Multimed Tools Appl (2017) 76:22671–22701
DOI 10.1007/s11042-016-4271-8

Efficient biometric palm-print matching on smart-cards for high security and privacy

Nadia Nedjah[1] · **Rafael Soares Wyant**[1] · **Luiza de Macedo Mourelle**[2]

Received: 17 October 2016 / Revised: 8 December 2016 / Accepted: 14 December 2016 / Published online: 3 January 2017

Abstract Smart control access to any service is at the very basis of any smart city project. Biometrics have been used as a solution for system access control, for many years now. However, the simple use of biometrics can not be considered as final and perfect solution. Most problems are related to the data transmission way between the medias, where the users require access and the servers where the biometric data, captured upon registration, are stored. In this paper, the use smart-cards is adopted as a possible effective yet efficient solution to this problem. Palm-prints have been used as a human identifier for a long time now. This biometric is considered one of the most reliable to distinguish a person from another as its unique yet stable over time. In this work, we propose an efficient implementation of palm-print verification on smart-cards. For this implementation, the matching is done on-card. Thus, the biometric characteristics are always kept in the owner's card, guaranteeing the maximum security and privacy. In a first approach, the False Acceptance Rate (FAR) and False Rejection Rate (FRR) are improved using upward, downward, leftward and rightward translations of the matched palm-codes. However, after thorough analysis of the achieved results, we show that the proposed method introduces a significant increase in terms of execution time of the matching operation. In order to mitigate this impact, we augmented the proposed technique with an acceptance threshold verification, thus decreasing drastically the execution time of the matching operation, and yet achieving considerably

✉ Nadia Nedjah
nadia@eng.uerj.br

Rafael Soares Wyant
wyant@eng.uerj.br

Luiza de Macedo Mourelle
ldmm@eng.uerj.br

[1] Department of Electronics Engineering and Telecommunications, Faculty of Engineering, State University of Rio de Janeiro, Rio de Janeiro, Brazil

[2] Department of Systems Engineering and Computation, Faculty of Engineering, State University of Rio de Janeiro, Rio de Janeiro, Brazil

low FAR and FRR. It is noteworthy to point out that these characteristics are at the basis of any access control successful usage.

Keywords Biometrics · Palm-print · Palm-code · Smart-card · Health-care · Security · Privacy

1 Introduction

Biometrics are studies of certain human physical or behavioral characteristics that are capable of distinguishing two different persons. There are certain properties that must be present in the features so that they can be used as a biometrics. The most fundamental propriety is that every person or at least the vast majority of people must possess this characteristic. This property is called *universality*. Moreover, two persons must differ considering this characteristic. This property is called *distinguishibility*. Furthermore, the characteristic should be invariant with respect to time. This property is called it perdurability. Finally, the characteristic should be measurable. This property is called *collectability* [18]. Currently, there are many characteristics or organs used as biometrics. To name only few that are most commonly used, we can cite DNA, face, hand veins, fingerprint hand geometry, iris, palm-print, voice, among many others. Biometrics are highly secure and convenient for identification and/or verification of individual identity, as they can not be stolen or forgotten besides the extremely high difficulty to be forged [13].

The main application of biometrics is related to access control, *i.e.* through the verification of biometrics, a person can be granted or denied access to the guarded service/information. In most cases, biometry have a big advantage when compared to other kind of identity certification because only biometric characteristics can really guarantee the authenticity of the claimant. Banking systems as well as healthcare systems often resort to the use of passwords and alphanumeric codes for letters besides the use of a card to grant an account access. Bothe systems must, at any cost, guarantee privacy, security, usability and performance of the various functionalities available in the system. The implementation of a biometric system considerably improves security against misuse. Despite the fact that the use of biometrics is a solution that aims to increase security, the risk of fraud can not be ignored. Many developers believe that the use of biometrics is the final and perfect solution for all identification problems [10].

Besides the problems regarding security, there is also the acceptance problem by users. The use of biometrics has been spreading rapidly and people are starting to think about their own safety when they are asked to register their biometrics, indiscriminately in various institutions. After all, their biometric details would be stored in many databases, which are susceptible to attacks. In such a case, biometrics, which is unique and invariable over time, could be forever compromised.

It is now well-established that the exploitation of smart-card based solutions augmented with biometrics verification provide more privacy and security when compared to a biometrics-only or smart-card-only solutions. With the biometrics details stored in the card's memory and performing the biometric match on-card, the privacy and security of smart-card biometric authentication are enhanced as well as system performance. In this paper, we study the approach that makes use of smart-cards together with biometrics aiming at increasing the security of access control systems. The objective of the overall project is to evaluate the possibility of using a multi-application smart-card that grants access by performing biometric comparisons. Thus, it would be possible to use a single card for several institutions and biometrics would always be stored in a single card in the possession of the owner. The biometrics details would be stored only in a unique smart-card and the matching is processed on-card.

In this paper, we propose an efficient and secure implementation of user authentication via palm-print verification. Palm-prints have been used as a human identifier for over 100 years and are still considered one of the most reliable ways to distinguish a person from another, because of its stability and uniqueness [31]. However, only recently, studies about using it as biometrics have emerged. Comparison of palm-prints, as described in [42], was chosen to be implemented on smart-cards because it requires small amount of memory and low computing effort to obtain the comparison results. It is noteworthy to point out that preliminary results appeared in [39].

The rest of this paper is organized in 6 sections. First, in Section 2, we give a brief introduction of biometrics main concepts. After that, in Section 3, we detail some important issue about the implementation of biometrics verification in smart-cards. Then, in Section 4, we give an overview of relevant related work. There follows, in Section 5, we describe the method used to obtain a palm-code for a given palm-print, define its proposed internal representation and describe the proposed algorithm used to perform palm-code comparison efficiently. Subsequently, in Section 6, we present and discuss the performance and effectiveness of the proposed implementation. Last but not least, in Section 7, we draw some conclusions and point out some future work.

2 Biometrics

Biometric technologies are defined as automated methods for identification and/or verification of unique features of a living being [2]. These can be physical or behavioral characteristics. Biometrics are highly secure and convenient for identification or identity verification of an individual. They can not be stolen or forgotten, besides the high difficulty to be forged [13]. There are several existing biometrics. Fingerprint, iris, hand or facial geometry, hand veins and voice are examples of physical characteristics while signing and cadence of typing on keyboards are behavioral characteristics.

The choice of a particular biometrics must take into account many factors such as the complexity of collection procedure, the performance of the technology, the entailed cost, the profile and user culture. These factors may also affect the acceptance of biometrics. Table 1[1] shows a comparison between existing different biometrics in relation to the underlying characteristics.

One of the key features for selecting the use of a specific biometrics is the distinguishability that will directly impact the accuracy of the verification system. Taking into consideration all the above aspects, the biometrics of fingerprint, iris and palm-print are the most likely to be used in smart-card based system.

The palm-print is used as a human identifier for over 100 years now, and is still regarded as one of the most reliable ways of distinguishing one person from another due to its stability and uniqueness [31]. However, only recently that is being studied and implemented for biometrics usage. This said, however, as it will be shown later in Section 4, there are already few methods based on different approaches.

[1]Table is taken from [18], where L, M and H represent low, medium and high, respectively. The results are based on the perception of the authors.

Table 1 Comparison between biometrics technologies

biometric ID	Universality	Distinguishibility	Perdurability	Collectability	Performance	Acceptability	Chance of fraud
DNA	H	H	H	L	H	L	L
Ear	M	M	H	M	M	H	M
Face	H	L	M	H	L	H	H
Facial thermogram	H	H	L	H	M	H	L
Fingerprint	M	H	H	M	H	M	M
Way of walking	M	L	L	H	L	H	M
Hand geometry	M	M	M	H	M	M	M
Hand veins	M	M	M	M	M	M	L
Iris	H	H	H	M	H	L	L
Typing cadence	L	L	L	M	L	M	M
Smell	H	H	H	M	L	M	M
Palm print	M	H	H	M	H	M	M
Retina	H	H	M	L	H	L	L
Signature	L	L	L	H	L	H	A
Voice	M	H	H	M	L	H	H

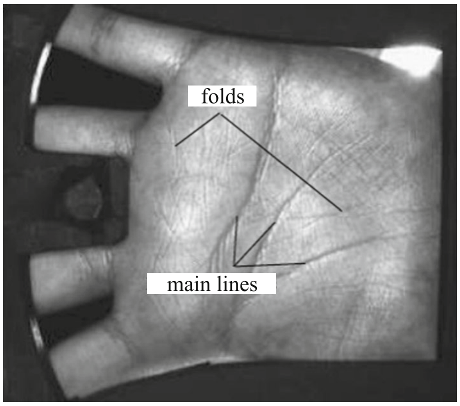


Fig. 1 Print of a hand print

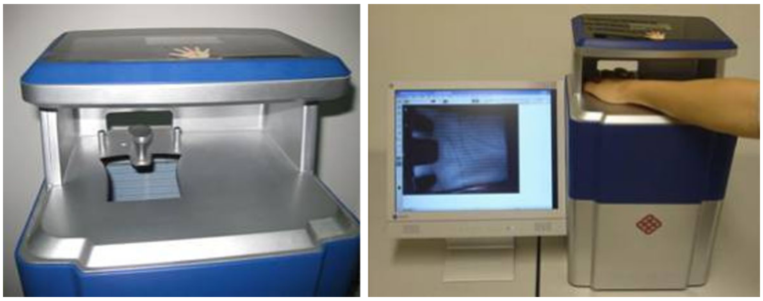
As illustrated in Fig. 1, the main features used on the palm-print biometrics are the main lines and secondary folds. As well as fingerprint, palm-print also has lines and details that can be used to distinguish between two individuals. Due to the size of a palm-print, the lines and minutiae can only be used with high resolution sensor images. Still, some methods go further and use the pores to make the distinction more precise.

A device used to acquire the palm-print images can be seen in Fig. 2. It is capable of making the acquisition of 2D image and/or 3D map of a hand. The palm-print biometrics does not have a great market penetration yet, but due to the results obtained by researchers, soon can become a highly used biometrics.

3 Biometrics in smart-cards

This section defines the aspects that the implemented biometric systems are required to have when using smart-cards. Nonetheless, recall that the focus of this work is study the feasibility of implementing biometric comparisons processed in smart cards. Biometric systems are basically composed of four components:

– A machine or mechanism responsible for the digital representation of the biometric characteristics of a person;



(a) *Acquisition equipment* (b) *Equipment in use*

Fig. 2 Equipment used for acquisition of a palm-print

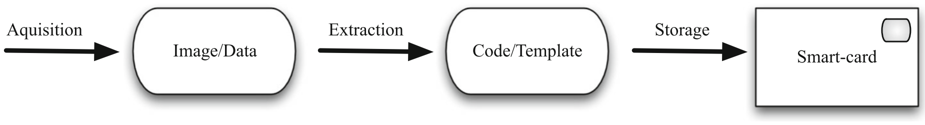


Fig. 3 Registration of a biometrics in the smart-card

- A standard extraction tool that will be used in the comparison;
- A verification tool to match the stored pattern and the input pattern;
- An interface to output of the result.

Biometric systems operate in two stages: the storage of the pattern that will serve as a basis for the comparison and the matching of the stored pattern and the input pattern.

Figure 3 illustrates the registration process, also known as *enrollment*. The sample of the individual, who is card user, is captured. For each specific method will be used biometrics (fingerprints for scanner, microphone for voice recognition, camera for face recognition camera for iris recognition etc.). The collected data is then processed to extract the unique characteristics of the user. The extracted biometric template that will be used in future comparisons is stored on the card.

Figure 4 illustrates the biometric verification process, also known as *matching*. The applicant's biometric sample is captured similarly to the process made during enrollment stage. The unique patterns of this sample are extracted and sent to the checker. The stored pattern is retrieved from the card and sent to the checker, which then runs the verification process, resulting in a score that establish whether both biometric samples are from the same individual. Biometric system's main purpose may be identification and/or verification. The

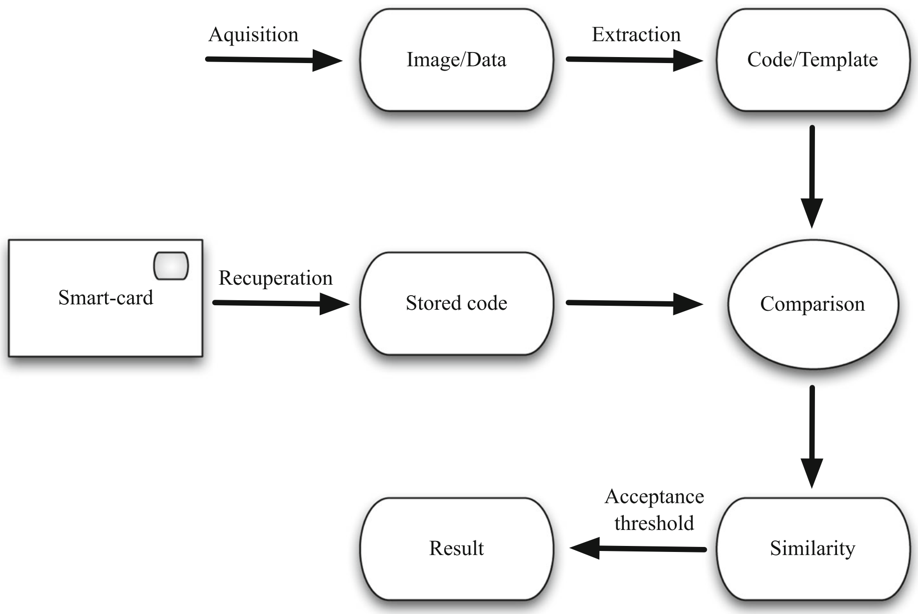


Fig. 4 Biometric verification.

identification is the search for a person from a given biometric sample. This encompasses large databases and requires high processing power. Indexing techniques for improving search may be used to improve performance. On the other hand, verification is the validation done given two biometric samples, resulting in the identification whether the samples belong to the same person. The biometric comparisons using smart-cards can occur in two ways:

- *Template on Card* (ToC), where the user's biometric sample is stored in the card's memory and the comparison is done externally on another machine. This requires cards that only have memory, which are much cheaper thank cards that are endowed with a micro-processor.
- *Match on Card* (MoC), where the user's biometric sample is stored in the card's memory and the comparison is also processed on the card. In this case, the smart card need to include at least one processor. The low processing frequency and small memory size included in nowadays smart-cards are the biggest obstacles in these implementations. In this work, we propose an implementation MoC.

In a biometric system, when the stored sample is compared to the captured information, a score of similarity is assigned and used to confirm the identity of an individual. When this score is compared with a pre-defined threshold, two types of error rate can be observed:

- False Acceptance Rate (FAR), which indicates the rate of false entries or incorrectly accepted fraudulent data.
- False Rejection Rate (FRR), which indicate the rate of correct individual entries that were incorrectly rejected.

The supra-cited rates are extremely important in choosing the limit of the score that should define the final decision of comparisons to be declared as false or true. When it comes to embedded systems, an extremely important factor is also the choice of the algorithm to be implemented. It is necessary to determine the complexity in terms of memory usage and runtime.

4 Related work

The palm-print biometrics began to be studied recently, but it has shown a promising biometric technology [44]. The palm is a very rich surface in terms of details, from the most prominent, as the main lines, to the tiniest details, like the minutiae, as studied and used for fingerprint, and even the palm pores.

In the case of comparisons in smart-cards, methods which use the tiny details, such as minutiae and pores, are not interesting, because they entail high-resolution sensors. Also, the number of extracted details is very large, requiring more processing in the comparison phase. Therefore, we only focus on methods that use low resolution images *i.e.* less than 100 dpi.

Currently there are 3 main types of palm-print recognition approaches: holistic, based on specific characteristics and hybrid.

4.1 Holistic approach

In this approach, the palm-print image is used as the basis of an extractor or holistic classifier. Its use introduces two main problems: the representation of the image and the classifier design. In the sequel, we briefly describe the most commonly used representations and some existing classifiers.

Palm-print images can be represented both in the space domain as well as in transformed domain, such as the frequency domain or using subspace among many others. The holistic features can be extracted using these representations in several ways [1, 4, 8].

Concatenating columns of a palm-print image into a vector of many dimensions, a variety of linear and nonlinear subspaces can be explored for the extraction of the characteristics [27, 36, 40]. Recently, stress analyzers have been developed treating the palm-print image as a second-order tensor [14, 45]. Common digital image processing techniques allowing the representation of a palm-print image were investigated. The Fourier transform, which is a classical technique for image transformation, has been successfully applied to feature extraction and classifier design [19, 25]. In [11], a multi-layer perceptron neural network with back-propagation was initially applied for palm-print authentication. However, the recognition of the palm is a typical multi-class learning problem, which is known to be very difficult to be dealt with the back-propagation algorithm. In [26], modular neural network is used to decompose the palm-print recognition problem into a series of smaller and simpler sub-problems of two classes each.

4.2 Local characteristics based approach

The exists a number of characteristics of the palm-print that can be used for recognition. Table 2 provides a classification of the key characteristics, regarding of the resolution required, collectability, perdurability and distinguishibility.

The main lines are not reliable to make a direct comparison, but they can be used for pre-alignment of the palm-print images before a more detailed comparison takes place, as used in [34]. Typically, the main lines are used in conjunction with other features to enhance the reliability of the comparison final result.

The palm folds can stay unchanged for period of times, but they are not permanent like the minutiae. For this reason, they are not useful in critical areas, such as criminology and forensics. However, they can be used for recognition in real-time systems, yielding high performance [32, 42].

In [43], the 3D structure of the hand palm is used to increase the reliability of recognition and invalidate attacks based on using false palm-prints. Even though, this representation increases the complexity of data acquisition, when combined with the 2D texture, it becomes highly very reliable and robust against fraud.

The use of minutiae has recently shown great potential in forensics and criminalistics [17]. To work correctly, only minimal resolution images of 500dpi are needed. A

Table 2 Specific characteristics for palm-print recognition

Characteristic	Resolution	Collectibility	Perdurability	Distinguishibility
Main lines	low	high	high	low
Folds	medium	high	medium	high
3D	medium	low	medium	medium
Minutiae	high	medium	high	high
Level 3	very high	low	medium	high

great advantage of using minutiae is the possibility of carrying out recognition with high reliability using only a small portion of the palm-print.

Level 3 features is the name given to the set of features that includes all of the above as well as the tiny details of the palm-print, such as lines, pores and scars [16]. The use of Level 3 features is even more important because only a part of the palm-print is used in the identification process. However, the images must be of very high resolution *i.e.* higher than 1000dpi). It was established that only 20 to 40 pores are sufficient to identify an individual [3].

As mentioned earlier, we considered only the methods that use low-resolution images. So, we only consider algorithms that are based on the main lines and folds of the palm. There are three main mechanisms for extraction and comparison. These are based on lines, codes or texture descriptors of the palm.

In [38], the second order derivative of the Gaussian is used to represent the magnitude of the line and the first order to detect the location of the line. All directional lines are combined to form the final result. The problem with this approach is the fact that it is inevitable to go through a process of alignment by rotation and translation during the comparison. This process that requires a high processing time. To circumvent these problems and improve performance, the extracted lines are dilated before the comparison.

The codes based methods use digital filters to convert the palm-print image into a binary codes. The exploitation of these binary codes present advantages, such as low memory requirements and swift comparison. Therefore, these codes have been very useful in representing and comparing palm-prints. Inspired by the IrisCode [6], palm-code method was developed in [42]. Initially, convolution of the palm-print image, using a 2D Gabor filter, is obtained; then the real and imaginary resulting images are coded according to their phase in a binary representation. To improve the performance of the method, it is possible to extract several palm-codes, applying the Gabor filter's transformation following different orientations. Hence, the FusionCode method was developed [21]. It allowed a significant reduction of the error rates. Recent advances in code based methods indicate that one of the most promising features for the palm-print recognition is the orientation of the lines [22, 37]. Such methods have been able to achieve near-zero error rates.

A typical palm-print texture descriptor divides the palm image into small blocks, computes the mean, variance, energy or histogram associated with each of these blocks as local characteristics [12, 24, 33, 35]. In [24], the palm-print image is divided into overlapping blocks, the Discrete Cosine Transform coefficients associated with each block are then computed, and their standard deviations are used to form a feature vector. Other texture descriptors, such as power of the directional element and histogram of the local direction have also been adopted in palm-print recognition [12, 35].

A dedicated hardware design for palm-print and palm-vein identification is presented in [30]. The proposed design is implemented in FPGAs to prove its functionality and evaluate its performance.

4.3 Hybrid approach

It has been argued that the human vision system uses both local characteristics and holistic perception to recognize and identify objects of interest and it is expected that systems with hybrid approaches of this kind are promising for recognizing palm-prints [44]. Hybrid systems have two main applications: high recognition accuracy [23] and efficient identification of palm-prints [41]. Using both approaches for multiple representations of the palm-print, several strategies for fusion different types of characteristics can be used. Decisions and scores can be exploited to denote the performance of comparisons [20]. In this purpose, a large number of approaches, using multiple palm-prints was introduced in [23, 29]. In [23], three representations are extracted: Gabor, line and subspace features. Then, a combination of the obtained results is used to deduce the comparison scores. In [41], multiple levels for hand palm-print identification are used: hand geometry, which is labeled as level 1, energy overall texture, labeled as level 2, fuzzy features of the lines, labeled as level 3, and local texture energy, labeled as level 4. These levels are used in a hierarchical classification process. A guided search scheme is proposed to achieve an efficient comparison. This method can be used for both recognition and for identification.

5 Proposed palm-print verification on smart-cards

In the work reported in this paper, we used as a basis the method proposed in [42]. The extraction of the palm-code of a region of interest of the palm-print image is done using a 2D Gabor's filter. The result of the convolution is a matrix of complex numbers and the palm-code consists of two binary matrices representing the real and imaginary parts.

5.1 Extraction

In all types of biometrics, the extraction is one of the most important steps, because the result obtained directly impacts the performance of the next steps as well as the final result of the comparison. Section 5.1.1 explains in greater detail the delimitation of the area of interest of the palm-print image and Section 5.1.2 details the method used for extraction of the binary code.

5.1.1 Area of interest

In order to obtain the best results during the generation of the binary code, it is important that the delimitation of the region of interest is always done in the same way in different images, regardless of their quality. In [42] a coordinated system was elaborated so as to perform the measurements using the joints of the fingers as reference points. There are five main steps, as shown in Fig. 5 to delimit the area of interest:

- **Step 1:** Apply a low-pass filter, such as Gaussian, to the original image. Using an intensity threshold in the resulting image to classify the pixels into white or black as shown in Fig. 5b.
- **Step 2:** Get the edges of the joints of the fingers (F_1x_j, F_1y_j) and (F_2x_j, F_2y_j) using a edge recognition algorithm, such as Sobel and Canny, as illustrated in Fig. 5c. The junction between the ring and middle fingers is not extracted because it is not used in the next steps.
- **Step 3:** Find the tangent between the two junctions. Consider (x_1, y_1) and (x_2, y_2) being the points at (F_1x_j, F_1y_j) and (F_2x_j, F_2y_j), respectively. If the line $y = mx+c$ passing through these two points satisfies the inequalities $F_iy_j \leq mF_ix_j + c$, for all values i and j, as in Fig. 5d, this line will be the tangent between the two junctions.
- **Step 4:** The line passing through (x_1, y_1) and (x_2, y_2) will be the Y-axis palm coordinate system. The line that is perpendicular to the Y-axis through the midpoint between the

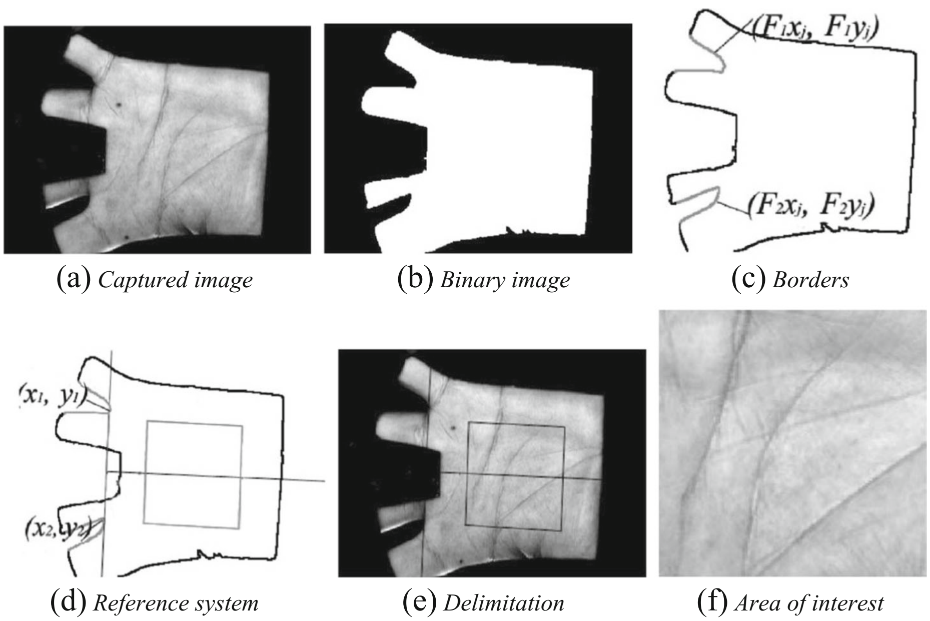


Fig. 5 Reference system for extraction of the area of interest

two points will the X-axis, thus determining the origin of the coordinate system, as shown in Fig. 5d.

– **Step 5:** The area of interest will be a sub-image of pre-defined size based on the coordinate system, as shown in Fig. 5e. This sub-image is used to extract the binary code, as shown in Fig. 5f.

5.1.2 Binary code of palm-print

The basis for comparison is the sub-image extracted with the aid of the coordinate system, as described in Section 5.1.1. Direct comparison of the image with another image is very susceptible to the light and quality of the captured image. Inspired by the work of Daugman iris biometrics [6], Zhang [42] proposed using Gabor 2D filter for the extraction of the main palm-print features. This filter neutralizes the brightness and quality differences, allowing for a direct comparison.

Due to the existence of an available tool to extract the palm-code from a plam-print image, we implemented in MATLAB the code of a Gabor 2D filter and the palm-code extractor, as explained in the next two sections.

5.1.3 2D Gabor's filter

Direct comparison of the extracted image with another image is very susceptible to brightness and quality of that image. Inspired by Daugman's work on Iris Biometrics [6], Zhang [42] proposed using 2D Gabor filter to extract the main features of the palm-print. This filter allows to neutralize the difference in brightness and quality, bringing forth the possibility of direct comparison.

The 2D Gabor function was proposed by Daugman [5, 7] as a simple model of the visual cortex cells. It is based on the discovery of the crystalline organization of the principal cells of the cortex in the brains of mammals [15]. The 2D Gabor function, as proposed by Daugman, is a spacial bandpass filter that achieves the theoretical limit for a resolution the associated to information in the 2D spatial and 2D Fourier domains.

Gabor [9] showed that there is a “quantum principle” for information: an association of time-frequency domain for 1D signals must necessarily be quantified so that no signal or filter can fit in an area that is less than a certain minimum area. This minimum area, which reflects the inevitable trade-off between time resolution and frequency, has a lower limit of the product, analogous to the Heisenberg uncertainty principle in physics. He found that complex exponential modulated by a Gaussian provides a better result. Equation (1) presents a general form of the 2D Gabor filter, used in [42], to extract the characteristics of a palm-print.

$$G(x, y, \theta, u, \sigma) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}} \exp^{2\pi i(uxcos\theta + uysen\theta)}, \tag{1}$$

wherein $i = \sqrt{-1}$, u i a the frequency of the senoidal wave, θ that controls the orientation of the function and σ represents the standadrd deviation of the Gaussian function.

Figure 6a shows the real part of the 2D Gabor filter and Fig. 6b shows the imaginary part. To make the filter more robust against brightness, it is transformed into zero CD (direct current), applying (2).

$$G'[s, y, \theta, u, \sigma] = G[x, y, \theta, u, \sigma] - \frac{\sum_{i=-n}^{n} \sum_{j=-n}^{n} G[i, j, \theta, u, \sigma]}{(2n+1)^2}, \tag{2}$$

wherein $(2n + 1)^2$ is the filter size. Due of the symmetry, the imaginary part of the filter already has zero DC. Certainly, the success of the extraction of code depends on the choice of parameters θ, u and σ. In [42], a process of refinement was applied to optimize these parameters and found out that $\theta = \pi/4$, $u = 0.0916$ and $\sigma = 5.6179$. These values were used in the implementation of 2D Gabor's filter, used in this paper.

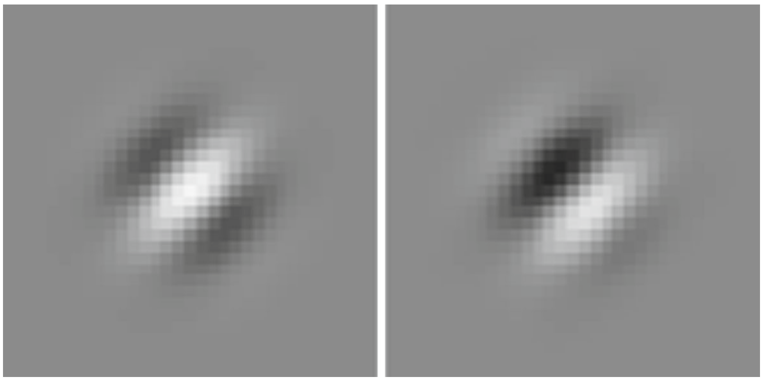

(a) *Real part* (b) *Imaginary part*

Fig. 6 Response of the 2D Gabor's filter to a pulse

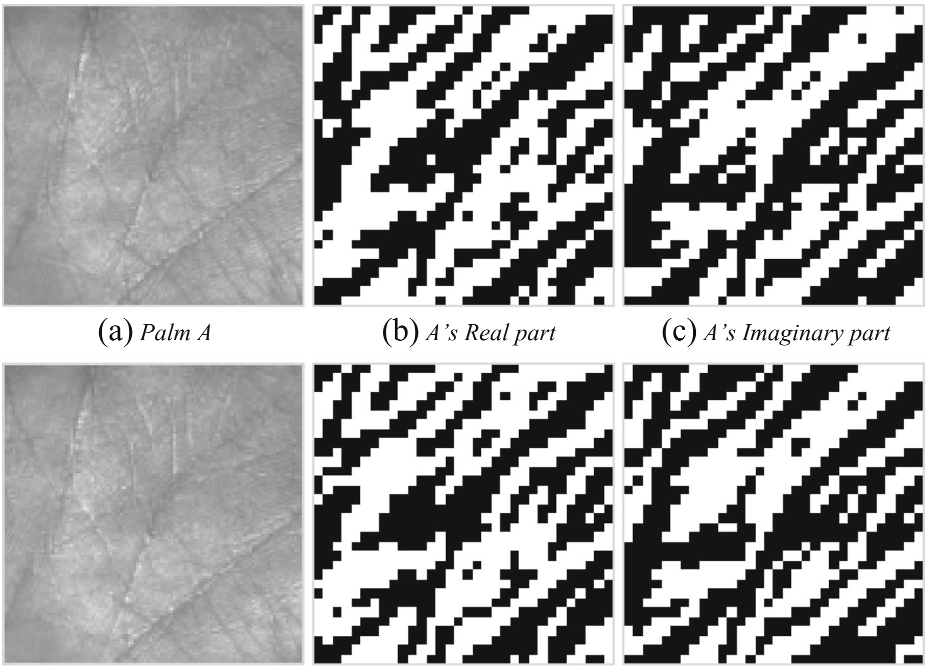
5.2 Palm-print comparison

After the application of the 2D Gabor filter, the result is an array of complex numbers. The size of the array palm-code is set to 32×32. The coding last step takes into account only the phase of each complex numbers. It generates two codes, one code for the resulting real part and one for the imaginary part, observing only the information about the quadrant, in which the complex number are mapped. If the real part of the imaginary number is greater than zero, then the code will be set to 1, and 0 otherwise. The same logic is applied to the imaginary part. The final result is defined by two arrays of 32×32 bits.

Figure 7 shows two examples of palm-code. Figures 7a and d show the area of interest, as described in Section 5.1.1, of two images captured from the same palm. Note that the second column shows the real part and the third column shows the imaginary part extracted from their respective images. Once extracted the codes are compared.

The comparison between two palm-codes is performed by computing the Hamming distance (HD) between them. The Hamming distance between two palm-codes $P : (P_R, P_I)$ and $Q : (Q_R, Q_I)$ is defined in (3), wherein $\oplus$ is the binary XOR operator and N^2 the size of the real or imaginary part of palm-codes. It provides the percentage of different bits between two palm-codes.

$$HD_{P,Q} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} P_R(i,j) \oplus Q_R(i,j) + P_I(i,j) \oplus Q_I(i,j)}{2N^2} \tag{3}$$


(a) *Palm A* (b) *A's Real part* (c) *A's Imaginary part*

(d) *Palm B* (e) *B's Real part* (f) *B's Imaginary part*

Fig. 7 Examples of palm-code extraction

This result can be improved using a relative displacement between the two compared matrices. This due the fact that often the error occurs because of a small phase shift between the stored and input templates. In order to improve the effectiveness of the palm-code matching process, we implemented both horizontal and vertical displacements as well as in both directions, i.e. left and right.

Consider the binary code of the two palms A and B of Fig. 7. Figure 8 shows the result of applying the XOR operation between them, where the pixels in white represent the points where the codes differ.

Calculating the normalized Hamming distance, *i.e.*, counting the white points and dividing by the total of points, we found a distance of 12.99. This distance confirms the fact that the two images were obtained from the same palm. Note that the Hamming distance can vary from 0 to 100. Distances closer to 0 indicate high similarity between the compared samples.

Often, the result of the comparison is not as close to 0 even when it is regarding images from the same palm. This is mainly due to the different positioning of the hand during image acquisition phase. Even a small difference can disrupt the identification process, thus generating incorrect results. In order to mitigate the impact of this kind of problem, we could perform additional checks when calculating the Hamming distance, varying the relative position between the two binary matrix of the compared palm-codes. These variations may be implemented by performing displacements of bits of the palm-code in both horizontal and vertical directions. For another example of palm-codes, Table 3 shows the variation of the result of using some Hamming distance offsets (HD). Note that all possible horizontal movements are performed (leftward, none and rightward) combined with the vertical displacements (downward, none and upward), totaling 9 possible cases. Throughout the remainder of this paper, the verification performed regarding the displacement of 1 bit in every way possible will be termed as the comparison with 1-bit translation.

In the aforementioned example, the best result of 19.50, *i.e.*, the shortest distance occurred using the horizontal displacement 1 bit leftward with no vertical displacement against 33.44 in the case where no translations are not applied . Note that the horizontal possible displacements of 1 bit leftward, none and rightward are listed as horizontal displacement of -1, 0 and $+1$ bit, respectively, while and the vertical displacements of 1 bit downward, none and upward are referred to as -1, 0, $+1$ bit. Similarly, the comparison can

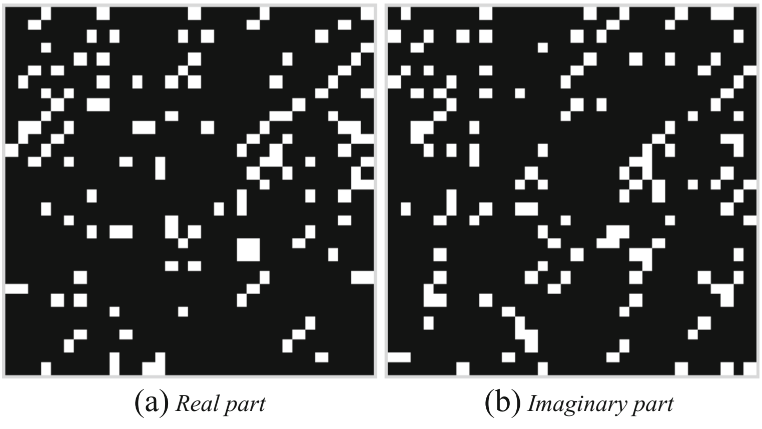

(a) *Real part* (b) *Imaginary part*

Fig. 8 Result of the Xor's application

Table 3 Comparison with 1-bit translation

Shift				HD (%)
Horizontal		Vertical		
leftward	-1	downward	-1	28.40
leftward	-1	none	0	19.50
leftward	-1	upward	$+1$	32.88
none	0	downward	-1	28.07
none	0	none	0	33.44
none	0	upward	$+1$	44.70
rightward	$+1$	downward	-1	40.06
rightward	$+1$	none	0	45.26
rightward	$+1$	upward	$+1$	50.36

be done considering the translation of 2 bits, which will consider all possible combinations of offsets -2, -1, 0, $+1$ and $+2$ for the horizontal and vertical axes, totalizing 25 possible cases. This is explained in further details with an illustrative example in the next section.

5.3 Algorithms for palm-code comparison

The palm-code is represented by two matrices of 32×32 bits: one for the real part and the other for the imaginary part. Thus, the total size to store the template is $2 \times 32 \times 32 = 2048$ bits or 256 bytes. Because of the limit of a single transmission within Java Card, which is of 128 bytes, the transfer of the code is be done in two transmission steps. It is noteworthy that there is no problem in terms of memory storage of the code because current smart-cards offer over 100kb EEPROM and 1 to 3Kb RAM.

The JavaCard does not allow the use of 2-dimensional arrays. Therefore, the palm-code is allocated in vector for future comparisons. Bytes are compared using the XOR operator using 128 iterations to go through all the bytes that compose the palm-code. The number of bits with value 1 is counted and the accumulated result is divided by the total number of bits. This division results in a decimal number. However, float variable are not allowed. The most sophisticated variable type is *short* (2 bytes). In order to mitigate the problem of accuracy, the dividend is multiplied by 100 before final division, as described in (3), thus resulting in a Hamming distance with an accuracy of two digits. Note that the maximum value of the dividend is 2048 and thus 204800 after multiplication by 100. This leads to an overflow problem because variables of type *short* can vary within $[-32768, +32767]$. To remedy to this new issue, the dividend is multiplied by 10 while divider is divided by 10 before the final division occurs. Thus, the maximum value of the dividend is $2048/10 = 20.480$ and the divider value is $2048/10 = 204$, avoiding any kind of overflow. After this maneuver, the division will provide the Hamming distance as a percentage.

Algorithm 1 computes the Hamming distance of the proposed modifications, where N^2 is the size of the matrix that is stored as a vector of *shorts* with 64 positions, $\mathbb{T}$ is the binary code stored in the card and $\mathbb{I}$ is the binary code of the individual who is requesting authentication. The palm-code has a real part and an imaginary that are represented as $(\mathbb{T}_R, \mathbb{T}_I)$ and $(\mathbb{I}_R, \mathbb{I}_I)$, respectively.

Algorithm 1 Computing hamming distance of two palm-codes

```
Require: Input palm-codes: $\mathbb{I}$ and template palm-code: $\mathbb{T}$
Ensure: Hamming distance between $\mathbb{I}$ and $\mathbb{T}$: $H$
1: $Nbits := 0$
2: for $i := 1 \to N^2$ do
3:     $xored := \mathbb{T}_R(i) \oplus \mathbb{I}_R(i)$
4:     $Nbits := Nbits + bitsCount(xored)$
5:     $xored := \mathbb{T}_I(i) \oplus \mathbb{I}_I(i)$
6:     $Nbits := Nbits + bitsCount(xored)$
7: end for;
8: $Nbits := 10 \times Nbits$
9: $H := Nbits/204$
```

The bit counting performed in Algorithm 1 (lines 4 and 6), can be implemented in several ways. The simplest, most intuitive yet most inefficient way consists of using a loop wherein the count is done for every bit that is equal to 1. This always requires 16 iterations. For a fast comparison, we devised Algorithm 2, in which the number of iterations coincides with that of bits equal to 1.

Algorithm 2 $bitsCount$: Efficient counting of 1-bits

```
Require: $xored$
Ensure: Total number of 1-bits in $xored$
1: $bitsCount := 0$
2: while $xored \neq 0$ do
3:     $xored := xored$ AND $(xored - 1)$
4:     $bitsCount := bitsCount + 1$
5: end while;
```

Thus, it is not always necessary to make 16 iterations to reach the final result. In the case where there are no bits equal to 1, no iteration occurs and the returned result is zero. This counting method reduces the comparison run time. We can infer that the complexity of Algorithm 1 two palm-codes represented by 2 NxN matrix each is $\mathcal{O}(N^2)$. However, the 1-bit counting procedure usually does the job in less than 16 iterations, assuming a normal distribution in short variables, it requires 8 iterations in average, which is half the time the straightforward counting requires.

As explained in Section 5.2, it is possible to translate the compared palm-codes to improve the result of the comparison. The implementation of comparison with bit translation takes into account the memory allocation strategy. It views the palm-codes as a set of vectors instead of matrices. The translation is done assuming that every set of consecutive 32 bits is the beginning of a new line. Also, it is noteworthy to point out that the number of compared bits is equal to $32 \times 32 \times 2 = 2048$ when no translation is used. However, this number decreases when translations occur. For instance, applying a rightward translation of 1 bit and no vertical translation, the number of the compared bits is reduced to $31 \times 32 \times 2 = 1984$.

Note that after the displacements in the vertical direction, the comparison of the resulting palm-codes is still based on the comparisons of short integers. However, the displacement of the horizontal direction are done using the bit-shift operations $\ll$ and $\gg$, requiring the comparison of bits belonging to different short integers.

The 1-bit translation consists of performing displacement -1 to 1 bit in both the horizontal and vertical directions, resulting in nine comparisons of palm-codes. To improve the performance of the comparisons with 1-bit translation, we consider all the vertical translations for each of the possible horizontal translations. This prevents the execution of bit displacement of the same line of the binary code several times. Although it requires three times the memory used to allocate the auxiliary variables that accumulate temporary results, it is not too expensive, because for more bit translation, more temporary counters must be used.

Figure 9 illustrates the translation process in a palm-code, showing only 6×32 bits. Note that a complete palm-code has 32 lines. Consider the set of red rectangles as the input palm-code and the set of black rectangles being the stored one. Each inner rectangle represents an a vector entry (a short integer) of 16 bits, amounting to 32 bits per line. In the first iteration, as in Fig. 9a, no translation is done horizontally. Therefore, no bit shift operation is required. The second iteration, as in Fig. 9b, uses the leftward shift operation of 1 bit and the third iteration applies a rightward shift operation also of 1 bit. When translations of 2 bits are considered, we will have 2 extra iterations for 2-bit translations: one leftwards and the other rightwards. Moreover, we will also have 2 extra stages in each iteration for 2 -bit translations: one upwards and the other downwards. Thus, there will be 5 iterations of 5 stages each, summing up 25 possibilities.

Figure 9c shows a blue highlight representing one of the short integer inputs, of the vector stored after applying the rightward bit shift operation. The extraction of the bits that form a given cell is performed only once. In the same iteration, comparisons are executed for the three cases of vertical translations.

6 Performance results

The aim of this work is to use the smart-card to process the matching operation and thereby increase the security level. During the card configuration, the palm-code of the owner is

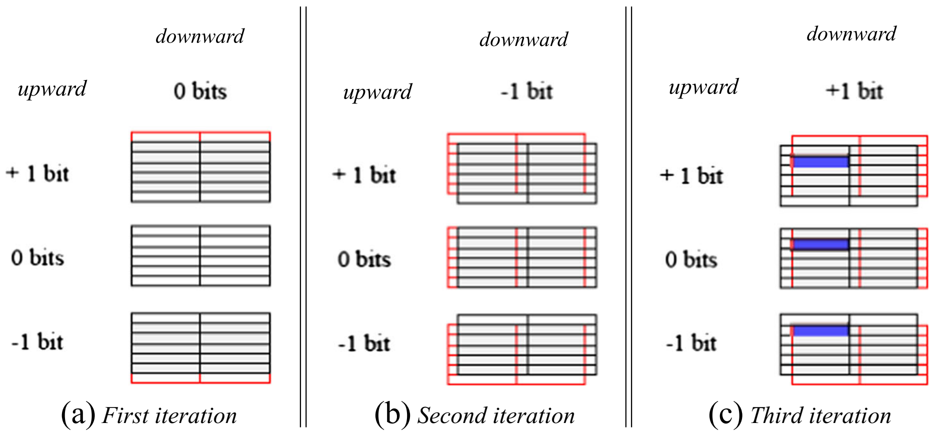


Fig. 9 Hamming distance with translations

transmitted to the card so that it is stored for future matching upon access requirement. To access the service being protected by the biometry, the card-holder must provide its palm-code, as an input, to confirm his/her identity through the computation of the Hamming distance to the stored template. For implementation purposes, we used the Java Card platform.

The images used to evaluate this work were taken from the database of Polytechnic University of Hong Kong and is available in [28]. The database contains 8000 samples of palm-print images of 400 different hands.

6.1 Palm-print database

The images used for performance evaluation are from the database of the Polytechnic University of Hong Kong and is available in [28]. The database includes 8000 samples of palm-prints of 400 different hands.

The samples were obtained through two acquisition sessions, with 10 samples acquired in the first session and 10 in the second. The average time between two sessions is one month to allow any possible changes to occur. The process of delimitation and cutting to obtain the area of interest presented in Section 5.1.1 was conducted by the university, being available only the final images of the area of interest of each sample.

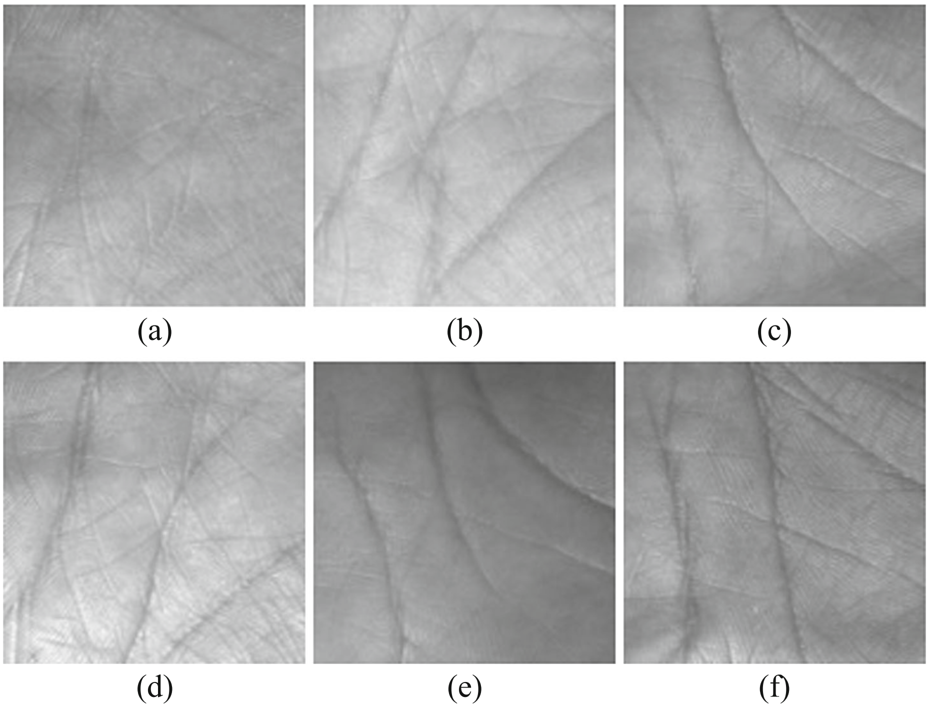


Fig. 10 Samples from palm-print images extracted from database POLYU

Figure 10 shows a number of samples obtained from the palm-prints of different hands. These smaples are part of the database. All images are in gray scale and have a resolution of 128×128 pixels. Although the database has both 2D and 3D image acquisition, this work is focused on the images of 2D acquisition because the implemented algorithm does not consider the data from the 3D acquisition.

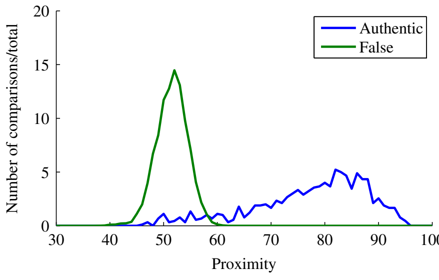
The results are presented in terms of the *proximity* defined as $100 - H$, wherein H is the Hamming distance. This metric indicates the percentage of similarity between the two compared palm-codes. In order to test and validate the proposed implementation on the smart-card, we compared 10 samples of 20 different hands selected randomly from the used database.

6.2 Direct comparison

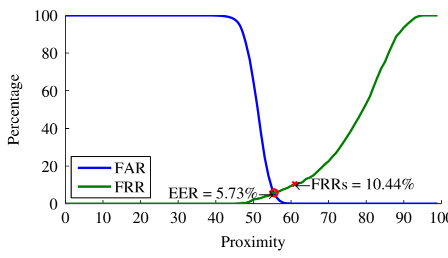
The first test consists of comparing palm-codes without any displacement. Figure 11a shows the distribution of the results of authentic *vs.* false comparisons. In order to be able to present both distributions of true and false comparisons in the same illustration, the distributions were normalized with reference to the total relevant comparisons.

From the data shown in Fig. 11a we can conclude that false comparisons are harmonically concentrated between proximities 40 and 60, while the authentic comparisons are distributed between 45 and 95, with higher concentration between 80 and 90. The ideal result occurs when the two curves do not intersect. This illustration can help choosing a threshold to be applied to declare authentic comparison results, *i.e.* originating from the comparison of the same palm.

Figure 11b shows the test results in a more elaborate way. For each choice of proximity, the illustration shows that the FRR and FAR, as defined in Section 3. The most appropriate threshold for the proximity, which separates the comparisons of the approved comparisons from the other ones can also be made from the data as depicted in this illustration. Note that in order to improve the system security, it is essential that different individuals are not mixed up. So, the FAR should be as small as possible, but without much impact on the FRR. In the illustration, we highlight two important points: the point at which the percentages are equal (ERR) and the point where FAR is less than 0.1 %, which defines a safe FRR. Thus, Figure 11b highlights the ERR of 5.73 % and the safe FRR of 10.44 %. The second point is the most interesting for a secure system, which means that, if the choice of the proximity



(a) *Proximity distribution*



(b) *FAR and FRR for different threshold*

Fig. 11 Comparison results when no translation is allowed

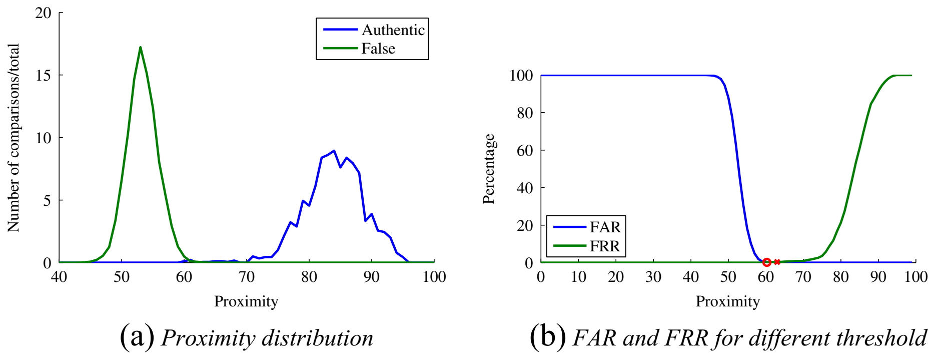


Fig. 12 Comparison results when a translation of 1 bit is allowed

of 68 % as access threshold, the result would be that at every 1000 comparisons of different hands, only one would be considered authentic while an individual would have 90 % chance of be granted the access in each trial.

6.3 Comparison with translations

As discussed in Section 5.2, the proximities can be decreased if bit displacements are carried out in one of the palm-codes before the comparison. The impact obtained when using applying translations of 1 and 2 bits will be discussed next.

Figure 12 shows the chart of the proximity distributions of all performed comparisons when translations of 1 bit in any possible directions, as well as the corresponding FAR and FRR of the obtained results. Note the number of possible translations sums up to 9 distinct combinations. In this case, the percentage of equal error rate dropped to EER = 0.17 %, compared to ERR = 5.73 % in the case of direct comparison i.e. without translation only. Moreover, the secure FRR dropped to 0.39 %. compared to FRR = 10.44 % in the case of direct comparisons only. Thereby, we validate the possibility of making use of the translation of bits to improve the result of the comparison.

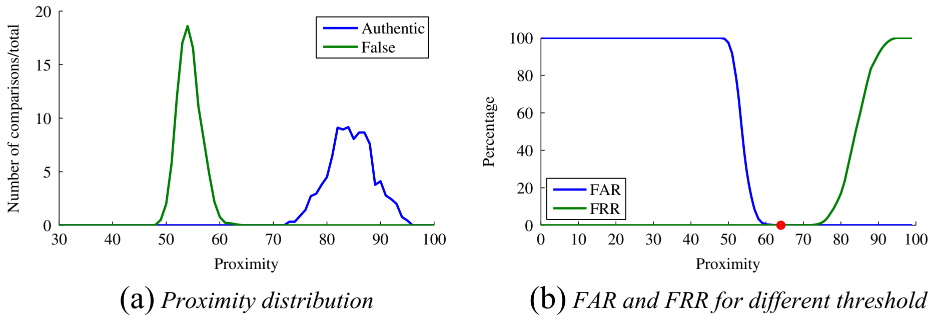


Fig. 13 Comparison results when a translation of 2 bit is allowed

As explained earlier, aiming at a higher improvement, it is possible to consider the translation of more bits. Figure 13 shows the chart of the proximity distributions of all performed comparisons with translations of up to 2 bits in any possible directions, as well as the corresponding FAR and FRR of the obtained results. Note the number of possible translations sums up to 25 distinct combinations: 16 comparisons augmented with the 9 combinations corresponding to the translations of 1 bit.

The chart of Fig. 13a shows an ideal distribution, wherein the results for authentic comparisons and those for false ones are totally separated. This allows for a perfect choice of the acceptance threshold. It is noteworthy to point out that the distribution of the proximity results regarding authentic comparisons is concentrated between 75 and 95. An interesting yet expected point is that the distribution of the results related to false comparisons became narrower with the increase of the number of translated bits. With direct comparisons only (*i.e.* no translation allowed), this distribution is concentrated between 40 to 60, while for comparisons, which allow the translation of 1 bit, this interval is narrowed to 45 to 60, and finally in the case of allowing a translation of 2 bits, the interval is further narrowed to 50 to 60. Furthermore, it is noteworthy to observe that for each tentative, there is chance to improve the comparison result. However, false comparisons do not achieve proximities higher that 60 %. Moreover, as shown in Fig. 13b, both important points EER and FRR are 0. This indicates that this biometry is getting closer and closer to perfection, meaning that there will be no errors during palm-code comparisons of distinct individuals and yet there will no failure in authentic tentatives either.

6.4 Processing time analysis

Although the comparison result is important for validating the implementation performance, it is not the only result that is worthy of analysis. The runtime due to a comparison is another very important factor as it will determine the waiting time of an individual at every authentication attempt.

The total time for a comparison takes into account the complete transmission of the palm-codes and the processing time required to compute the Hamming distance. Comparisons with translation of bits can be done to achieve optimal error rates but certainly the corresponding processing time has increases the execution time of a comparison. The transmission time and storage in the smartcard EEPROM of each part of the palm-code is about 850 ms. Two transmissions must be made to store the whole code. Thus, the total storage time is 1700 ms. This time is relatively high compared to the time of transmission and storage in the RAM of the part of the palm-code to be compared, which about 185 ms. Although relatively slow, the time required during the storage in the card EEPROM is not significant since it is a process that is performed only once during the lifetime of this card.

Figure 14 allows us to analyze the execution time of the comparisons when no translation is allowed. Figure 14a depicts the the execution time distribution for all the performed 40 thousands comparisons, which include 2 thousands authentic cases and 38 thousands false cases. As before, the results are normalized (percentages) are used so as to include all the processing times in the same chart. Figure 14b illustrates the relationship between the execution time and the obtained comparison result. It can be observed that the execution time of authentic comparisons varies between 440 and 560 ms while that of false comparisons

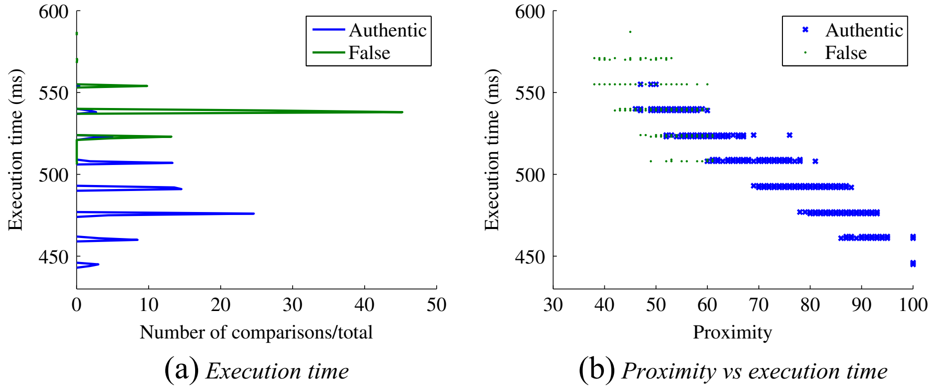


Fig. 14 Execution time when no translation is allowed

varies between 500 and 580 ms. Note that in general, authentic comparisons take smaller time to be declared than false ones.

Similarly, Figure 15 allows us to analyze the execution time of the comparisons when translations of 1 bit are allowed. Once again, the tendency of the execution times regarding authentic comparisons, which varies between 1350 and 1800 ms, are smaller than the required times to determine false comparisons, which varies between 1650 and 1950 ms. Furthermore, from Fig. 15b, it can be clearly observed that the authentic and false comparisons are more separated with respect to proximity than in the case wherein no translation is allowed, as illustrated in Fig. 14b.

The charts of Fig. 16 show distinctively that the execution times regarding authentic comparisons are smaller than those regarding false comparisons. From Fig. 16a, it is noteworthy to point out that there exists two distinct concentrations of execution times for authentic *vs.* false comparisons. Furthermore, form Fig. 16b, the separation between the two kind of comparisons is more neat than in the previous analyzed cases.

Another interesting characteristic to be observed is the relationship between the decrease in comparison time and the increase of the proximity measure. This relationship is due to the bit counting algorithm used (see Algorithm 2). The algorithm requires less iterations to

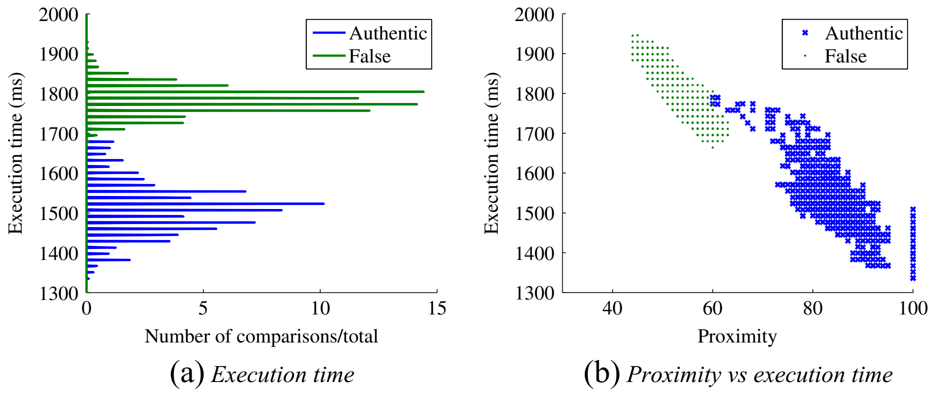


Fig. 15 Execution time when a translation of 1 bit is allowed

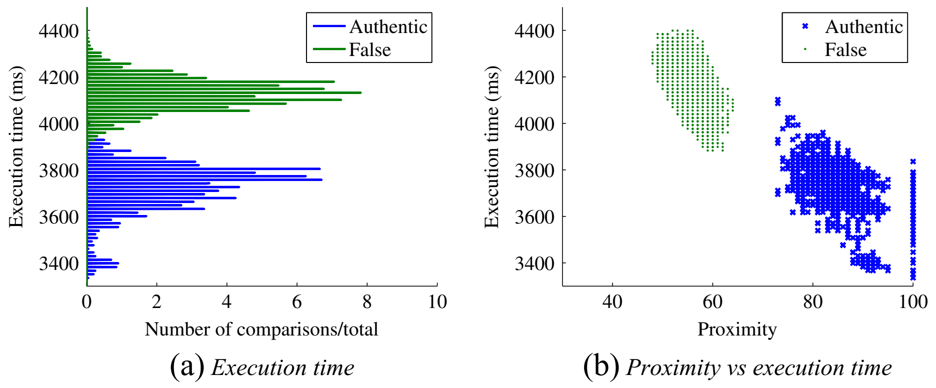


Fig. 16 Execution time when a translation of 2 bit is allowed

count the bits of inputs that are more similar in terms of the proximity measure, resulting in a smaller execution time. In Fig. 16, observe that the points in Figure that are associated with proximity of 100 % are the samples that compared with themselves. These comparisons were used for the only purpose to confirm the the comparison of more similar samples are completed in the smaller time.

Table 4 shows the means and standard deviations of the execution times of the comparisons without translation *vs.* with translations of 1 and 2 bits. Figure 17 compares the average execution times, considering the three analyzed scenarios (*i.e.* no translations or translations of 1 or 2 bits allowed). It is noteworthy to point out that average execution time when translations of 2 bits are allowed is almost 7× bigger than that obtained when no translations are allowed. However, the average execution times when only translations of 1 bit are allowed is only 3× bigger than that obtained when no translations are allowed. Due to the fact that biometric systems based on smartcards require an answer in real time, the translation of 2 bits can be considered as too slow even though it offers very reduced error rates.

Figure 17 presents a comparison chart the average execution times. The average execution time of the comparisons with translations of 2 bits is nearly 7× larger than the average execution time without translation while with the translation of 1 bit, the time increases approximately 3×. As biometric systems using smart-cards require a real-time response, the translation of 2 bit presents a negative factor in spite of the fact that it causes reduced error rates. IN the next section, we provide a viable solution to make a reasonable trade-off between correct comparison result and its underlying execution time.

Table 4 Execution time of comparisons varying the number of translated bits

#Translated bits	Authentic comparison		False comparison	
	Mean (ms)	Std. Dev.	Mean (ms)	Std. Dev.
0	489	22	538	8
1	1520	74	1783	37
2	3725	109	4127	79

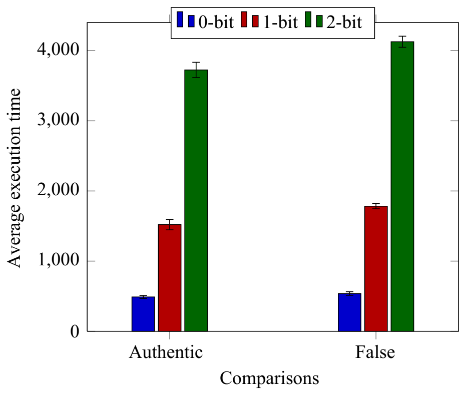


Fig. 17 Execution time of comparisons varying the number of translated bits

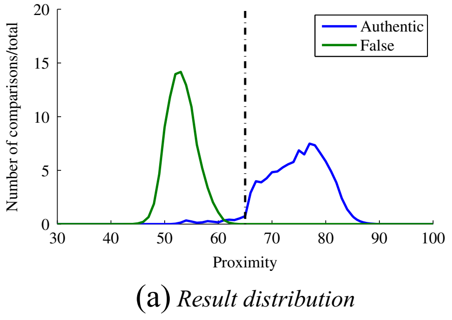
6.5 Comparison with a acceptance threshold

The results presented so far indicate that a smart-card is able to execute a biometric-based method of the palm-print and provide a high performance and high reliability. Nonetheless, it can be weighted whether the given system should prioritize the speed of comparison or the low error rates. From the analysis of the results and in order to mitigate the impact of the execution time, it is possible to use comparisons with translation to achieve the lowest error rates, but imposing an acceptance threshold so that the processing of various displacements of bits can be halted in case of a positive authentication and thereby reducing the overall runtime.

The acceptance threshold is a specific value of proximity, from which the comparisons are considered to be correct, *i.e.* after reaching a predefined proximity, the smart card will have to compare the outcome and may terminate the execution, returning the available result. The definition of the acceptance threshold can only reduce the execution time of authentic comparisons, but does not affect the execution time of false comparisons since, in this case, the execution would not be interrupted. Comparisons with no translation perform the Hamming distance computation only once, so cannot be interrupted. The acceptance threshold was implemented for comparisons with translation of 1 and 2 bits. Here, only the execution times are shown since no improvement would be achieved in terms of comparison results.

The choice of the acceptance threshold is based on the comparison results obtained for when translations of 0, 1 and 2 bits are allowed. The comparison results obtained when translation of 2 bits are allowed, as shown in Fig. 13b, indicate that false comparisons do not occasion proximities greater than 65. Valuing security, this proximity value is used to set up the acceptance threshold.

The utilization of the acceptance threshold will not change the test results. Therefore, we chose new samples fro the database to perform the test whose results are presented in this section. For this purpose, we used 20 samples of 2 different palms, summing up 400 samples and 160,000 comparisons, wherein 8,000 comparisons are authentic and 152,000 false ones. The distribution of the achieved results, when translations of 1 bit are allowed, are shown in Fig. 18a, wherein the straight vertical line indicates the used acceptance threshold. Figure 18b shows the relation between different proximity values and the corresponding FAR and FRR.



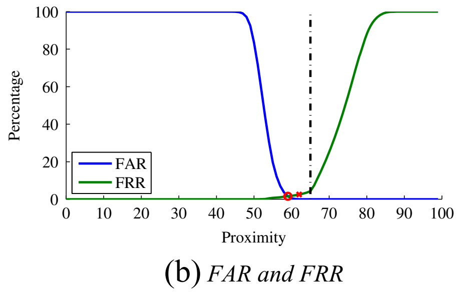
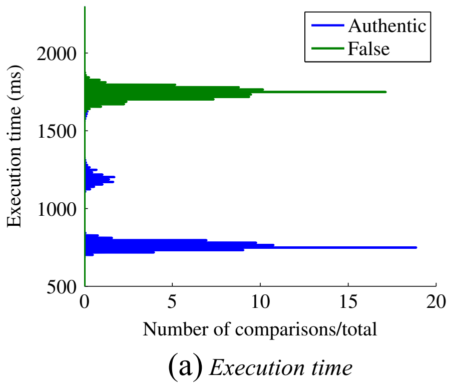


Fig. 18 Comparison result with acceptance threshold when a translation of 1 bit is used

For the chosen acceptance threshold, FAR is 0 % and the corresponding FRR is 3.55 %. The increase in terms of error rate regarding the testes when translations of 1 bit are allowed are due to the random selection of the new samples. This represents a small error rate and considered safe regarding false comparisons. It is possible to observe in both charts, a sudden increase in the curve slope immediately after the chosen acceptance threshold, thus validating the existence of an acceptance threshold since, starting form that point, there is no more result improvement. Figure 18b indicates that it is possible to decrease the acceptance threshold without big impact on the authentication result. Note that the point where FRR is the same as FAR is 1.62 % and the safe FRR occurs at 2.99 %

Figure 19 presents the charts that show the distribution of the execution times during comparisons when translations of 1 bit are allowed with acceptance threshold. It is easy to observe in the chart that are 3 clusters of time to indicate the accepted values and each comparison. As the false comparisons are all denied they are in the cluster with longer execution time while accepted authentic comparisons are clustered in two groups with lower execution times. The average execution time for authentic comparisons is of 868ms with a standard deviation of 249 while the average execution time for false comparisons is of 1744ms with a standard deviation of 37. The latter is similar to the time of the comparisons with translation 1 bit, as expected.



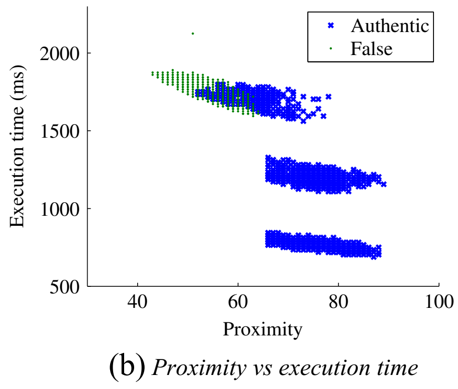


Fig. 19 Execution time with acceptance threshold when a translation of 1 bit is used

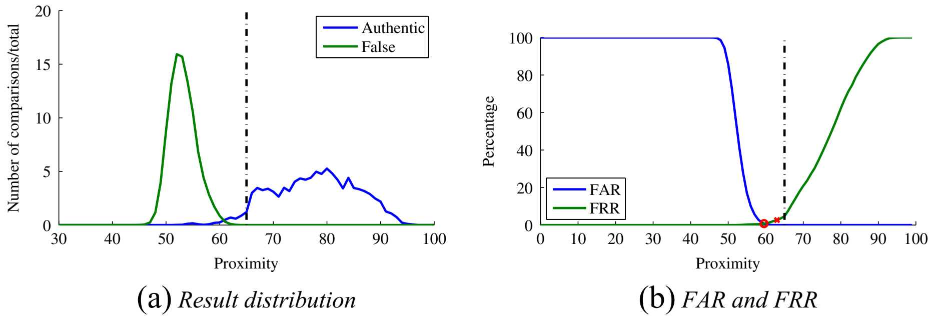


Fig. 20 Comparison result with acceptance threshold when a translation of 2 bit is used

Figure 20 depicts the charts regarding the proximity distribution and FAR *vs*. FRR when comparison allow translations of up to 2 bits as well as the acceptance threshold. For the chosen acceptance threshold, FAR is 0 % and the corresponding FRR is 3.55 %, which coincide with those achieved in the case with translations of 1 bit.

Figure 21 shows the chart related to the runtime comparisons when translations of up 2 bits are allowed as well as the acceptance threshold. There are 5 clusters representing the 5 iterations that actually took place during the comparison process. It is possible to observe that few comparisons performed the last two iterations. The average execution time in the case of authentic comparisons is of 1217ms. When compared to the average execution time of comparisons when translations of 1 bit are allowed without acceptance threshold, which is 1520ms, it can be concluded that the benefit is significant compared to the processing cost.

Figure 22 evaluates the impact of the acceptance threshold usage on the execution times obtained. For the authentic comparisons, the execution times of the comparisons with translation of 1 or 2 bits have shown considerable improvement when using the threshold of acceptance. In the case of an actual utilization, it is expected that only the card-holder

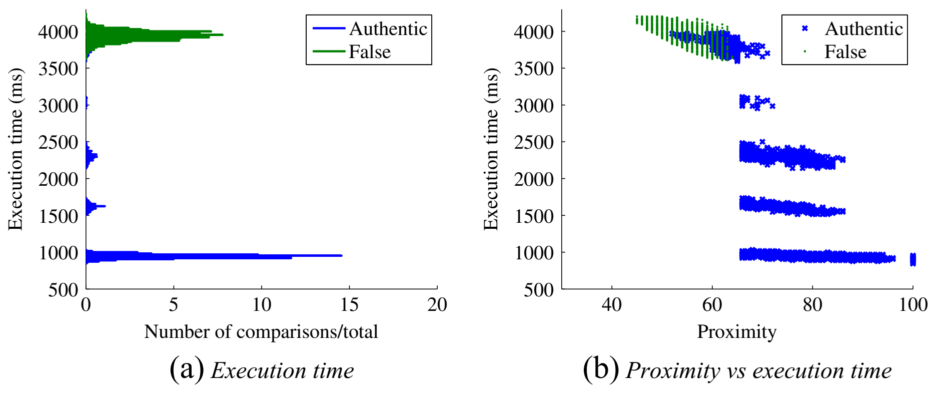


Fig. 21 Execution time with acceptance threshold when a translation of 2 bit is used

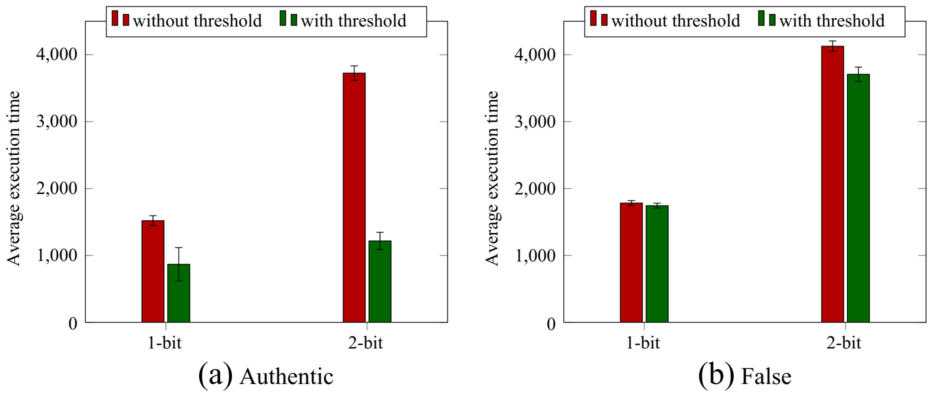


Fig. 22 Execution time of comparisons with threshold varying the number of translated bits

attempts an authentication. Therefore, the execution time for authentic comparisons has a greater relevance in relation to false comparisons. However, false comparisons cannot impose a very high runtime because it has a negative impact on the acceptance of the technology of smart-card with biometrics. Taking this into consideration, comparisons with translation of 1 bit and acceptance threshold would be declared as the best option to use in a practical application.

Last but not least, it is noteworthy to point out that no comparison with third party implementation is possible to intellectual property protection on commercial palm-print verification on smart-cards. Note that any other kind of implementation either on general purpose or deicated hardware would be somehow biased.

7 Conclusions

This study was motivated by the growing need for ever more robust security systems. Biometrics are known as high reliability security tools as well as smart cards, which have been inserted into multiple segments to add new functionalities with security. Therefore, these are two security-related technologies being used together to give rise to an even more robust tool.

The Java Card technology was chosen because of its vast documentation, various tools and community that could aid is in the development process. The proposed implementation followed a common flow, *i.e.* selection of the algorithm; database selection; algorithm implementation; debugging, testing and performance evaluation.

Comparison of palm-print implemented in a smart-card achieved good results both in terms of execution time and effectiveness. The Hamming distance is used to compute the similarity between the stored and input palm-codes. This allowed us to establish a reliable evaluation metric regarding the proximity of the two codes. Using the translations of the palm-code, it is possible to achieve an EER 0 %, *i. e.* using a specific acceptance threshold, there would be no error in comparison results. In other words, all false comparisons will be detected and all true comparisons will be accepted. However, the use of translations causes an increase in execution time. To mitigate the impact of this problem, a threshold of acceptance is proposed. The result is returned once the limit is reached. The results showed that the use of 1-bit translation is sufficient to achieve reduced execution times. Using this strategy, it was possible to improve the average execution time from 1520 ms to 868 ms in comparisons with 1-bit translation. For comparisons with 2-bit translation, which reached the EER rate of 0 %, the average execution time was about 3725 ms with no acceptance limit and 1217 ms when the acceptance threshold is imposed. The safe FRR achieved the case of palm-code comparisons when 1-bit and 2-bit tanslation with acceptance threshold are almost the same, indicating that the use of 1-bit translation is sufficient to ensure reasonable security levels.

Furthermore, regarding the execution time and the accuracy of the implemented algorithms in conjunction with the exploited databases, the biometrics of the palm-print can be considered ideal, as it achieves comparison times shorter than 1 s and still reaches a flase rejection rate of less than 5 % yet this value can reach 0 %, when increasing slightly the comparison execution time.

As a future work, other types of biometrics may be studied in order to verify the possibility of the implementation in a smart-card. As a multi-application card, it is possible that the same card offers more than one type of biometric verification. The implementation of merging of different biometrics can also improve system security.

References

1. Alsmirat M, Jararweh Y, Al-Ayyoub M, Shehab MA, Gupta BB (2016) Accelerating compute intensive medical imaging segmentation algorithms using gpus. Multimedia Tools and Applications
2. Alsmirat MA, Jararweh Y, Obaidat I, Gupta BB (2016) Automated wireless video surveillance framework: Design and evaluation. Journal of Real-Time Image Processing
3. Ashbaugh DR (1999) Quantitative-qualitative friction ridge analysis: an introduction to basic and advanced ridgeology. CRC press, Boca Raton, Estados Unidos
4. Atawneh S, Almomani A, Al Bazar H, Sumari P, Gupta B (2016) Secure and imperceptible digital image steganographic algorithm based on diamond encoding in dwt domain. Multimedia tools and applications
5. Daugman JG (1980) Two-dimensional spectral analysis of cortical receptive field profiles. Vis Res 20(10):847–856
6. Daugman JG (1993) High confidence visual recognition of persons by a test of statistical independence. IEEE Trans Pattern Anal Mach Intell 15(11):1148–1161
7. Daugman JG, et al. (1985) Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. Opt Soc Amer, J A: Opt Image Sci 2(7):1160–1169
8. Emerich S, Lup E, Belean B, Crisan S (2016) Image analysis and coding based on ordinal data representation. In: Image Feature Detectors and Descriptors. Springer-Verlag, Berlin, pp 281–303
9. Gabor D (1946) Theory of communication. part 1: The analysis of information. J Inst of Electr Eng-Part III: Radio Commun Eng 93(26):429–441
10. Hachez G, Quisquater J-J, Koeune F (2000) Biometrics, access control, smart cards: a not so simple combination. In: Smart Card Research and Advanced Applications. Springer, Bristol, Inglaterra, pp 273–288
11. Han C-C, Cheng H-L, Lin C-L, Fan K-C (2003) Personal authentication using palm-print features. Pattern Recogn 36(2):371–381
12. Han Y, Tan T, Sun Z (2007) Palmprint recognition based on directional features and graph matching. In: Advances in Biometrics. Springer, Seoul, Coreia do Sul, pp 1164–1173
13. Healthcare Council. Smart cards and biometrics in healthcare identity applications, 2012
14. Hu D, Feng G, Zhou Z (2007) Two-dimensional locality preserving projections (2dlpp) with its application to palmprint recognition. Pattern Recogn 40(1):339–342
15. Hubel DH, Wiesel TN (1977) Ferrier lecture: Functional architecture of macaque monkey visual cortex. Anais of the Royal Society of London. Series B, Biological Sciences, pp 1–59

16. Jain AK, Chen Y, Demirkus M (2007) Pores and ridges: High-resolution fingerprint matching using level 3 features 29(1):15–27
17. Jain AK, Feng J (2009) Latent palmprint matching. IEEE Trans Pattern Anal Mach Intell 31(6):1032–1047
18. Jain AK, Ross A, Prabhakar S (2004) An introduction to biometric recognition. IEEE Trans Circ Syst Video Technol 14(1):4–20
19. Jing X-Y, Tang Y-Y, Zhang D (2005) A fourier-lda approach for image recognition. Pattern Recogn 38(3):453–457
20. Kittler J, Hatef M, Duin RPW, Matas J (1998) On combining classifiers. IEEE Trans Pattern Anal Mach Intell 20(3):226–239
21. Kong A, Zhang D, Kamel M (2006) Palmprint identification using feature-level fusion. Pattern Recogn 39(3):478–487
22. Kong AW-K, Zhang D (2004) Competitive coding scheme for palmprint verification. In: Anais. 17th International Conference on Pattern Recognition, 2004, vol 1. IEEE, Cambridge, Inglaterra, pp 520–523
23. Kumar A, Zhang D (2005) Personal authentication using multiple palmprint representation. Pattern Recogn 38(10):1695–1704
24. Kumar A, Zhang D (2006) Personal recognition using hand shape and texture. IEEE Trans Image Process 15(8):2454–2461
25. Li W, Zhang D, Xu Z (2002) Palmprint identification by fourier transform. Int J Pattern Recogn Artif Intell 16(04):417–432
26. Li Y, Wang K, Zhang D (2005) Palmprint recognition based on translation invariant zernike moments and modular neural network, pp 177–182
27. Lu G, Zhang D, Wang K (2003) Palmprint recognition using eigenpalms features. Pattern Recogn Lett 24(9):1463–1467
28. PolyU (2013) Polyu 3d palmprint database, 2008 Acessado em fevereiro de
29. Poon C, Wong DCM, Shen HC (2004) Personal identification and verification: fusion of palmprint representations. In: Biometric Authentication. Springer, Hong Kong, China, pp 782–788
30. Pudzs M, Ruskuls RRF, Eglitis T, Kadikis A, Greitans M (2013) Fpga based palmprint and palm vein biometric system. In: Proceedings of IEEE International Conference on Biometrics Special Interest Group (BIOSIG)
31. Shu W, Zhang D (1998) Automated personal identification by palmprint. Opt Eng 37(8):2359–2362
32. Sun Z, Tan T, Wang Y, Li SZ (2005) Ordinal palmprint represention for personal identification representation. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Washington, DC, USA, IEEE Computer Society, vol 1, pp 279–284
33. Wang X, Gong H, Zhang H, Li B, Zhuang Z (2006) Palmprint identification using boosting local binary pattern. In: 18th International Conference on Pattern Recognition, 2006. ICPR 2006, vol 3. IEEE, Hong Kong, China, pp 503–506
34. Wei L, Zhang B, Lei Z, Yan J (2012) Principal line-based alignment refinement for palmprint recognition. IEEE Trans Syst, Man, Cybern, Part C 42(6):1491–1499
35. Wu X, Wang K, Zhang D (2002) Fuzzy directional element energy feature (FDEEF) based palmprint identification. In: Proceedings of the 16th International Conference on Pattern Recognition, Québec City, QC, Canada, IEEE Computer Society, vol 1, pp 95–98
36. Wu X, Zhang D, Wang K (2003) Fisherpalms based palmprint recognition. Pattern Recogn Lett 24(15):2829–2838
37. Wu X, Zhang D, Wang K (2006) Fusion of phase and orientation information for palmprint authentication. Pattern Anal Appl 9(2-3):103–111
38. Wu X, Zhang D, Wang K (2006) Palm line extraction and matching for personal authentication. IEEE Trans Syst, Man Cybern Part A: Syst Humans 36(5):978–987
39. Wyant RS, Nedjah N, De Macedo Mourelle L (2014) Efficient biometric palm-print matching on smart-cards. In: Computational Science and Its Applications - ICCSA 2014 - 14th International Conference. Proceedings, Part VI, Guimarães, Portugal, pp 236–247
40. Yang J, Zhang D, Yang J-Y, Niu B (2007) Globally maximizing, locally minimizing: unsupervised discriminant projection with applications to face and palm biometrics. IEEE Trans Pattern Anal Mach Intell 29(4):650–664
41. You J, Kong W-K, Zhang D, Cheung KH (2004) On hierarchical palmprint coding with multiple features for personal identification in large databases. IEEE Trans Circ Syst Video Technol 14(2):234–243
42. Zhang D, Kong W-K, You J, Wong M (2003) Online palmprint identification. IEEE Trans Pattern Anal Mach Intell 25(9):1041–1050
43. Zhang D, Lu G, Li W, Zhang L, Luo N (2009) Palmprint recognition using 3-d information. IEEE Trans Syst, Man, Cybern, Part C: Appl Rev 39(5):505–519

44. Zhang D, Zuo W, Yue F (2012) A comparative study of palmprint recognition algorithms. ACM Comput Surv (CSUR) 44(1):2
45. Zuo W, Wang K, Zhang D (2005) Bi-directional pca with assembled matrix distance metric. In: IEEE International Conference on Image Processing, 2005. ICIP 2005, vol 2. IEEE, Genoa, Itlia, pp II–958

Nadia Nedjah is an associate professor in the Department of Electronics Engineering and Telecommunications at the Faculty of Engineering, State University of Rio de Janeiro, Brazil. Her research interests include functional programming, embedded systems and reconfigurable hardware design as well as cryptography. Nedjah received her Ph.D. in Computation from the University of Manchester - Institute of Science and Technology (UMIST), England, her M.Sc. in System Engineering and Computation from the University of Annaba, Algeria and her Engineering degree in Computer Science also from the University of Annaba, Algeria. Contact her at nadia@eng.uerj.br.

Rafael Soares Wyant holds an Engineering degree in Electronics from University of Brasilia in Brazil. He also holds a M.Sc. degree in Electronics engineering at the State University of Rio de Janeiro in Brazil. His research interests include programming, smart cards, biometry, and artificial intelligence in general. Contact her at dabonillac@gmail.com.

Luiza de Macedo Mourelle is an associate professor in the Department of System Engineering and Computation at the Faculty of Engineering, State University of Rio de Janeiro, Brazil. Her research interests include computer architecture, embedded systems design, hardware/software co-design and reconfigurable hardware. Mourelle received her PhD in Computation from the University of Manchester - Institute of Science and Technology (UMIST), England, her MSc in System Engineering and Computation from the Federal University of Rio de Janeiro (UFRJ), Brazil and her Engineering degree in Electronics also from UFRJ, Brazil. Contact her at ldmm@eng.uerj.br.