans. Circ. Syst. Video Technol.* **18**(1), 36–47 (January 2008). Q4
21. C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *IEEE Trans. Image Process.* **11**, 467–476 (2002).
22. O. Ayinde and Y. H. Yang, "Face recognition approach based on rank correlation of gabor-filtered images," *Pattern Recogn.* **35**(6), 1275–1289 (2002).
23. A. Noore, R. Singh, and M. Vatsa, "Robust memory-efficient data level information fusion of multi-modal biometric images," *Inf. Fusion* **8**(4), 337–346 (2007).
24. W. K. Kong, D. Zhang, and W. Li, "Palmprint feature extraction using 2-D gabor filters," *Pattern Recogn.* **36**(10), 2339–2347 (2003).
25. T. Savic and N. Pavesic, "Personal recognition based on an image of the palmar surface of the hand," *Pattern Recogn.* **40**(11), 3152–3163 (2007). Q5
26. Texas instruments, <http://www.ti.com/>.
27. Xilinx, <http://www.xilinx.com/>, (2009).
28. PolyU palmprint database, <http://www.comp.polyu.edu.hk/> biometrics (2006).
29. A. M. Martinez and R. Benavente, "The AR face database Technical Report No. 24, CVC (june 1998).
30. D. Zhang, W. K. Kong, J. You, and M. Wong, "Online palmprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(9), 1041–1050 (2003).

Audrey Poinsot received her MS from the University of Bordeaux I, France in August 2007. Since September 2007, she has been a PhD student in image processing and instrumentation at the University of Burgundy. Her research interests consist of multimodal biometrics, with particular emphasis on their hardware implementations on embedded systems.

Fan Yang is a full professor and member of LE2I CNRS-UMR, Laboratory of Electronic, Computing, and Imaging Sciences at the University of Burgundy, France. Her research interests are in the areas of patterns recognition, neural network, motion estimation based on spatiotemporal Gabor filters, parallelism and real-time implementation, and, more specifically, automatic face image-processing algorithms and architectures.

Vincent Brost is an associate professor and member of LE2I CNRS-UMR, Laboratory of Electronic, Computing, and Imaging Sciences at the University of Burgundy, France. Before joining the LE2I Laboratory, he was an engineer in electronics and embedded systems for ten years (for Renault and France Telecom). His research topics are specific processor optimization and real-time hardware implementations on the DSP and the FPGA. Q6

Queries

Q1: Au: Please check the authors affiliation for correctness.
Q2: Au: Please check Ref. 11 for content errors, or provide the DOI.
Q3: Au: Please provide page range for Ref. 15
Q4: Au: Please check Ref. 20 for content errors, or provide the DOI.
Q5: Au: Please provide date accessed and title document for Ref. 25
Q6: Au: Please supply photograph of author Vincent Brost.