

Biometric Binary String Generation with Detection Rate Optimized Bit Allocation

C. Chen, R.N.J. Veldhuis
Signals and Systems Group, Electrical Engineering
University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
{c.chen, r.n.j.veldhuis}@utwente.nl

T.A.M. Kevenaar, A.H.M. Akkermans
Philips Research
Prof. Holstlaan 4, 5656 AA, Eindhoven, The Netherlands
{tom.kevenaar, ton.h.akkermans}@philips.com

Abstract

Extracting binary strings from real-valued templates has been a fundamental issue in many biometric template protection systems. In this paper, we present an optimal bit allocation method (OBA). By means of it, a binary string at a pre-defined length with maximized overall detection rate is generated. Experiments with the binary strings and a Hamming distance classifier on FRGC and FERET databases show promising performance in terms of FAR and FRR.

1. Introduction

Biometric recognition is popular in many applications such as access control, surveillance and law enforcement. Currently most of these applications use real-valued biometric representations, such as fingerprint minutiae locations and face image pixel values. These templates may demand huge storage capability and computational complexity during matching. Besides, storing the raw biometric data introduces privacy concerns [1]. To overcome these drawbacks, template protection systems, such as fuzzy commitment [2], fuzzy extractor [3] and helper data systems [4] have been proposed. All these systems develop their biometric protection techniques based on the assumption that a real-valued biometric template can be transformed into a fixed length binary string. Hence, the performance, measured as false acceptance rate (FAR) and false rejection rate (FRR), are evaluated by the similarity of the binary strings. Therefore, it is crucial to generate binary strings that can meet the low FAR and FRR requirements in such systems.

A common way to obtain a binary string is via quantiza-

tion. Usually a vector of independent feature components is first extracted from the original real-valued template. Afterwards, a quantizer is applied to every feature component. The quantization interval in which the genuine feature component falls is coded and concatenated to construct the binary string. The final decision is made on the similarity between the binary strings, by means of Hamming distance.

So far, some work regarding the quantizer design of a single feature component has been published [5] [6] [7] [8]. In order to cope with external noise and user variations, the quantization aims to divide the feature domain into quantization intervals, with a binary code assigned to each interval. The interval where a feature of the genuine user is expected to fall is the genuine interval, and the assigned code of this interval represents the code of the genuine user. The construction of the quantization intervals always relies on two probability density functions (PDFs): the background PDF p_b and the genuine user PDF p_g , representing a single feature's density of the whole population and of a genuine user, respectively. Tuyls *et al.* [5] first introduced a 1-bit fixed quantizer (FQ), and Chen *et al.* [8] extended it into multi-bits fixed quantizer. Generally, an n -bits fixed quantizer constructs fixed boundaries (independent of the genuine user PDF), with the same background probability mass 2^{-n} in each interval. Having the same probability mass in all intervals yields independent output bits, which is beneficial to privacy protection. Zhang *et al.* [7] introduced a user-specific multi-bits quantizer (ZQ). In this method, a genuine interval is first established as $[\mu - k\sigma, \mu + k\sigma]$, where μ and σ are the mean and the standard deviation of the genuine user PDF. The remaining intervals are constructed with the same length $2k\sigma$. Chen *et al.* [8] later introduced a user-specific

likelihood ratio based multi-bits quantizer (LQ). Same as fixed quantizer, an n -bits likelihood ratio quantizer assigns equal 2^{-n} background probability mass to each interval, but the genuine interval is determined by the likelihood ratio of p_g and p_b .

In general, the false acceptance rate α_i of feature component i with b_i -bits quantization is defined as:

$$\alpha_i(b_i) = \int_{Q_{\text{genuine}}(b_i)} p_b(v) dv, \quad (1)$$

where Q_{genuine} represents the genuine user interval. As mentioned earlier, both the fixed quantizer and the likelihood ratio based quantizer generate intervals that equally divide the background probability mass [8]. That means the *FAR* for one feature of both quantizers becomes:

$$\alpha_{\text{FQ},i}(b_i) = \alpha_{\text{LQ},i}(b_i) = 2^{-b_i}. \quad (2)$$

Similarly, the detection rate δ_i of feature component i with b_i -bits quantization is defined as:

$$\delta_i(b_i) = \int_{Q_{\text{genuine}}(b_i)} p_g(v) dv, \quad (3)$$

where p_g is the genuine user PDF. Therefore the quantizer is designed to meet the Neyman-Pearson criterion, that is at a given *FAR* value in (2), the detection rate in (3) is maximized. Among all the above quantizers, the likelihood ratio based quantizer satisfies this criterion best.

Assuming D independent feature components, the overall false acceptance rate α and false rejection rate β are:

$$\alpha = \prod_{i=1}^D \alpha_i(b_i), \quad (4)$$

$$\beta = 1 - \prod_{i=1}^D \delta_i(b_i). \quad (5)$$

Given (2) and the binary string length L , the overall *FAR* of the fixed quantizer and the likelihood ratio based quantizer becomes:

$$\alpha_{\text{FQ}} = \alpha_{\text{LQ}} = 2^{-L}. \quad (6)$$

In the existing implementations, a fixed number of b -bits (*e.g.* 2) is allocated to each feature component. As a result, given the overall *FAR* performance in (4), the *FRR* performance in (5), with respect to the binary string length, is not optimized, since one would prefer to use more bits for a reliable feature and fewer bits for an unreliable feature.

In this paper, we present an optimal bit allocation method. This method determines how many bits should be extracted from every feature component, such that the overall detection rate is maximized at a given binary string

length. To solve this optimization problem we propose a recursive dynamic programming approach. An implementation of this approach in combination with the fixed quantizer for the FRGC and FERET face databases show promising equal error rate (*EER*) performance. Furthermore, the performances are not much degraded as compared to the real-value based likelihood ratio classifier.

This paper proceeds as follows. In Section 2, the overall detection rate optimized bit allocation method, with a dynamic programming approach, are introduced. In Section 3, experiments with the optimal bit allocation method in combination with the fixed quantizer on three data sets are presented and the results are shown. In Section 4, some discussions are presented, and in Section 5 conclusions are drawn.

2. Detection Rate Optimized Bit Allocation

Let D denote the number of feature components to be quantized; L , the desired binary string length; $b_i \in \{0, \dots, b_{\text{max}}\}$, $i = 1, \dots, D$, the possible number of bits assigned to component i ; and $\delta_i(b_i)$, $i = 1, \dots, D$, the corresponding detection rate of component i , respectively. Assuming that all the D feature components are independent, the overall detection rate (δ) can be written as:

$$\delta = \prod_{i=1}^D \delta_i(b_i). \quad (7)$$

Our goal is to find a set of allocated bits $\{b_i^*\}$ that maximizes the above overall detection rate δ :

$$\{b_i^*\} = \arg \max_{b_i} \prod_{i=1}^D \delta_i(b_i), \quad (8)$$

under the constraint that

$$\sum_{i=1}^D b_i^* = L. \quad (9)$$

To solve this problem, we propose the following dynamic programming approach by adding one feature at a time. It can be seen that the procedure to find the optimal bit allocation is recursive. That is, given the optimal detection rates $\delta^{(j-1)}(l)$ for $j-1$ features at string length l , $l = 0, \dots, (j-1) \times b_{\text{max}}$:

$$\delta^{(j-1)}(l) = \max_{b_i | \sum b_i = l, b_i \in \{0, \dots, b_{\text{max}}\}} \prod_{i=1}^{j-1} \delta_i(b_i), \quad (10)$$

the optimal detection rates $\delta^{(j)}(l)$ for j features is computed as:

$$\delta^{(j)}(l) = \max_{\substack{b' + b'' = l, \\ b' \in \{0, \dots, (j-1) \times b_{\text{max}}\}, \\ b'' \in \{0, \dots, b_{\text{max}}\}}} \delta^{(j-1)}(b') \delta_j(b''), \quad (11)$$

for $l = 0, \dots, j \times b_{\max}$. Note that $\delta^{(j)}(l)$ will be computed for all string lengths $l \in \{0, \dots, j \times b_{\max}\}$. Eq. (11) says that the optimal detection rate for j features at string length l is derived from maximizing the product of an optimized detection rate for $j - 1$ features at string length b' and the detection rate of the j^{th} feature quantized to b'' bits, while $b' + b'' = l$. In each iteration step, for each value of l in $\delta^{(j)}(l)$, the specific optimal bit assignments of features must be maintained. Let $\{b_i(l)\}, i = 1, \dots, j$ denote the optimal bit assignments for j features at binary string length l such that the i^{th} entry corresponds to the i^{th} feature. Note that the sum of all entries in $\{b_i(l)\}$ equals l ($\sum_{i=1}^j b_i(l) = l$). If \hat{b}' and \hat{b}'' denote the values of b' and b'' that correspond to the maximum value $\delta^{(j)}(l)$ in (11). Then the optimal assignments are updated by:

$$b_i(l) = b_i(\hat{b}'), i = 1, \dots, j - 1, \quad (12)$$

$$b_j(l) = \hat{b}'' . \quad (13)$$

The iteration procedure is initialized with $j = 0, b_0(0) = 0$, and $\delta^{(0)}(0) = 1$ and is terminated when $j = D$. The final solution is $\{b_i^*\} = \{b_i(L)\}, i = 1, \dots, D$. This iteration procedure can be formalized into a dynamic programming approach, as described in Algorithm 1.

Algorithm 1 Dynamic programming approach to maximize the overall detection rate.

Input:

$$D, L, \delta_i(b_i), b_i \in \{0, \dots, b_{\max}\}, i = 1, \dots, D,$$

Initialize:

$$j = 0,$$

$$b_0(0) = 0,$$

$$\delta^{(0)}(0) = 1,$$

while $j \neq D$ **do**

$$j = j + 1,$$

$$\hat{b}', \hat{b}'' = \arg \max \delta^{(j-1)}(b') \delta_j(b''),$$

$$b' + b'' = l,$$

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

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

$$l = 0, \dots, j \times b_{\max},$$

$$b_i(l) = b_i(\hat{b}'), i = 1, \dots, j - 1,$$

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

end while

Output:

$$\{b_i^*\} = \{b_i(L)\}, i = 1, \dots, D .$$

Essentially, the dynamic programming approach optimizes (7), given L and $\delta_i(b_i)$. This means this approach is independent of the specific type of the quantizer, which determines the behavior of $\delta_i(b_i)$. The optimal solution $\{b_i^*\}$ is user-specific and feasible as long as $0 \leq L \leq (D \times b_{\max})$. 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 that of a brute force search.

In Fig 1 we give an example of how the optimization procedure is conducted on three feature components. Fig. 1(a) plots the detection rate at possible quantization bits $b_i \in \{0, \dots, 3\}$ for each of the three feature components (e.g. $b_{\max} = 3$). Fig. 1(b), 1(c) and 1(d) plot the computed overall detection rate at iteration step $j = 1, 2, 3$, respectively. In each iteration step, a maximum detection rate (\bullet) is found at every possible string length $l, l \in \{0, \dots, 3 \times j\}$, labeled with the corresponding bits assignments $\{b_1\}$ ($j = 1$), $\{b_1, b_2\}$ ($j = 2$), $\{b_1, b_2, b_3\}$ ($j = 3$). Only these maximum detection rates and their bits assignments are needed for the optimization in the next iteration step.

Given $\{b_i^*\}$, the theoretical overall false acceptance rate α^* and false rejection rate β^* performance of the optimal bits allocated L -bits binary string are computed as:

$$\alpha^* = \prod_{i=1}^D \alpha_i(b_i^*), \quad (14)$$

$$\beta^* = 1 - \prod_{i=1}^D \delta_i(b_i^*). \quad (15)$$

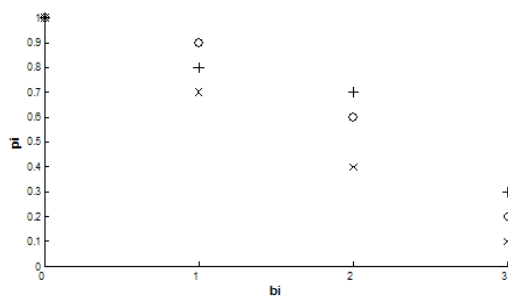
If our optimal bit allocation method is applied on the fixed quantizer or the likelihood ratio based quantizer, we have (2). Thus the overall FAR becomes:

$$\alpha^* = 2^{-L}. \quad (16)$$

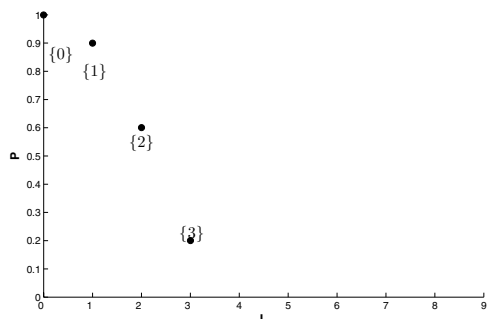
3. Experiments

We tested our optimal bit allocation method on three data sets, derived from FRGC (version 1) [9] and FERET [10] face databases.

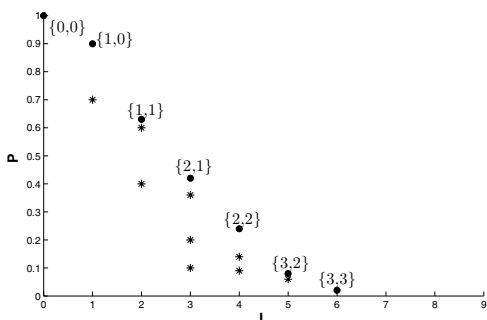
- **FRGC_T**: This is the total FRGC (version 1) data set, containing variable number of images from 275 users. The images were taken under both controlled and uncontrolled conditions and were aligned using manually labeled landmarks. A normalized region of interest (ROI) was extracted from every 128 by 128 image, resulting in 8762 pixel values as the raw data (Fig. 2).
- **FRGC_S**: This is a subset of FRGC_T, containing 198 users with at least 2 images per user. The images were taken under uncontrolled conditions (Fig. 2).
- **FERET_S**: This is a subset of the FERET data set, containing 237 users with at least 4 images per user. The images were normalized and 51 fiducial points were extracted to model the shape of six key objects: left and right eye, left and right eyebrow, mouth and nose. For every fiducial point, texture information was



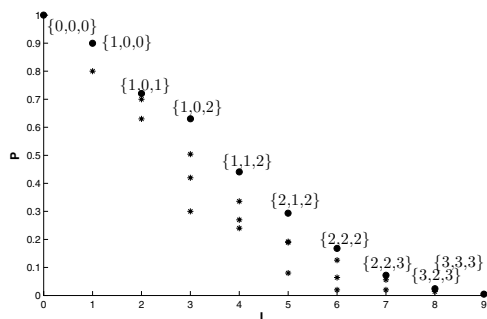
(a)



(b)



(c)



(d)

Figure 1. An example of the optimization procedure for three feature components. (a) the detection rate at $b_1, b_2, b_3 \in \{0, \dots, 3\}$ for the three feature components (feature 1: \circ ; feature 2: \times ; feature 3: $+$); (b-d) All the computed (*) and the maximum (\bullet) overall detection rate at step 1, 2, 3, with the bits assignments.

derived by using the Gabor kernels with 5 frequencies and 8 orientations. This resulted in a total 2040 length raw data [6] (Fig. 2).



Figure 2. Example of data sets used in the experiment. left: FRGC image in controlled conditions; middle: FRGC image in uncontrolled conditions; right: FERET image.

The experiments consist of three steps: training, enrollment and verification. During the initial off-line training step, a principle component analysis and linear discriminant analysis based method (PCA/LDA) [11] was applied on the training data to extract independent feature components with reduced dimensionality. The same trained transformation was applied to both the enrollment and verification data. In the enrollment phase, every individual feature component i was quantized by a set of b_i -bits fixed quantizers ($b_i \in \{0, \dots, 3\}$), as illustrated in Fig. 3. As an implementation, the background PDF p_b can be modeled as a Normal density $p_b(v) = N(v, 0, 1)$ (as shown in Fig. 3), and the genuine user PDF can be modeled as a Gaussian density $p_g(v) = N(v, \mu, \sigma)$, where μ and σ represent the mean and standard deviation, respectively. Applying these two models in (3), the detection rate $\delta_i(b_i)$ of every b_i -bits fixed quantization was computed. Given these detection rates, our optimal bit allocation method was implemented, resulting in an optimal set of allocated bits $\{b_i^*\}$, where b_i^* indicates the optimal quantization bits of feature i . With $\{b_i^*\}$, every single feature component was quantized and assigned with a Gray code [12]. The concatenation of the codes from D feature components constructed the L -bits reference binary string C . Both C and $\{b_i^*\}$ were stored. In the verification phase, every individual feature component i in the verification data was quantized and coded with a b_i^* -bits fixed quantizer, where b_i^* belongs to the the claimed identity, and this resulted in a binary string C' . The final decision was made by comparing C' with the reference string C , by using a Hamming distance classifier. The verification performance therefore relies on the Hamming distance threshold. Due to the lack of samples, we used the same data for training and enrollment in our experiment. Assuming N data samples of a user, we randomly select the samples following the division of the training (enrollment)

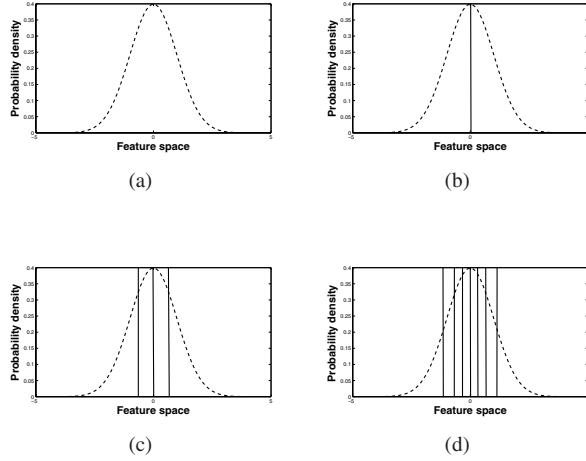


Figure 3. An illustration of the fixed quantizer: background PDF (dashed); quantization intervals (solid). (a) $b_i = 0$; (b) $b_i = 1$; (c) $b_i = 2$; (d) $b_i = 3$.

and verification data, as depicted in Table 1. To evaluate the error with a cross-validation procedure, we repeated our experiment with a number of trials, as listed in Table 1.

Table 1. Training, Enrollment and verification data division per user and the number of trials for the three data sets in the experiments.

	Training	Enrollment	Verification	Trials
FRGC _T		N/2	N/2	5
FRGC _S		N/2	N/2	5
FERET _S		3	N-3	4

In Experiment I, we fixed the number of feature components to $D = 50$, and investigated the performances of the binary strings generated through the fixed quantizer based optimal bit allocation method (FQ-OBA), by using a Hamming distance classifier, at binary string lengths $L = 10, 30, 64, 80, 100$. Table 2 shows the *EER* performances (defined as the performance at which the *FAR* and the *FRR* are the same) of the Hamming distance classifier for the three data sets, compared to a real-value based likelihood ratio classifier (LC) with the same feature components. Results show that as L increases, the *EERs* of all three data sets first decrease, and then increase again. The best performance of the three data sets occurs at $L = 30, 30, 64$, respectively. In Fig. 4, 5 and 6, we plot several ROC curves for FRGC_T, FRGC_S and FERET_S. In the case of FRGC_T and FERET_S, optimal bit allocation method gives somewhat worse performance compared to the real-value based likelihood ratio classifier. But the optimal bit allocation method shows better performance than the real-value based likelihood ratio classifier for FRGC_S.

Table 2. *EER* performances of the real-value based likelihood ratio classifier (LC) and the Hamming distance classifier on FQ-OBA, at $D = 50$, for FRGC_T, FRGC_S and FERET_S.

	LC (%)	FQ-OBA (%)				
		L=10	30	64	80	100
FRGC _T	2.7	4.2	3.4	4.1	4.3	4.7
FRGC _S	7.0	5.7	3.2	4.4	5.1	6.5
FERET _S	1.5	4.7	3.3	2.9	3.5	4.2

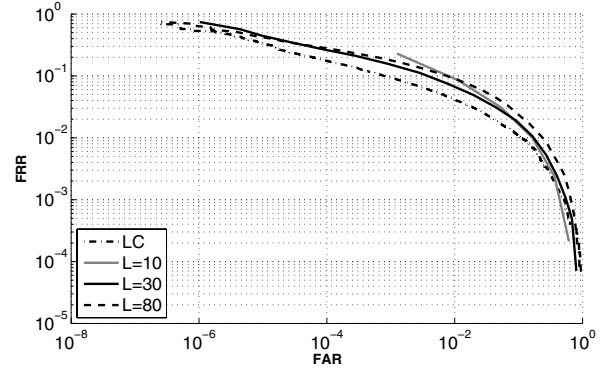


Figure 4. ROC performances of LC and FQ-OBA ($L = 10, 30, 80$) at $D = 50$, for FRGC_T.

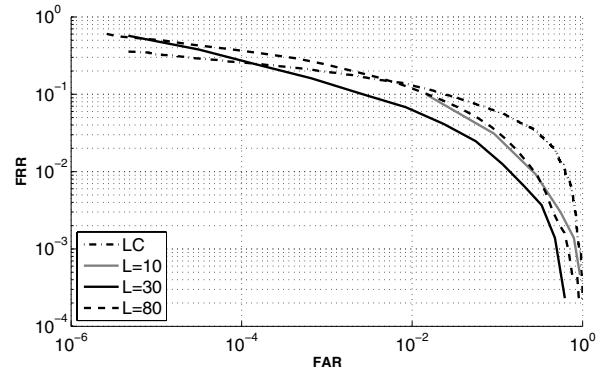


Figure 5. ROC performances of LC and FQ-OBA ($L = 10, 30, 80$) at $D = 50$, for FRGC_S.

In Experiment II, we compared the performance of the fixed quantizer based optimal bit allocation FQ-OBA, to the performance of the fixed quantizer based fixed b -bits allocation FQ- b , given the same $D = 64$ and $L = 64, 128$. Results in Table 3 show that FQ-OBA outperforms FQ- b for FRGC_T and FERET_S data set, but does not outperform FQ- b for FRGC_S data set.

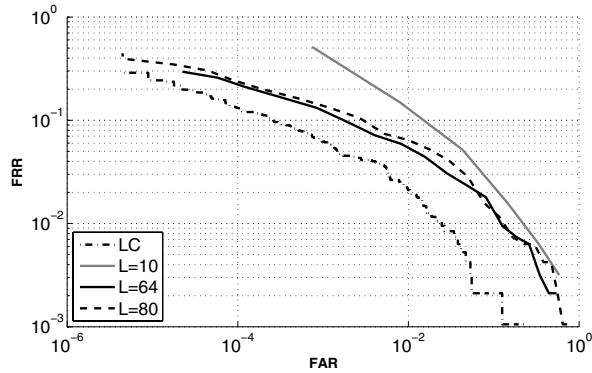


Figure 6. ROC performances of LC and FQ-OBA ($L = 10, 64, 80$) at $D = 50$, for FERETS.

Table 3. EER performances of FQ- b and FQ-OBA, at $D = 64$, $L = 64, 128$, for FRGC_T, FRGC_S and FERETS.

	D=64, L=64		D=64, L=128	
	FQ-1	FQ-OBA	FQ-2	FQ-OBA
FRGC _T	4.5%	3.7%	4.8%	4.5%
FRGC _S	3.8%	4.6%	5.0%	8.0%
FERETS	3.3%	2.6%	3.7%	3.3%

4. Discussion

The FAR and FRR performances of the optimal bits allocated L -bits binary string are in (16) and (15). Given D , when L is small, FAR is relatively high in (16), contrarily, when L is large, FRR becomes high in (15). This explains the result in Experiment I that usually the binary string with an intermediate length gives the best performance.

Table 2 shows that in general binary strings generated through the optimal bit allocation method give similar performance compared to the real-valued templates. Particularly on unreliable data sets, such as FRGC_S, binary representation derived from quantization shows more robustness to noise, compared to the real-value representation. This suggests the employment of binary representation in practical applications where the biometric data are noisy.

According to (6) and (16), we know that the optimal bit allocation method and the fixed b -bits allocation method have equal FAR performance. At the same time, the optimal bit allocation method optimizes the detection rate of every feature component ($\delta_i(b_i^*) \geq \delta_i(b)$). Therefore, it gives lower FRR in (15) compared to fixed b -bits allocation method in (5). This leads to the better performance of optimal bit allocation method for FRGC_T and FERETS data sets, as shown in Table 3. Unfortunately, for unreliable data sets, such as FRGC_S, the parameters of the genuine user PDF estimated from enrollment data become unreliable. Therefore, using extra genuine PDF may bring more error,

compared to the fixed b -bits allocation method where the unreliable genuine user PDF is not used ($\delta_i(b_i^*) < \delta_i(b)$). One way to solve this problem is to increase the number of components provided for the optimal bit allocation. Our experiments with FQ-OBA on FRGC_S at $D = 80, 100$ and $L = 64$ show that the EER reduce to 3.7%, which is lower than the fixed b -bits allocation method ($EER = 3.8%$).

Our bit allocation method in fact generates an optimal binary string as the input features of the Hamming distance classifier, thus the whole system performance depends on the performance of the Hamming distance classifier. Therefore, optimizing the performance of the Hamming distance classifier is our direction of future work.

5. Conclusion

The problem of quantizing real-valued biometric templates into high quality binary strings has been raised recently.

In this paper we presented a method to generate binary strings from biometric feature vectors in the form of a user-specific optimal bit allocation method (OBA). The method assigns bits to individual features so as to optimize the overall detection rate and is independent of the quantizer design.

Experiments using OBA on FRGC and FERET face databases show promising results. The use of OBA will bring substantial benefits to many biometric applications with limited storage, severe matching constraints, and privacy protection.

6. ACKNOWLEDGMENTS

This research is supported by the research program Sentinels (www.sentinels.nl). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

References

- [1] U. Uludag, S. Pankanti, S. Prabhakar, and A.K. Jain. Biometric cryptosystems: Issues and challenges. *Proceedings of the IEEE*, 92(6):948–960, 2004. 1
- [2] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In G. Tsudik, editor, *Sixth ACM Conference on Computer and Communications Security*, pages 28–36. ACM Press, 1999. 1
- [3] Y. Dodis, M. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology — Eurocrypt 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer-Verlag, 2004. 1
- [4] J.M. G. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In

- J. Kittler and M. Nixon, editors, *Conference on Audio and Video Based Person Authentication*, volume 2688 of *Lecture Notes in Computer Science*, pages 238–250. Springer-Verlag, 2003. 1
- [5] P. Tuyls, A.H.M. Akkermans, T.A.M. Kevenaar, G.J. Schrijen, A.M. Bazen, and R.N.J. Veldhuis. Practical biometric authentication with template protection. In Takeo Kanade, Anil K. Jain, and Nalini K. Ratha, editors, *AVBPA*, volume 3546 of *Lecture Notes in Computer Science*, pages 436–446. Springer-Verlag, 2005. 1
- [6] T.A.M. Kevenaar, G.J. Schrijen, M. van der Veen, A.H.M. Akkermans, and F. Zuo. Face recognition with renewable and privacy preserving binary templates. In *IEEE Workshop on Automatic Identification Advanced Technologies (AutoID 2005)*, pages 21–26. IEEE Computer Society, 2005. 1, 4
- [7] Y. Chang, W. Zhang, and T. Chen. Biometrics-based cryptographic key generation. In *ICME*, pages 2203–2206, 2004. 1
- [8] C. Chen, R.N.J. Veldhuis, T.A.M. Kevenaar, and A.H.M. Akkermans. Multi-bits biometric string generation based on the likelihood ratio. In *IEEE Conference on Biometrics: Theory, Applications and Systems*, 2007. 1, 2
- [9] P. J. Phillips, P. J. Flynn, W. T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W.J. Worek. Overview of the face recognition grand challenge. In *CVPR (1)*, pages 947–954, 2005. 3
- [10] P. J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1090–1104, 2000. 3
- [11] R.N.J. Veldhuis, A. Bazen, J. Kauffman, and P. Hartel. Biometric verification based on grip-pattern recognition. *Security, Steganography, and Watermarking of Multimedia Contents VI. Edited by Delp, Edward J., III; Wong, Ping W. Proceedings of the SPIE*, 5306:634–641, 2004. 4
- [12] M. Gardner. The binary gray code. In *Knotted Doughnuts and Other Mathematical Entertainments*, chapter 2. W. H. Freeman and Co., 1986. 4