EXTRACTING BIOMETRIC BINARY STRINGS WITH MINIMAL AREA UNDER THE FRR CURVE FOR THE HAMMING DISTANCE CLASSIFIER

Chun Chen, Raymond Veldhuis

Signals and Systems Group, Electrical Engineering, University of Twente
P.O. Box 217, 7500AE Enschede, The Netherlands
email: {c.chen, r.n.j.veldhuis}@utwente.nl

ABSTRACT

Quantizing real-valued templates into binary strings is a fundamental step in biometric compression and template protection. In this paper, we introduce the area under the FRR curve optimize bit allocation (AUF-OBA) principle. Given the bit error probability, AUF-OBA assigns the numbers of quantization bits to every feature, in such way that the analytical area under the false rejection rate (FRR) curve for a Hamming distance classifier (HDC) is minimized. Experiments on the FRGC face database yield good performances.

1. INTRODUCTION

Binary biometric representations are used in data compression and template protection [1]. For the recognition purpose, the binary strings should achieve low false acceptance rate (FAR) and false rejection rate (FRR). 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 binary strings is by quantizing and coding the real-valued biometric templates: Firstly, independent features are extracted from the raw measurements. Afterwards, features are quantized individually. The final binary string is then the concatenation of the bits from every feature. To obtain *i.i.d.* bits, some equal-probability quantizers have been introduced [2], [3], [4]. Furthermore, independent of the quantizer design, a detection rate optimized bit allocation (DROBA) principle [5] was proposed to assign the number of quantization bits, based on the density distribution of every feature, so that the analytical overall detection rate of the binary string is maximized.

Often, binary strings are compared by a Hamming distance classifier (HDC) that makes decision on the Hamming distances between two strings. Thus, theoretically DROBA only provides the optimal solution at zero Hamming distance threshold, and the performances at the operational range are not optimized. In this paper, we first give the analytical performances of the HDC, based on the features' bit error probability. Furthermore, we propose an area under the FRR curve optimized bit allocation (AUF-OBA) principle that minimizes the area under the FRR curve for the HDC.

In Section 2 we demonstrate the performance of a Hamming distance classifier, given the features' bit error probability. In Section 3 we present the AUF-OBA principle. In Section 4, we give some experimental results of AUF-OBA on the FRGC (version 1) face database and conclusions are drawn in Section 5.

2. HAMMING DISTANCE CLASSIFIER (HDC)

In this section, we demonstrate that the analytical FAR and FRR performances of a HDC is predictable, once the bit error probabilities of both the genuine user and the imposters are known.

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

During the enrollment, let $s_{\text{g},i}$ denote the b_i bits generated by the genuine user for the i^{th} feature. The entire L-bit string for the genuine user $\boldsymbol{s}_\text{g}$ is then the concatenation of the bits extracted from every single feature, i.e. $\boldsymbol{s}_\text{g} = s_{\text{g},1} \dots s_{\text{g},D}$. Similarly, during the verification, let $s'_{\text{g},i}$ and $s'_{\text{i},i}$ be the bits generated by the genuine user and the imposters, respectively, for the i^{th} feature, and $\boldsymbol{s}'_\text{g}$ and $\boldsymbol{s}'_\text{i}$ be their corresponding entire L-bit string. We know that during the verification, due to the inter- and intra-class variation, the genuine user might not extract the same string as the enrollment template, i.e. $s'_{\text{g},i} \neq s_{\text{g},i}$. Contrarily, the imposter might end up with the same string as that of the genuine user in the enrollment, i.e. $s'_{\text{i},i} = s_{\text{g},i}$. For these reasons, we can compute the bit error probabilities for $\boldsymbol{s}'_\text{g}$ and $\boldsymbol{s}'_\text{i}$ as compared to $\boldsymbol{s}_\text{g}$. Therefore, for the i^{th} feature, we define:

$$P_{\text{g},i}(k_i;b_i) \stackrel{\text{def}}{=} \mathscr{P}\{d_H(s_{\text{g},i}, s'_{\text{g},i}) = k_i\}\,, \quad (1)$$

$$P_{\text{i},i}(k_i;b_i) \stackrel{\text{def}}{=} \mathscr{P}\{d_H(s_{\text{g},i}, s'_{\text{i},i}) = k_i\}, k_i \in 0,\dots,b_i\,, \quad (2)$$

where d_H computes the Hamming distance between two input bit strings. Hence $P_{\text{g},i}$ and $P_{\text{i},i}$ represent – for the genuine user and the imposters, respectively – the probability of having k_i bits error among the b_i bits extracted for the i^{th} feature during the verification, as compared to the genuine enrollment bit string.

Regarding a total of D features, we define:

$$\phi_\text{g}(k;\{b_i\}) \stackrel{\text{def}}{=} \mathscr{P}\{d_H(\boldsymbol{s}_\text{g}, \boldsymbol{s}'_\text{g}) = k\}\,, \quad (3)$$

$$\phi_\text{i}(k;\{b_i\}) \stackrel{\text{def}}{=} \mathscr{P}\{d_H(\boldsymbol{s}_\text{g}, \boldsymbol{s}'_\text{i}) = k\}, k \in 0,\dots,L\,, \quad (4)$$

where $\phi_\text{g}(k)$ and $\phi_\text{i}(k)$ represent – for the genuine user and the imposters, respectively – the probability of having k bits error among the entire L bits extracted during the verification, as compared to the enrollment bit string. Assume that the features are statistically independent, thus their bit errors are also independent. Therefore, the error probability of the whole feature sets equals to the convolution of their individual probabilities. Thus ϕ_g and ϕ_i can be computed from the convolution of $P_{g,i}$ and $P_{i,i}$:

$$\phi_g(k;\{b_i\}) = (P_{g,1} * P_{g,2} * \ldots * P_{g,D})(k;\{b_i\}) , \quad (5)$$
$$\phi_i(k;\{b_i\}) = (P_{i,1} * P_{i,2} * \ldots * P_{i,D})(k;\{b_i\}) . \quad (6)$$

Expressions in (5) and (6) are defined as the bit error probability models 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. Thus, the FAR ($\alpha(t;\{b_i\})$ at the Hamming distance threshold t is:

$$\begin{aligned} \alpha(t;\{b_i\}) &\stackrel{\text{def}}{=} \mathscr{P}\{d_H(\mathbf{s}_g,\mathbf{s}'_i) \le t\} , \\ &= \sum_{k=0}^{t} \phi_i(k;\{b_i\}) , \qquad i = 1,\ldots,D . \quad (7) \end{aligned}$$

Similarly, the FRR ($\beta(t;\{b_i\})$ at the Hamming distance threshold t is:

$$\begin{aligned} \beta(t;\{b_i\}) &\stackrel{\text{def}}{=} \mathscr{P}\{d_H(\mathbf{s}_g,\mathbf{s}'_g) > t\} , \\ &= \sum_{k=t+1}^{L} \phi_g(k;\{b_i\}) , \qquad i = 1,\ldots,D . (8) \end{aligned}$$

3. AREA UNDER FRR CURVE OPTIMIZED BIT ALLOCATION (AUF-OBA)

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 single feature, a background probability density function (PDF) and a genuine user PDF – indicating the feature density of the imposters and the genuine user respectively – are known. Moreover, a quantizer is employed to quantize the i^{th} feature into b_i bits, $i = 1,\ldots,D$, $b_i \in \{0,\ldots,b_{\max}\}$. From (7) and (8) we observe that the FAR and FRR performances of the strings depend on the bits assignment $\{b_i\}$. Therefore, the goal of the bit extraction is to determine $\{b_i\}$, so that the verification performance in (7) and (8) is optimal.

Furthermore, to obtain *i.i.d* imposter bits, an equal-probability quantizer is required for the quantization of every feature. We know that an equal-probability quantizer gives equally 2^{-b_i} probability mass for every interval. Thus, for the i^{th} feature, when assigned with 2^{b_i} code words, the k_i-bit error probability $P_{i,i}(k_i;b_i)$ for the imposters becomes:

$$P_{i,i}(k_i;b_i) = 2^{-b_i} \binom{b_i}{k_i} . \quad (9)$$

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

$$\begin{aligned} \alpha(t;\{b_i\}) &= \sum_{k=0}^{t} \phi_i(k;\{b_i\}) , \\ &= 2^{-L} \sum_{k=0}^{t} \binom{L}{k} . \quad (10) \end{aligned}$$

Expression (10) suggests that when quantized by an equal-probability quantizer, the FAR only depends on the string length L and becomes independent of the bits assignment $\{b_i\}$. Therefore, to optimize the FAR and FRR performances, we propose to minimize the area under the FRR curve. The optimization problem is then formulated as:

$$\begin{aligned} \{b_i^*\} &= \arg \min_{\sum_{i=1}^{D} b_i = L} A_{\text{FRR}} , \\ &= \arg \min_{\sum_{i=1}^{D} b_i = L} \sum_{t=0}^{L} \beta(t;\{b_i\}) , \quad (11) \end{aligned}$$

3.2 AUF-OBA Solution

We first reformulate $\beta(t;\{b_i\})$ in (8) into the following expression:

$$\begin{aligned} \beta(t;\{b_i\}) &= \sum_{k=t+1}^{L} \phi_g(k;\{b_i\}) , \\ &= \sum_{k=0}^{L} \mathrm{u}(k-(t+1))\phi_g(k;\{b_i\}) , \quad (12) \end{aligned}$$

with

$$\mathrm{u}(k) \stackrel{\text{def}}{=} \begin{cases} 1, & k \ge 0 , \\ 0, & k < 0 . \end{cases} \quad (13)$$

Therefore the area under the FRR curve becomes:

$$\begin{aligned} A_{\text{FRR}} &= \sum_{t=0}^{L} \beta(t;\{b_i\}) , \\ &= \sum_{t=0}^{L} \sum_{k=0}^{L} \Big[\mathrm{u}(k-(t+1))\phi_g(k;\{b_i\}) \Big] , \\ &= \sum_{k=0}^{L} \Big[\phi_g(k;\{b_i\}) \sum_{t=0}^{L} \mathrm{u}(k-(t+1)) \Big] , \\ &= \sum_{k=0}^{L} k\phi_g(k;\{b_i\}) , \\ &= \mathscr{E}[k;\{b_i\}] . \quad (14) \end{aligned}$$

Hence, A_{FRR} equals to the expected value of the number of bit errors $\mathscr{E}[k;\{b_i\}]$. Furthermore, we know that the k-bit error of a L-bit binary string come from D real-valued features. Thus with k_i ($i = 1,\ldots,D$) bits error per feature. Furthermore, we have that the expected value of a sum equals the sum of the expected values. Therefore,

$$\begin{aligned} A_{\text{FRR}} &= \mathscr{E}[k;\{b_i\}] , \\ &= \sum_{i=1}^{D} \mathscr{E}[k_i;b_i] , \quad (15) \end{aligned}$$

where $\mathscr{E}[k_i;b_i]$ is defined as the expected value of the number of errors for the i^{th} feature:

$$\mathscr{E}[k_i;b_i] = \sum_{k_i=0}^{b_i} k_i P_{g,i}(k_i;b_i) . \quad (16)$$

Let $G_i(b_i)$ be a gain factor:

$$G_i(b_i) = -\mathscr{E}[k_i;b_i] = -\sum_{k_i=0}^{b_i} k_i P_{g,i}(k_i;b_i) , \quad (17)$$

we can now formulate AUF-OBA principle in (11) into:

$$\begin{aligned}\{b_i^*\} &= \arg\min_{\sum_{i=1}^{D} b_i = L} \sum_{i=1}^{D} \mathscr{E}[k_i;b_i] \,, \\ &= \arg\max_{\sum_{i=1}^{D} b_i = L} \sum_{i=1}^{D} G_i(b_i) \,. \end{aligned} \quad (18)$$

This optimization can be solved by a common dynamic programming approach that is also used in DROBA [5], see Appendix A. The computational complexity is about $O(D^2 \times b_{\max}^2)$.

3.3 The genuine user k_i-bit error probability

Computing the gain factor G_i in (17) relies on the genuine user k_i-bit error probability $P_{g,i}(k_i;b_i)$, as defined in (1). Given the real-valued genuine user PDF $p_{g,i}$ as well as a quantizer, we can compute $P_{g,i}(k_i;b_i)$ as:

$$\begin{aligned} P_{g,i}(k_i;b_i) &\stackrel{\text{def}}{=} \mathscr{P}\{d_H(s_{g,i},s'_{g,i}) = k_i\} \,, \\ &= \int_{Q(k_i;b_i)} p_{g,i}(v)dv \,, \end{aligned} \quad (19)$$

where $Q(k_i;b_i)$ indicates the quantization intervals with k_i-bit error as compared to the genuine code $s_{g,i}$. An example of these intervals based on Gray code is illustrated in Fig. 1.

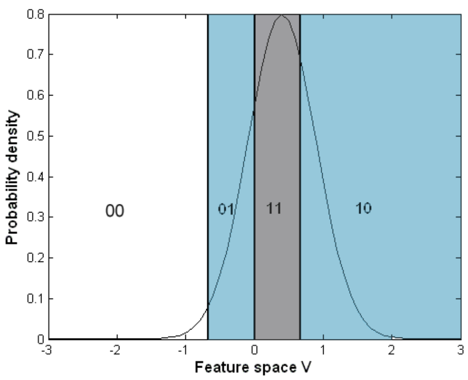


Figure 1: An example of computing $P_{g,i}(k_i;b_i)$ for the i^{th} feature, assigned with $b_i = 2$ bits Gray code. The genuine user PDF $p_{g,i}$ (black curve); $Q(0;2)$ with the genuine code '11' (grey); $Q(1;2)$ with 1-bit error (blue); and $Q(2;2)$ with 2-bit error (white).

4. EXPERIMENTS

4.1 Experimental setup

We tested the AUF-OBA principle on the FRGC(version 1) [6] face database. A standard landmark based registration method, i.e. eyes, nose and mouth, was used to align the images. Afterwards, a measurement of 8762 gray pixel values were extracted. We made two subsets: FRGC_H and FRGC_L. Set FRGC_H contains 275 users with various numbers of high quality images (n from 4 to 36), taken under controlled conditions. Set FRGC_L contains 198 users with low quality images (n from 4 to 16), taken under uncontrolled conditions. We randomly selected different users for training and testing and repeated our experiment with 5 partitionings. With, in total, n samples per user, the division of the data is stated in Table 1.

Table 1: Data division (number of users $\times$ number of samples per user) and the number of trials for FRGC_H and FRGC_L.

	Training	Enrollment	Verification	Partitioning
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

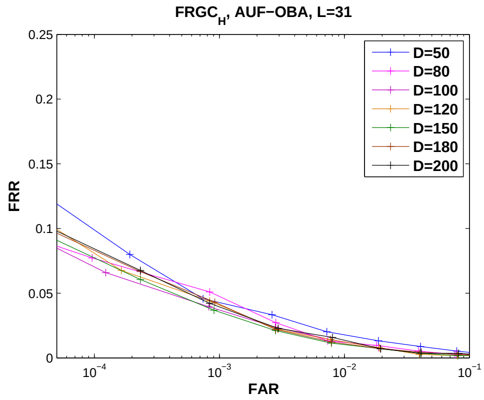
In the training step, we first applied a combined PCA/LDA method [7] on a training set. The obtained transformation was then applied to both the enrollment and verification sets. We assume that the measurements have Gaussian density, thus after the PCA/LDA, the extracted D features are statistically independent. In the enrollment step, we applied the AUF-OBA for every target user. We set $b_{\max} = 3$. The gain factor G_i was computed from the fixed quantizer [2], [3], [4]. Additionally, the background PDF and the genuine user PDF were modeled as Gaussian density, e.g. $p_b = N(v,0,1)$, $p_g = N(v,\mu,\sigma)$, respectively. The AUF-OBA outputs the bit assignment $\{b_i^*\}$, based on which the features were quantized and coded with a Gray code. In the verification step, the features of the query user were quantized and coded according to the $\{b_i^*\}$ of the claimed identity, resulting in a query binary string. Finally the query binary string was compared with the target binary string by using a HDC.

4.2 Experimental results

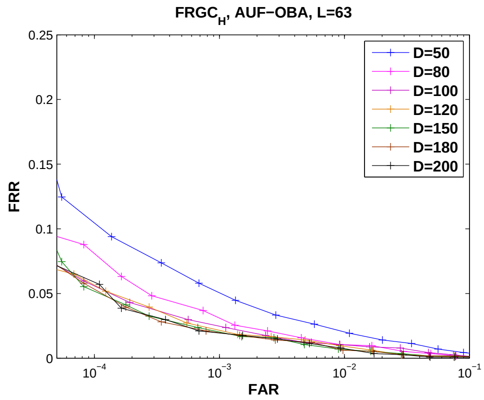
We evaluated the performances of the binary strings with $L = 31$, 63 and 127, extracted from various numbers of features D. The FAR/FRR performances for FRGC_H and FRGC_L are shown in Fig. 2 and Fig. 3, where the FAR is plotted in log scale. Since the HDC is evaluated at integer Hamming distance threshold, the FAR/FRR performances are discrete. Figure 2 suggests that for the high quality data FRGC_H, given L, when the number of features D increases, the overall FAR/FRR performance improves and becomes stable. This result proves that AUF-OBA can effectively extract distinctive bits when the feature dimensionality is high. Contrarily, Fig. 3 suggests that for the low quality data FRGC_L, given L, when the number of features D increases, the overall FAR/FRR performance improves. However, when $D \gg L$, as seen with $L = 31$ and 63 in Fig. 3(a), 3(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 built on such features are not accurate.

To further investigate the performances at the operational points, i.e. FAR $\approx 10^{-4}$, we picked the D-L settings with the best performances around such operational points. The FAR/FRR performances for FRGC_H and FRGC_L are listed in Table 2 and Table 3. Results show that the FRR performances at FAR $\approx 10^{-4}$ are good regarding a compression or template protection system. Additionally, A lower FRR is achieved when the binary string length L is larger, e.g. $L = 127$.

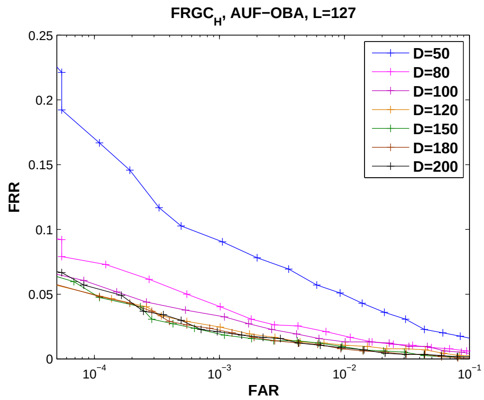
To compare the performances of AUF-OBA with DROBA, in Fig. 4 we illustrated their performances at the same D-L settings. Results show that AUF-OBA is slightly better than DROBA.



(a)

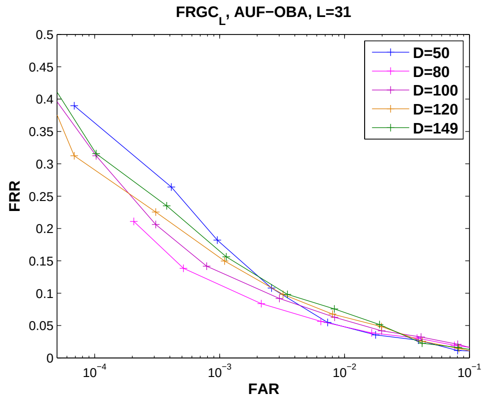


(b)

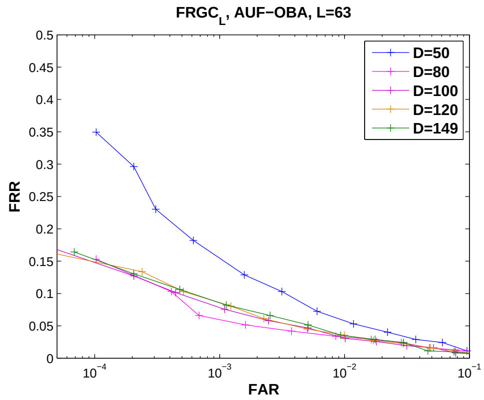


(c)

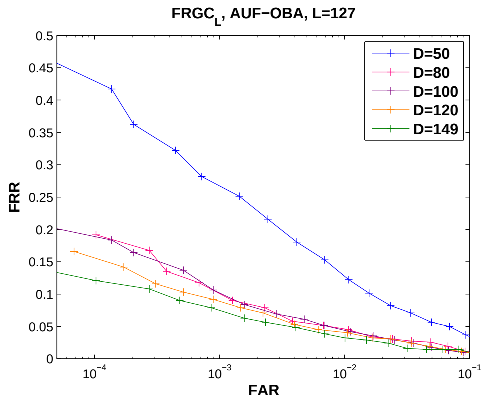
Figure 2: The FAR/FRR performances for $FRGC_H$ extracted under AUF-OBA principle, from various numbers of features D, at (a) $L = 31$; (b) $L = 63$ and (c) $L = 127$.



(a)



(b)



(c)

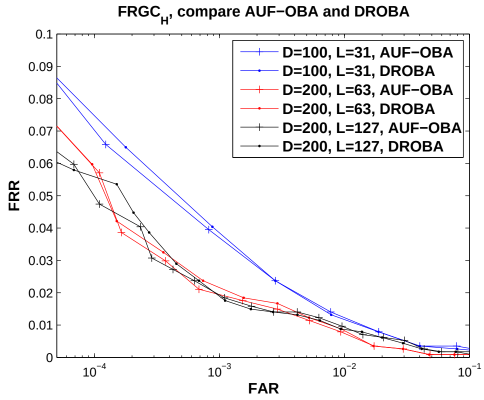
Figure 3: The FAR/FRR performances for $FRGC_L$ extracted under AUF-OBA principle, from various numbers of features D, at (a) $L = 31$; (b) $L = 63$ and (c) $L = 127$.

Table 2: The FAR/FRR performances for $FRGC_H$.

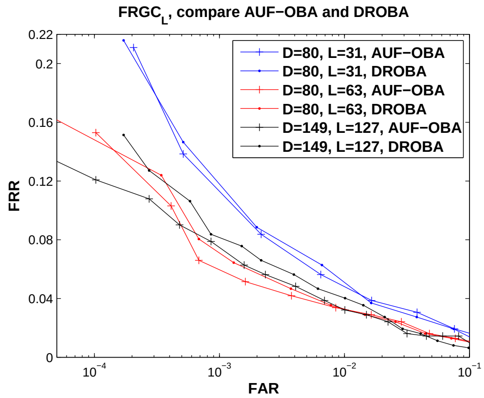
$FRGC_H$	FRR	FAR	FRR	FAR	FRR	FAR
	(%)		(%)		(%)	
D=100, L=31	6.5	0.01	2.3	0.2	0.7	1.8
D=200, L=63	5.7	0.01	1.7	0.1	0	1.7
D=200, L=127	4.7	0.01	1.8	0.1	0	1.4

Table 3: The FAR/FRR performances for $FRGC_L$.

$FRGC_L$	FRR	FAR	FRR	FAR	FRR	FAR
	(%)		(%)		(%)	
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



(a)



(b)

Figure 4: The FAR/FRR performances of AUF-OBA, compared with DROBA, for (a) FRGC_H and (b) FRGC_L.

5. CONCLUSION

Quantizing real-valued templates into binary strings is a fundamental step in biometric compression and template protection. In this paper, we propose the AUF-OBA principle. Given the features' bit error probability, AUF-OBA assigns the numbers of quantization bits to every feature, in such way that the analytical area under the FRR curve of a Hamming distance classifier is minimized. Experiments on the FRGC face database yield good performances. Theoretically, AUF-OBA is superior to DROBA and this is proved by the experimental results. However, the improvements is not very significant.

6. ACKNOWLEDGMENTS

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

A. DYNAMIC APPROACH

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

Input:

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

Initialize:

$$\begin{aligned} j &= 0\ ,\\ b_0(0) &= 0\ ,\\ G^{(0)}(0) &= 1\ , \end{aligned}$$

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

$$\begin{aligned} j &= j+1\ ,\\ \hat{b}', \hat{b}'' &= \mathop{\arg\max}_{\substack{b'+b''=l,\\ b' \in \{0,\ldots,(j-1)\times b_{\max}\},\\ b'' \in \{0,\ldots,b_{\max}\}}} G^{(j-1)}(b') + G_j(b'')\ ,\\ & \quad l = 0,\ldots,j \times b_{\max}\ ,\\ G^{(j)}(l) &= G^{(j-1)}(\hat{b}') + G_j(\hat{b}'')\ ,\\ b_i(l) &= b_i(\hat{b}'), i = 1,\ldots,j-1\ ,\\ b_j(l) &= \hat{b}''\ , \end{aligned}$$

end while

Output:

$$\{b_i^\star\} = \{b_i(L)\}, i = 1,\ldots,D\ .$$

REFERENCES

[1] A.K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 8(2), 2008.

[2] 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 *Proc. Audio-and Video-Based Biometric Person Authentication (AVBPA 2005)*, pages 436–446, NY, USA, 2005.

[3] 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 *Proc. IEEE Workshop on Automatic Identification Advanced Technologies (AutoID 2005)*, pages 21–26, NY, USA, 2005.

[4] 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 *Proc. IEEE Conference on Biometrics: Theory, Applications and Systems (BTAS07)*, 2007.

[5] C. Chen, R.N.J. Veldhuis, T.A.M. Kevenaar, and A.H.M. Akkermans. Biometric binary string generation with detection rate optimized bit allocation. In *Computer Vision and Pattern Recognition Workshops (CVPR)*, 2008.

[6] 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 *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 947–954, MD, USA, 2005.

[7] R.N.J. Veldhuis, A. Bazen, J. Kauffman, and P. Hartel. Biometric verification based on grip-pattern recognition. In *Proc. SPIE Security, Steganography, and Watermarking of Multimedia Contents VI (SSWMC 2004)*, volume 5306, pages 634–641, CA, USA, 2004.