LETTER

Error-Correcting Output Codes Guided Quantization for Biometric Hashing

Cagatay KARABAT[†,††a)], ***Student Member and*** **Hakan ERDOGAN**[††], ***Nonmember***

SUMMARY In this paper, we present a new biometric verification system. The proposed system employs a novel biometric hashing scheme that uses our proposed quantization method. The proposed quantization method is based on error-correcting output codes which are used for classification problems in the literature. We improve the performance of the random projection based biometric hashing scheme proposed by Ngo *et al.* in the literature [5]. We evaluate the performance of the novel biometric hashing scheme with two use case scenarios including the case where an attacker steals the secret key of a legitimate user. Simulation results demonstrate the superior performance of the proposed scheme.

key words: *biometric hashing, biometric security, privacy*

1. Introduction

Recent years have seen increased usage of biometric verification systems in many applications. In these systems, an input biometric template is compared to the reference biometric template either stored in a database server or a smart card for verification. The reference biometric template is stored as plaintext in a database or a smart card in most such systems. These systems are deemed insecure and raise about security and privacy concerns [1], [2]. A proposed solution to handle aforementioned threats is to encrypt the reference biometric template stored in a smart card or a database by using cryptographic algorithms [3], [4]. The main problem of such solutions is that the encrypted reference biometric template must be decrypted to compare it with the claimer's input biometric template. This makes the systems weak against possible attacks at the verification stage. Cancellable biometrics that combine the biometric with a secret key to enable randomized biometric hashing is a promising solution to cope with such problems [5], [8].

In this paper, we propose a novel biometric hashing scheme which depends on Error-Correcting Output Codes (ECOC). We improve the performance of the random projection based biometric hashing scheme by introducing a new quantization method that attempts to optimize biometric hash vectors by using the ideas from ECOC classifiers. The proposed scheme shows superior performance in comparison with Ngo *et al.*'s scheme [5] on four databases.

Manuscript received July 15, 2011.
Manuscript revised December 9, 2011.

[†]The author is with TUBITAK BILGEM (Center of Research for Advanced Technologies on Informatics and Information Security), Turkey.
[††]The authors are with the Faculty of Engineering and Natural Sciences, Sabanci University, Turkey.
a) E-mail: cagatay@uekae.tubitak.gov.tr

DOI: 10.1587/transinf.E95.D.1707

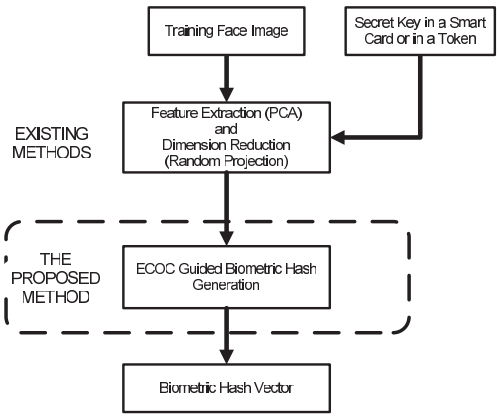

Fig. 1 The basic steps of the proposed biometric hashing scheme.

2. The Proposed Biometric Verification System

In this section, we introduce our new biometric verification system based on the proposed ECOC guided biometric hash generation method as illustrated in Fig. 1.

2.1 Enrollment Stage

Here, we explain the enrollment stage which consists of two main phases: 1) Feature extraction and dimension reduction, 2) ECOC guided biometric hash generation.

2.1.1 Feature Extraction and Dimension Reduction

At this phase, we use face images in the training set. The training set has training face images belonging to registered users, $\mathbf{I}_{i,j} \in \Re^{m \times n}$ where $i = 1, 2, \ldots, K$ and K denotes the number of users, $j = 1, 2, \ldots, L$ and L denotes the number of training images per user. We lexicographically re-order the face images and obtain training face vectors, $x_{i,j} \in \Re^{(mn)\times 1}$. Then, we employ Principle Component Analysis (PCA) to the face images in the training set for feature extraction as follows:

$$\boldsymbol{y}_{i,j} = \mathbf{A}(\boldsymbol{x}_{i,j} - \boldsymbol{\mu}), \tag{1}$$

where $\mathbf{A} \in \Re^{k \times (mn)}$ is the PCA matrix trained by the face images in the training set, $\boldsymbol{\mu}$ is the mean face vector and $\boldsymbol{y}_{i,j} \in \Re^{k \times 1}$ is a vector containing PCA coefficients belonging to the j^{th} training image of the i^{th} user.

We generate a Random Projection (RP) matrix, $\mathbf{R}_i \in \Re^{\ell \times k}$ $\forall i$, for each user to reduce the dimension of the face images in the training set. The RP matrix elements are identically and independently (*i.i.d*) generated from a Gauss distribution with zero mean and unit variance by using a Random Number Generator (RNG) with a seed derived from the user's secret key. We apply Gram-Schmidt (GS) procedure to obtain an orthonormal projection matrix $\mathbf{R}_{GS,i} \in \Re^{\ell \times k}$ from $\mathbf{R}_i$ to have more distinct projections. We project the PCA coefficient vectors onto a lower ℓ-dimensional subspace as follows:

$$\boldsymbol{z}_{i,j} = \mathbf{R}_{GS,i}\boldsymbol{y}_{i,j}, \tag{2}$$

where $\boldsymbol{z}_{i,j} \in \Re^{\ell \times 1}$ is an intermediate biometric hash vector belonging to the j^{th} training image of the i^{th} user.

2.1.2 ECOC Guided Biometric Hash Generation

At this phase, we first calculate a representative raw biometric hash vector, $\boldsymbol{E}_i$, for each user:

$$E_i(m) = \frac{1}{L}\sum_{j=1}^{L} \boldsymbol{z}_{i,j}(m), \tag{3}$$

where $m \in \{1, 2, \ldots, \ell\}$ and ℓ is the length of the raw biometric hash vector. Next, we map the elements of $\boldsymbol{E}_i$ to the interval $[0, 1]$ by employing min-max normalization [9] and obtain a representative intermediate biometric hash vector, $\boldsymbol{V}_i \in \Re^{\ell \times 1}$, for each user as follows:

$$V_i(m) = \frac{E_i(m) - \min(\boldsymbol{E}_i)}{\max(\boldsymbol{E}_i) - \min(\boldsymbol{E}_i)}, \tag{4}$$

where $\boldsymbol{V}_i$ denotes a representative intermediate biometric hash vector of the i^{th} user, min(.) function computes minimum value of its input vector and max(.) function computes maximum value of its input vector.

Conventionally, the $\boldsymbol{V}_i$ vector is binary-quantized by thresholding to obtain the final biometric hash vector for each user. In Ngo *et al.*'s scheme [5], a different quantization threshold (t_g) for each user is obtained by computing the average value of each associated vector, that is $t_g = 1/\ell \sum_{m=1}^{\ell} V_i(m)$. Note that the threshold is the same for each bit position, therefore we can call it a *global* threshold.

In contrast, we employ bit-adaptive quantization to improve the performance of the biometric hashing scheme by generating a more diverse set of biometric hashes for authorized users. We define $\boldsymbol{C}$ as the *codeword matrix* which is formed by stacking biometric hashes of all users in its rows. As shown in Fig. 2, the i^{th} row of $\boldsymbol{C}$ is obtained by quantizing $\boldsymbol{V}_i$ using a set of thresholds (one for each bit) which we aim to optimize.

$$C(i,m) = \begin{cases} 1 & \text{if} \quad \boldsymbol{V}_i(m) \geq \boldsymbol{t}(m) \\ 0 & \text{Otherwise.} \end{cases} \tag{5}$$

In the literature, ECOC is proposed to cope with multi-class classification problems using multiple binary classifiers [10], [11]. Here, our aim is to reduce verification errors

	1^{st} element	2^{nd} element	...	l^{th} element
Representative Intermediate Biometric Hash Vector for the 1^{st} User	0.65	0.21	...	0.34
Representative Intermediate Biometric Hash Vector for the 2^{nd} User	0.33	0.11	...	0.62
.....	0.87	0.59	...	0.75
Representative Intermediate Biometric Hash Vector for the K^{th} User	0.53	0.67	...	0.11

Quantize with respect to $\boldsymbol{t}$

	1^{st} bit	2^{nd} bit	...	l^{th} bit
Biometric Hash Vector for the 1^{st} User	1	0	...	0
Biometric Hash Vector for the 2^{nd} User	0	0	...	1
.....	1	1	...	1
Biometric Hash Vector for the K^{th} User	0	1	...	0

Fig. 2 The illustration of the ECOC guided quantization step in the proposed biometric hashing scheme.

by employing separation criteria used in ECOC classifiers to optimize the biometric hash codeword matrix $\boldsymbol{C}$ by modifying the threshold vector $\boldsymbol{t}$.

The ECOC matrices are optimized on two main criteria [12]: 1) Row separation, 2) Column separation. In the proposed method, we use row and column separation criteria described below to optimize the biometric hash vectors.

Row Separation: The Hamming distance between the biometric hash vectors, which belong to different users, should be maximized to reduce errors. The minimum Hamming distance between any pair of biometric hash vectors is called the row separation:

$$H_r(\boldsymbol{t}) = \min_{i,j,i\neq j} \sum_{m=1}^{\ell} |C(i,m) - C(j,m)|. \tag{6}$$

H_r is dependent on $\boldsymbol{t}$ since the thresholds determine and change the codeword matrix $\boldsymbol{C}$. An ECOC matrix with minimum Hamming distance, H_r, between any pair of biometric hash vectors will correct up to $\left\lfloor \frac{H_r - 1}{2} \right\rfloor$ bit errors [10]. Thus, it is beneficial to maximize this minimum distance to obtain better biometric hash vectors.

Column Separation: Column separation is defined as the minimum Hamming distance between all the columns of the codeword matrix $\boldsymbol{C}$. The aim in ECOC matrix design is to maximize the column separation. In calculating the column separation we should also consider the distance to the complement of a column as well since it gives the same split of the set of biometric hash bits.

$$H_c(\boldsymbol{t}) = \min_{m,n,m\neq n} \left\{ \sum_{i=1}^{K} |C(i,m) - C(i,n)|, \sum_{i=1}^{K} |1 - C(i,m) - C(i,n)| \right\} \tag{7}$$

where $m, n \in \{1, \ldots, \ell\}$ and K denotes the number of users.

Maximizing the column separation will increase the verification accuracy of the system by decreasing correlation between classification errors [11], [12] and makes the system more robust against attacks. We define $H(\boldsymbol{t}) = H_r(\boldsymbol{t}) + H_c(\boldsymbol{t})$ as the optimization criterion to maximize. Hence, we have to solve $\hat{\boldsymbol{t}} = \arg\max_{\boldsymbol{t}} H(\boldsymbol{t})$.

Since the user cannot give exactly the same biometric template for each attempt to enter the system due to sensor imperfections and/or user dependent mistakes errors occur in biometric hashes. To decrease such errors, Hamming distance between the biometric hash vectors belonging to different users should be maximized. Besides, Hamming distance between each bit position of the biometric hash vectors should be maximized to reduce redundancy and to increase security against attacks.

We proceed as follows to optimize this complex objective. Initially, we find an optimum system level quantization threshold $\hat{t_s} \in [0, 1]$ that maximizes $H(t_s) = H_r(t_s) + H_c(t_s)$ by using the Golden section search (GSS) algorithm [13] in the range $[0, 1]$. The optimum system level threshold, $\hat{t_s} = \arg\max_{t_s} (H(t_s))$, is a quantization threshold which can be used for all bit positions of the biometric hash vectors.

Next, using the optimal system level threshold as an initial value, we find an optimum threshold for each bit position (m) of the biometric hash vectors that maximizes $H(\boldsymbol{t}) = H_r(\boldsymbol{t}) + H_c(\boldsymbol{t})$ by using the Golden section search algorithm [13] within the range $[0, 1]$. We perform this by using the coordinate descent method where we update one coordinate at a time while keeping the rest of the threshold vector constant. So, at each iteration, we solve optimization problem $\hat{t}(m) = \arg\max_{t(m)} (H(\tilde{\boldsymbol{t}}))$ for $m = 1, \ldots, \ell$ where $\tilde{\boldsymbol{t}} \in \Re^{1\times\ell}$ is the latest threshold vector which contains the latest values of the thresholds for all bit positions and $\hat{t}(m) \in [0, 1]$ is the optimum threshold value for m^{th} bit position of the biometric hash vector. We go through all the coordinates multiple times until the iterations stop changing the objective value $H(\boldsymbol{t})$. The vector obtained in the end is the optimal bit-adaptive threshold vector $\hat{\boldsymbol{t}}$.

The pseudo-code of the optimization algorithm performed in the enrollment phase is given as Algorithm 1.

2.1.3 Relation with ECOC Classification

In our scheme, we employ the column and row separation criteria used in ECOC matrix design to optimize the codeword matrix obtained from the biometric hash vectors. So, we do not pre-specify the codeword matrix and design classifiers afterwards as regularly done in ECOC classification. Random projection followed by binary quantization that is used in biometric hashing can be seen as using a set of random linear classifiers $\boldsymbol{w}^T\boldsymbol{x} - b \gtrless 0$ where $\boldsymbol{w}$ corresponds to a single row of the random projection matrix and the bias term b corresponds to the bit-specific threshold t. Our method can be seen as using a number of random linear classifiers and modifying their bias value (the threshold) to optimize the row and column separation obtained by them. So, our method is not a direct application of ECOC multi-class classification, rather an innovative idea where random linear classifiers are optimized in their bias terms to obtain a better codeword matrix that will result in better verification performance.

Algorithm 1 Pseudo Code of the Enrollment Phase

K : number of users and L : number of face images per user
ℓ : Length of biometric hash vector
Inputs: Training face images, $\mathbf{I}_{i,j}$, and secret keys of the users
Outputs: The binary codeword matrix, $\boldsymbol{C}$, and threshold vector $\boldsymbol{t}$

Compute PCA matrix $\mathbf{A}$ by using all the training images $\mathbf{I}_{i,j}$
for $i \leftarrow 1$ **to** K **do**
    Generate RP matrix $\mathbf{R}_{GS,i}$ by using the secret key of the i^{th} user
    for $j \leftarrow 1$ **to** L **do**
        Compute PCA coefficient vectors $\boldsymbol{x}_{i,j}$
        Compute $\boldsymbol{y}_{i,j} = \mathbf{A}(\boldsymbol{x}_{i,j} - \boldsymbol{\mu})$
        Compute $\boldsymbol{z}_{i,j} = \mathbf{R}_{GS,i}\boldsymbol{y}_{i,j}$
    end for
end for
for $i \leftarrow 1$ **to** K **do**
    for $m \leftarrow 1$ **to** ℓ **do**
        Compute $\boldsymbol{E}_i(m) = \frac{1}{L}\sum_{j=1}^{L} \boldsymbol{z}_{i,j}(m)$
    end for
end for
for $i \leftarrow 1$ **to** K **do**
    for $m \leftarrow 1$ **to** ℓ **do**
        Compute $\boldsymbol{V}_i(m) = \frac{E_i(m) - \min(\boldsymbol{E}_i)}{\max(\boldsymbol{E}_i) - \min(\boldsymbol{E}_i)}$
    end for
end for
$t_s^0 \leftarrow 0.5$ (set initial value of quantization threshold)
Solve $\hat{t_s} = \arg\max_{t_s} H_r(t_s) + H_c(t_s)$ using GSS algorithm
$t^0(m) = \hat{t_s}, m = 1, \ldots, \ell$ (set initial value of the threshold vector to the system level threshold)
Solve $\hat{\boldsymbol{t}} = \arg\max_{\boldsymbol{t}} H_r(\boldsymbol{t}) + H_c(\boldsymbol{t})$ using coordinate descent and GSS algorithm for each coordinate
Compute codeword matrix $\boldsymbol{C}$ by using the optimal threshold vector $\hat{\boldsymbol{t}}$
Store binary codeword matrix $\boldsymbol{C}$ and the threshold vector $\hat{\boldsymbol{t}}$

2.2 Test Stage

At this stage, a claimer sends his face image $\tilde{\mathbf{I}} \in \Re^{m\times n}$ and his secret key to the system. Then, the system computes the claimer's test biometric hash vector by using the same procedures in the enrollment phase with the optimum threshold for each bit position, $t(m)$. Finally, the system computes the Hamming distance [14] between the test biometric hash vector and the claimed user's reference biometric hash vector stored in the database. If the Hamming distance is below the pre-determined distance threshold, the claimer is accepted; otherwise, the claimer is rejected.

Since the biometric hash vectors can only be computed by the system, a user typically does not know her biometric hash vector. Whenever a new user wants to enroll in the proposed system, the threshold vector, $\boldsymbol{t}$, and the codeword matrix, $\boldsymbol{C}$, which contains reference biometric hash vectors, stored in the database need to be updated. Initially, for new users, the system can use the existing threshold vector to determine their biometric hash vectors. When the number

of new users reach a pre-defined specific number, the system will update itself (e.g. at night time when the system is idle) and generate a new threshold vector and a new codeword matrix which are optimal for the new population.

In the proposed system, only the threshold vector, $\boldsymbol{t}$, is stored additionally in comparison with the Ngo *et. al.*'s system [5]. It is used for computing biometric hash vectors at the test stage. Even if an attacker obtains it, he cannot get any more information since the security of the system depends on the RP matrix and secret key of the users as in Ngo *et. al.* 's system [5]. The attacker can obtain neither the feature vector nor the biometric template of the user by using the threshold vector, $\boldsymbol{t}$, and the codeword matrix, $\boldsymbol{C}$, since there are infinitely many choices when getting back from binary biometric hash to the face image.

3. Simulation Results

In this section, we test and discuss the performance of the proposed scheme on Carnegie Mellon University (CMU) face database [15], Cambridge university AT&T face database [16], Multi Modal Verification for Teleservices and Security applications (M2VTS) face database [17], [18], and the Sheffield (previously UMIST) face databases [19]. Preprocessing methods such as eye marking, alignment and head region masking are not applied to the face images. Table 2 shows the number of face images used in the enrollment and test phases.

Table 3 shows the total number of genuine and imposter pairs. Note that all enrollment images for a person are used to generate a single reference biometric hash vector for a person. Each test image indicated in Table 2 is used once in a genuine test and as an impostor for all other users.

Table 2 Databases and experimental set-up.

Database	Number of Face Images	Enrollment Phase	Test Phase
CMU	975 images from 13 people	The first 15 images of each user	The following 30 images of each user
AT&T	400 images from 40 people	The first 5 images of each user	The rest 5 images of each user
M2VTS	1480 images from 37 people	The first 20 images of each user	The rest 20 images of each user
Sheffield	564 images from 20 people	The first 8 images of each user	The following 8 images of each user

Table 1 EER performance comparison between the proposed biometric hashing scheme and Ngo et al.'s scheme [5].

Length of Face Hash Vector	EER (%) of Ngo *et al.*'s Scheme [5] (PCA+RP)	EER (%) of The Proposed Scheme	Scenario	Database
64 bit	% 2.05	**% 0.15**	Key Unknown	CMU
128 bit	% 0.98	**% 0.00**	Key Unknown	CMU
256 bit	% 0.60	**% 0.00**	Key Unknown	CMU
512 bit	% 0.22	**% 0.00**	Key Unknown	CMU
64 bit	% 4.10	**% 2.50**	Key Stolen	CMU
128 bit	% 2.46	**% 0.74**	Key Stolen	CMU
256 bit	% 1.72	**% 0.08**	Key Stolen	CMU
512 bit	% 1.18	**% 0.07**	Key Stolen	CMU
64 bit	% 12.19	**% 6.44**	Key Unknown	AT&T
128 bit	% 7.36	**% 4.25**	Key Unknown	AT&T
256 bit	% 5.81	**% 1.13**	Key Unknown	AT&T
512 bit	% 3.79	**% 0.04**	Key Unknown	AT&T
64 bit	% 16.93	**% 13.07**	Key Stolen	AT&T
128 bit	% 13.97	**% 9.71**	Key Stolen	AT&T
256 bit	% 12.76	**% 7.64**	Key Stolen	AT&T
512 bit	% 12.34	**% 8.01**	Key Stolen	AT&T
64 bit	% 18.10	**% 10.01**	Key Unknown	M2VTS
128 bit	% 14.56	**% 7.55**	Key Unknown	M2VTS
256 bit	% 11.15	**% 6.38**	Key Unknown	M2VTS
512 bit	% 9.23	**% 5.81**	Key Unknown	M2VTS
64 bit	% 21.36	**% 14.17**	Key Stolen	M2VTS
128 bit	% 18.08	**% 13.01**	Key Stolen	M2VTS
256 bit	% 16.72	**% 10.50**	Key Stolen	M2VTS
512 bit	% 16.09	**% 10.00**	Key Stolen	M2VTS
64 bit	% 17.09	**% 12.30**	Key Unknown	Sheffield (UMIST)
128 bit	% 16.38	**% 7.87**	Key Unknown	Sheffield (UMIST)
256 bit	% 15.05	**% 6.25**	Key Unknown	Sheffield (UMIST)
512 bit	% 14.97	**% 2.93**	Key Unknown	Sheffield (UMIST)
64 bit	% 21.40	**% 17.74**	Key Stolen	Sheffield (UMIST)
128 bit	% 21.92	**% 16.91**	Key Stolen	Sheffield (UMIST)
256 bit	% 22.53	**% 15.25**	Key Stolen	Sheffield (UMIST)
512 bit	% 23.47	**% 14.21**	Key Stolen	Sheffield (UMIST)

Table 3 Genuine and imposter pairs in each database.

Database	Number of Genuine Pairs	Number of Imposter Pairs
CMU	$1 \times 30 \times 13 = 390$	$1 \times 30 \times ((12 \times 13) \div 2) = 2340$
AT&T	$1 \times 5 \times 40 = 200$	$1 \times 5 \times ((39 \times 40) \div 2) = 3900$
M2VTS	$1 \times 20 \times 37 = 740$	$1 \times 20 \times ((36 \times 37) \div 2) = 13320$
Sheffield	$1 \times 8 \times 20 = 160$	$1 \times 8 \times ((19 \times 20) \div 2) = 1520$

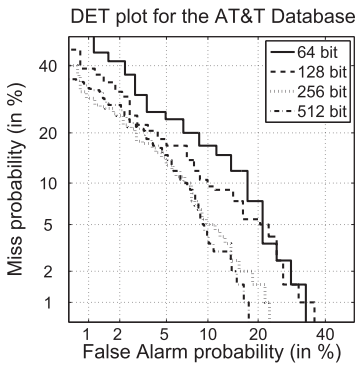

Fig. 3 DET plots of the proposed method for key stolen scenario - AT&T database.

3.1 Equal Error Rate (EER) Performances

In this part, we test the performance of the proposed scheme. Kong *et al*. state that if unauthorized people steal the secret key and the RNG, the performances of biometric hashing schemes get worse [8]. Therefore, we simulate two scenarios, in our experiments as shown in Table 1.

1. Key Unknown Scenario: An imposter user wants to impersonate a genuine user. However, she neither has biometric template nor secret key of a genuine user. She sends her own biometric template and a secret key to the system to be authenticated as a genuine user. In tests, we have used each impostor's own key for their impostor attempts as well.

2. Key Stolen Scenario: An imposter user obtains secret key of a genuine user. She sends her own biometric template and the secret key of the genuine user to the system to be authencated as a genuine user.

As shown in Table 1, the proposed scheme has lower EER in comparison with [5]. Moreover, we show the detection error trade-off (DET) curves [20] of the proposed method for key stolen scenario in Figs. 3-6. Besides, we show genuine-imposter distance histograms and false accept rate (FAR) - false reject rate (FRR) plots in key stolen scenario for the proposed method in Figs. 7-10. We attribute the performance improvements to the better scattering of biometric hash vectors due to the maximization of row and column separation in the codeword matrix in our method.

The proposed method more dramatically reduces the errors as the length of the biometric hash vector increases as shown in Table 1. The proposed scheme approximately reduces the EER by half in most of the cases. Furthermore, even in some cases, the proposed scheme perfectly separates the genuine and imposter users with no errors.

The proposed method maximizes the Hamming dis-

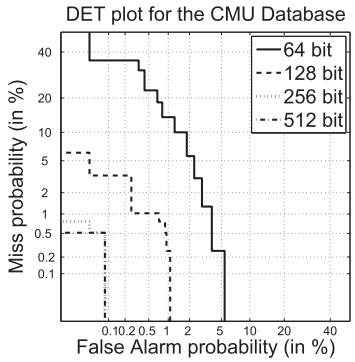

Fig. 4 DET plots of the proposed method for key stolen scenario - CMU database.

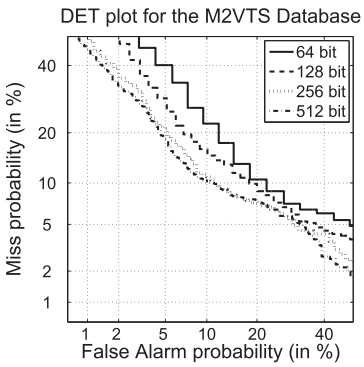

Fig. 5 DET plots of the proposed method for key stolen scenario - M2VTS database.

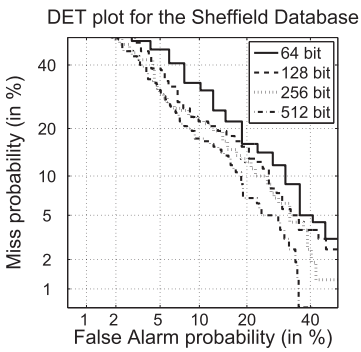

Fig. 6 DET plots of the proposed method for key stolen scenario - Sheffield database.

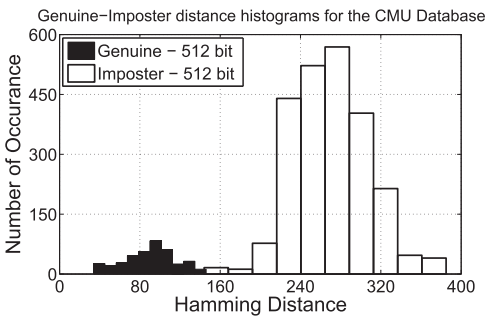

Fig. 7 Genuine-Imposter distance histograms of the proposed method for key stolen scenario in the CMU database - 512 bit.

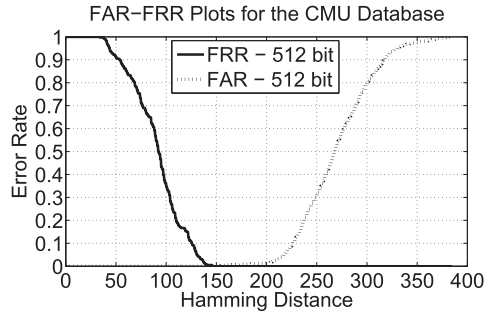

Fig. 8 FAR-FRR plots of the proposed method for key stolen scenario in the CMU database - 512 bit.

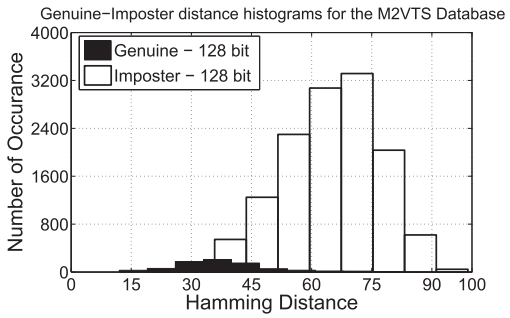

Fig. 9 Genuine-Imposter distance histograms of the proposed method for key stolen scenario in the M2VTS database - 128 bit.

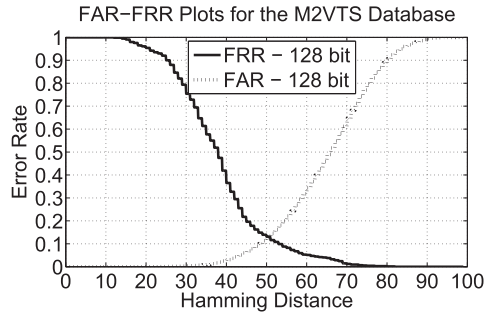

Fig. 10 FAR-FRR plots of the proposed method for key stolen scenario in the M2VTS database - 128 bit.

tance between the biometric hashes belonging to the different users at the enrollment stage. Thus, we achieve lower EERs in comparison with [5]. Even in the key stolen scenario, we see improvements in performance which is possibly due to better placement of reference biometric hash vectors in the space of all possible hashes.

4. Conclusion

In this paper, we propose a novel biometric hashing scheme based on the proposed quantization method that maximizes the row and the column separation of the code matrix as in ECOC classifiers. We maximize the distance between the genuine-impostor pairs as well as decrease the correlation between the bit positions in biometric hash vectors belonging to different users. The proposed method has superior performance in comparison with [5]. The proposed quantization method can be applied to the other biometric hashing schemes that employs various feature extraction techniques.

References

[1] G. Tomko, "Biometrics as a privacy enhancing technology: Friend or foe of privacy?," Proc. 9th Privacy Comm./Data Proct. Auth. Workshop, Spain, Sept. 1998.

[2] M. Crompton, "Biometrics and privacy: The end of the world as we know it or white knight of privacy?," Proc. 1st Biometrics Inst. Conf., Sydney, March 2002.

[3] G.I. Davida, Y. Frankel, and B.J. Matt, "On enabling secure applications through off-line biometric identification," Proc. IEEE Symp. Security and Privacy, pp.148–157, Oakland, Ca., 1998.

[4] G.I. Davida, Y. Frankel, B.J. Matt, and R. Peralta, "On the relation of error correction and cryptography to an off line biometrics based identification scheme," Proc. Workshop on Coding and Cryptography, pp.129–138, France, 1999.

[5] D.C.L. Ngo, A.B.J. Teoh, and A. Goh, "Biometric hash: High confidence face recognition," IEEE Trans. Circuits. Syst. Video Technol., vol.16, no.6, pp.771–775, June 2006.

[6] A.B.J. Teoh, A. Goh, and D. Ngo Chek Ling, "Random multispace quantisation as an analytic mechanism for bioHashing of biometric and random identity inputs," IEEE Trans. Pattern. Anal. Mach. Intell., vol.28, no.12, pp.1892–1901, Dec. 2006.

[7] A. Lumini and L. Nanni, "An improved bioHashing for human authentication," Pattern Recognit., vol.40, no.3, pp.1057–1065, March 2007.

[8] A. Kong, K.H. Cheung, D. Zhang, M. Kamel, and J. You, "An analysis of BioHashing and its variants," Pattern Recognit., vol.39, no.7, pp.1359–1368, 2007.

[9] J. Han and M. Kamber ed., Data mining: Concepts and techniques, Morgan Kaufmann Publishers, 2006.

[10] D.W. Aha and R.L. Blankert, "Cloud classification using error-correcting output codes," Art. Intel. Applications: Natural Resources, Agr. and Env. Sci., vol.11, no.1, pp.13–28, 1997.

[11] T.G. Dietterich and G. Bakiri, "Solving multi class learning problems via error-correcting outputcodes," J. Art. Intel. Research, vol.2, pp.263–286, 1995.

[12] L.I. Kuncheva ed., Combining Pattern Classifiers: Methods and Algorithms, John Wiley & Sons, 2004.

[13] J. Kiefer, "Sequential minimax search for a maximum," Amer. Math. Soc., vol.4, pp.502–506, 1953.

[14] R.W. Hamming, "Error detecting and error correcting codes," Bell Syst. Tech. J., vol.29, no.2, pp.147–160, 1950.

[15] Carnegie Mellon University Face Database, http://amp.ece.cmu.edu.

[16] Cambridge University AT&T face database, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.

[17] S. Pigeon and L. Vandendorpe, "The M2VTS multimodal face database (Release 1.00)," Proc. 1st International Conference on Audio and Video Based Biometric Person Authentication, pp.403–409, London, UK, 1997.

[18] B. Topcu, Feature extraction and fusion techniques for patchbased face recognition, MS thesis, Sabanci University, Turkey, 2009.

[19] D.B. Graham and N.M. Allinson, "Face recognition: From theory to applications," NATO ASI Series F, Computer and Systems Sciences, vol.163, pp.446–456, 1998.

[20] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M.A. Przybocki, "The det curve in assessment of detection task performance," Eurospeech, pp.1895–1898, Rhodes, Greece, Sept. 1997.