Signal Processing 91 (2011) 906–918

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Extracting biometric binary strings with minimal area under the FRR curve for the hamming distance classifier

C. Chen *, R. Veldhuis

Department of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands

ARTICLE INFO

Article history:
Received 24 March 2010
Received in revised form
7 July 2010
Accepted 8 September 2010
Available online 17 October 2010

Keywords:
Area under the FRR curve
Hamming distance classifier
Quantization
Bit allocation
Biometric compression and protection
Dynamic programming
Fingerprint and face recognition

ABSTRACT

Extracting binary strings from real-valued biometric templates is a fundamental step in template compression and protection systems, such as fuzzy commitment, fuzzy extractor, secure sketch and helper data systems. Quantization and coding are the straightforward way to extract binary representations from arbitrary real-valued biometric modalities. Afterwards, the binary strings can be compared by means of a Hamming distance classifier (HDC). One of the problems of the binary biometric representations is the allocation of quantization bits to the features. In this paper, we first give a theoretical model of the HDC, based on the features' bit error probabilities after the quantization. This model predicts the false acceptance rate (FAR) and the false rejection rate (FRR) as a function of the Hamming distance threshold. Additionally, we propose the area under the FRR curve optimized bit allocation (AUF-OBA) principle. Given the features' bit error probabilities, AUF-OBA assigns variable numbers of quantization bits to features, in such way that the analytical area under the FRR curve for the HDC is minimized. Experiments of AUF-OBA on the FVC2000 fingerprint database and the FRGC face database yield good verification performances. AUF-OBA is applicable to arbitrary biometric modalities, such as fingerprint texture, iris, signature and face.

1. Introduction

Binary representations for biometrics have drawn considerable interest for their merits in template compression, and particularly template protection [1,2]. Unprotected storage and transfer of biometric information allow direct steal-and-use impersonation, leading to identity theft, since biometric data are closely linked to individuals and cannot be replaced.

Several biometric template protection concepts have been published, such as biohashing [3–7], cancelable biometrics [8,9], biometric key generation [10–16], and biometric key binding [17–28]. Biohashing transforms biometric features according to a user-specific secret key. Cancelable biometrics distort the image of a face or a fingerprint by using a computationally non-invertible geometric distortion function. Biometric key generation schemes directly generate a crypto key from the biometric features. Biometric key binding schemes, including fuzzy commitment, helper data, fuzzy vault, secure sketch, use biometric template to bind a crypto key. In the key generation and key binding schemes, biometric templates are represented as binary strings.

In this paper, we focus on extracting binary biometric strings for a key binding verification scheme [20]. Thus, before being used for template protection purpose, the biometric features need to be transformed into a binary string. Therefore, as shown in Fig. 1, a template protected biometric verification system with binary representations can be generalized into three modules.

Feature extraction: This module aims to extract independent, reliable and discriminative real-valued features from raw measurements. Independent features are highly

* Corresponding author.
E-mail addresses: c.chen@nki.nl (C. Chen), r.n.j.veldhuis@utwente.nl (R. Veldhuis).

0165-1684/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2010.09.008

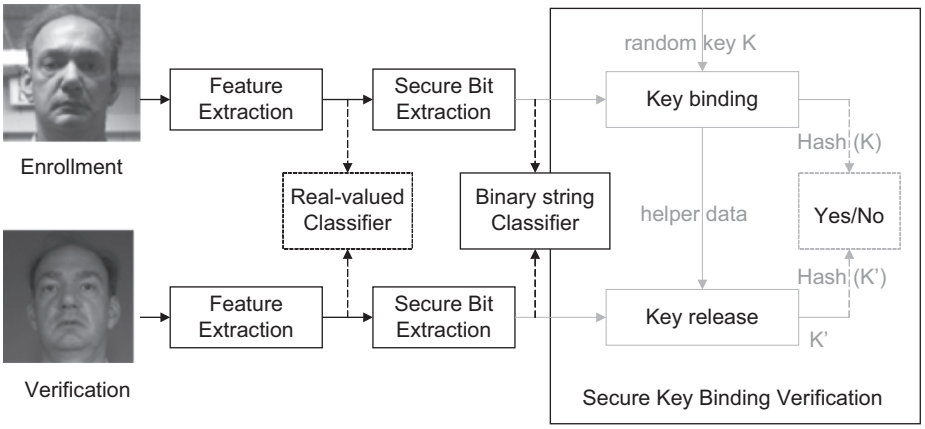

Fig. 1. The scheme of a template protected biometric verification system with binary representations.

desirable for template protection. Independent features are a condition for achieving that the extracted bits in the next secure bit extraction module are independent, which is a requirement considering template security. In this paper we apply classical techniques such as principle component analysis (PCA) and linear discriminant analysis (LDA) [29] as an example, in order to achieve independent features, but other more advanced feature extraction methods can also be used. In a standard biometric system, the extracted features are compared through a real-valued classifier.

Secure bit extraction: This module aims to transform the real-valued features into a fixed-length binary string, which is used to bind a crypto key. Biometric information is well known for its uniqueness. Unfortunately, due to sensor and user behavior, it is inevitably noisy, which leads to intra-class variations. Therefore, it is desirable to extract binary strings that are not only discriminative, but also have low intra-class variations. Such requirements translate to low false acceptance rate (FAR) and false rejection rate (FRR), respectively. Additionally, in order to maximize the attacker's efforts in guessing the target template, the bits should be statistically independent and identically distributed (*i.i.d.*). The straightforward way to extract bits is by quantization and coding.

Secure key binding verification: This module, as presented in [20], aims to provide verification when the target biometric string is protected and bound to a cypto key. In the enrollment stage, a random crypto key K is encoded by an error-correcting encoder into a codeword C. This codeword is further bound to the genuine binary biometric string S through $W = S \oplus C$. In the verification stage, a noisy version C' is released by the operation $C' = W \oplus S'$ of W and the query biometric string S'. Afterwards, C' is decoded into K' through error-correcting decoding. The final 'Yes/No' decision is made by comparing K' and the original K. Essentially, the key binding verification process functions as a Hamming distance classifier (HDC) to the binary biometric strings. That is, the access is granted if and only if the number of bit errors between the target and the query strings is below a Hamming distance threshold.

In this paper we focus on the secure bit extraction module by quantizing and coding every feature individually. To extract bits from every feature involves two tasks: designing the quantization intervals and determining the number of quantization bits. The final binary string is then the concatenation of the output bits from all the features.

First we give an overview of some bits extraction methods. As illustrated in Fig. 2, designing a quantizer relies on two probability density functions (PDFs) that are analyzed for each feature: the background PDF and the genuine user PDF, representing the probability densities of the imposters and the genuine user, respectively. The PDFs are estimated from training or enrollment samples, sometimes under Gaussian assumptions. So far, a number of one-dimensional quantizers have been proposed [19–21,14,30,15,31]. Quantizers in [19–21] are user-independent, constructed merely from the background PDF, whereas quantizers in [14,30,15,31] are user-specific, constructed from both the genuine user PDF and the background PDF. Theoretically, user-specific quantizers provide better FAR and FRR performances. Particularly, the likelihood-ratio based quantizer [31], which is optimal in the Neyman–Pearson sense. Quantizers in [19,14,30,15] have equal-width intervals. Unfortunately, this leads to potential threats. Features obtain higher probabilities in certain quantization intervals than others, thus attackers can more easily find the genuine interval by continuously guessing the one with the highest probability. To avoid this problem, quantizers in [20,21,31] have equal-probability intervals, which meets the *i.i.d.* bit requirements mentioned above.

Once the quantizer type has been determined, a bit allocation principle is desired to determine the number of quantization bits for every single feature. So far, a fixed bit allocation (FBA) principle [20,21,31] and a detection rate optimized bit allocation (DROBA) principle [32] have been proposed. The FBA principle assigns a fixed number of bits to every feature. As seen in Fig. 2, in order to obtain a low overall error probability, it is efficient to extract more bits for a distinctive feature and fewer bits for a non-distinctive feature [33]. The DROBA principle solves this problem by assigning a variable number of bits based on

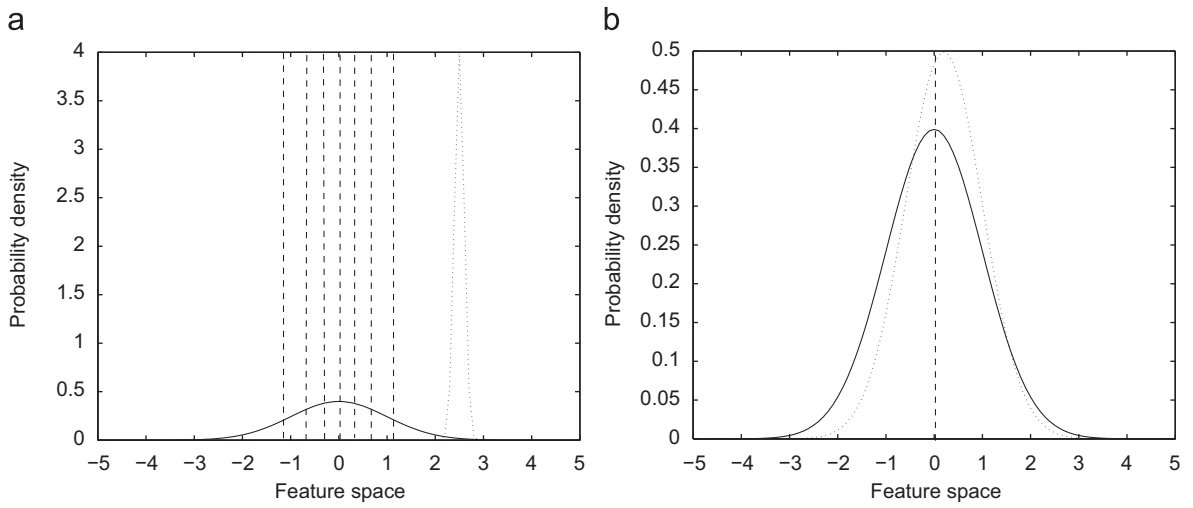

Fig. 2. Two examples of quantizer, given the background PDF (solid), the genuine user PDF (dot), and the quantization intervals (dash). (a) The distinctive genuine user PDF can be quantized into 3 bits. (b) The non-distinctive genuine user PDF is only quantized into 1 bit.

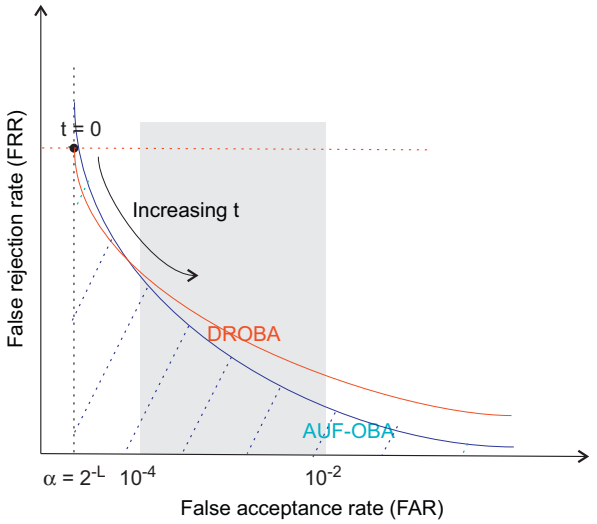

Fig. 3. Illustration of DROBA and AUF-OBA principles.

the statistical properties of every feature, so that the theoretical overall detection rate at the zero Hamming distance threshold is maximized. It is worth mentioning that binary biometrics are also used outside the context of template protection, such as the iris code [34,35] quantized by the iris features. Iris code uses a fixed bit allocation method based on the approximation that the features are equally distinctive.

Although DROBA yields reasonably good performances, in Fig. 3 we illustrate that in principle it only minimizes the FRR performance at zero Hamming distance threshold. Thus it does not provide the optimal solution at the commonly used operational points with a FAR between 10^{-4} and 10^{-2}. Furthermore, as mentioned before, it is important to extract binary strings that provide good performances for the Hamming distance classifier, since it models the secure classification that allows a certain number of errors. Therefore, in this paper, we propose an area under the FRR curve optimized bit allocation (AUF-OBA) principle for the Hamming distance classifier.

We first show that given the features' bit error probabilities after the quantization, we can predict the analytical area under the FRR curve for the Hamming distance classifier (HDC). Then we define the AUF-OBA problem and present a dynamic programming approach to search for the solution.

This paper is organized as follows. In Section 2 we give the analytical performance of a HDC, given the features' bit error probability. In Section 3 we present the AUF-OBA principle. Simulation results are illustrated in Section 4. In Section 5, we give some experimental results of AUF-OBA on the FVC2000 fingerprint database and the FRGC face database. In Section 6 the results are discussed and conclusions are drawn in Section 7.

2. Hamming distance classifier (HDC)

A HDC compares the target string and the query string by computing their Hamming distance. As a result, the query string is accepted if and only if the Hamming distance is smaller than a threshold. Consequently, by varying the threshold, the trade-off between FAR and FRR can be varied. In this section, we show that for a biometric verification problem, the FAR and FRR performance of a HDC can be analytically computed, once the bit error probabilities for both the genuine user and the imposters are known.

We begin by defining the bit error probabilities for the binary strings. Suppose a sequence of L bits is extracted from D independent real-valued features, i.e. $\sum_{j=1}^{D} b_j = L$, where b_j bits are extracted from the jth feature.

During the enrollment, let $s_{g,j}$ denote the string of b_j bits generated by the genuine user for the jth feature. The entire L-bit string for the genuine user $\boldsymbol{s}_g$ is then the

concatenation of the bits extracted from every single feature, i.e. $\boldsymbol{s}_g = s_{g,1} \ldots s_{g,D}$. Similarly, during the verification, let $s'_{g,j}$ and $s'_{i,j}$ be the bits generated by the genuine user and the imposters, respectively, for the jth feature, and $\boldsymbol{s}'_g$ and $\boldsymbol{s}'_i$ be their corresponding entire L-bit string. We know that during the verification, due to the intra-class variation, the genuine user might not extract the same string as the enrollment template, i.e. $s'_{g,j} \neq s_{g,j}$. Contrarily, the imposter might end up with the same string as that of the genuine user in the enrollment, i.e. $s'_{i,j} = s_{g,j}$. Therefore, we introduce the following definitions.

Definition 1. For the jth feature, we define the bit error probabilities for $s'_{g,j}$ and $s'_{i,j}$ when compared to $s_{g,j}$:

$$P_{g,j}(k_j; b_j) = P\{d_H(s_{g,j}, s'_{g,j}) = k_j\}, \quad k_j \in 0, \ldots, b_j, \tag{1}$$

$$P_{i,j}(k_j; b_j) = P\{d_H(s_{g,j}, s'_{i,j}) = k_j\}, \quad k_j \in 0, \ldots, b_j, \tag{2}$$

where d_H is the Hamming distance between two input bit strings. Hence $P_{g,j}$ and $P_{i,j}$ represent – for the genuine user and the imposters, respectively – the probability of having k_j bits error in the b_j bits extracted for the jth feature during the verification.

Definition 2. Regarding a total of D features, we define the bit error probabilities for $\boldsymbol{s}'_g$ and $\boldsymbol{s}'_i$ when compared to $\boldsymbol{s}_g$:

$$\phi_g(k; \{b_j\}_{j=1}^D) = P\{d_H(\boldsymbol{s}_g, \boldsymbol{s}'_g) = k\}, \quad k \in 0, \ldots, L, \tag{3}$$

$$\phi_i(k; \{b_j\}_{j=1}^D) = P\{d_H(\boldsymbol{s}_g, \boldsymbol{s}'_i) = k\}, \quad k \in 0, \ldots, L, \tag{4}$$

where $\phi_g(k)$ and $\phi_i(k)$ represent – for the genuine user and the imposters, respectively – the probability of having k bits error in the entire L bits extracted during the verification.

Note that the bit assignment $\{b_j\}_{j=1}^D$ determines the binary strings. Consequently the bit error probabilities (e.g. $P_{g,j}$, $P_{i,j}$, ϕ_g, ϕ_i) depend on the bit assignment as well. Assuming that the features are statistically independent, their bit errors will also be independent. The total number of bit errors will be the sum of the bit errors of the individual, independent features. Therefore, according to the sum rule for independent random variables [36], the error probability of the whole feature set equals the convolution of the individual probabilities of the features. Thus ϕ_g and ϕ_i can be computed from the convolution of $P_{g,j}$ and $P_{i,j}$:

$$\phi_g(k; \{b_j\}_{j=1}^D) = (P_{g,1} * P_{g,2} * \cdots * P_{g,D})(k; \{b_j\}_{j=1}^D), \tag{5}$$

$$\phi_i(k; \{b_j\}_{j=1}^D) = (P_{i,1} * P_{i,2} * \cdots * P_{i,D})(k; \{b_j\}_{j=1}^D). \tag{6}$$

Expressions in (5) and (6) are the bit error probabilities of the binary string for the genuine user and the imposters. Based on these, we can further compute the analytical FAR and FRR performances of the HDC.

Definition 3. The FAR (α) at the Hamming distance threshold t, ($0 \le t \le L$), is defined as

$$\alpha(t; \{b_j\}_{j=1}^D) = P\{d_H(\boldsymbol{s}_g, \boldsymbol{s}'_i) \le t\}. \tag{7}$$

Given (4), we have

$$\alpha(t; \{b_j\}_{j=1}^D) = \sum_{k=0}^{t} \phi_i(k; \{b_j\}_{j=1}^D). \tag{8}$$

Furthermore, to obtain *i.i.d.* bits, an equal-probability quantizer [20,21,31], with 2^{-b_j} probability mass for every interval, is required for the quantization of every feature. Thus, for the jth feature, when assigned with 2^{b_j} code words, the $P_{i,j}(k_j; b_j)$, as defined in (2), becomes

$$P_{i,j}(k_j; b_j) = 2^{-b_j} \binom{b_j}{k_j}. \tag{9}$$

Subject to $\sum_{j=1}^D b_j = L$, the FAR in (7) becomes

$$\alpha(t; \{b_j\}_{j=1}^D) = \sum_{k=0}^{t} \phi_i(k; \{b_j\}_{j=1}^D) = 2^{-L} \sum_{k=0}^{t} \binom{L}{k}. \tag{10}$$

The proof of (10) is given in Appendix A. This expression shows that when quantized by an equal-probability quantizer, the FAR only depends on the string length L and becomes independent of the bit assignment $\{b_j\}_{j=1}^D$.

Definition 4. Similarly, we define the FRR (β) at the Hamming distance threshold t, ($0 \le t \le L$), as

$$\beta(t; \{b_j\}_{j=1}^D) = P\{d_H(\boldsymbol{s}_g, \boldsymbol{s}'_g) > t\}. \tag{11}$$

Given (3), we have

$$\beta(t; \{b_j\}_{j=1}^D) = \sum_{k=t+1}^{L} \phi_g(k; \{b_j\}_{j=1}^D). \tag{12}$$

3. Area under the FRR curve optimized bit allocation (AUF-OBA)

Given the analytical FRR performance in (11), we compute the area under the FRR curve as a criterion for the overall HDC performance. Furthermore, the performance relies on the features' bit error probability $P_{g,j}(k_j; b_j)$ after quantization, more precisely the bit assignment $\{b_j\}_{j=1}^D$. Therefore, in this section, we give the $\{b_j\}_{j=1}^D$ solution that optimizes the area under the FRR curve.

3.1. Problem formulation

The optimization problem is defined for every genuine user. Suppose we need to extract L bits from D independent real-valued features. For every feature, the background PDF and the genuine user PDF are assumed to be known, usually estimated from the training or enrollment samples. Moreover, a quantizer is employed to quantize the jth feature into b_j bits, $j=1,\ldots,D$, $b_j \in \{0, \ldots, b_{\max}\}$.

To minimize the area under the FRR curve, the optimization problem is formulated as

$$\{b_j^*\}_{i=1}^D = \arg \min_{\sum_{j=1}^D b_j = L} A_{FRR}$$

$$= \arg \min_{\sum_{j=1}^{D} b_j = L} \sum_{t=0}^{L} \beta(t; \{b_j\}_{j=1}^{D}). \quad (13)$$

3.2. AUF-OBA solution

We first reformulate the FRR in (12) into the following expression:

$$\beta(t; \{b_j\}_{j=1}^{D}) = \sum_{l=0}^{L} u(l-(t+1))\phi_g(l; \{b_j\}_{j=1}^{D}) \quad (14)$$

with

$$u(l) = \begin{cases} 1, & l \geq 0, \\ 0, & l < 0. \end{cases} \quad (15)$$

The newly introduced function u allows us to enlarge the summation index range from $[k+1,L]$ to $[0,L]$, which simplifies the computation. Therefore the area under the FRR curve becomes

$$\begin{aligned} A_{\mathrm{FRR}} &= \sum_{t=0}^{L} \beta(t; \{b_j\}_{j=1}^{D}) \\ &= \sum_{t=0}^{L} \sum_{l=0}^{L} [u(l-(t+1))\phi_g(l; \{b_j\}_{j=1}^{D})] \\ &= \sum_{l=0}^{L} \left[\phi_g(l; \{b_j\}_{j=1}^{D}) \sum_{t=0}^{L} u(l-(t+1)) \right] \\ &= \sum_{l=0}^{L} l\phi_g(l; \{b_j\}_{j=1}^{D}). \end{aligned} \quad (16)$$

Expression (16) is the expected value of the number of bit errors k, which we denote by $E[k;\{b_j\}_{j=1}^{D}]$. Hence, A_{FRR} equals $E[k;\{b_j\}_{j=1}^{D}]$.

$$A_{\mathrm{FRR}} = E[k; \{b_j\}_{j=1}^{D}]. \quad (17)$$

Furthermore, we know that the k-bit error of a L-bit binary string come from D real-valued features. Thus with k_j ($j=1,\ldots,D$) bits error per feature. Furthermore, we have that the expected value of a sum equals the sum of the expected values. Therefore,

$$A_{\mathrm{FRR}} = E[k; \{b_j\}_{j=1}^{D}] = \sum_{j=1}^{D} E[k_j; b_j], \quad (18)$$

where $E[k_j;b_j]$ is the expected value of the number of errors k_j for the jth feature:

$$E[k_j; b_j] = \sum_{l=0}^{b_j} lP_{g,j}(l; b_j). \quad (19)$$

We can now reformulate the AUF-OBA problem as

$$\{b_j^*\}_{j=1}^{D} = \arg \min_{\sum_{j=1}^{D} b_j = L} \sum_{j=1}^{D} E[k_j; b_j]. \quad (20)$$

Furthermore, let $G_j(b_j)$ be a gain factor, defined as

$$G_j(b_j) = -E[k_j; b_j]. \quad (21)$$

The AUF-OBA then becomes a maximization problem:

$$\begin{aligned} \{b_j^*\}_{j=1}^{D} &= \arg \min_{\sum_{j=1}^{D} b_j = L} \sum_{j=1}^{D} E[k_j; b_j], \\ &= \arg \max_{\sum_{j=1}^{D} b_j = L} \sum_{j=1}^{D} G_j(b_j). \end{aligned} \quad (22)$$

With the gain factor defined in (21), the problem in (22) has the same form as the DROBA optimization problem presented in [32]. Therefore, solving (22) involves two steps: (1) computing $G_j(b_j)$ for every feature j; (2) finding the optimal $\{b^*_j\}_{j=1}^{D}$ through the same dynamic programming procedure as proposed in DROBA [32].

3.3. Computing $G_j(b_j)$

To compute $G_j(b_j)$, the genuine user bit error probability $P_{g,j}(k_j;b_j)$ is required. As defined in (1), given the feature's genuine user PDF $p_{g,j}$, the quantizer and the number of quantization bits b_j, we can compute $P_{g,j}(k_j;b_j)$ as

$$P_{g,j}(k_j; b_j) = \int_{Q(k_j;b_j)} p_{g,j}(v)\, dv, \quad (23)$$

where $Q(k_j;b_j)$ indicates the quantization intervals with k_j-bit error as compared to the genuine code $s_{g,j}$. An example of these intervals encoded by a Gray code [37] is illustrated in Fig. 4.

3.4. Dynamic programming approach

The optimization problem in (22) has the same form as DROBA [32]. Therefore, once the $G_j(b_j)$ is computed, (22) can be solved by a common recursive dynamic programming approach, as described in Appendix B. As explained in [32], the essential concept is that the optimal bits assignment for j features can be computed directly from

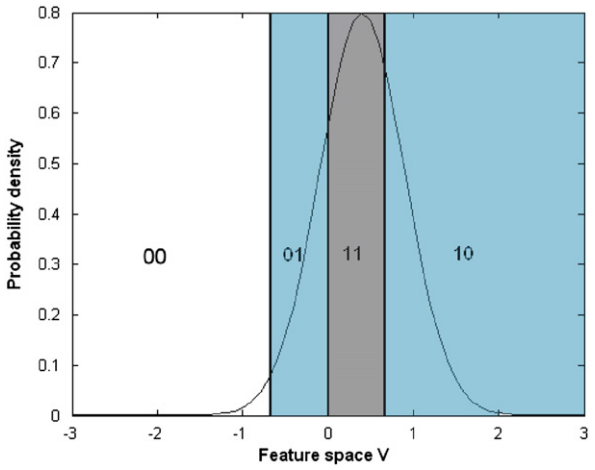

Fig. 4. An example of computing $P_{g,j}(k_j;b_j)$ for the jth feature, assigned with $b_j=2$ bits Gray code. The genuine user PDF $p_{g,j}$ (black curve); $Q(0;2)$ with the genuine code '11' (gray); $Q(1;2)$ with 1-bit error (blue); and $Q(2;2)$ with 2-bit error (white). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the optimal bits assignment for $j-1$ features. Therefore, the final optimal bits assignment can be computed through an iterative procedure. The number of operations per iteration step is about $O((j-1)\times b_{\max}^2)$, leading to a total number of operations of $O(D^2\times b_{\max}^2)$, which is significantly less than a brute force search.

4. Simulations on synthetic data

In this section we test the HDC performances of the bit strings extracted with AUF-OBA, on randomly generated independent features. The background PDF of every feature is generated as a Gaussian density with zero-mean and unit-variance, i.e. $p_{\mathrm{b},j}=N(v,0,1)$. Additionally, the genuine user PDF of every feature is generated as a Gaussian density with user-specific mean and standard deviation, i.e. $p_{\mathrm{g},j}=N(v,\mu_j,\sigma_j)$. The quantizer that we employed to compute $P_{\mathrm{g},j}(k_j;b_j)$ in (23) is the user-independent equal-probability quantizer [20,21,31], defined as

$$B_0=-\infty, \tag{24}$$

$$B_m=\arg\underset{B}{}\left[\int_{B_{m-1}}^{B_m} p_{\mathrm{b},j}\,dv=2^{-b_j}\right],\quad m=1,\ldots,2^{b_j}, \tag{25}$$

where $(B_{m-1},B_m]$ represents the mth quantization interval. The quantization symbols are assigned with Gray code, and we set $b_{\max}=3$. Thus, given D features and a predetermined length L, we search for the $\{b_j\}_{j=1}^{D}$ through the DP process in Appendix B. Afterwards, we compute the corresponding FAR and FRR performances for HDC according to (10) and (11).

Fig. 5 shows the FAR vs. FRR performances by increasing the binary string length (L=31, 63, 127), given a fixed set of features (D=50). Results show that there exist a number of bits (e.g. close to L=63) that gives the optimal trade-off in terms of FAR and FRR.

Fig. 6 shows the FAR vs. FRR performances by increasing the input features (D=50, 100, 150), at a predetermined string length (L=127). The FAR performance merely depends on L and thus is fixed. While increasing the number of features, the FRR performances always improve. This result suggests that AUF-OBA tends to extract distinctive bits as the number of input features increases.

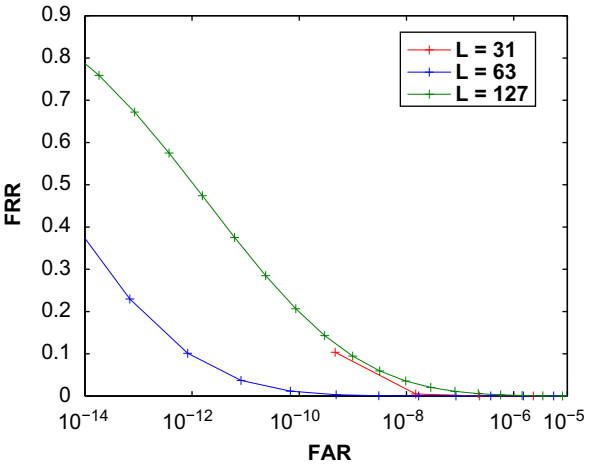

Fig. 5. The FAR vs. FRR performances of AUF-OBA on the synthetic features, when the output L=31, 63 and 127, at D=50.

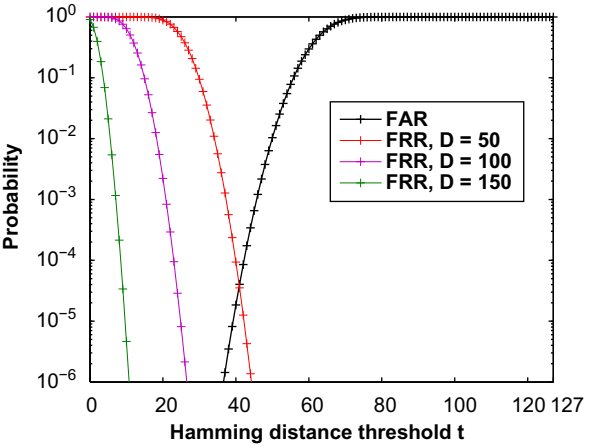

Fig. 6. The FAR vs. FRR performances of AUF-OBA on the synthetic features, when the input D=50, 100 and 150, at L=127.

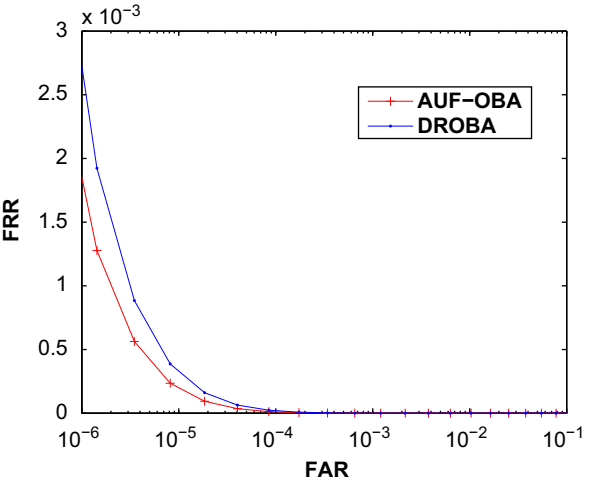

Fig. 7. The FAR vs. FRR performances of AUF-OBA on the synthetic features, compared with DROBA, at D=50, L=127.

In Fig. 7, we further compare the FAR vs. FRR performances between AUF-OBA and DROBA, at D=50, L=127. Although DROBA minimizes the highest FRR at zero Hamming distance threshold, AUF-OBA obtains lower FRR at the operational area where FAR is between 10^{-4} and 10^{-2}.

In the simulations, both the background PDF and the genuine user PDF are assumed to be Gaussian. In Section 5.2.4 we tested this Gaussian assumption on real data.

5. Real data experiments

In this section we conduct the experiments with AUF-OBA on real data. We first investigate the verification performances while varying the input feature dimensionality D and

the output binary string length L. From the best D–L settings we analyze the bits capacity of the features. Afterwards, we compare AUF-OBA with DROBA. Finally, we discuss the independent Gaussian hypothesis by comparing the empirical results with the predicted FAR and FRR performances.

5.1. Experimental setup

We tested the AUF-OBA on three datasets, derived from the FVC2000(DB2) fingerprint database [38] and the FRGC(version 1) face database [39]. One important consideration for biometric protection system is that it is not allowed to conduct the user-specific image alignment, since the reference image, as a template, is encrypted. Therefore, we could only rely on absolute alignment methods or alignment-free measurements. In this paper, we applied basic absolute alignment methods.

- *FVC2000*: This is the FVC2000(DB2) fingerprint dataset, containing eight images of 110 users. Images are aligned to an automatically detected standard core point position through translation. As illustrated in Fig. 8, the raw measurements contain two categories: the squared directional field in both x and y directions, and the Gabor response in four orientations (0, $\pi/4$, $\pi/2$, $3\pi/4$). Determined by a regular grid of 16 by 16 points with spacing of eight pixels, measurements are taken at 256 positions, leading to a total of 1536 elements [20].
- *FRGC$_H$*: This is a subset of FRGC(version 1), containing 275 users with various numbers of high quality images, taken under controlled conditions. The number of samples n per user ranges from 4 to 36. As illustrated in Fig. 9, a set of four standard landmarks, i.e. eyes, nose and mouth, is used to align the faces to a standard reference face. The measurements with 8762 elements are the gray pixel values, picked from a region of interest (ROI) with size 128×128.
- *FRGC$_L$*: This is a subset of FRGC(version 1), containing 198 users with low quality images (n from 4 to 16), taken under uncontrolled conditions. The alignment and measurements are the same as FRGC$_H$.

We randomly selected different users for training and testing and repeated our experiment with a number of trials. The data division is described in Table 1.

Our experiments involved three steps: training, enrollment and verification. According to the requirement for the feature extraction module, independent features are necessary. Thus, any method that extracts independent

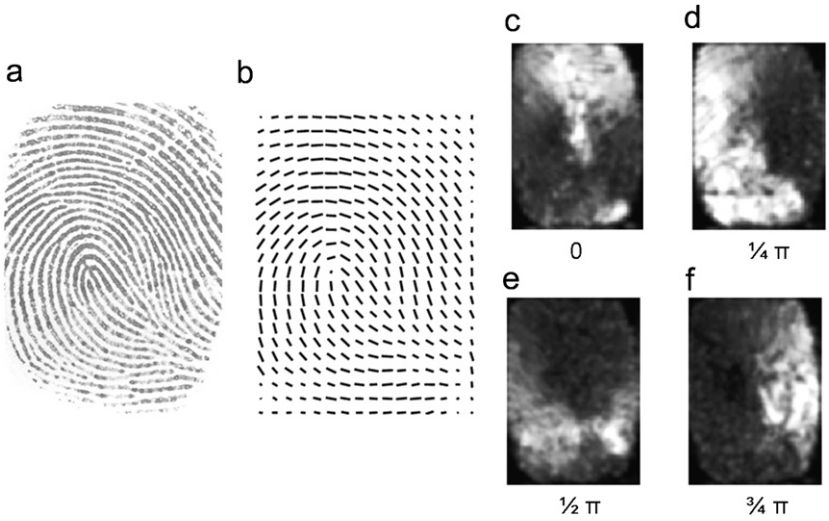

Fig. 8. (a) Fingerprint image, (b) directional field, (c)–(f) the absolute values of Gabor responses for different orientations θ.

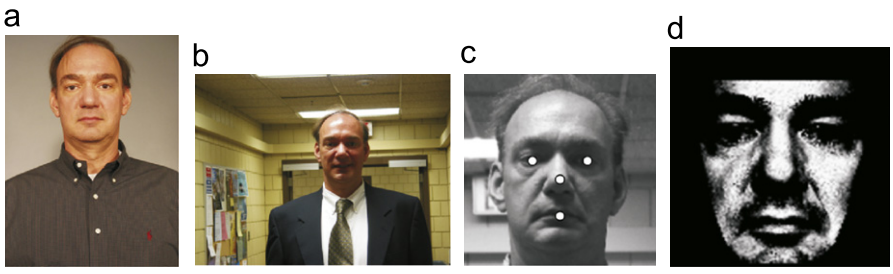

Fig. 9. (a) Controlled image, (b) uncontrolled image, (c) landmarks and (d) the region of interest (ROI).

features can be applied. During the training step in our experiment, we applied a common PCA/LDA [40] method on the training set. That is, we first applied PCA to obtain the projections on the eigenvectors at a reduced dimensionality. Based on which we further applied LDA to pick the eigenvectors that yield the largest within and between class scatters. The obtained transformation was then applied to both the enrollment and verification sets. We assume that the measurements are with Gaussian density, thus after the PCA transformation, the extracted features are statistically independent. Additionally, the LDA method we applied assumes user-independent intra-class variance, so that the extracted features are statistically independent for every genuine user as well. In the enrollment step, for the jth feature, we first have to estimate both the background PDF $p_{b,j}$ and the genuine user PDF $p_{g,j}$. In [32], it is shown that modeling every feature as Gaussian density gives reasonably good performances. Therefore, we model both PDFs as Gaussian density $p_{b,j}=N(\nu,0,1)$, $p_{g,j}=N(\nu,\mu_j,\sigma_j)$. Additionally, we set $b_{max}=3$, and the gain factor G_j was computed from the fixed quantizer in (25). Afterwards, we applied the AUF-OBA for every genuine user. Based on the output bit assignment $\{b^*_j\}_{j=1}^{D}$, the features were coded with Gray code. In the verification step, the features of the query user were quantized and coded according to the $\{b^*_j\}_{j=1}^{D}$ of the target user, resulting in a query binary string. Finally the query binary string was compared with the target binary string by using a HDC.

Table 1
Data division: number of users × number of samples per user (n), and the number of trials for FVC2000, $FRGC_H$ and $FRGC_L$.

	Training	Enrollment	Verification	Trials
FVC2000	$80\times n$	$30\times 3n/4$	$30\times n/4$	20
$FRGC_H$	$210\times n$	$65\times 3n/4$	$65\times n/4$	5
$FRGC_L$	$150\times n$	$48\times 2n/3$	$48\times n/3$	5

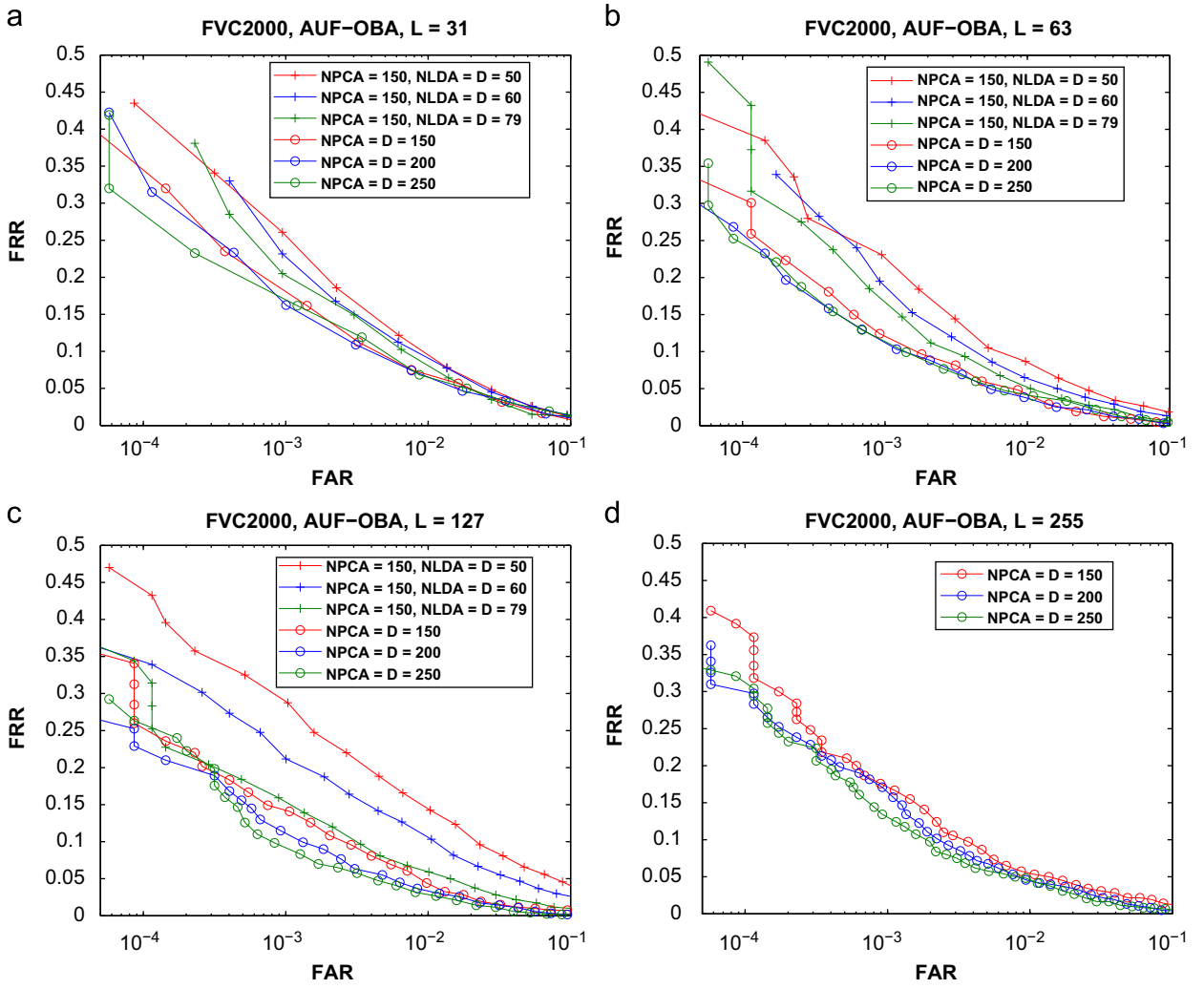

Fig. 10. The FAR vs. FRR performances for FVC2000 extracted with AUF-OBA, from various numbers of features D, at (a) L=31; (b) L=63; (c) L=127 and (d) L=255.

5.2. Experimental results

5.2.1. Verification performance

We tested the binary strings at length L=31, 63, 127 and 255, extracted from various numbers of features D. The FAR vs. FRR performances for FVC2000, $FRGC_H$ and $FRGC_L$ are shown in Figs. 10–12, where the FAR is plotted as a log scale. Since the Hamming distance threshold is an integer, the FAR and FRR performances are discrete.

We first investigate the performances at fixed L by increasing D. For FVC2000, we first applied both PCA and LDA transformation, given L, when the number of features D increases, the performance improves, yet still not satisfying. The reason might be the dimensionality limit (D_{max}=number of training user−1=79) from LDA. To solve this problem, we relax the independency constraint for the genuine user by only applying the PCA transformation, and the performance improves. Fig. 11 suggests that for the high quality data $FRGC_H$, given L, when the number of features D increases, the overall FAR vs. FRR performance improves and becomes stable. These results are consistent to the synthetic data performances in Fig. 6 and prove that AUF-OBA can effectively extract distinctive bits when the feature dimensionality is high. Contrarily, Fig. 12 suggests that for the low quality data $FRGC_L$, given L, when the number of features D increases, the overall FAR vs. FRR performance improves. However, when $D \gg L$, as seen with L=31 and 63 in Fig. 12(a) and (b), the performance starts to deteriorate. The reason is that at a high dimensionality after PCA/LDA transformation, the features of the low quality data become less reliable, and the error probabilities estimated from such features are not accurate. Consequently, AUF-OBA no longer provides the effective bit assignment.

We then investigate the performances at fixed D by increasing L. All three datasets show that given D features, the moderate length L=127 gives the best performances. These results are consistent to the synthetic data performances in Fig. 5. It proves that given a number of features, a maximum number of bits can be extracted that gives the best performances in terms of FAR vs. FRR.

To further investigate the performances at the operational points, we picked the D–L settings with the best performances around the operational points. The FAR vs. FRR performances for FVC2000, $FRGC_H$ and $FRGC_L$ are listed in Table 2. Results show that regarding a compression or template protection system, the FRR performances at FAR $\approx 10^{-4}$ are reasonably good, especially for the high quality data $FRGC_H$.

5.2.2. Bit capacity of features

Since AUF-OBA enables more quantization bits for distinctive features than for non-distinctive feature, the bit assignment to some extent indicates the feature distinctiveness. Therefore, we take the best D–L settings in Table 2, and in Fig. 13 we plot the bit assignment histogram for the features, averaged over all genuine

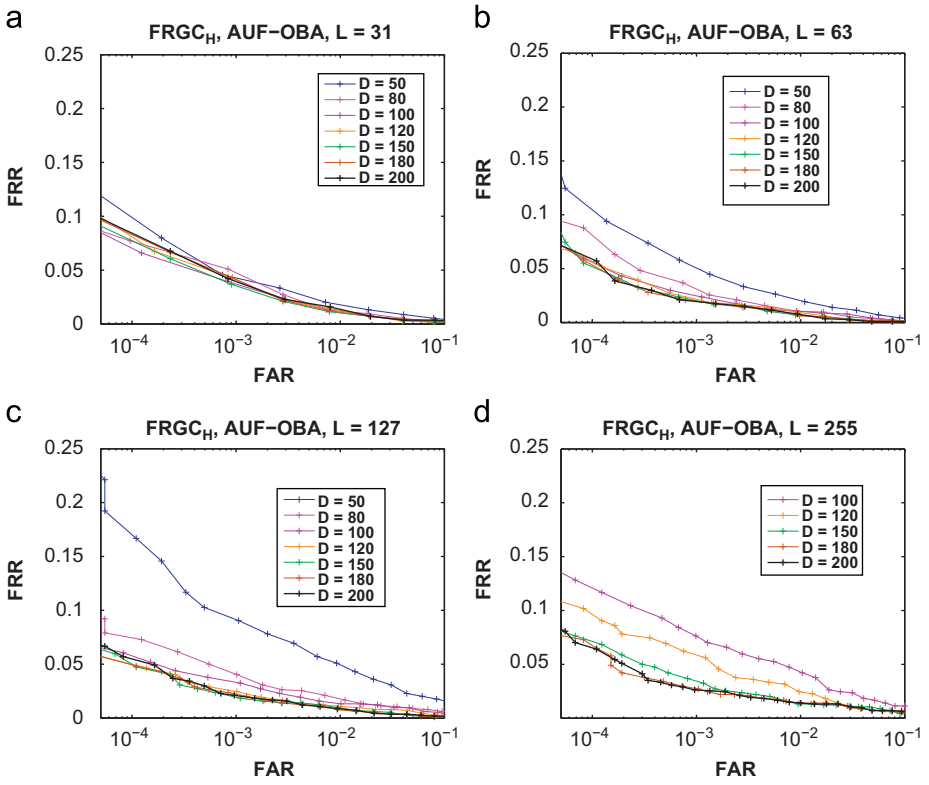

Fig. 11. The FAR vs. FRR performances for $FRGC_H$, with varying D ($NPCA$=250, $NLDA$=D), at (a) L=31; (b) L=63; (c) L=127 and (d) L=255.

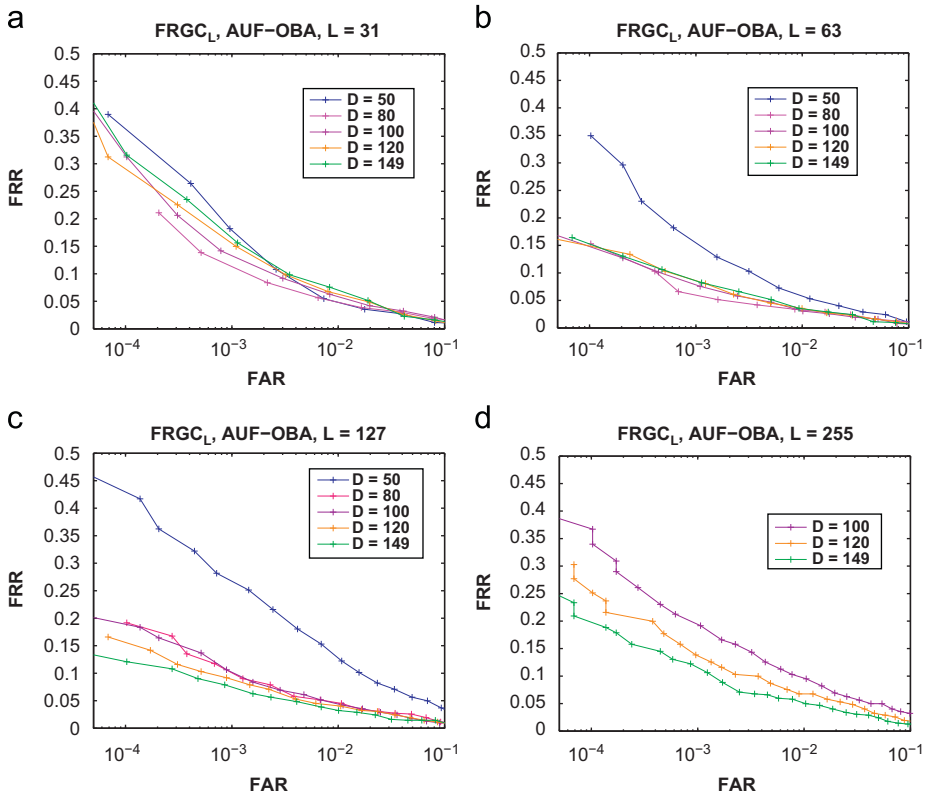

Fig. 12. The FAR vs. FRR performances for FRGC$_L$, with varying D ($NPCA$=250, $NLDA$=D), at (a) L=31; (b) L=63; (c) L=127 and (d) L=255.

Table 2
The FAR vs. FRR performances for (a) FVC2000, (b) FRGC$_H$ and (c) FRGC$_L$.

FVC2000	FRR	FAR	FRR	FAR	FRR	FAR
	(%)		(%)		(%)	
(a)						
D=250, L=31	23.2	0.02	16.1	0.1	5.0	1.8
D=250, L=63	22.0	0.01	9.9	0.1	4.0	1.0
D=250, L=127	**22.0**	**0.01**	**8.3**	**0.1**	**2.6**	**1.0**
D=250, L=255	29.4	0.01	12.4	0.1	4.1	1.1
FRGC$_H$	FRR	FAR	FRR	FAR	FRR	FAR
	(%)		(%)		(%)	
(b)						
D=100, L=31	6.5	0.01	2.3	0.2	0.7	1.8
D=200, L=63	5.7	0.01	1.7	0.1	0	1.7
D=200, L=127	**4.7**	**0.01**	**1.8**	**0.1**	**0**	**1.4**
D=200, L=255	6.4	0.01	2.6	0.1	1.4	1.0
FRGC$_L$	FRR	FAR	FRR	FAR	FRR	FAR
	(%)		(%)		(%)	
(c)						
D=80, L=31	21	0.02	8	0.2	3	1.6
D=80, L=63	15	0.01	5	0.1	3	1.6
D=149, L=127	**12**	**0.01**	**6**	**0.1**	**3**	**1.0**
D=149, L=255	18	0.01	10	0.1	5	1.0

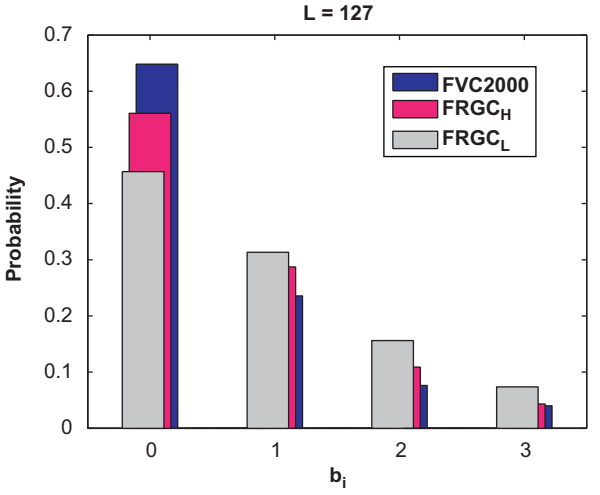

Fig. 13. An example of the bit assignment histogram for the features, averaged over all genuine users, for FVC2000, FRGC$_H$ and FRGC$_L$.

users. All three datasets show consistent results. A large proportion of features are assigned with 0 bits or discarded, which means these features are not distinctive.

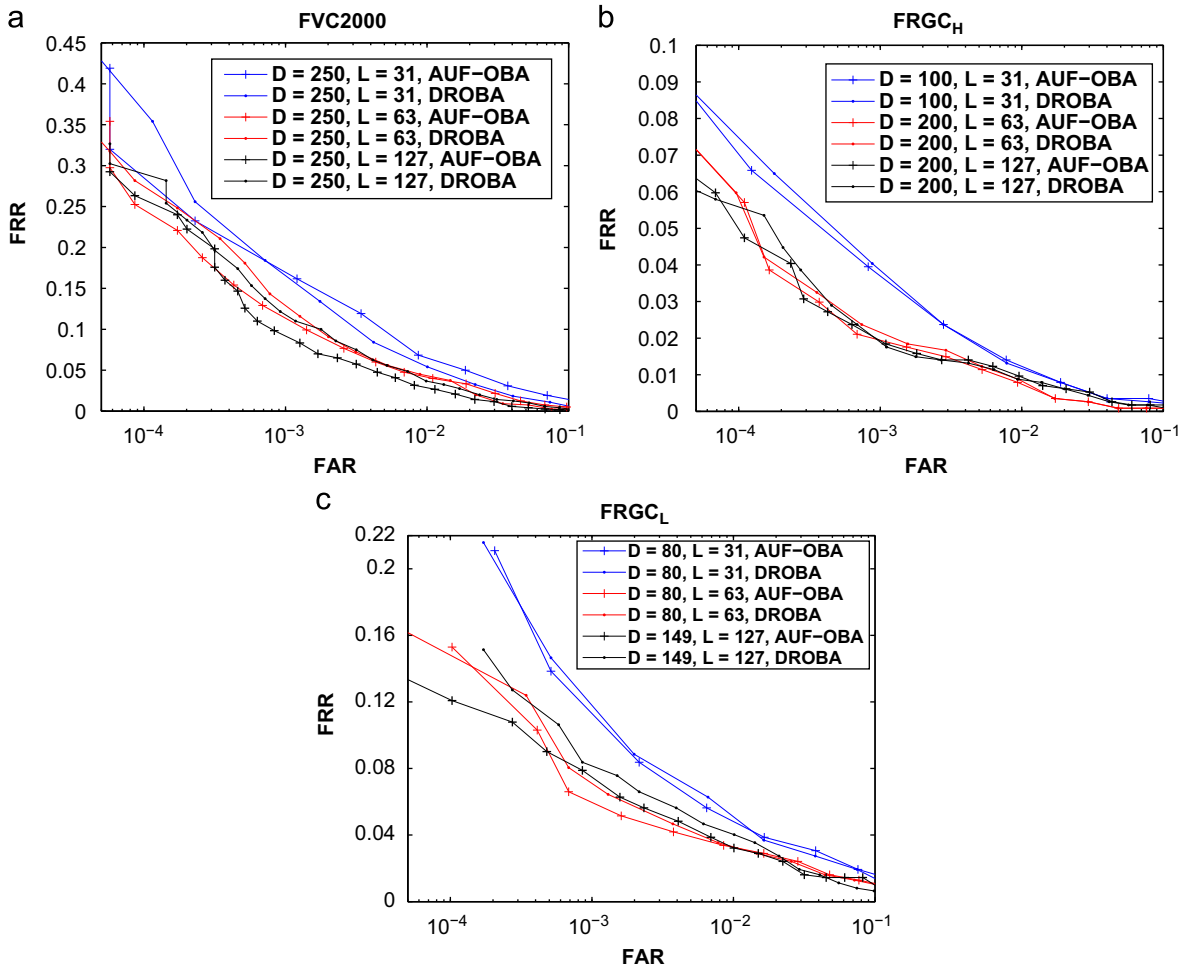

Fig. 14. The FAR vs. FRR performances of AUF-OBA, compared with DROBA, for (a) FVC2000, (b) $FRGC_H$ and (c) $FRGC_L$.

However, only few features are distinctive enough to extract 2 or 3 bits.

5.2.3. Comparison with DROBA

In Fig. 7 we showed that theoretically AUF-OBA is superior to DROBA concerning the performances at the operational points. Now we further compare their performances on the real data. In Fig. 14 we illustrate their performances at the same D–L settings. Results show that AUF-OBA is indeed slightly better than DROBA.

5.2.4. Considerations about the independent Gaussian assumption

One important assumption in AUF-OBA is – for both the imposters and the genuine user – the independency among the features. In our experiments, we assume that the measurements are with Gaussian density, thus after the PCA transformation, the extracted features are independent Gaussian density. Furthermore, in our LDA transformation, we assume that every feature has user-independent intra-class variance, so that the extracted features are also independent for every genuine user. Now we investigate whether the real data comply with these assumptions. However, formally testing the independent Gaussian hypothesis is not within the scope of this paper.

As in the previous experiments, computing the $\{b^*_j\}_{j=1}^{D}$ output of AUF-OBA is based on the independent Gaussian density $p_{g,j}$, $p_{b,j}$. Then, according to (10) and (11), we can compute the theoretical FAR as well as the theoretical averaged FRR performances over all the genuine users. Furthermore, given the $\{b^*_j\}_{j=1}^{D}$, we can evaluate the FAR vs. FRR performance on both the enrollment and the verification datasets. Thus, by comparing the real data and the theoretical performances, we could evaluate whether the real data comply with the independency and the Gaussian density assumptions. In Fig. 15 we give an example of the performances for $FRGC_H$, at D=200, L=127. The overall FAR performance of both the enrollment and verification sets are consistent to the theoretical result, showing that the background PDF fits the Gaussian density and the independency assumption. This results further suggests that the extracted bits are *i.i.d.* However, the empirical averaged FRR performance is higher than the theoretical prediction, suggesting that the features for the genuine user is not fully Gaussian or independent.

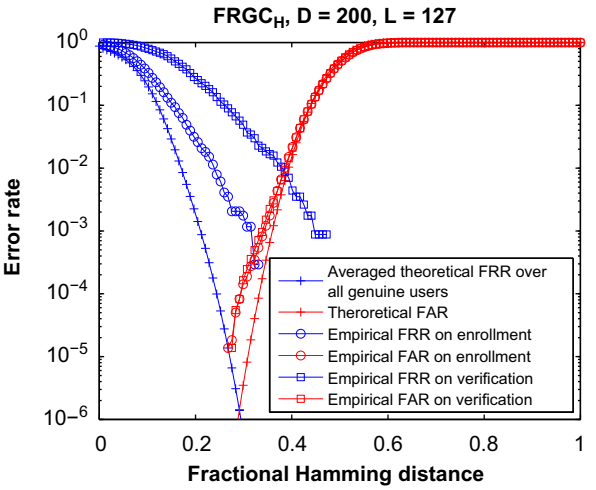

Fig. 15. Comparing the theoretical FAR vs. FRR performances with the FAR vs. FRR performances on the enrollment and verification data.

6. Discussion

An important assumption in AUF-OBA is that after feature extraction (e.g. PCA/LDA), the features are independent among both the entire populations and the genuine user. In Section 5.2.4 we proved the independency among the entire populations. Although it is not true for the genuine user, we see that AUF-OBA still works in such relaxed condition and provides reasonably good FRR performances.

Although AUF-OBA provides an optimal way to extract variable bits, the performances of a template protection biometric system relies on the other factors as well. For instance, aligning the query image for the specific biometric modality, reliably extracting independent features, and applying the error-correcting technique with higher correction capability. From the template protection system perspective, these methods still need further investigation.

7. Conclusion

Binary biometric representations are becoming popular for its benefits in template compression and protection. Quantization and coding are the common way to achieve the binary representation from arbitrary biometric modalities. One of the problems in the quantization is the allocation of quantization bits to the features. In this paper, we first give a theoretical model of the HDC, based on the bit error probability after quantization. This model predicts the FAR and the FRR as a function of the Hamming distance threshold. Additionally, we propose the AUF-OBA principle. Given the features' bit error probabilities after quantization, AUF-OBA assigns variable numbers of quantization bits to features, in such a way that the analytical area under the FRR curve for the HDC is minimized. AUF-OBA is capable of achieving low FRR at a wide range of Hamming distances thresholds, rather than the DROBA principle which optimizes the FRR at Hamming distance threshold zero. Experiments of AUF-OBA on the FVC2000 fingerprint database and the FRGC face database yield good verification performances.

Appendix A. Derivation of the FAR

In order to prove (10), we only need to prove

$$\phi_{\mathrm{i}}(k;\{b_j\}_{j=1}^{D}) = 2^{-L}\binom{L}{k}. \tag{26}$$

Proof. Note that for binomial coefficients $\binom{m}{q}$ and $\binom{n}{p-q}$ Vandermonde's identity states that

$$\sum_{q=0}^{p}\binom{m}{q}\binom{n}{p-q} = \binom{m+n}{p}. \tag{27}$$

Thus, for instance, by using (9) we obtain

$$\begin{aligned}\sum_{l=0}^{k} P_{\mathrm{i},1}(l;b_1)P_{\mathrm{i},2}(k-l;b_2) &= \sum_{l=0}^{k} 2^{-b_1}\binom{b_1}{l}2^{-b_2}\binom{b_2}{k-l}\\ &= 2^{-(b_1+b_2)}\binom{b_1+b_2}{k}.\end{aligned} \tag{28}$$

Expression (28) in fact computes the convolution of the bit error probabilities of two features. In the case of D features, as in (6), ϕ_{i} is the convolution from all the D features. Therefore, we can apply (28) repetitively to all the D features. For instance, to convolve with the third feature, we have

$$\begin{aligned}&\sum_{m=0}^{k}\left[\sum_{l=0}^{m} P_{\mathrm{i},1}(l;b_1)P_{\mathrm{i},2}(m-l;b_2)\right]P_{\mathrm{i},3}(k-m;b_3)\\ &= \sum_{m=0}^{k} 2^{-(b_1+b_2)}\binom{b_1+b_2}{m}2^{-b_3}\binom{b_2}{k-m}\\ &= 2^{-(b_1+b_2+b_3)}\binom{b_1+b_2+b_3}{k}.\end{aligned} \tag{29}$$

Applying this convolution for all D features with $\sum_{j=1}^{D} b_j = L$, we finally leads to the desired result in (26).

This result can also be found by realizing that, for L *i.i.d.* bits with error probability 2^{-1}, the probability of a given set of precisely k bits to be erroneous is 2^{-L} and that there are $\binom{L}{k}$ possibilities to select k bits. □

Appendix B. Dynamic programming approach

Algorithm 1. The dynamic programming approach to solve AUF-OBA principle.

Input:
$D, L, G_j(b_j), b_j \in \{0,\ldots,b_{\max}\}, j = 1,\ldots,D,$

Initialize:
$n = 0,$

$b_0(0) = 0,$

$G^{(0)}(0) = 1,$

while $n \neq D$ **do**
$n = n+1,$

$\hat{b}', \hat{b}'' = \arg\max G^{(n-1)}(b') + G_n(b''),$

$b' + b'' = l,$

$b' \in \{0,\ldots,(n-1)\times b_{\max}\}$,

$b'' \in \{0,\ldots,b_{\max}\}$,

$l = 0,\ldots,n\times b_{\max}$,

$G^{(n)}(l) = G^{(n-1)}(\hat{b}') + G_n(\hat{b}'')$,

$b_j(l) = b_j(\hat{b}'), j = 1,\ldots,n-1$,

$b_n(l) = \hat{b}''$,

end while

Output:

$\{b_j^*\} = \{b_j(L)\}, j = 1,\ldots,D$.

References

[1] U. Uludag, S. Pankanti, S. Prabhakar, A. Jain, Biometric cryptosystems: issues and challenges, Proceedings of the IEEE 92 (6) (2004) 948–960, doi:10.1109/JPROC.2004.827372.

[2] A. Jain, K. Nandakumar, A. Nagar, Biometric template security, EURASIP Journal on Advances in Signal Processing (113).

[3] T. Connie, A. Teoh, M. Goh, D. Ngo, Palmhashing: a novel approach for dual-factor authentication, Pattern Analysis and Applications 7 (3) (2004) 255–268.

[4] A.B.J. Teoh, A. Goh, D.C.L. Ngo, Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (12) (2006) 1892–1901, doi:10.1109/TPAMI.2006.250.

[5] A. Kong, K. Cheung, D. Zhang, M. Kamel, J. You, An analysis of biohashing and its variants, Pattern Recognition 39 (7) (2006) 1359–1368 doi:http://dx.doi.org/10.1016/j.patcog.2005.10.025.

[6] A. Lumini, L. Nanni, An improved biohashing for human authentication, Pattern Recognition 40 (3) (2007) 1057–1065 doi:http://dx.doi.org/10.1016/j.patcog.2006.05.030.

[7] L. Nanni, A. Lumini, Random subspace for an improved biohashing for face authentication, Pattern Recognition Letters 29 (3) (2008) 295–300 doi:http://dx.doi.org/10.1016/j.patrec.2007.10.005.

[8] N.K. Ratha, J.H. Connell, R.M. Bolle, Enhancing security and privacy in biometrics-based authentication systems, IBM Systems Journal 40 (3) (2001) 614–634.

[9] N.K. Ratha, S. Chikkerur, J.H. Connell, R.M. Bolle, Generating cancelable fingerprint templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (4) (2007) 561–572.

[10] I. Buhan, J. Doumen, P. Hartel, R. Veldhuis, Fuzzy extractors for continuous distributions, in: Proceedings of the Second ACM Symposium on Information, Computer and Communications Security (ASIACCS), Singapore, ACM, 2007, pp. 353–355.

[11] F. Monrose, M. Reiter, S. Wetzel, Password hardening based on keystroke dynamics, in: Proceedings of the Sixth ACM Conference on Computer and Communications Security, Singapore, 1999, pp. 73–82.

[12] F. Monrose, M. Reiter, Q. Li, S. Wetzel, Cryptographic key generation from voice, in: Proceedings of the IEEE Symposium on Security and Privacy (S&P 2001), CA, USA, 2001, pp. 202–213.

[13] H. Feng, C. Wah, Private key generation from on-line handwritten signatures, Information Management and Computer Security 10 (4) (2002) 159–164.

[14] C. Vielhauer, R. Steinmetz, A. Mayerhofer, Biometric hash based on statistical features of online signatures, in: Proceedings of the 16th International Conference on Pattern Recognition (ICPR 2002), vol. 1, Quebec, Canada, 2002, pp. 123–126.

[15] Y. Chang, W. Zhang, T. Chen, Biometrics-based cryptographic key generation, in: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2004), vol. 3, Taipei, Taiwan, 2004, pp. 2203–2206.

[16] W. Zhang, T. Chen, Generalized optimal thresholding for biometric key generation using face images, in: Proceedings of the IEEE International Conference on Image Processing (ICIP 2005), vol. 3, PA, USA, 2005, pp. 784–787.

[17] A. Juels, M. Wattenberg, A fuzzy commitment scheme, in: Proceedings of the Sixth ACM Conference on Computer and Communications Security, 1999, pp. 28–36.

[18] A. Juels, M. Sudan, A fuzzy vault scheme, Designs, Codes and Cryptography 38 (2) (2006) 237–257.

[19] J.G. Linnartz, P. Tuyls, New shielding functions to enhance privacy and prevent misuse of biometric templates, in: Proceedings of the Audio-and Video-Based Biometrie Person Authentication (AVBPA 2003), Lecture Notes in Computer Science, vol. 2688, Guildford, UK, 2003, pp. 238–250.

[20] P. Tuyls, A. Akkermans, T. Kevenaar, G. Schrijen, A. Bazen, R. Veldhuis, Practical biometric authentication with template protection, in: Proceedings of the Audio-and Video-Based Biometric Person Authentication (AVBPA 2005), NY, USA, 2005, pp. 436–446.

[21] T. Kevenaar, G. Schrijen, M. van der Veen, A. Akkermans, F. Zuo, Face recognition with renewable and privacy preserving binary templates, in: Proceedings of the IEEE Workshop on Automatic Identification Advanced Technologies (AutoID 2005), NY, USA, 2005, pp. 21–26.

[22] F. Hao, R. Anderson, J. Daugman, Combining cryptography with biometrics effectively, IEEE Transactions on Computers 55(9) (2006) 1081–1088, doi:10.1109/TC.2006.138.

[23] E.-C. Chang, S. Roy, Robust extraction of secret bits from minutiae, in: Proceedings of the Second International Conference on Biometrics, ICB, 2007, pp. 750–759.

[24] K. Nandakumar, A. Jain, S. Pankanti, Fingerprint-based fuzzy vault: implementation and performance, IEEE Transactions on Information Forensics and Security 2 (4) (2007) 744–757.

[25] Q. Li, E.-C. Chang, Robust, short and sensitive authentication tags using secure sketch, in: Proceedings of the Eighth Workshop on Multimedia and security, ACM, New York, NY, USA2006, pp. 56–61 doi:http://doi.acm.org/10.1145/1161366.1161377.

[26] Q. Li, Y. Sutcu, N. Memon, Secure sketch for biometric templates, in: Advances in Cryptology Asiacrypt, Springer-Verlag2006, pp. 99–113.

[27] Y. Sutcu, Q. Li, N.D. Memon, Protecting biometric templates with sketch: theory and practice, IEEE Transactions on Information Forensics and Security 2 (3–2) (2007) 503–512.

[28] Y. Sutcu, Q. Li, N.D. Memon, Secure biometric templates from fingerprint-face features, in: CVPR, 2007.

[29] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., John Wiley & Sons, LTD, New York, 2000.

[30] F. Hao, C. Wah, Private key generation from on-line handwritten signatures, Information Management & Computer Security 10 (4) (2002) 159–164.

[31] C. Chen, R. Veldhuis, T. Kevenaar, A. Akkermans, Multi-bits biometric string generation based on the likelihood ratio, in: Proceedings of the IEEE Conference on Biometrics: Theory, Applications and Systems (BTAS07), 2007.

[32] C. Chen, R. Veldhuis, Extracting biometric binary strings with minimal area under the frr curve for the hamming distance classifier, in: The 17th European Signal Processing Conference (EUSIPCO09), 2009.

[33] Y. Shoham, A. Gersho, Efficient bit allocation for an arbitrary set of quantizers, IEEE Transactions on Acoustics, Speech, and Signal Processing 36 (9) (1988) 1445–1453.

[34] J. Daugman, Biometric decision landscapes, Technical Report No. TR482, University of Cambridge Computer Laboratory.

[35] J. Daugman, The importance of being random: statistical principles of iris recognition, Pattern Recognition 36 (2) (2003) 279–291.

[36] A. Papoulis, Probability Random Variables and Stochastic Processes, third ed., Tata McGraw Hill, 1991.

[37] M. Gardner, The Binary Gray Code, W.H. Freeman and Co., NY, USA, 1986.

[38] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, A. Jain, FVC2000: fingerprint verification competition, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3) (2002) 402–412.

[39] P.J. Phillips, P.J. Flynn, W.T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), MD, USA, 2005, pp. 947–954.

[40] R. Veldhuis, A. Bazen, J. Kauffman, P. Hartel, Biometric verification based on grip-pattern recognition, in: Proceedings of the SPIE Security, Steganography, and Watermarking of Multimedia Contents VI (SSWMC 2004), vol. 5306, CA, USA, 2004, pp. 634–641.