ON BINARY REPRESENTATIONS FOR BIOMETRIC TEMPLATE PROTECTION

Chun Chen

De promotiecommissie:

voorzitter en secretaris:		
	Prof.dr.ir. A.J. Mouthaan	Universiteit Twente
promotor:		
	Prof.dr.ir. C.H. Slump	Universiteit Twente
assistent promotor:		
	Dr.ir. R.N.J. Veldhuis	Universiteit Twente
referenten:		
	Dr.ir. T.A.M. Kevenaar	GenKey Europe
leden:		
	Prof.dr.ir S. Etalle	Universiteit Twente
	Dr.ir. M. Bentum	Universiteit Twente
	Prof.dr.ir. P. Campisi	Universita' degli Studi Roma Tre
	Prof.dr.ir. J.W.M. Bergmans	Technische Universiteit Eindhoven

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.

Signals & Systems group,
EEMCS Faculty, University of Twente
P.O. Box 217, 7500 AE Enschede, the Netherlands

Printed by Gildeprint, Enschede, The Netherlands
Typesetting in LaTeX2e

ISBN 978-90-365-2830-6
DOI 10.3990/1.9789036528306

ON BINARY REPRESENTATIONS FOR BIOMETRIC TEMPLATE PROTECTION

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. H. Brinksma,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op 7 December 2011 om 14:45

door

Chun Chen
geboren op 20 March 1981
te Nanjing, China

Dit proefschrift is goedgekeurd door:

De promotor:	Prof.dr.ir. C.H. Slump
De assistent promotor:	Dr.ir. R.N.J. Veldhuis

Contents

Nomenclature

Abbreviations

APQ	Adaptive phase quantizer
AUF-OBA	Area under the FRR curve optimized bit allocation
DET	Detection error tradeoff curve
DROBA	Detection rate optimized bit allocation
DP	Dynamic programming
ECC	Error-correcting code
EER	Equal error rate
FAR	False acceptance rate
FBA	Fixed bit allocation
FQ	Fixed quantizer
FRR	False rejection rate
GAR	Genuine acceptance rate
GHD	Genuine Hamming distance
GS	Greedy search
HDC	Hamming distance classifier
IHD	Imposter Hamming distance
LC	Likelihood ratio classifier
LDA	Linear discriminant analysis
LL	Long-long
LQ	Likelihood ratio based quantizer
LS	Long-short

MC	Mahalanobis distance classifier
PCA	Principle component analysis
PDF	Probability density function
QIM	Quantization index modulation
ROC	Receiver operating characteristic curve
WAUF-OBA	Weighted area under the FRR curve optimized bit allocation
ZQ	Zhang's multi-bits quantizer

1 Introduction

1.1 Biometric systems

Biometrics is the science of establishing the identity of an individual based on the physical, chemical or behavioral attributes of the person, commonly referred to as fingerprint, face, hand geometry, iris, signature, voice, gait, or DNA information [1]. Biometrics is becoming increasingly incorporated in various applications, such as access control, data management, national ID, passport control, and forensics.

Unlike traditional means of identity establishment (e.g. passwords and ID cards), which can easily be lost, shared, manipulated or stolen, biometrics offers a natural and reliable solution to certain aspects of identity management, by utilizing fully automated or semi-automated schemes based on an individual's unique biological characteristics [2]. In this way, using biometrics could guarantee that an identity who accesses a system can not later deny it. Besides, biometrics also enhances user convenience by alleviating the need to design and remember passwords or to carry tokens.

Figure 1.1 illustrates how a biometric system works. An enrollment stage is first passed to generate the biometric templates of the users. Before being stored in the database, the captured *biometric raw measurement* needs to pass quality assessment and feature extraction steps. These steps yield a compact collection of *biometric features*, called the *biometric template*. A biometric system may be used for verification of an identity or identification of an individual [1]. In a verification system, a user's identity is verified by comparing his/her biometric template to that of the claimed identity. This is a one-to-one comparison. In an identification system, a user's identity is established through comparing his/her template to those of all the users in the database. This is a one-to-many comparison. The decision of choosing a verification

or an identification system depends on the application context.

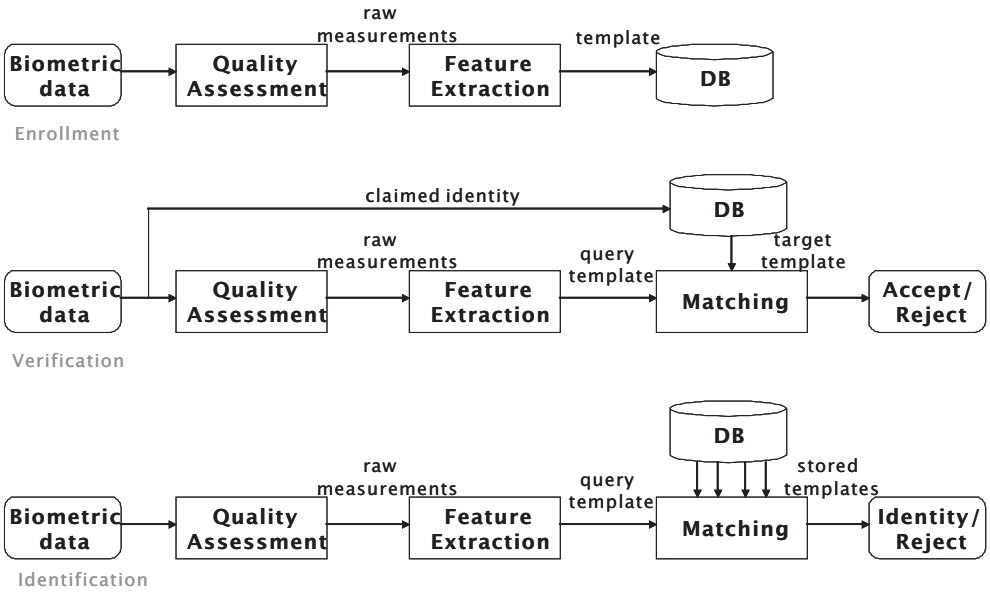


Figure 1.1: *The enrollment, verification and identification stages of a biometric system.*

The recognition performance of a biometric system is often presented as the false rejection rate (FRR) and the false acceptance rate (FAR). The FRR is defined as the probability that a system will incorrectly reject an access attempt by a genuine user. An alternative measurement of FRR is the detection rate or the genuine acceptance rate (GAR), defined as the probability that a system will correctly accept a genuine user. Thus:

$$GAR = 1 - FRR \ , \tag{1.1}$$

The other performance measurement is the FAR, defined as the probability that a system will incorrectly accept an access attempt by an imposter. Both a low FRR and a low FAR are favorable. However, a system aiming for a lower FAR usually has a higher FRR and vice versa. Therefore, in designing a biometric system, the goal is to optimize the system parameters in order to obtain a better trade-off between the FAR and the FRR. Often this trade-off is illustrated either as a receiver operating characteristic (ROC) curve showing the GAR against the FAR, or as a detection error tradeoff (DET) curve showing the FRR against the FAR, at various parameter values. The Equal Error Rate (EER), defined as the point in the DET curve where the FAR equals the FRR, is also used to measure the system performance in comparing different sets of parameters.

1.2 Biometric template protection

1.2.1 Vulnerabilities of biometric systems

Unlike passwords or ID cards, biometrics are unique, irrevocable, and may even contain sensitive private information. Unfortunately, in most of the current applications, biometric templates are stored merely as a compact collection of features that are directly extracted from the raw measurements. As a result, biometric templates are exposed to a number of threats: First, it is possible to recover the biometric measurements from the stored template. For instance, a hill-climbing attack can be conducted by iteratively adjusting a candidate's face image according to the matching score of this image and a target image in the database [3]. Second, if a sufficiently similar biometric template of the same individual is stored in multiple application databases, it is susceptible to cross-matching between two or more reference templates from the same subject across different applications. Finally, biometric templates may contain sensitive private information. In many countries, the widespread biometric applications have given rise to legislations on privacy protection of personal biometric data.

As a countermeasure to these threats, biometric template protection has become an important issue, and therefore is the motivation of this research.

1.2.2 Requirements for a template protection system

Generally speaking, a biometric template protection system aims to prevent the abuse of private biometric information, while maintaining the biometric recognition performances. A biometric template protection system should satisfy the following properties [4]:

- **Diversity**: It should be possible to generate multiple templates from the same user, in order to prevent cross-matching over different databases.
- **Revocability**: It should be straightforward to revoke a compromised template and reissue a new one for the same user.
- **Security and privacy**: From the security perspective, it must be computationally hard to recover *a* set of biometric features that can gain access to the biometric system. From the privacy perspective, it must be computationally hard to recover *the* set of biometric features that are similar enough to those of the user to prevent revealing private personal information.
- **Recognition performance**: The biometric template protection system should not degrade the FAR and the FRR performances of the unprotected biometric.

Both diversity and revocability require the capability of generating multiple protected templates from the same user. This could be achieved by associating the biometric template with random variations. For instance a function with random variables or a random key.

The security of a biometric system is quantified as the average effort for an attacker to obtain a set of biometric features that is similar enough to gain access. Privacy is

quantified as the average effort for an attacker to obtain a set of biometric features that is similar enough to reveal the private information. Although defined in a similar way, security and privacy are two different concepts and they are dependent on the accuracy of the template protection system. For instance, in an extreme case where the template protection scheme quantifies every user's biometric data into a single bit of 0 or 1, an attacker only needs to guess a 0 or 1 in order to gain access, which gives very low security. However, the privacy is well preserved in this case, because a single bit hardly tells anything about what the original biometric data (e.g. face or fingerprint) looks like. Another issue about security and privacy is the quantification of the effort. Security can be quantitatively measured in terms of the effort of recovering an accessible version of the real-valued biometric data. Privacy, however, is even more difficult to quantify, because it is unclear how accurately a biometric template must be determined in order to reveal private information. This, of course, also depends on the kind of information that is looked for.

With additional template protection, the FAR and the FRR performances of a template protection system often degrade as compared to an unprotected biometric system. Therefore, a biometric template protected scheme is desired to maintain low FAR and low FRR. Note that the FAR also indicates the security of the biometric system. Forcing the system to make a false accept is sometimes called a zero-effort attack [4].

1.2.3 Overview of template protection schemes

At present, most of the biometric template protection schemes are designed for verification. Therefore, we give an overview of template protection methods in the context of a verification system. The major challenge of a biometric template protection system comes from the intra-user variations, i.e., the biometric measurements of the same user change from instant to instant. For these reasons, it is not possible to directly apply one-way hash functions to the extracted biometric features, as in the traditional password based identity establishment systems. However, there are attempts to directly generate a cryptographic key from biometric features, such as biometric key generation and fuzzy extractor. An alternative solution is to acquire a user-specific key and use it as a guidance to generate a cryptographic key from biometric features, such as BioHashing. Contrarily, other template protection schemes are aiming to design a computationally non-invertible function or a hash that involves error-correcting codes (ECC), to be applied to the biometric features. These schemes are Cancelable biometrics, Fuzzy Commitment, Helper Data, Secure Sketch and Fuzzy Vault. A summary of the properties of these schemes are given in Table 1.1, which is a revision from a table in [4]. A more detailed description is given below.

Biometric key generation [5], [6], [7], [8], [9], [10] and *fuzzy extractor* [11], [12], [13] belong to a category of template protection schemes that directly generates a cryptographic key from biometric features. To overcome intra-user variations, user-specific quantization is employed in these schemes. Information about the quantization boundary and the quantized codes are stored. Comparison is done in the discrete domain. However, it is only possible to generate one quantized template for every

Table 1.1: *Summary of different template protection schemes, where $\mathcal{F}$ represents an invertible transformation function; $\widetilde{\mathcal{F}}$ represents a computationally non-invertible transformation function; T denotes the target biometric features; Q denotes the query biometric features; K_ω is a user-specific key; K is a random key.*

	Diversity depends on	Stored data	Matching is done by	Security depends on
Biometric Key Generation Fuzzy Extractor	no	public: $\mathcal{F}(T)$	$\mathcal{F}(T)$; $\mathcal{F}(Q)$	information revealed by $\mathcal{F}$
BioHashing	K_ω	public: $\mathcal{F}(T;K_\omega)$ secret: K_ω	$\mathcal{F}(T;K_\omega)$; $\mathcal{F}(Q;K_\omega)$	secret key K_ω
Cancelable biometrics	K_ω	public: $\widetilde{\mathcal{F}}(T;K_\omega)$ K_ω	$\widetilde{\mathcal{F}}(T;K_\omega)$; $\widetilde{\mathcal{F}}(Q;K_\omega)$	non-invertible $\widetilde{\mathcal{F}}$
Fuzzy Commitment Helper Data Secure Sketch Fuzzy Vault	K	public: $\mathcal{F}(T;K)$ $\widetilde{\mathcal{F}}(K)$	$\widetilde{\mathcal{F}}(K)$; $\widetilde{\mathcal{F}}(\mathcal{F}^{-1}(Q;\mathcal{F}(T;K))^a$	non-invertible $\widetilde{\mathcal{F}}$ and information revealed by $\mathcal{F}$

[a] $\mathcal{F}^{-1}$ here refers to a general reverse process to retrieve K from Q and $\mathcal{F}(T;K)$. Thus $\mathcal{F}^{-1}$ is not restricted to be a mathematically defined inverse function.

user. To what extent the biometric features can be recovered from the stored template depends on the quantization process.

BioHashing [14], [15], [16], [17], [18] is a template protection scheme that transforms biometric features under the guidance of a user-specific key. These transformed features are then stored as the template. In the verification stage, the transformation is applied to the query biometric features according to the query user-specific key. The resulting query template is then compared with the stored target template. Usually, the transformation function is known. Hence the user-specific key needs to be securely stored or remembered by the user. If this key is compromised, the template is compromised as well. Since one user could have multiple secrete keys, BioHashing enables multiple templates for the same user. However, introducing extra user-specific keys gives security responsibility to users.

Cancelable biometrics [19], [20] distort the image of a face or a fingerprint by using a non-invertible geometric distortion function. Unlike the traditional hash function, the non-invertible transform refers to a one-way function that is "easy to compute" but "hard to invert" [4]. The parameters of the transform function are recorded as a user-specific key, and therefore enables multiple templates for the same user. In the verification stage, the user-specific key, combined with the transformation function, is applied to the query biometric features and the result is matched against the target template. Compared to BioHashing, even though the user-specific key is compromised, it is still computationally hard to recover the original biometric features. To overcome the intra-user variations, features from the same user should be similar and features from different users should be dissimilar in the transformed feature space. However, it is difficult to find a transformation function that provides non-invertibility and overcome intra-user variability.

Fuzzy Commitment [21], *Helper Data*, [22], [23], [24], [25], *Secure Sketch* [26], [27], [28], [29], [30] and *Fuzzy Vault* [31], [32], [33] use the noisy biometric features to bind an error-correcting encoded random key. In the enrollment stage, a random key (K) is generated. The key is hashed ($\mathcal{H}(K)$) and stored. In the mean time, it is encoded into a codeword C by the encoder of an error-correcting system. The codeword is then bound with the biometric features and stored as well. In the verification stage, the stored template releases a noisy version C' through the query user's biometric features. If C' is similar to C, C' can be correctly decoded into K within the error-correcting capability. Thus, a direct "Yes/No" match can be conducted based on $\mathcal{H}(K)$. The data stored in the database include $\mathcal{H}(K)$ as well as the bound information between the biometric features and the codeword. The random key provides multiple templates for the same user.

1.3 Research context

The context of this research is the development of a generic template protection scheme for biometric verification applications. As summarized in Table 1.1, Fuzzy Commitment, Helper Data, Secure Sketch and Fuzzy Vault are preferable, because in these systems the diversity and the revocability of biometric templates do not depend

on user-specific keys. From these possibilities, we choose the Helper Data scheme in this research. Helper Data scheme is basically a Fuzzy Commitment with additional quantization and coding of biometric features, which leads to the main topic of the thesis. To start with, in Section 1.3.1, we present the Helper Data scheme. Once it is adopted, the whole template protection system can be divided into three functional modules: feature extraction, reliable bit extraction and secure key binding verification. These modules are summarized in Section 1.3.2. Among the three modules, reliable bit extraction is crucial for the template protection performance. Therefore, extracting fixed-length secure binary strings from biometric features is defined as the main purpose of this research. Finally, in Section 1.3.3, the research objectives are presented in detail.

1.3.1 The selected template protection scheme

The Helper Data scheme [22], basically a Fuzzy commitment with additional quantization and coding of biometric features, is adopted for this research. The framework is illustrated in Fig. 1.2. S_ω and S' represent the binary strings of an enrolled user and a query user, respectively. They are derived from the real-valued biometric features through a quantization and coding procedure. During the enrollment and the verification phase, error correcting techniques are integrated in order to successfully retrieve a randomly generated key K, when the query template S' and the target template S_ω are within a certain number of errors. During the enrollment, the random key K is first encoded into a codeword C of an ECC. This codeword C and the enrolled user biometric template S_ω are bound in $W_{\omega,1}$ by the XOR operation ($W_{\omega,1} = C \oplus S_\omega$). During the verification, a noisy version of C' is released by the same XOR operation of $W_{\omega,1}$ and the query biometric string S' ($C' = W_{\omega,1} \oplus S'$). Afterwards, the C' is decoded into K' through the error-correcting decoding. The final "Yes/No" decision is made by comparing K' and the original K in their hashed manner. Thus, the access is granted only when C' can be correctly decoded into $K' = K$. Furthermore, extra quantization information may be desirable, for instance the quantization intervals or the number of quantization bits. In general, we denote such quantization information as *helper data* $W_{\omega,2}$. The helper data $W_{\omega,2}$, together with $W_{\omega,1}$ and the hashed key $\mathcal{H}(K)$, are stored publicly for every enrolled user ω.

To summarize, the Helper Data scheme uses a binary biometric string to bind a random key. ECC are applied to correct the errors in these binary strings due to the intra-class variations.

To meet the requirements of a template protection system as described in Section 1.2.2, the Helper Data system has to consider the following aspects: (1) Since the quantization intervals and quantization bits, as helper data $W_{\omega,2}$, are stored publicly, it is desirable that $W_{\omega,2}$ reveals minimum information of S_ω. Otherwise, an attack can search for S_ω by guessing the code with the highest probability within the quantization intervals presented in $W_{\omega,2}$. Retrieving S_ω would breach the security and privacy. Therefore, it is important to design quantization without revealing information of S_ω. (2) The length of the random key K determines how many keys a biometric binary string can bind. Thus, increasing the length of K gives higher diversity and

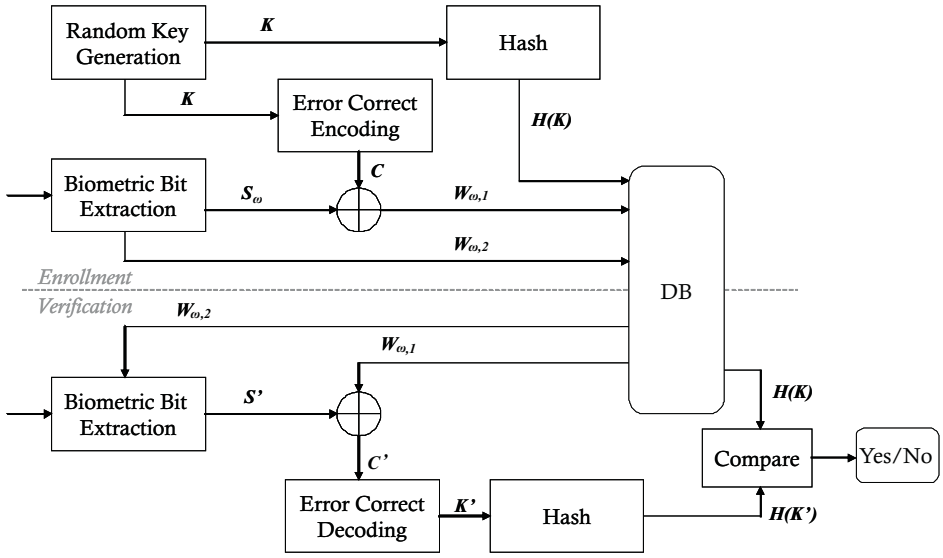


Figure 1.2: *The framework of the Helper Data scheme. S_ω and S' represent the binary strings of an enrolled user and a query user, respectively. K denotes a random key. C and C' are the error-correcting codewords. C and S_ω are bound into $W_{\omega,1}$. $W_{\omega,2}$ denotes helper data.*

revocability of the template protection system. Moreover, the length of K also tells how difficult it is to guess the random key. Since the ECC and $W_{\omega,1}$ are public, the compromise of K directly leads to the compromise of S_ω, which also brings security and privacy threats. Efforts to increase the length of K involves improving the error-correcting capability or extracting more reliable biometric bits. (3) The recognition performance FRR indicates how a genuine key K can be correctly retrieved through the error-correcting procedure, even when the biometric strings S_ω and S' of the same user are different. Contrarily, the FAR indicates how a genuine key K can be falsely retrieved, even when the biometric strings S_ω and S' are from two different users. Obviously, the FRR and the FAR depend on both the error-correcting capability and the reliability of the biometric bits. Thus, designing advanced ECC or extracting reliable biometric bits would improve the recognition performances.

1.3.2 The complete template protection system and the subject of this research

As has been described above, the Helper Data scheme is chosen as the subject of this research. Furthermore, we show that extracting biometric bits and ECC design are two key aspects that influence the performances of the template protection. In this Section, by taking a perspective of the entire verification system, we generalize a template protection system into three functional modules: feature extraction, reliable

bit extraction and secure key binding verification, as shown in Fig. 1.3. Optimizing each of the three modules would contribute to the final performances of the template protection system.

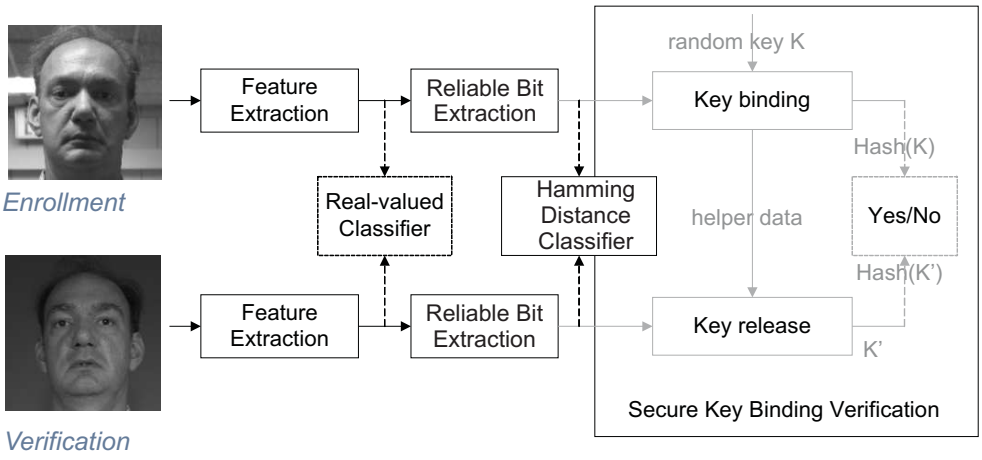


Figure 1.3: *Three modules generalized for the Helper Data based verification system: feature extraction, reliable bit extraction and secure key binding verification.*

1. **Feature extraction**: This module aims to extract independent, reliable and discriminative real-valued features from raw measurements. Strictly speaking, it involves quality control, image alignment, feature processing, and finally feature extraction. While quality control, image alignment and feature processing depend on the application and the specific biometric feature modality, the feature extraction techniques can be quite common. Classical feature extraction methods are, among others, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [34]. In fact, this feature extraction module, together with the real-valued classifier applied afterwards, constitutes the conventional biometric verification system.

2. **Reliable bit extraction**: This module aims to transform the real-valued features into a fixed-length binary string, through quantization and coding, such that the binary strings have a small Hamming distance if the real-valued features are close. 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, the extracted bits are desired to maintain low intra-class variations, leading to a low FRR. In the mean time, the extracted bits should provide sufficient security. First, a low FAR. Second, 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.*).

3. **Secure key binding verification**: This module aims to provide verification

when the target biometric string is protected and bound to a secret key. A realization of such module is the Helper Data scheme presented in Section 1.3.1: In the enrollment phase, a biometric binary string is used to bind an error-correcting codeword, encoded from a secret key K. In the verification phase, the key is correctly released, only when the erroneous bits in the query biometric binary string can be corrected by the error-correcting decoding.

Usually binary ECCs are evaluated by the $[n, k, t]$ parameters, where n, k, t represent the length of the codeword, the number of secret bits, and the error-correcting capability, respectively. In case of the Helper Data scheme, the length of the codeword equals the length of the biometric strings; the number of secret bits equals the length of the random key K. The error-correcting capability t refers to the maximum allowed number of erroneous bits, also called Hamming distance, between the codeword C and the noisy version C'. In the Helper Data system, C is directly linked to the biometric binary string S_ω. Thus, t also equals the number of erroneous bits or the Hamming distance that the ECC can correct on the biometric strings. Therefore, the secure key binding verification essentially functions as a Hamming distance classifier (HDC) that is applied to the n-bit biometric binary strings with a Hamming distance threshold t. More specifically, a HDC that grant access when the Hamming distance between two binary strings is lower than t, and vice versa.

Well-developed methods are available for both feature extraction and secure key binding verification, such as PCA and the Helper Data scheme with a BCH code. However, the capability of quantifying real-valued biometric features into binary strings has not yet been thoroughly studied. Therefore, in this research, we focus on the reliable bit extraction module which aims to extract a binary string from the biometric features, via a quantization and coding process. Furthermore, there are a variety of ECCs that could be applied to evaluate the performance of the binary strings. As we know, a $[n, k, t]$ ECC functions as a HDC applied to the n-bit binary strings with Hamming distance threshold t. Therefore, as a generalization of a variety of ECCs, we directly evaluate the performances of the biometric binary strings through a HDC.

1.3.3 Research objectives

This research focuses on the reliable bit extraction module. To summarize the contents in Section 1.3.1 and 1.3.2, the research question is refined as:

How can real-valued biometric features, in a Helper Data scheme based template protection system, be converted to a binary string, with the following requirements?

I. Since we adopt the Helper Data scheme, the binary strings extracted from the real-valued biometric features should be of fixed-length.

II. 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.*).

III. In order to maximize the length of the random key, the extracted bits should be as reliable as possible, i.e. for a given user the probability of bit errors should be as low as possible.

IV. The verification via binary strings should not degrade the FAR and the FRR performances.

Each of these requirements is translated into a corresponding research objective. To summarize, in this research, we aim to extract fixed-length reliable binary strings which have *i.i.d.* bits, while maintaining good FAR and FRR verification performance.

1.4 Overview of the thesis

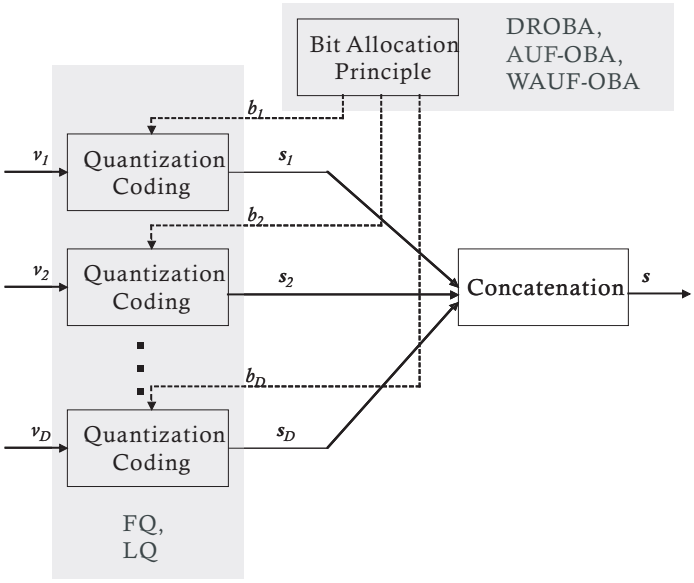
1.4.1 Main contributions

The main contributions of the thesis include two aspects: (1) how to optimize the quantization intervals for the biometric features and (2) how to allocate the number of quantization bits to features: First, we propose a one-dimensional quantization scheme, as shown in Fig. 1.4(a), where every feature is individually quantized and then concatenated into a binary string. In particular, two new one-dimensional quantizers, the fixed quantizer (FQ) and the likelihood ratio based quantizer (LQ), are presented in Chapter 2. In addition to optimizing the quantization intervals for every feature, assigning various numbers of bits to features with different discriminative power could also optimize the final binary string performance. Therefore, three new bit allocation principles, the detection rate optimized bit allocation (DROBA), the area under the FRR curve optimized bit allocation (AUF-OBA) and the weighted area under the FRR curve optimized bit allocation (WAUF-OBA), are presented in Chapters 3, 4 and 5. Moreover, as shown in Fig. 1.4(b), a two-dimensional quantization scheme is proposed. The two-dimensional polar quantizer, including the phase and the magnitude, are presented in Chapter 6. Additionally, two new pairing strategies, the long-short (LS) and the long-long (LL) pairing strategies are designed for phase and magnitude, respectively. In Chapter 7, an advanced phase quantizer, the adaptive phase quantizer (APQ) with LS pairing strategy is presented.

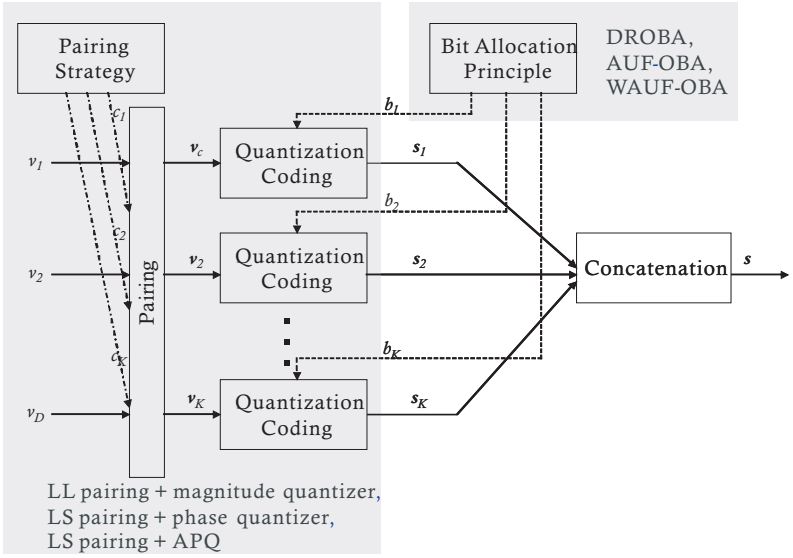
1.4.2 Chapters overview

The chapters of the thesis are based on published papers. The main chapters are Chapters 2-7, of each consists of one or more papers in their originally published format. These papers have been published in a period of more than 4 years, during which notations and terminologies have evolved. This has led to some notational inconsistencies across the papers, for which we apologize. The main contributions of the thesis and the knowledge diagram are illustrated in Fig. 1.5.

In Chapter 2, two one-dimensional quantizers, the fixed quantizer (FQ) and the likelihood ratio based quantizer (LQ), are presented. Both quantizers are able to extract multiple bits per biometric feature. The FQ determines the quantization intervals merely by equally dividing the probability mass of the background probability



(a)



(b)

Figure 1.4: *The reliable bit extraction design based on (a) one-dimensional and (b) two-dimensional quantization and coding, where $v_i, i = 1, \ldots, D$ denotes D real-valued features. In the two-dimensional case, c_i denotes the feature index for the i^{th} feature pair. b_i and s_i denote the number of quantization bits and the output bits for the i^{th} feature or feature pair. The final string s is the concatenation of s_i.*

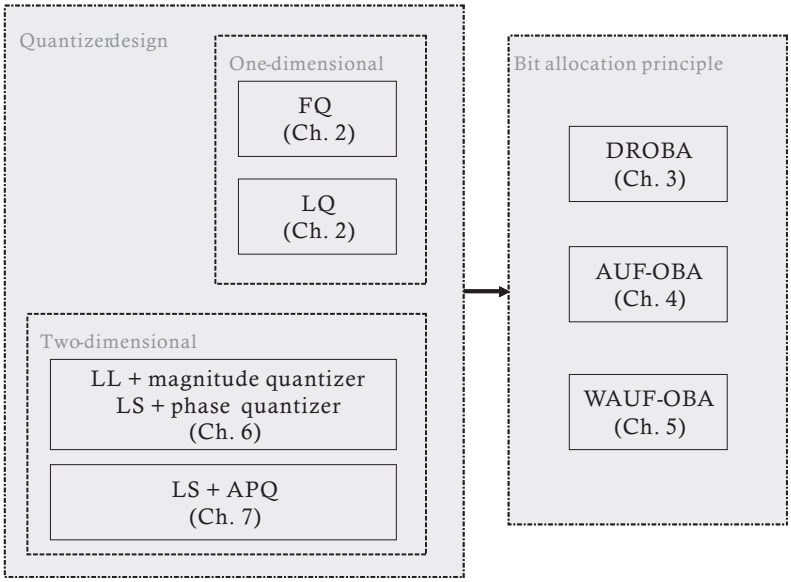


Figure 1.5: *The main contributions of the thesis and the knowledge diagram according to chapters.*

density function (PDF). The LQ determines the quantization intervals from the likelihood ratio between the genuine user PDF and the background PDF of the feature. As a result, both quantizers are able to extract *i.i.d.* bits. Superior to FQ, LQ optimizes the theoretical FRR of a feature, given a prescribed number of quantization bits. Here the theoretical FRR refers to a theoretical quantity that is optimized based on models. It is different from the actual recognition performance that is achieved on the real data experiments.

In Chapter 3, the detection rate optimized bit allocation (DROBA) principle is presented. Subject to a prescribed total length of the binary string, DROBA assigns user-dependent numbers of bits to every feature, in such way that the theoretical overall detection rate at zero Hamming distance threshold for a HDC is optimized. Both a dynamic programming and a greedy approach are then proposed to search for the optimal solution. Compared to quantizing every feature into a prescribed fixed number of bits, combining quantizers with DROBA yields better FAR and FRR performances of the entire binary strings.

In Chapter 4, the area under the FRR curve optimized bit allocation (AUF-OBA) principle is presented. Given the bit error probabilities of the biometric features, AUF-OBA assigns user-dependent numbers of bits to every feature, in such way that the theoretical area under the FRR curve for a HDC is minimized. A dynamic programming approach is then proposed to search for the optimal solution. Superior to DROBA, AUF-OBA optimizes the overall FRR performances, rather than the FRR at zero Hamming distance threshold.

In Chapter 5, the weighted area under the FRR curve optimized bit allocation (WAUF-OBA) principle is presented. Given the bit error probabilities of the biometric features, WAUF-OBA assigns user-dependent numbers of bits to every feature, in such way that the theoretical weighted area under the FRR curve for a HDC is minimized. Depending on the value of the parameter in the weighting function, different ranges of the Hamming distance thresholds are emphasized, which makes WAUF-OBA a generalization of DROBA and AUF-OBA. Superior to DROBA or AUF-OBA, WAUF-OBA optimizes the overall FRR performances in the emphasized range of Hamming distance thresholds.

In Chapter 6, a two-dimensional pairwise polar quantizer that quantizes the magnitude and the phase is introduced. Quantization intervals in both domains are selected dependent on the background PDFs of the pairwise features. Furthermore, aiming to optimize the discrimination between the genuine Hamming distance (GHD) distribution and the imposter Hamming distance (IHD) distribution, two heuristic feature pairing strategies are proposed: the long-short (LS) strategy for the phase quantization, as well as the long-long (LL) strategy for the magnitude quantization. The phase quantizer combined with the LS pairing gives low FAR and FRR performances.

In Chapter 7, a two-dimensional pairwise adaptive phase quantizer (APQ), together with an improved long-short (LS) pairing strategy, is presented. The APQ adjust the phase quantization intervals in order to maximize the theoretical detection rate of a given feature pair. The LS pairing strategy composes feature pairs in order to maximize the overall detection rate, for the total binary strings, at zero Hamming distance threshold. With APQ and LS pairing, the extracted binary strings obtain better FAR and FRR performances than the phase quantizer without adjustment in Chapter 6.

In Chapter 8, conclusions and future work are given.

1.4.3 Biometric data sets

In this research, a generic Helper Data scheme is chosen, so that the template protection is not limited to a certain biometric type. Two publicly available and accepted databases: fingerprint database FVC2000 [35], [36] and face database FRGC [37], [38] are used in evaluation. Furthermore, in order to extract fixed-length binary strings, the biometric features are extracted as following.

- **FVC2000**: The FVC2000(DB2) fingerprint data set contains 8 images of 110 users. The features were extracted in a fingerprint recognition system that was used in [22]. As illustrated in Fig. 1.6, the raw features contain two types of information: the squared directional field in both x and y directions, and the Gabor response in 4 orientations (0, $\pi/4$, $\pi/2$, $3\pi/4$). Determined by a regular grid of 16 by 16 points with spacing of 8 pixels, measurements are taken at 256 positions, leading to a total of 1536 elements.

- **FRGC**: The FRGC(version 1) face data set contains 275 users with a different number of images per user, taken under both controlled and uncontrolled conditions. The number of samples s per user ranges from 4 to 36. The image size

was 128×128. From that a region of interest (ROI) with 8762 pixels was taken as illustrated in Fig. 1.7.

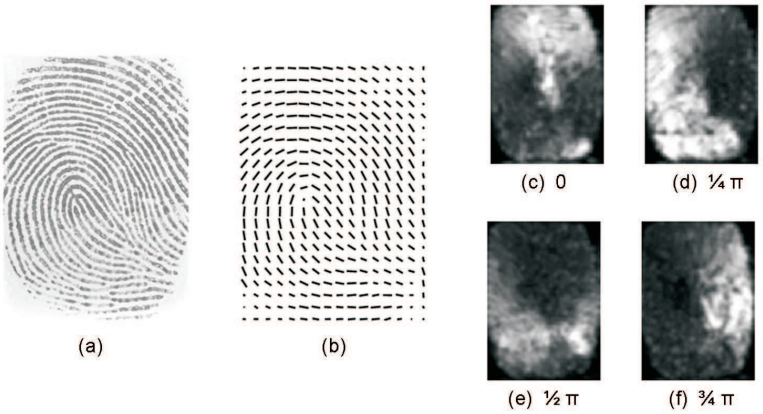


Figure 1.6: *(a) Fingerprint image, (b) directional field, (c)-(f) the absolute values of Gabor responses for different orientations θ.*

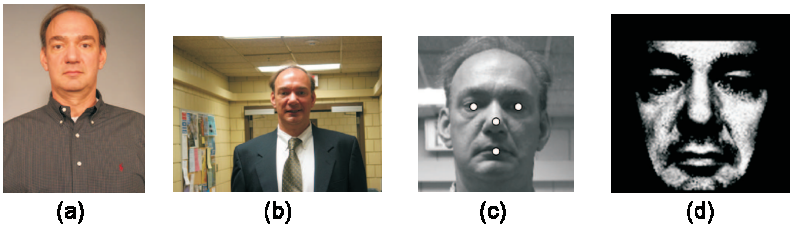


Figure 1.7: *(a) Controlled image, (b) uncontrolled image, (c) landmarks and (d) the region of interest (ROI).*

2

One-dimensional Quantizer

2.1 Chapter introduction

PURPOSE. This chapter deals with one-dimensional quantization and coding. The purpose of this chapter is to design one-dimensional quantizers for each of the biometric features, given a prescribed fixed number of bits per feature. The quantizers should be capable of extracting multiple bits, that are statistically independent and identically distributed (*i.i.d.*). After every feature is quantized into a prescribed number of bits, these bits concatenate into the biometric binary string. When applied to a Hamming distance classifier (HDC), these binary strings should result in good recognition performance.

CONTENTS. A fixed quantizer (FQ) and a likelihood ratio based quantizer (LQ) are presented in this chapter. As illustrated in Fig. 2.1, the FQ or the LQ are designed to quantize features with a number of bits that is the same for every feature. The FQ is user-independent: For every feature, the quantization intervals are merely determined by equally dividing the probability mass of the background probability density function (PDF), representing the probability density of the entire population. The interval where the mean of the genuine user PDF is located, is referred to as the genuine user interval. In contrast, LQ is user-dependent and superior to FQ: For every feature of an enrolled user, LQ determines equal probabilistic quantization intervals from the likelihood ratio between the genuine user PDF and the background PDF, where the genuine user PDF represents the probability density of the genuine user for one feature. Based on a required number of quantization intervals, the genuine user interval is first determined by applying a threshold to the likelihood ratio. Afterwards, the remaining intervals are expanded towards both tails of

the background PDF, in such way that all the quantization intervals have equal background probability mass. The left and the right tail constitute one wrap-around interval. As a result, LQ minimizes the theoretical FRR per feature at Hamming distance zero, subject to a prescribed number of quantization bits. For both quantizers, Gray codes, in which the Hamming distance of two adjacent codewords is limited to one single bit, are then assigned to the quantization intervals. This reduces the number of bit errors due to the within-class variation. Because the intervals have equal background probability, the bits assigned to each feature are *i.i.d.*. The bits in the entire binary string are then *i.i.d.*, if the biometric features are statistically independent. Figure 2.2 shows the contribution of this chapter in the context of the thesis.

PUBLICATION(S). The content of Section 2.2 has been published in [39]. In this paper the term 'side-information' is used for what is defined as 'helper data' in Chapter 1.

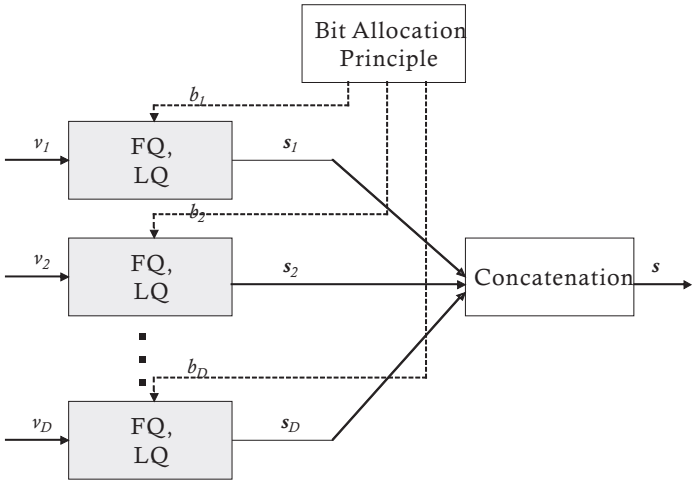


Figure 2.1: *Block diagram of a one-dimensional quantization and coding scheme, highlighted in FQ and LQ design. The $v_i, i = 1 \dots D$ denote D independent biometric features. Since bit allocation (in gray) is not discussed in this chapter, every feature is prescribe to a fixed length of b-bit. The quantized bits $s_i, i = 1 \dots D$ from all D features are then concatenated into the binary string s.*

2.2 Multi-bits biometric string generation based on the likelihood ratio

Abstract

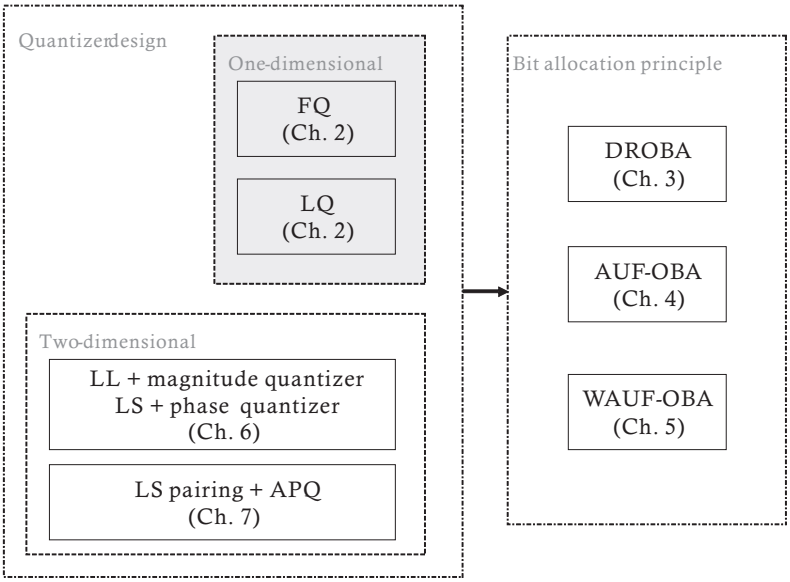


Figure 2.2: *Block diagram of the main contributions, highlighted in Chapter* 2.

Preserving the privacy of biometric information stored in biometric systems is becoming a key issue. An important element in privacy protecting biometric systems is the quantizer which transforms a normal biometric template into a binary string. In this paper, we present a user-specific quantization method based on a likelihood ratio approach (LQ). The bits generated from every feature are concatenated to form a fixed length binary string that can be hashed to protect its privacy. Experiments are carried out on both fingerprint data (FVC2000) and face data (FRGC). Results show that our proposed quantization method achieves a reasonably good performance in terms of FAR/FRR (when FAR is 10^{-4}, the corresponding FRR are 16.7% and 5.77% for FVC2000 and FRGC, respectively).

2.2.1 Introduction

Use of biometrics has brought considerable benefits in the area of access control and ICT security. Recently, however, protection of biometric template is becoming more important [40], because a biometric template may reveal personal information. Additionally, unprotected storage and transfer of biometric information allows direct steal-and-use impersonation. Once the biometric template is compromised, it can not be re-issued.

Biometric template protection aims to protect biometric reference information stored in biometric systems from abuse. In the past years, several techniques were developed to protect biometric information. In [19], [20] the authors discuss an approach known as 'cancelable biometrics'. Before storing the image of a face or a

fingerprint in a biometric system, it is distorted using a parametrized one-way geometric distortion function. The fuzzy vault method as introduced in [32] is a general cryptographic construction allowing to store a secret in a vault that can be locked using an unordered set of features. An initial attempt to use the fuzzy vault scheme in the setting of fingerprints is given in [31]. A third group of techniques, containing fuzzy commitments [21], fuzzy extractors [11] and helper data systems [24], derive a key from a biometric measurement and store an irreversibly hashed version of the key in the biometric system. It is the purpose of all these methods to protect the privacy of biometric information without reducing the performance of the biometric system in terms of False Acceptance Rate (*FAR*) and False Rejection Rate (*FRR*).

In this paper we will concentrate on the third group of methods. In order to extract a key, these methods assume that a biometric template can be represented as a fixed length binary string. In effect, these methods define the similarity of two binary templates in terms of Hamming distance [23]. A binary template is usually obtained by quantizing the original biometric template using a quantizer. In order to work properly, many quantizers produce and use side-information [24], [23], [22] that must be stored in the biometric system. Since this side-information is user dependent, it may leak information about the original template. Side-information with low privacy leakage is therefore a design objective.

So far, few quantization-based template methods have been proposed. Tuyls et al. [22] first introduced the fixed-interval quantization (FQ) with one bit per feature, in which two intervals are separated at the mean of the background distribution. However, they report an Equal Error Rate (*EER*) which is quite high (5.3%) when compared with the *EER* of a likelihood ratio classifier (LC) on the same data. Moreover, the one-bit per feature quantization generates only short binary strings which may be vulnerable to a brute force attack. Zhang et al. [9] introduced fixed interval quantization with multi-bits per feature (ZQ), in which the quantization intervals are determined by the mean and the standard deviation of the feature. However, the quantization method they proposed is not optimal in terms of *FAR* and *FRR*, and the security issue is not addressed by them.

Therefore, in this paper, we propose a user-specific, likelihood ratio based quantizer (LQ) that allows to extract multiple bits from a single feature. Experiments are carried out on both fingerprint data (FVC2000) and face data (FRGC). Results show that our proposed quantization method achieves a reasonably good performance in terms of *FAR*/*FRR* (when *FAR* is 10^{-4}, the corresponding *FRR* are 16.7% and 5.77% for FVC2000 and FRGC, respectively). In the mean time, the stored side-information retains high security.

In Section 2.2.2, our algorithm is presented. In Section 2.2.3, experiments on synthetic and real data are explained. In Section 2.2.4, we discuss the method while conclusions and directions for further research are given in Section 2.2.5.

2.2.2 Multi-bits quantization

The framework that we describe is similar to the Helper Data scheme proposed in [22]. It basically includes three parts: (1) extracting features; (2) quantization and

coding per feature and concatenating the output codes; (3) applying error correction coding (ECC) and hashing. However, in this paper, we propose a new approach for the first two items.

2.2.2.1 Extracting reliable, distinctive and independent features

One important step before applying quantization is to extract reliable, distinctive and independent features. In this paper our models assume Gaussian distributions and equal within-class variations. Therefore, a sufficient number of samples is required to provide reliable Gaussian parameters. Additionally, we require distinctive features, with small within-class variation and large between-class variation [41], to reduce quantization errors. Furthermore, we require features that are independent, with respect to both the background distributions and the genuine user distribution. Independent features can reduce the quantization error and subsequently generate independent bits. To extract features which meet the above requirements, we choose the PCA/LDA processing method described in [42].

2.2.2.2 Quantization and concatenation

The user-specific quantization is applied independently to each feature dimension, and the output codes are concatenated as the binary string. The idea of using likelihood ratio is driven by its optimal *FAR/FRR* performance in many biometric applications [43]. In a one-dimensional feature space $\mathbb{V}$ the likelihood ratio of user ω is defined as:

$$L_\omega = \frac{G(v, \mu_\omega, \sigma_\omega)}{G(v, \mu_0, \sigma_0)} \ , \tag{2.1}$$

where v, μ and σ are scalars. Due to the PCA/LDA processing, we have $G(v, \mu_0, \sigma_0)$ with $(\mu_0 = 0; \sigma_0 = 1)$ as the background probability density function (PDF) and $G(v, \mu_\omega, \sigma_\omega)$ as the genuine user PDF [43].

Fig. 2.3 shows an example of constructing a one-dimensional quantizer, given both probability density functions. By applying a threshold $t \in [0, \infty)$ to the likelihood ratio L_ω, a genuine quantization interval $Q_{\text{genuine},\omega}$ is determined in the feature space $\mathbb{V}$, in which the genuine user ω is assigned:

$$Q_{\text{genuine},\omega} = \{v \in \mathbb{V} \mid L_\omega \geq t\} \ . \tag{2.2}$$

With $Q_{\text{genuine},\omega}$, the probability P_ω for an impostor to be inside the genuine quantization interval can be calculated:

$$P_\omega = \int_{Q_{\text{genuine},\omega}} G(v, 0, 1) dv \ . \tag{2.3}$$

We construct the rest of the quantization intervals such that they have the same probability mass P_ω in the background distribution. This gives an attacker no additional information on which is the genuine interval. Furthermore, it can be seen that

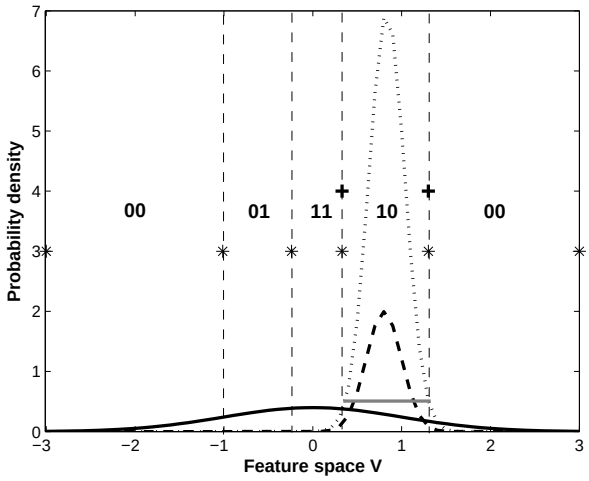


Figure 2.3: *An example of constructing a one-dimensional quantizer based on the likelihood ratio L_ω (dotted). The background PDF is $G(v, 0, 1)$ (solid), the genuine user PDF is $G(v, \mu_\omega = 0.8, \sigma_\omega = 0.2)$ (dashed), threshold t (grey). $+$ illustrates the genuine user interval, whilst $*$ illustrates the complete quantization intervals and the intervals are labeled with Gray code.*

this might lead to independent bits derived from a single feature. Thus we have:

$$\begin{aligned} \bigcup_{k=1}^{K_\omega} Q_{k,\omega} &= \mathbb{V} \ , \\ Q_{k,\omega} \bigcap Q_{l,\omega} &= \emptyset, k \neq l \ , \\ Q_{k,\omega} = Q_{\text{genuine},\omega}, &\text{ for certain } k \ , \\ \int_{Q_{k,\omega}} G(v, 0, 1) dv &= P_\omega \ , \end{aligned} \tag{2.4}$$

where K_ω is the number of quantization intervals and $Q_{k,\omega}$ is the quantization interval. In the following part, we will see that P_ω presented in (2.3) equals the FAR for a single feature.

Given an arbitrary t, it is not always possible to let each quantization interval have this P_ω probability. Usually the left-end and the right-end interval have a probability mass less than P_ω. Therefore, we address them as one wrap-around interval. In order to meet (2.4), only thresholds t that can generate

$$P_\omega = 1/K_\omega \ , \tag{2.5}$$

are applicable in our algorithm. Based on the above procedure, a K_ω-interval quantizer is established ($*$ in Fig. 2.3). Note that K_ω might not be an exponential of 2 and

it varies with different users. In most of the applications, we need to obtain a fixed code length L for all the users. For this reason, the code length need to be extended from $\log_2 K_\omega$ to L, $L = \lceil \log_2 K_\omega \rceil$.

Quantization intervals are labeled with a Gray code [44] which limits the Hamming distance of two adjacent code words to a single bit (see Fig. 2.3). This reduces the number of bit errors due to within-class variation.

Besides the binary code generated above, the quantizer information (known as side-information Q_ω) has to be stored for user ω as well. Since the background PDF is known, we only have to randomly select one quantization interval $(Q_{k,\omega} \mid k \in [1, K_\omega])$ as the side-information to be stored.

To extend the quantization to the m-dimensional case, we simply need to apply the above method to each feature dimension. The output binary string S_ω is a concatenation of binary codes corresponding to the genuine intervals of each dimension, and the side-information is the collection of quantizer information for each dimension.

2.2.2.3 FAR/FRR and security

Given a threshold t, the false acceptance rate $FAR_{i,\omega}(t)$ and false rejection rate $FRR_{i,\omega}(t)$ of user ω with the one-dimensional feature i is given by:

$$FAR_{i,\omega}(t_i) = \int_{Q_{\text{genuine},\omega}} G(v, 0, 1)dv \ , \tag{2.6}$$

$$FRR_{i,\omega}(t_i) = 1 - \int_{Q_{\text{genuine},\omega}} G(v, \mu_\omega, \sigma_\omega)dv \ . \tag{2.7}$$

Assuming that the PCA/LDA process results in independent features, the FAR and FRR in the m-dimensional feature space $\mathbb{V}^m$ for user ω, with the threshold vector $\mathbf{T} = [t_1 \dots t_m]$, is defined as:

$$FAR_\omega(\mathbf{T}) = \prod_{i=1}^{m} FAR_{i,\omega}(t_i) \ , \tag{2.8}$$

$$FRR_\omega(\mathbf{T}) = 1 - \prod_{i=1}^{m} (1 - FRR_{i,\omega}(t_i)) \ . \tag{2.9}$$

In a conventional biometric system, FAR represents the security at the real-valued biometric representation level. In our system, since we derive a binary string as the output representation, it is necessary to consider the security at the binary string level as well. Thus ideally the entropy of the output string $H(S_\omega)$ should be high, and the mutual information $I(S_\omega; Q_\omega)$ between the output binary string and the published side-information should be zero [22].

For one-dimensional feature i, given the number of quantization intervals $K_{i,\omega}$, the way to achieve a high binary string entropy and a mutual information zero is to build the quantization according to (2.4), which means an equal probability P_ω for each quantization interval. This requires a threshold t that gives $FAR_{i,\omega} = 1/K_{i,\omega}$. Under this condition, the binary string entropy $H_i(S_{i,\omega})$ and its relation with $FAR_{i,\omega}$

is given by (2.10). In our implementation, the wrap-around interval, with less than P_ω probability mass for each of the left-end and right-end interval, will never be a genuine interval. Due to this effect, the mutual information is (2.11).

$$H_i(S_{i,\omega}) = \log_2 K_{i,\omega} = -\log_2 FAR_{i,\omega} \ , \tag{2.10}$$

$$I_i(S_{i,\omega}; Q_{i,\omega}) = \log_2 K_{i,\omega} - \log_2 (K_{i,\omega} - 1) \ . \tag{2.11}$$

In the m-dimensional feature space $\mathbb{V}^m$, the m features are independent because of the PCA/LDA process. Hence, the binary string entropy and the mutual information becomes:

$$H = \sum_{i=1}^{m} H_i \ , \tag{2.12}$$

$$I = \sum_{i=1}^{m} I_i \ . \tag{2.13}$$

2.2.2.4 Optimization

A good biometric system requires low FAR_ω/FRR_ω with high H. A well-defined method is to construct a receiver operating characteristic (ROC) curve based on all possible m-dimensional FAR_ω and FRR_ω [9]. Every point on the ROC curve corresponds to a threshold vector $\mathbf{T}$. An optimal system can be found by minimizing the overall FRR_ω given the FAR_ω constraint:

$$\arg\min_{\mathbf{T}} \ (FRR_\omega(\mathbf{T})), \text{ given } FAR_\omega(\mathbf{T}) = \alpha \ . \tag{2.14}$$

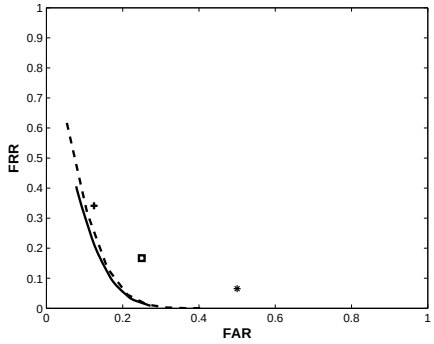
The above optimization procedure needs a full range of $\mathbf{T}$ vectors, while in our case, only some $\mathbf{T}$ vectors are acceptable according to requirement (2.5). To solve this problem, we proposed a sub-optimal method. We will explain the detail of this method in Section 2.2.3.

2.2.3 Experiments and results

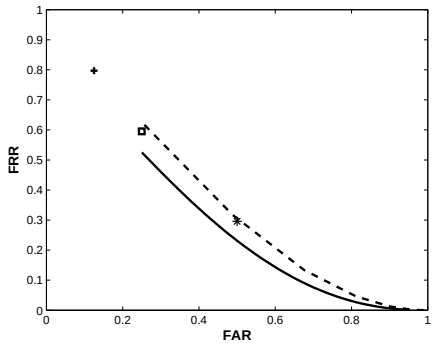
To examine the performance of this likelihood ratio based quantization method, we conducted experiments on both synthetic and real data sets.

2.2.3.1 Synthetic data experiments

We first carried out an experiment on the synthetic Gaussian data, with six methods: (1) likelihood ratio classifier (LC); (2) Zhang's multi-bits quantization (ZQ) [9]. In this method, each feature component is quantized with multiple intervals and each interval has the same fixed size ($k\sigma$), where σ denotes the standard deviation of the genuine user PDF; (3) fixed one-bit quantization (FQ1) [22]. In this method, each feature component is quantized with 2 fixed intervals which have equally 0.5 background probability mass; (4) fixed two-bits quantization (FQ2). In this method,



(a)



(b)

Figure 2.4: *One-dimensional simulation result: (a) Overall ROC with* $\sigma = 0.2$*; (b) Overall ROC with* $\sigma = 0.8$*. ZQ (dashed); LQ and LC (solid); FQ1 (*$*$*); FQ2 (*$\square$*); FQ3 (+).*

each feature component is quantized with 4 fixed intervals which have equally 0.25 background probability mass; (5) fixed three-bits quantization (FQ3). In this method, each feature component is quantized with 8 fixed intervals which have equally 0.125 background probability mass; (6) our likelihood ratio based multi-bits quantization (LQ).

We first performed a one-dimensional simulation on both a distinctive ($\sigma = 0.2$) and a non-distinctive ($\sigma = 0.8$) feature example. Fig. 2.4 shows the ROC performance of the overall user population. Our LQ method has the best FAR/FRR performance, the same as a likelihood ratio classifier. For fixed quantization FQ1, FQ2 and FQ3, it is not possible to tune any parameter, and their performance is worse than our LQ method. When the user within-class variation is small (e.g. $\sigma = 0.2$) , LQ has similar performance as ZQ, when the user within-class variation is large (e.g. $\sigma = 0.8$), LQ outperforms ZQ.

We applied the LQ method on two-dimensional synthetic data, based on the as-

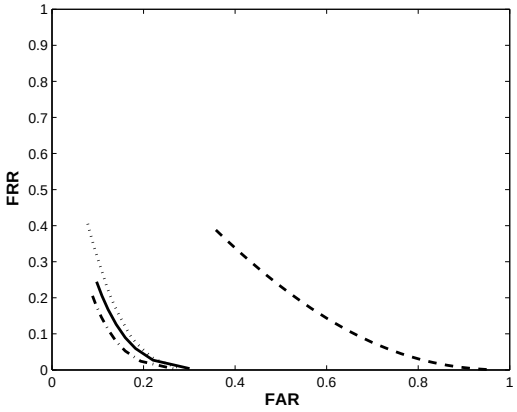


Figure 2.5: *Two-dimensional simulation result: ROC of the one-dimensional feature $\sigma_1 = 0.2$ (dotted); ROC of the one-dimensional feature $\sigma_2 = 0.8$ (dashed); ROC of the two-dimensional features $\sigma_1 = 0.2$ and $\sigma_2 = 0.8$ from LQ (solid); ROC of the same two-dimensional features from LC (dash-dotted).*

sumption that the user within-class variance for the first two dimensions was $\sigma_1 = 0.2$ and $\sigma_2 = 0.8$ respectively. The optimal ROC curve was constructed by the process described in Section 2.2.2. Fig. 2.5 plots the two-dimensional overall ROC performance, and it suggests that the combined ROC curve constructed from our LQ method does not introduce a large degradation compared to the performance of LC.

2.2.3.2 Real data experiments

The real data experiments were conducted on two data sets: a fingerprint data set FVC2000 (DB2) [35], [36] and a face data set FRGC (version 1) [38]. Both data sets were extracted into fixed length feature vectors.

- **FVC2000(DB2)**: This fingerprint data set contains 8 images of 110 different users. The original feature vector length extracted from the image was 1536 [22]. Features include the squared directional field and the Gabor response.
- **FRGC(ver1)**: This face data set contains variable images of 275 different users. The images were taken under controlled conditions and they were aligned using manually labeled landmarks. The original feature vector length extracted from the image was 8762. Features are the grey value of the face images.

The experiments consist of three steps: training, enrollment and verification. During the initial off-line training step, PCA/LDA was applied on the training data to reduce the feature dimension. Afterwards, an enrollment step was conducted in which the quantizers were constructed based on the enrollment data, in particular the means of the features after dimensionality reduction. The output reference binary string and the side-information were stored. In the verification step, verification

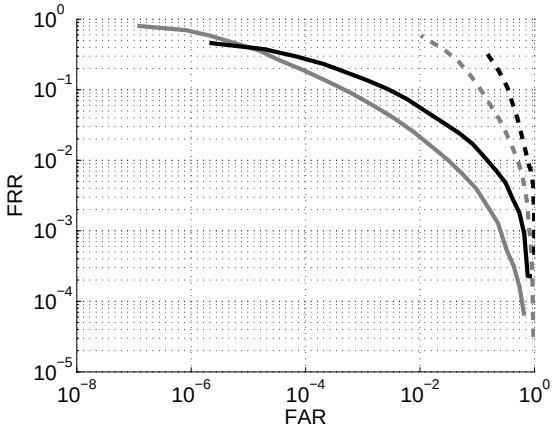


Figure 2.6: *Results of PCA/LDA feature extraction compared to the reliable bits selection method on FVC2000 (black) and FRGC (grey) (feature dimension for PCA is* 100 *and feature dimension for LDA is* 50*). Reliable bits selection method (dashed); PCA/LDA/FQ1 method (solid).*

data were quantized based on the quantizer side-information, and the output query string was compared to the reference string for the final decision. To split the data, 75% (FVC2000) and 50% (FRGC) of the samples per user were used for both training and enrollment, and the rest 25% (FVC2000) and 50% (FRGC) of the samples were used for verification. For both data sets, we extracted 50 features from their original measurements. To compare the query and the reference binary strings, we applied a Hamming distance classifier, in which the Hamming distance represents the number of different bits between the enrollment and verification binary string. The Hamming distance classifier replaces the ECC present in many template protection methods (e.g. [22]). Assigning a threshold D to the distance has the same effect as applying an ECC that can correct at most D bits. By varying the threshold D, a ROC curve on the verification data can be constructed. To obtain a reasonable error on the results, we repeated the above procedure with 20 random splits of enrollment and verification data.

We conducted two types of experiments. In the first experiment, we examined the feature extraction performance via the PCA/LDA process, followed by the FQ1 quantization. The result was compared to the reliable bits selection method proposed in [22], in which the output binary strings are selected directly from the original feature measurements, with a pre-selection based on the reliability of FQ1 results on each enrollment sample and a selection based on the ratio of within-class variation and between-class variation. Fig. 2.6 plots the log-ROC curves derived from both PCA/LDA method and reliable bits selection method. For both FVC2000 and FRGC, the performance increases dramatically with PCA/LDA. Such result suggests that features extracted from PCA/LDA method are more reliable and distinctive, which

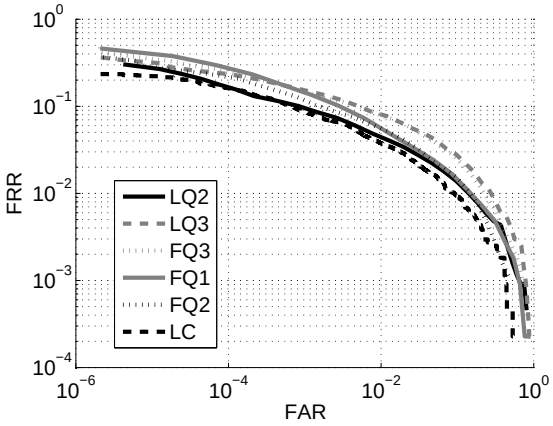


Figure 2.7: *Log-ROC curve of the fingerprint FVC2000 data.*

provides a crucial precondition for the upcoming quantization step.

In the second experiment, we examined the different quantization performances. To do a high-dimensional quantization experiment, we need to construct a ROC curve for high-dimensional features, but the optimization method described by (2.14) in Section 2.2.2 is not feasible and constructing an optimal ROC curve is a point of further research. However, since a fixed length binary string as output is often preferred, we propose an alternative sub-optimal LQn method. The core idea is to quantize each feature dimension into n bits, which also means that the FAR per dimension is fixed to 2^{-n}. As a result, the output string will have a fixed length.

We performed the experiments of LQ2 ($n = 2$) and LQ3 ($n = 3$) on both data sets, followed by the three-step procedure described above. The feature dimension after feature extraction was set to 50. Consequently, each user ended up with 100 and 150 bit string. (Note that the above likelihood ratio based quantization is user customized, which means each user has his own optimized quantization configuration.) Afterwards, we compared the LQ2 and LQ3 performance with FQ1, FQ2, FQ3 and LC methods.

Fig. 2.7 and 2.8 show the ROC plots for FVC2000 and FRGC data sets. It can be seen that results from all the methods are consistent on both data sets. LC is superior to all the quantization based methods. Apparently, FQ1, FQ3 and LQ3 do not provide comparable performance to LQ2 and FQ2. Compared to LQ2, FQ2 has a slightly worse performance. That means LQ2 consistently outperforms all the quantization methods, and its performance is not significantly degraded compared to the LC result. Table 2.1 lists the performance of LQ2 under different FAR/FRR requirements, compared to the LC performance. For a reasonable application requiring $FAR = 10^{-4}$, the corresponding FRR are 16.7% (FVC2000) and 5.77% (FRGC) respectively, which is acceptable as compared to the performance of the LC classifier. The Hamming distance threshold needed to achieve such system performance is 29

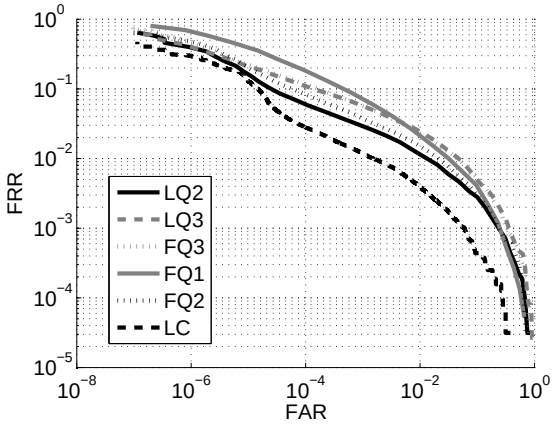


Figure 2.8: *Log-ROC curve of the face FRGC data.*

Table 2.1: *The performance of LC, LQ2 under different system requirements*

	$FAR = 10^{-2}$		$FAR = 10^{-3}$		$FAR = 10^{-4}$	
	FRR	D	*FRR*	D	*FRR*	D
FVC2000-LC	3.8%	N/A	8.7%	N/A	16.2%	N/A
FVC2000-LQ2	4.3%	37	8.7%	33	16.7%	29
FRGC-LC	0.41%	N/A	1.20%	N/A	2.80%	N/A
FRGC-LQ2	1.03%	37	2.60%	33	5.77%	29

from 100 bits for both data sets.

Now we analyze the security of the output binary string. Under the assumption of independent features, the output average string entropy for FQ2, LQ2, FQ3 and LQ3 are 100, 100, 150 and 150 respectively. However, in practice these numbers will be lower due to dependency of the individual features. The mutual information I between the output binary string and the side-information is zero for the FQ method, but not zero for our LQ method. For instance, the mutual information for LQ2 is 0.415 bit per feature component. This can be viewed as a sacrifice of security since we introduced more user-specific information in the LQ quantization.

2.2.4 Discussion

The performance of the quantization methods is affected by two factors: the quality of the features and the quantization interval size. In our case, the quality of the features is defined as the within-class variation of each feature component after the PCA/LDA process, and the quantization interval size is driven by the number of quantization bits per feature dimension: quantization into 1 bit per feature (FQ1); quantization into 2 bits per feature (FQ2/LQ2) and quantization into 3 bits per feature (FQ3/LQ3). An

investigation on the within-class variation of the feature components after PCA/LDA process demonstrates that for both FVC2000 and FRGC data sets, the within-class variance of the 50 features range from 0.14^2 to 0.60^2. If FQ1 is applied, which has relatively large quantization intervals compared to the feature variation, the *FRR* per feature dimension is low. However, in this case the *FAR* of 0.5 per dimension is quite high. This results also in a high *FAR* in the high dimensional experiment (2.8). If FQ3 and LQ3 are applied, which have relatively small quantization intervals compared to feature variation, the *FAR* reduces to 0.125 per feature dimension. In contrast, the *FRR* per feature dimension will be high. This results in a high *FRR* in the high dimensional experiment (2.9). Therefore, FQ2 and LQ2 turn out to be a good compromise with respect to the *FAR*/*FRR* requirements. This explains why in Fig. 2.7 and Fig. 2.8, LQ2 and FQ2 outperforms FQ1, FQ3 and LQ3.

2.2.5 Conclusions

In this paper we discussed the problem of transforming biometric feature vectors into binary strings which are to be used in recently introduced methods for privacy protection of biometric information. We proposed to pre-process the feature vectors using a PCA/LDA transformation followed by a quantizer based on a likelihood ratio approach. Depending on the setting, our quantizer allows to extract multiple bits from a single feature. Comparison of our approach with a number of quantizers known from the literature, using both synthetic and real-life data, shows that the likelihood quantizer outperforms the other quantizers. Moreover, its performance is not significantly degraded as compared to a traditional likelihood classifier.

In our current experiments we extracted the same number of bits for every feature. In practice, however, not all features are equally distinctive. Therefore, an adaptive coding method, in which more bits are assigned to distinctive features and less bits to non-distinctive features, is a point of future research.

2.3 Chapter conclusion

In this chapter, one-dimensional quantizers FQ and LQ are presented. Regarding the research objectives, both quantizers are capable of extracting multiple *i.i.d.* bits. Compared to FQ, LQ extracts more reliable bits of a prescribed length, and thus optimizes the FAR and the FRR performances for every feature. Furthermore, with more reliable bits extracted from every feature, the length of the random key K can be increased.

3

Detection Rate Optimized Bit Allocation

3.1 Chapter introduction

PURPOSE. In Chapter 2, we aimed to design one-dimensional quantizers, given a prescribed fixed number of quantization bits per biometric feature. For example, the likelihood ratio based quantizer (LQ) determines the quantization intervals, so as to maximize the detection rate at a prescribed number of quantization bits. However, even with optimized quantization intervals per feature, the overall recognition performance of the entire binary strings is not yet optimal, because the features are all quantized with an equal number of bits. Given the same total length of the binary string, it is desirable to extract more bits from more discriminative features and fewer bits from less discriminative features. Therefore, the purpose of this chapter is to design such an adaptive bit allocation principle. This bit allocation principle should be independent of the quantizers, which means it can be applied together with any types of quantizer, e.g. one-dimensional quantizers like FQ and LQ, or two-dimensional quantizers. Given independent features and quantizers that can extract statistically independent and identically distributed (*i.i.d.*) bits, the bit allocation principle should preserve the *i.i.d.* bits property. When applied to a Hamming distance classifier (HDC), the binary strings should result in good recognition performance.

CONTENTS. A detection rate optimized bit allocation (DROBA) principle is presented in this chapter. Independent of the actual quantization intervals, DROBA can be applied to both one- and two-dimensional quantization schemes. In this chapter, it is presented in combination with the one-dimensional quantizers FQ and LQ, as illustrated in Fig. 3.1. Given any chosen type of quantizer, for every feature

of an enrolled user, the detection rates at zero Hamming distance is computed for a range of allowed allocated bits. These detection rates can be theoretically computed based on the modeling of the genuine user probability density function, or as an approximated value. Given these detection rates for every feature, DROBA aims to maximize the overall theoretical detection rate of the binary strings, subject to a fixed total number of bits. A dynamic programming or a greedy search approach is then applied to search for the optimal solution. As a result, DROBA assigns more bits to more discriminative features and fewer bits to less discriminative features. Essentially, DROBA optimizes the theoretical overall detection rate for the HDC, when the Hamming distance threshold is zero. Figure 3.2 shows the contribution of this chapter in the context of the thesis.

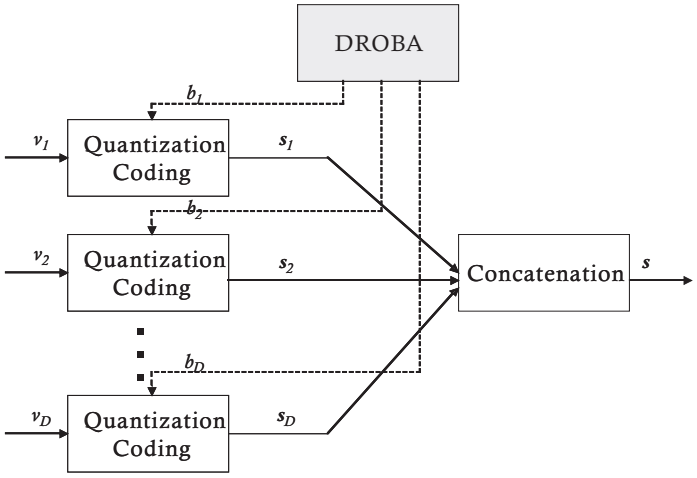


Figure 3.1: *Block diagram of the one-dimensional quantization and coding scheme, highlighted in DROBA design. The $v_i, i = 1 \dots D$ denote D independent biometric features, and b_i denotes the assigned number of bits to the i^{th} feature. The quantized bits $s_i, i = 1 \dots D$ from all D features are then concatenated into the binary string s.*

PUBLICATION(S). The content of Section 3.2 has been published in [45].

3.2 Biometric quantization through detection rate optimized bit allocation

Abstract

Extracting binary strings from real-valued biometric templates is a fundamental step in many biometric template protection systems, such as fuzzy commitment, fuzzy extractor, secure sketch and helper data systems. Previous work has been focusing

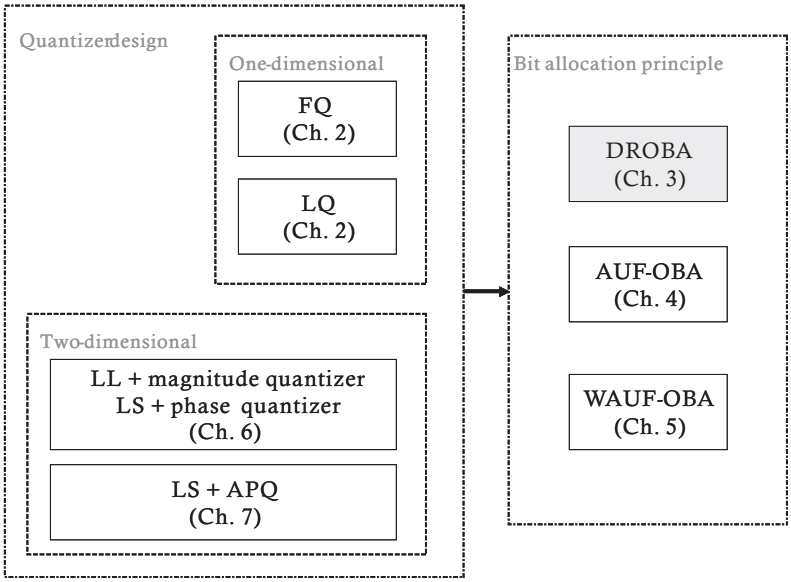


Figure 3.2: *Block diagram of the main contributions, highlighted in Chapter* 3.

on the design of optimal quantization and coding for each single feature component, yet the binary string – concatenation of all coded feature components – is not optimal. In this paper, we present a detection rate optimized bit allocation principle (DROBA), which assigns more bits to discriminative features and fewer bits to non-discriminative features. We further propose a dynamic programming approach (DP) and a greedy search approach (GS) to achieve DROBA. Experiments of DROBA on the FVC2000 fingerprint database and the FRGC face database show good performances. As a universal method, DROBA is applicable to arbitrary biometric modalities, such as fingerprint texture, iris, signature and face. DROBA will bring significant benefits not only to the template protection systems, but also systems with fast matching requirements or constrained storage capability.

3.2.1 Introduction

The idea of extracting binary biometric strings was originally motivated by the increasing concern about biometric template protection [40]: Some proposed systems, such as fuzzy commitment [21], fuzzy extractor [11], [13], secure sketch [26] and helper data systems [24], [22], [23], [25], employ a binary biometric representation. Thus, the quality of the binary string is crucial to their performances. Apart from the template protection perspective, binary biometrics also merit fast matching and compressed storage, facilitating a variety of applications utilizing low-cost storage media. Therefore, extracting binary biometric strings is of great significance. As shown in Fig. 3.3, a biometric system with binary representation can be generalized into the following

three modules.

Figure 3.3: *Three modules of a biometric system with binary representation.*

Feature extraction: This module aims to extract independent, reliable and discriminative features from biometric raw measurements. Classical techniques used in this step are, among others, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [42].
Bit extraction: This module aims to transform the real-valued features into a fixed-length binary string. 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. In other words, both a low false acceptance rate (FAR) and a low false rejection rate (FRR) are required. Additionally, from the template protection perspective, the bits, generated from an imposter, should be statistically independent and identically distributed (*i.i.d.*), in order to maximize the effort of an imposter in guessing the genuine template. Presumably, the real-valued features obtained from the feature extraction step are independent, reliable and discriminative. Therefore, a quantization and coding method is needed to keep such properties in the binary domain. So far, a variety of such methods have been published, of which an overview will be given in Section 3.2.2.
Binary string classification: This module aims to verify the binary strings with a binary string based classifier. For instance, the Hamming distance classifier which bases its decision on the number of errors between two strings. Alternatively, the binary strings can be verified through a template protection process, e.g. fuzzy commitment [21], fuzzy extractor [11], [13], secure sketch [26] and helper data systems [24], [22], [23], [25]. Encrypting the binary strings by using a one-way function, these template protection systems verify binary strings in the encrypted domain. Usually the quantization methods in the bit extraction module can not completely eliminate the intra-class variation. Thus employing a strict one-way function will result in a high FRR. To solve this problem, error correcting techniques are integrated to further eliminate the intra-class variation in the binary domain. Furthermore, randomness is embedded to avoid cross-matching.

This paper deals with the bit extraction module, for which we present a detection rate optimized bit allocation principle (DROBA) that transforms a real-valued biometric template into a fixed-length binary string. Binary strings generated by DROBA yield a good FAR and FRR performance when evaluated with a Hamming distance classifier.

In Section 3.2.2 an overview is given of known bit extraction methods. In Section 3.2.3 we present the DROBA principle with two realization approaches: dynamic programming (DP) and greedy search (GS), and their simulation results are illustrated in Section 3.2.4. In Section 3.2.5, we give the experimental results of DROBA on the

FVC2000 fingerprint database [35] and the FRGC face database [37]. In Section 3.2.6 the results are discussed and conclusions are drawn in Section 3.2.7.

3.2.2 Overview of bit extraction methods

A number of bit extraction methods, based on quantization and coding, have been proposed in biometric applications [24], [22], [23], [8], [46], [9], [39]. In general these methods deal with two problems: (1) how to design an optimal quantization and coding method for a single feature, and (2) how to compose an optimal binary string from all the features.

So far, most of the published work has been focusing on designing the optimal quantization intervals for individual features. It is known that, due to the inter- and intra-class variation, every single feature can be modeled by a background probability density function (PDF) p_{b} and a genuine user PDF p_{g}, indicating the probability density of the whole population and the genuine user, respectively. Given these two PDFs, the quantization performance of a single feature i, with an arbitrary b_i-bit quantizer, is then quantified as the theoretical FAR α_i:

$$\alpha_i(b_i) = \int_{Q_{\mathrm{genuine},i}(b_i)} p_{\mathrm{b},i}(v)dv \ , \tag{3.1}$$

and FRR β_i, given by:

$$\delta_i(b_i) = \int_{Q_{\mathrm{genuine},i}(b_i)} p_{\mathrm{g},i}(v)dv \ , \tag{3.2}$$

$$\beta_i(b_i) = 1 - \delta_i(b_i) \ , \tag{3.3}$$

where $Q_{\mathrm{genuine},i}$ represents the genuine user interval into which the genuine user is expected to fall, and δ_i represents the corresponding detection rate. An illustration of these expressions is given in Fig. 3.4. Hence, designing quantizers for a single feature is to optimize its FAR (3.1) and FRR (3.3).

Linnartz et al. proposed a method inspired by Quantization Index Modulation [24]. As depicted in Fig. 3.5(a), the domain of the feature v is split into fixed intervals of width q. Every interval is alternately labeled using a '0' or a '1'. Given a random bit string s, a single bit of s is embedded per feature by generating an offset for v so that v ends up in the closest interval that has the same label as the bit to be embedded.

Vielhauer et al. [8] introduced a user-specific quantizer. As depicted in Fig. 3.5(b), the genuine interval $[I_{\min}(1-t), I_{\max}(1+t)]$ is determined according to the minimum $I_{\min}$ and maximum $I_{\max}$ value of the samples from the genuine user, together with a tolerance parameter t. The remaining intervals are constructed with the same width as the genuine interval.

Hao and Wah [46] and Chang et al. [9] employed a user-specific quantizer as shown in Fig. 3.5(c). The genuine interval is $[\mu - k\sigma, \mu + k\sigma]$, where μ and σ are the mean and the standard deviation of the genuine user PDF, and k is an optimization parameter. The remaining intervals are constructed with the same width $2k\sigma$.

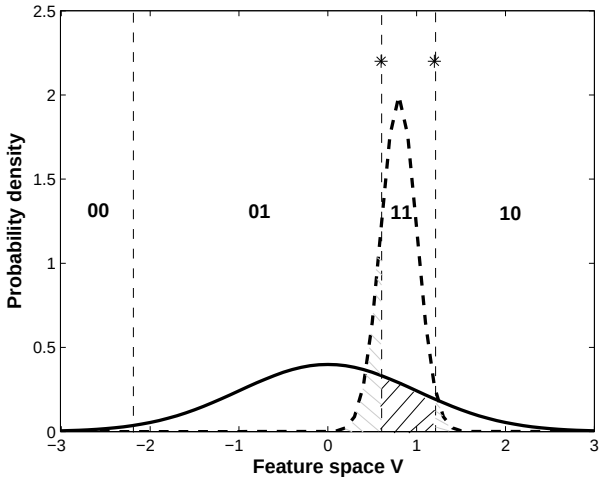


Figure 3.4: *An illustration of the FAR (black) and the FRR (gray), given the background PDF (solid), the genuine user PDF (dot), and the quantization intervals (dash), where the genuine user interval is marked as* $*$.

The quantizers in [24], [8], [46] and [9] have equal-width-intervals. However, considering a template protection application, this leads to potential threats, because samples tend to have higher probabilities in some quantization intervals and thus an imposter can search the genuine interval by guessing the one with the highest probability. Therefore, quantizers with equal-probability-intervals or equal-frequency-intervals [22], [39] have been proposed.

Tuyls et al. [22] and Teoh et al. [15] employed a 1-bit fixed quantizer as shown in Fig. 3.5(d). Independent of the genuine user PDF, this quantizer splits the domain of the feature v into two fixed intervals using the mean of the background PDF as the quantization boundary. As a result, both intervals contain 0.5 background probability mass. The interval that the genuine user is expected to fall into is referred to as the genuine interval.

Chen et al. [39] extended the 1-bit fixed quantizer into multi-bits. A b-bit fixed quantizer contains 2^b intervals, symmetrically constructed around the mean of the background PDF, with equally 2^{-b} background probability mass. Fig. 3.5(e) illustrates an example of $b = 2$. In the same paper [39], a user-specific likelihood ratio based multi-bits quantizer was introduced, as shown in Fig. 3.5(f). For a b-bit quantizer, a likelihood ratio threshold first determines a genuine interval with 2^{-b} background probability mass. The remaining intervals are then constructed with equal 2^{-b} background probability mass. The left- and right-tail are combined as one wrap-around interval, excluding its possibility as a genuine interval. The likelihood ratio based quantizer provides the optimal FAR and FRR performances in the Neyman-Pearson sense.

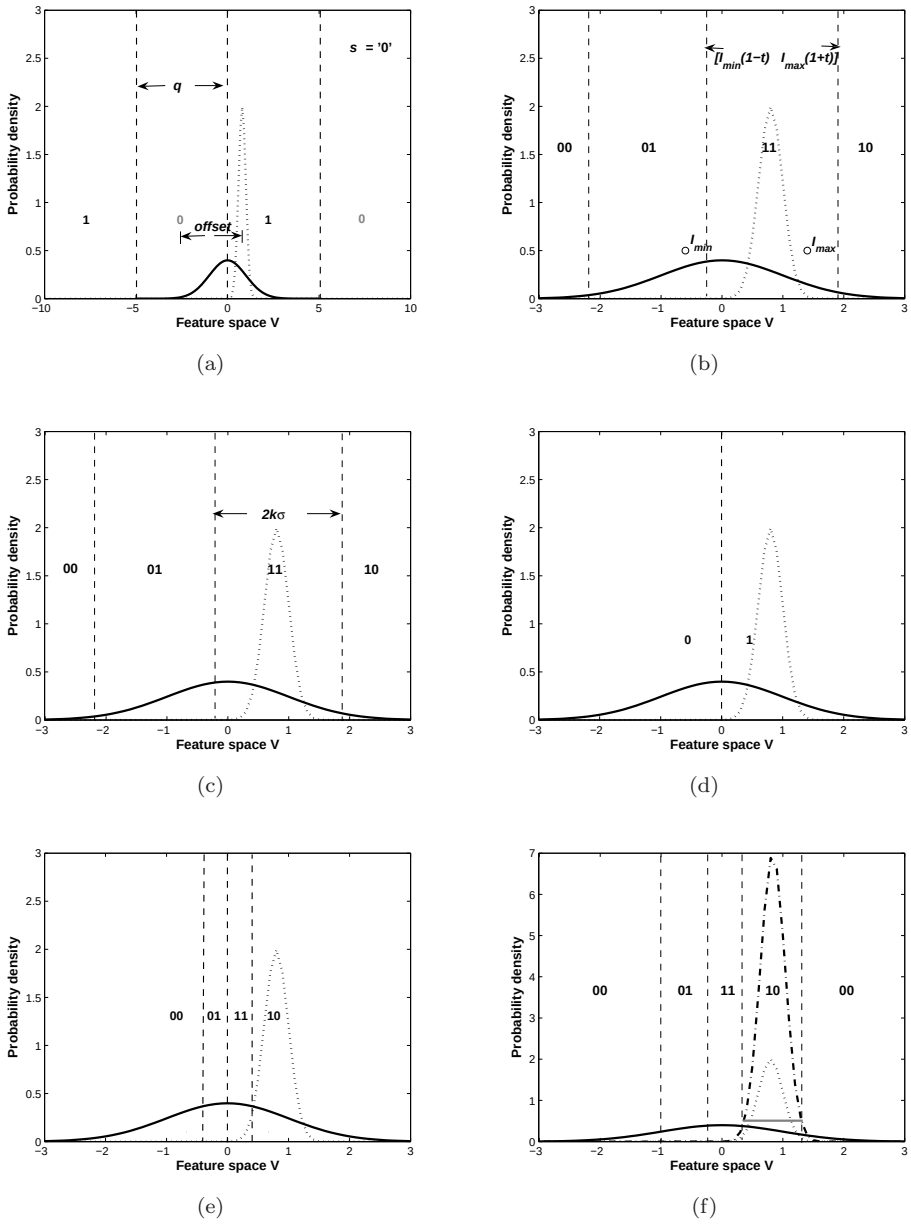


Figure 3.5: *Illustration of the quantizers for a single feature i, and the corresponding Gray codes. The background PDF $p_{\mathrm{b}}(v, 0, 1)$ (solid); the genuine user PDF $p_{\mathrm{g}}(v, \mu, \sigma)$ (dot); the quantization intervals (dash). (a) QIM quantization; (b) Vielhauer's quantizer; (c) Chang's multi-bits quantizer; (d) Fixed one-bit quantizer; (e) Fixed two-bits quantizer; (f) Likelihood ratio based quantizer, the likelihood ratio (dash-dot), threshold (gray).*

The equal-probability-intervals in both the fixed quantizer and the likelihood ratio based quantizer ensure independent and identically distributed bits for the imposters, which meets the requirement of template protection systems. For this reason, we take these two quantizers into considerations in the following sections. Additionally, because of the equal-probability-intervals, the FAR of both quantizers for feature i becomes:

$$\alpha_i(b_i) = 2^{-b_i} \ . \tag{3.4}$$

With regard to composing the optimal binary string from D features, the performance of the entire binary string can be quantified by the theoretical overall FAR α and detection rate δ:

$$\alpha(b_1, \ldots, b_D) = \prod_{i=1}^{D} \alpha_i(b_i) \ , \tag{3.5}$$

$$\delta(b_1, \ldots, b_D) = \prod_{i=1}^{D} \delta_i(b_i) \ , \quad \sum_{i=1}^{D} b_i = L \ . \tag{3.6}$$

Given (3.4), the overall FAR in (3.5) shows a fixed relationship with L:

$$\alpha(b_1, \ldots, b_D) = 2^{-L} \ . \tag{3.7}$$

Hence composing the optimal binary string is to optimize the detection rate at a given FAR value. In [22], [23] and [39], a fixed bit allocation principle (FBA) – with a fixed number of bits assigned to each feature – was proposed. Obviously, the overall detection rate of the FBA is not optimal, since we would expect to assign more bits to discriminative features and fewer bits to non-discriminative features. Therefore, in the next section, we propose the DROBA principle, which gives the optimal overall detection rate.

3.2.3 Detection rate optimized bit allocation (DROBA)

In this section, we first give the description of the DROBA principle. Furthermore, we introduce both a dynamic programming and a greedy search approach to search for the solution.

3.2.3.1 Problem Formulation

Let D denote the number of features to be quantized; L, the specified binary string length; $b_i \in \{0, \ldots, b_{\max}\}$, $i = 1, \ldots, D$, the number of bits assigned to feature i; and $\delta_i(b_i)$, the detection rate of feature i, respectively. Assuming that all the D features are independent, our goal is to find a bit assignment $\{b_i^*\}$ that maximizes the overall detection rate in (3.6):

$$\begin{aligned} \{b_i^*\} &= \arg \max_{\sum_{i=1}^{D} b_i = L} \delta(b_1, \ldots, b_D) \\ &= \arg \max_{\sum_{i=1}^{D} b_i = L} \prod_{i=1}^{D} \delta_i(b_i) \ . \end{aligned} \tag{3.8}$$

Note that by maximizing the overall detection rate, we in fact maximize the probability of all the features simultaneously staying in the genuine intervals, more precisely, the probability of a zero bit error for the genuine user. Furthermore, considering using a binary string classifier, essentially the overall FAR α in (3.5) and the overall detection rate δ in (3.6) correspond to the point with the minimum FAR and minimum detection rate on its theoretical receiver operating characteristic curve (ROC), as illustrated in Fig. 3.6. We know that α is fixed in (3.7), by maximizing δ, DROBA in fact provides a theoretical maximum lower bound for the ROC curve. Since DROBA only maximizes the point with minimum detection rate, the rest of the ROC curve, which relies on the specific binary string classifier, is not yet optimized. However, we would expect that with the maximum lower bound, the overall ROC performance of any binary string classifier is to some extent optimized.

The optimization problem in (3.8) can be solved by a brute force search of all possible bit assignments $\{b_i\}$ mapping D features into L bits. However, the computational complexity is extremely high. Therefore, we propose a dynamic programming approach with reasonable computational complexity. To further reduce the computational complexity, we also propose a greedy search approach, for which the optimal solution is achieved under additional requirements to the quantizer.

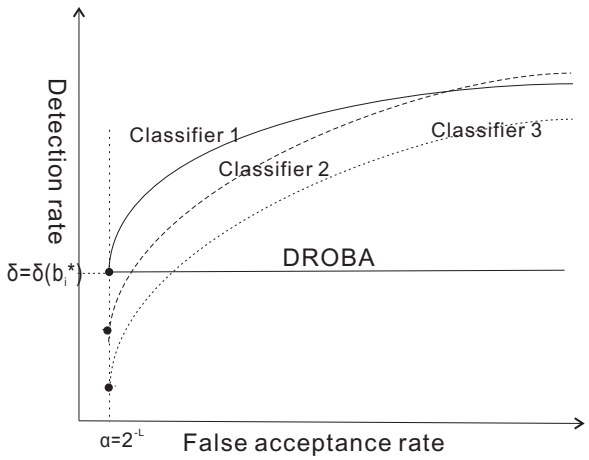


Figure 3.6: *Illustration of the maximum lower bound for the theoretical ROC curve provided by DROBA.*

3.2.3.2 Dynamic Programming Approach (DP)

The procedure to search for the optimal solution for a genuine user is recursive. That is, given the optimal overall detection rates $\delta^{(j-1)}(l)$ for $j-1$ features at string length

$l, l = 0, \ldots, (j-1) \times b_{\max}$:

$$\delta^{(j-1)}(l) = \max_{\sum b_i = l,\ b_i \in \{0,\ldots,b_{\max}\}} \prod_{i=1}^{j-1} \delta_i(b_i) \ , \tag{3.9}$$

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

$$\delta^{(j)}(l) = \max_{\substack{b' + b'' = l, \\ b' \in \{0,\ldots,(j-1)\times b_{\max}\}, \\ b'' \in \{0,\ldots,b_{\max}\}}} \delta^{(j-1)}(b')\delta_j(b'') \ , \tag{3.10}$$

for $l = 0, \ldots, j \times b_{\max}$. Note that $\delta^{(j)}(l)$ needs to be computed for all string lengths $l \in \{0, \ldots, j \times b_{\max}\}$. Eq. (3.10) tells 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 jth 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 $\{b_i(l)\}$, $i = 1, \ldots, j$ denote the optimal bit assignments for j features at binary string length l such that the ith entry corresponds to the ith feature. Note that the sum of all entries in $\{b_i(l)\}$ equals l, i.e. $\sum_{i=1}^{j} b_i(l) = l$. If $\hat{b}'$ and $\hat{b}''$ denote the values of b' and b'' that correspond to the maximum value $\delta^{(j)}(l)$ in (3.10), the optimal assignments are updated by:

$$b_i(l) = b_i(\hat{b}'), \quad i = 1, \ldots, j-1 \ , \tag{3.11}$$

$$b_j(l) = \hat{b}'' \ . \tag{3.12}$$

The iteration procedure is initialized with $j = 0$, $b_0(0) = 0$, and $\delta^{(0)}(0) = 1$ and terminated when $j = D$. After D iterations, we obtain a set of optimal bit assignments for every possible bit length $l = 0, \ldots, D \times b_{\max}$, we only need to pick the one that corresponds to L: The final solution $\{b_i^*\} = \{b_i(L)\}, i = 1, \ldots, D$. This iteration procedure can be formalized into a dynamic programming approach [47], as described in Algorithm 1.

Essentially, given L and arbitrary $\delta_i(b_i)$, the dynamic programming approach optimizes (3.8). The proof of its optimality is presented in Appendix A. This approach is independent of the specific type of the quantizer, which determines the behavior of $\delta_i(b_i)$. The user-specific optimal solution $\{b_i^*\}$ is 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. However, this approach becomes inefficient if $L \ll D \times b_{\max}$, because a D-fold iteration is always needed, regardless of L.

3.2.3.3 Greedy Search Approach (GS)

To further reduce the computational complexity, we introduce a greedy search approach. By taking the logarithm of the detection rate, the optimization problem in

Algorithm 1 Dynamic programming approach for DROBA

Input:

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

Initialize:

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

while $j < D$ **do**

$$\begin{aligned} j &= j + 1\ , \\ \hat{b}', \hat{b}'' &= \arg\max \delta^{(j-1)}(b')\delta_j(b'')\ , \\ &\qquad \substack{b' + b'' = l, \\ b' \in \{0, \ldots, (j-1) \times b_{\max}\}, \\ b'' \in \{0, \ldots, b_{\max}\},} \\ \delta^{(j)}(l) &= \delta^{(j-1)}(\hat{b}')\delta_j(\hat{b}'')\ , \\ b_i(l) &= b_i(\hat{b}'), i = 1, \ldots, j-1\ , \\ b_j(l) &= \hat{b}''\ , \\ &\quad \text{for } l = 0, \ldots, j \times b_{\max}\ , \end{aligned}$$

end while
Output:

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

(3.8) is now equivalent to finding a bit assignment $\{b_i^*\}$, $i = 1, \ldots, D$ that maximize:

$$\sum_{i=1}^{D} \log(\delta_i(b_i)) \ , \tag{3.13}$$

under the constraint of a total number of L bits. In [48], an equivalent problem of minimizing quantizer distortion, given an upper bound to the bit rate, is solved by first rewriting it as an unconstrained Lagrange minimization problem. Thus in our case we define the unconstrained Lagrange maximization problem as:

$$\max_{b_i, \lambda \geq 0} \left[\sum_{i=1}^{D} \log(\delta_i(b_i)) - \lambda \sum_{i=1}^{D} b_i \right] . \tag{3.14}$$

We know that the detection rate of a feature is monotonically non-increasing with the number of quantization bits. Therefore, we can construct an L-bit binary string, by iteratively assigning an extra bit to the feature that gives the minimum detection rate loss, as seen in Algorithm 2. Suppose $\{b_i(l)\}, i = 1, \ldots, D$, gives the bit assignments of all D features at binary string length l, we compute $\Delta_i(l)$ for each feature, representing the loss of the log detection rate by assigning one more bit to that feature:

$$\Delta_i(l) = \log(\delta_i(b_i(l))) - \log(\delta_i(b_i(l) + 1)), i = 1, \ldots, D \ . \tag{3.15}$$

Hence the extra bit that we select to construct the $(l + 1)$-bit binary string comes from the feature $i_{\max}$ that gives the minimum detection rate loss, and no extra bit are assigned to the unchosen feature components:

$$i_{\max} = \arg\min_i \Delta_i(l) \ , \tag{3.16}$$

$$b_i(l+1) = \begin{cases} b_i(l) + 1, & i = i_{\max} \ , \\ b_i(l), & \text{otherwise} \ . \end{cases} \tag{3.17}$$

The iteration is initialized with $l = 0$, $b_i(0) = 0$, $\log(\delta_i(b_i(0))) = 0$, $i = 1, \ldots, D$ and terminated when $l = L$. The final solution is $\{b_i^*\} = \{b_i(L)\}, i = 1, \ldots, D$.

To ensure the optimal solution of this greedy search approach, the quantizer has to satisfy the following two conditions:

1. $\log(\delta_i)$ is a monotonically non-increasing function of b_i.

2. $\log(\delta_i)$ is a concave function of b_i.

The number of operations of the greedy search is about $\mathrm{O}(L \times D)$, which is related with L. Compared with the dynamic programming approach with $O(D^2 \times b_{\max}^2)$, greedy search becomes significantly more efficient if $L \ll D \times b_{\max}^2$, because only an L-fold iteration needs to be conducted.

The DROBA principle provides the bit assignment $\{b_i^*\}$, indicating the number of quantization bits for every single feature. The final binary string for a genuine user is the concatenation of the quantization and coding output under $\{b_i^*\}$.

Algorithm 2 Greedy search approach for DROBA

Input:

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

Initialize:

$$\begin{aligned} l &= 0\ , \\ b_i(0) &= 0\ , \\ \log(\delta_i(b_i(0))) &= 0\ , \end{aligned}$$

while $l < L$ **do**

$$\begin{aligned} \Delta_i(l) &= \log(\delta_i(b_i(l))) - \log(\delta_i(b_i(l)+1))\ , \\ i_{\max} &= \arg\min_i \Delta_i(l)\ , \\ b_i(l+1) &= \begin{cases} b_i(l)+1, & i = i_{\max}\ , \\ b_i(l), & \text{otherwise}\ , \end{cases} \\ l &= l+1\ , i = 1, \ldots, D\ , \end{aligned}$$

end while
Output:

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

3.2.4 Simulations

We investigated the DROBA principle on five randomly generated synthetic features. The background PDF of each feature was modeled as a Gaussian density $p_{\mathrm{b},i}(v) = N(v, 0, 1)$, with zero mean and unit standard deviation. Similarly, the genuine user PDF was modeled as Gaussian density $p_{\mathrm{g},i}(v) = N(v, \mu_i, \sigma_i), \sigma_i < 1, i = 1, \ldots, 5$, as listed in Table 3.1. For every feature, a list of detection rates $\delta_i(b_i), b_i \in \{0, \ldots, b_{\max}\}$ with $b_{\max} = 3$, was computed from (3.2). Using these detection rates as input, the bit assignment was generated according to DROBA. Depending on the quantizer type and the bit allocation approach, the simulations were arranged as follows:

- FQ-DROBA(DP): fixed quantizer combined with DROBA, by using the dynamic programming approach;
- FQ-DROBA(GS): fixed quantizer combined with DROBA, by using the greedy search approach;
- LQ-DROBA(DP): likelihood ratio based quantizer combined with DROBA, by using the dynamic programming approach;
- LQ-DROBA(GS): likelihood ratio based quantizer combined with DROBA, by using the greedy search approach;
- FQ-FBA(b): fixed quantizer combined with the fixed b-bit allocation principle [39];
- LQ-FBA(b): likelihood ratio based quantizer combined with the fixed b-bit allocation principle.

Table 3.1: *The randomly generated genuine user PDF $N(v, \mu_i, \sigma_i), i = 1, \ldots, 5$.*

i	1	2	3	4	5
μ_i	−0.12	−0.07	0.49	−0.60	−0.15
σ_i	0.08	0.24	0.12	0.19	0.24

We computed the overall detection rate (3.6), based on the bit assignment corresponding to various specified string length L. The logarithm of the overall detection rate are in illustrated in Fig. 3.7. Results show that DROBA principle generates higher quality strings than the FBA principle. Moreover, DROBA has the advantage that an arbitrary length binary string can always be generated. Regarding the greedy search approach, we observe that the likelihood ratio based quantizer seems to satisfy the monotonicity and concaveness requirements, which explains the same optimal detection rate performance of LQ-DROBA(DP) and LQ-DROBA(GS). However, in the case of the fixed quantizer, some features in Table 3.1 do not satisfy the concaveness requirement for an optimal solution of GS. This explains the better performance of FQ-DROBA(DP) than FQ-DROBA(GS). Note that the performance of LQ-DROBA(DP) consistently outperforms FQ-DROBA(DP). This is because of the better performance of the likelihood ratio based quantizer.

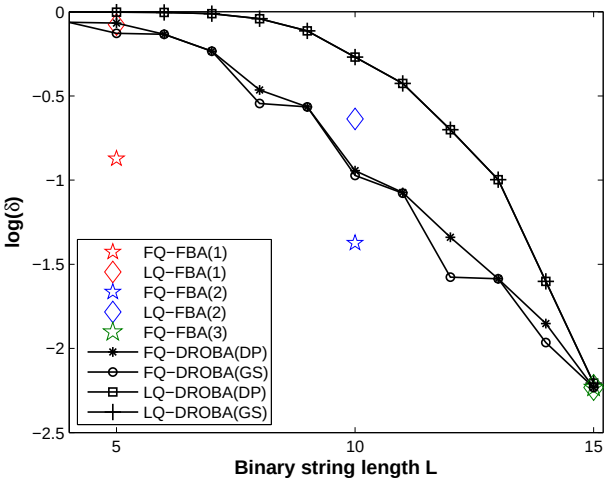


Figure 3.7: *The* $\log(\delta)$ *computed from the bit assignment, through model FQ-DROBA(DP), FQ-DROBA(GS), LQ-DROBA(DP), LQ-DROBA(GS), FQ-FBA(b), LQ-FBA(b),* $b = 1, 2, 3$*, on* 5 *synthetic features, at* $L, L = 1, \ldots, 15$*.*

Table 3.2 gives the bit assignment $\{b_i^*\}$ of FQ-DROBA(DP) and FQ-DROBA(GS), at $L = 1, \ldots, 15$. The result shows that the DROBA principle assigns more bits to discriminative features than the non-discriminative features. We observe that the dynamic programming approach sometimes shows a jump of assigned bits (e.g. from $L = 7$ to $L = 8$ of feature 5, with $\delta = 0.34$ at $L = 8$), whereas the bits assigned through the greedy search approach have to increase one step at a time (with $\delta = 0.28$ at $L = 8$). Such inflexibility proves that the greedy search approach does not provide the optimal solution in this example.

3.2.5 Experiments

We tested the DROBA principle on three data sets, derived from the FVC2000(DB2) fingerprint database [35] and the FRGC(version 1) [37] face database.

- **FVC2000**: This is the FVC2000(DB2) fingerprint data set, containing 8 images of 110 users. Images are aligned according to a standard core point position, in order to avoid a one-to-one alignment. The raw measurements contain two categories: the squared directional field in both x and y directions, and the Gabor response in 4 orientations (0, $\pi/4$, $\pi/2$, $3\pi/4$). Determined by a regular grid of 16 by 16 points with spacing of 8 pixels, measurements are taken at 256 positions, leading to a total of 1536 elements [22].

- **FRGCt**: This is the total FRGC(version 1) face data set, containing 275 users with various numbers of images, taken under both controlled and uncontrolled

Table 3.2: *The bit assignment $\{b_i^*\}$ of FQ-DROBA(DP) and FQ-DROBA(GS) at binary string length L, $L = 1, \ldots, 15$.*

L	$\{b_i^*\}$ of FQ-DROBA(DP)	$\{b_i^*\}$ of FQ-DROBA(GS)
0	[0 0 0 0 0]	[0 0 0 0 0]
1	[0 0 1 0 0]	[0 0 1 0 0]
2	[0 0 1 1 0]	[0 0 1 1 0]
3	[2 0 1 0 0]	[1 0 1 1 0]
4	[2 0 1 1 0]	[2 0 1 1 0]
5	[3 0 1 1 0]	[2 0 2 1 0]
6	[3 0 2 1 0]	[3 0 2 1 0]
7	[3 0 3 1 0]	[3 0 3 1 0]
8	[3 0 2 1 2]	[3 0 3 1 1]
9	[3 0 3 1 2]	[3 0 3 1 2]
10	[3 0 3 1 3]	[3 0 3 2 2]
11	[3 2 3 1 2]	[3 0 3 3 2]
12	[3 3 3 1 2]	[3 1 3 3 2]
13	[3 2 3 3 2]	[3 2 3 3 2]
14	[3 3 3 3 2]	[3 2 3 3 3]
15	[3 3 3 3 3]	[3 3 3 3 3]

conditions. A set of standard landmarks, i.e. eyes, nose and mouth, are used to align the faces, in order to avoid a one-to-one alignment. The raw measurements are the gray pixel values, leading to a total of 8762 elements.

- **FRGCs**: This is a subset of FRGCt, containing 198 users with at least 2 images per user. The images are taken under uncontrolled conditions.

Our experiments involved three steps: training, enrollment and verification. In the training step, we extracted D independent features, via a combined PCA/LDA method [42] from a training set. The obtained transformation was then applied to both the enrollment and verification sets. In the enrollment step, for every target user, the DROBA principle was applied, resulting in a bit assignment $\{b_i^*\}$, with which the features were quantized and coded with a Gray code. The advantage of the Gray code is that the Hamming distance between two adjacent quantization intervals is limited to one, which results in a better performance of a Hamming distance classifier. The concatenation of the codes from D features formed the L-bit target binary string, which was stored for each target user together with $\{b_i^*\}$. In the verification step, the features of the query user were quantized and coded according to the $\{b_i^*\}$ of the claimed identity, and this resulted in a query binary string. Finally the verification performance was evaluated by a Hamming distance classifier. A genuine Hamming distance was computed if the target and the query string originate from the same identity, otherwise an imposter Hamming distance was computed. The detection error trade-off (DET) curve or the equal error rate (EER) was then constructed from these distances.

The users selected for training are different from those in the enrollment and verification. We repeated our experiment with a number of random partitionings. With, in total, n samples per user ($n = 8$ for FVC2000, n ranges from 6 to 48 for FRGCt, and n ranges from 4 to 16 for FRGCs), the division of the data is indicated in Table 3.3.

Table 3.3: *Training, enrollment and verification data, number of users×number of samples per user(n), and the number of partitionings for FVC2000, FRGCt and FRGCs.*

	Training	Enrollment	Verification	Partitionings
FVC2000	$80 \times n$	$30 \times 3n/4$	$30 \times n/4$	20
FRGCt	$210 \times n$	$65 \times 2n/3$	$65 \times n/3$	5
FRGCs	$150 \times n$	$48 \times 2n/3$	$48 \times n/3$	5

In our experiment, the detection rate was computed from the fixed quantizer (FQ) [22], [39]. According to the Central Limit Theorem, we assume that after the PCA/LDA transformation, with sufficient samples from the entire populations, the background PDF of every feature can be modeled as a Gaussian density $p_{\mathrm{b},i}(v) = N(v, 0, 1)$. Hence the quantization intervals are determined as illustrated in Fig. 3.8. Furthermore, in DROBA, the detection rate plays a crucial role. Equation (3.2) shows

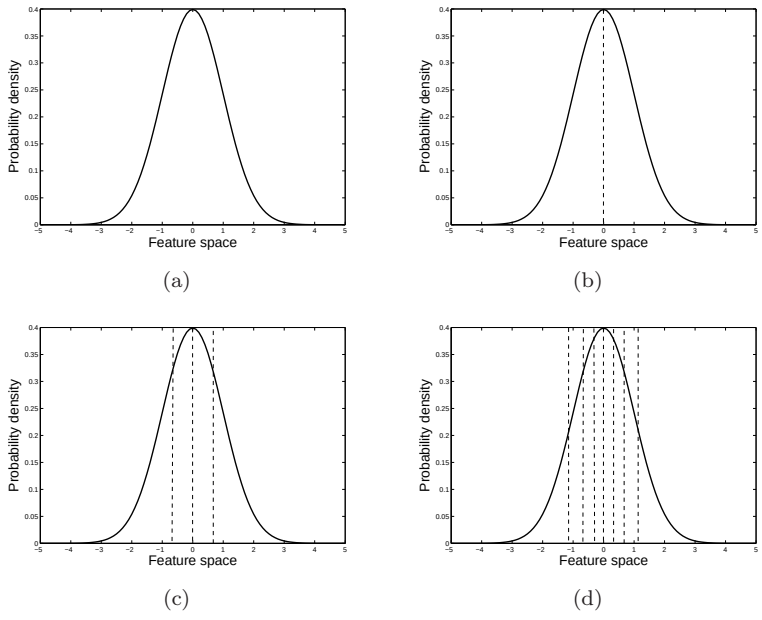


Figure 3.8: *Illustration of the fixed quantizer with equal background probability mass in each interval: background PDF $p_{\mathrm{b},i}(v) = N(v, 0, 1)$ (dashed); quantization intervals (solid). (a) $b_i = 0$; (b) $b_i = 1$; (c) $b_i = 2$; (b) $b_i = 3$.*

that the accuracy of the detection rate is determined by the underlying genuine user PDF. Therefore, we applied the following four models:

- Model 1: We model the genuine user PDF as a Gaussian density $p_{\mathrm{g},i}(v) = N(v, \mu_i, \sigma_i), i = 1, \ldots, D$. Besides, the user has sufficient enrollment samples, so that both the mean μ_i and the standard deviation σ_i are estimated from the enrollment samples. The detection rate is then calculated based on this PDF;

- Model 2: We model the genuine user PDF as a Gaussian density $p_{\mathrm{g},i}(v) = N(v, \mu_i, \sigma_i), i = 1, \ldots, D$, but there are not sufficient user-specific enrollment samples. Therefore, for each feature, we assume that the entire populations share the same standard deviation and thus the σ_i is computed from the entire populations in the training set. The μ_i, however, is still estimated from the enrollment samples. The detection rate is then calculated based on this PDF;

- Model 3: In this model we do not determine a specific genuine user PDF. Instead, we compute a heuristic detection rate $\overline{\delta_i}$, based on the μ_i, estimated from the enrollment samples. The $\overline{\delta_i}$ is defined as:

$$\overline{\delta_i}(b_i) = \begin{cases} 1 \ , & d_{\mathrm{L},i}(b_i) \times d_{\mathrm{H},i}(b_i) > 1 \ , \\ d_{\mathrm{L},i}(b_i) \times d_{\mathrm{H},i}(b_i) \ , & \text{otherwise} \ , \end{cases} \tag{3.18}$$

  where $d_{\mathrm{L},i}(b_i)$ and $d_{\mathrm{H},i}(b_i)$ stand for the Euclidean distance of μ_i to the lower and the higher genuine user interval boundaries, when quantized into b_i bits;

- Model 4: In this model the global detection rates are empirically computed from the entire populations in the training set. For every user, we compute the mean of feature i and evaluate this feature with the samples from the same user, at various quantization bits $b_i = 0, \ldots, b_{\max}$. At each b_i, the number of exact matches $n_{i,\mathrm{m}}(b_i)$ as well as the total number of matches $n_{i,\mathrm{t}}(b_i)$ are recorded. The detection rate of feature i with b_i bits quantization is then the ratio of $n_{i,\mathrm{m}}(b_i)$ and $n_{i,\mathrm{t}}(b_i)$ averaged over all users:

$$\widehat{\delta}_i(b_i) = \frac{\sum_{\text{all users}} n_{i,\mathrm{m}}(b_i)}{\sum_{\text{all users}} n_{i,\mathrm{t}}(b_i)} \ . \tag{3.19}$$

  We then repeat this process for all the features $i = 1, , \ldots, D$. The detection rates $\widehat{\delta}_i(b_i)$ are then used as input of DROBA. As a result, all the users share the same bit assignment.

Following the four models, experiments with DROBA were carried out and compared to the real-value based Mahalanobis distance classifier (MC), likelihood ratio classifier (LC), and the fixed bit allocation principle (FBA). Thus, in short, the experiments are described as:

- DROBA+Model 1/2/3/4: Generate the binary strings based on the fixed quantizer and the DROBA principle via the dynamic programming approach, where the detection rates are derived from Model 1, 2, 3 or 4, respectively. The binary strings are then compared with a Hamming distance classifier. Notation DROBA here refers to FQ-DROBA(DP) in Section 3.2.4.

- FBA: Generate the binary strings based on the fixed quantizer and the fixed bit allocation principle [22], [23], [39], which assigns the same number of bits to all features. The binary strings are then compared with a Hamming distance classifier. Notation FBA here refers to FQ-FBA(b) in Section 3.2.4.

- MC+Model 1/2: Employ a Mahalanobis (norm2) distance classifier [49] on the real-valued features, where the genuine user PDF is derived from Model 1 or 2, respectively;

- LC+Model 1/2: Employ a likelihood ratio classifier [43] on the real-valued features, where the genuine user PDF is derived from Model 1 or 2, respectively.

In the experiments the maximum number of quantization bits for each feature was fixed to $b_{\max} = 3$. This allows us to investigate the impact of the $D - L$ configuration on the DROBA performances. We conducted two experiments: In Experiment I, given D features, we evaluated the verification performances at various binary string lengths L; In Experiment II, given a budget of L bits, we investigated the verification performances with various numbers of features D. Additionally, since experimental results of DP and GS approaches are almost the same, we only present the result of DP.

In Experiment I, Fig. 3.9(a) (c) (e) and Table 3.4 show the corresponding EER performances for FVC2000, FRGCt and FRGCs, given $D = 50$ features after PCA/LDA transformation. All DROBA+Model 1/2/3/4 show similar behavior: As L increases, the performance first improves, and then starts to degrade. This could be explained by (3.6) and (3.7) that given D, a low L ends up in a high FAR bound, contrarily a high L ends up in a low detection rate bound. Therefore, a moderate L might provide a good trade-off between FAR and FRR. For FVC2000 and FRGCs, DROBA+Model 1 and DROBA+Model 2 reveal similar performances, whereas DROBA+Model 3 has slightly worse performance. In the case of FRGCt, DROBA+Model 1 constantly outperforms DROBA+Model 2/3. As a global implementation, DROBA+Model 4 performs worse than DROBA+Model 1/2 for all three data sets, but the difference decreases as L increases. When compared to DROBA+Model 3, despite a rather poor performance at small L, DROBA+Model 4 gives comparable performances at large L. To summarize, given D features, by applying DROBA, there exists a L that gives the optimal FAR/FRR performances of a Hamming distance classifier. The optimal L depends on the Model 1/2/3/4. Furthermore, we observe that at a low bit budget, user-specific models (Model 1/2/3) have advantages over global models (Model 4). Unfortunately, when the bit budget becomes too high, all models become poor. Fig. 3.9(b), (d) and (f) plot the DET curves of their best performances.

Comparing the performances of DROBA to FBA in Fig. 3.9(a) (c) (e), we observe that both DROBA+Model 1/2 outperform FBA for all three data sets. As an example of the FRR/FAR for FRGCt in Fig. 3.10, an explanation might be that DROBA maximizes the detection rate bound of the Hamming distance classifier, leading to averagely lower FRR than FBA. At a low L, DROBA+Model 3 outperforms FBA. However, at high L, it might lose its superiority, as seen in Fig. 3.9(a) (c). This implies that at a high L, the approximate detection rates – computed only from

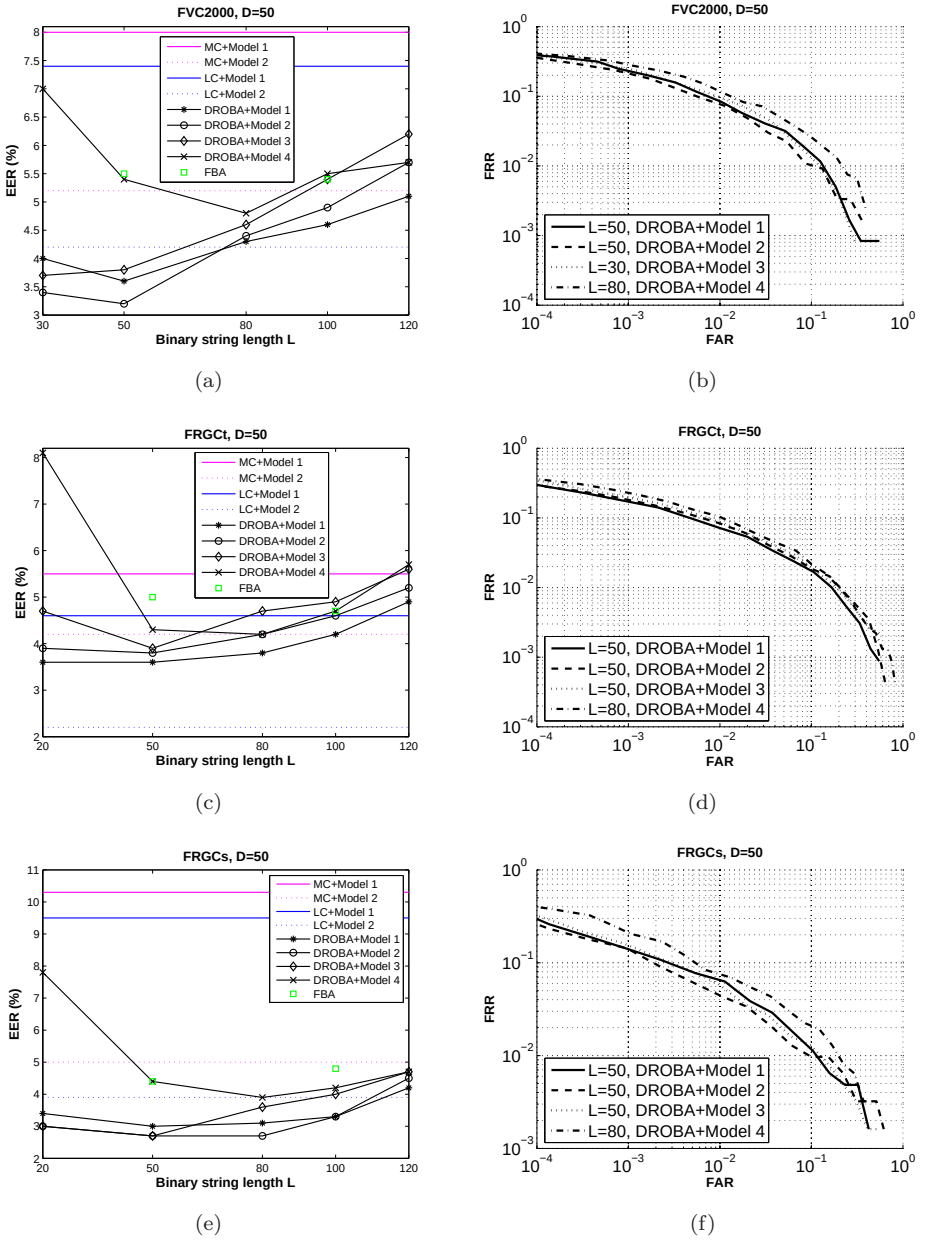


Figure 3.9: *Exp. I: the EER performances of the binary strings generated under DROBA and FBA principles, compared with the real-value feature based Mahalanobis distance classifier (MC) and likelihood-ratio classifier (LC), at $D = 50$, for (a) FVC2000, (c) FRGCt and (e) FRGCs, with the DET of their best performances in (b), (d), and (f), respectively.*

Table 3.4: *Exp. I: the EER performances of DROBA+Model 1/2/3/4, FBA, MC+Model 1/2 and LC+Model 1/2, at $D = 50$, for (a) FVC2000, (b) FRGCt and (c) FRGCs.*

FVC2000	D=50 *EER* = (%)				
	L=30	50	80	100	120
DROBA+Model 1	4.0	**3.6**	4.3	4.6	5.1
DROBA+Model 2	3.4	**3.2**	4.4	4.9	5.7
DROBA+Model 3	**3.7**	3.8	4.6	5.4	6.2
DROBA+Model 4	7.0	5.4	**4.8**	5.5	5.7
FBA	–	5.5	–	5.4	–
MC+Model 1	8.0				
MC+Model 2	5.2				
LC+Model 1	7.4				
LC+Mode 2	4.2				

(a)

FRGCt	D=50 *EER* = (%)				
	L=20	50	80	100	120
DROBA+Model 1	3.6	**3.6**	3.8	4.2	4.9
DROBA+Model 2	3.9	**3.8**	4.2	4.6	5.2
DROBA+Model 3	4.7	**3.9**	4.7	4.9	5.6
DROBA+Model 4	8.1	4.3	**4.2**	4.7	5.7
FBA	–	5.0	–	4.7	–
MC+Model 1	5.5				
MC+Model 2	4.2				
LC+Model 1	4.6				
LC+Model 2	2.2				

(b)

FRGCs	D=50 *EER* = (%)				
	L=20	50	80	100	120
DROBA+Model 1	3.4	**3.0**	3.1	3.3	4.2
DROBA+Model 2	3.0	**2.7**	2.7	3.3	4.5
DROBA+Model 3	3.0	**2.7**	3.6	4.0	4.7
DROBA+Model 4	7.8	4.4	**3.9**	4.2	4.7
FBA	–	4.4	–	4.8	–
MC+Model 1	10.3				
MC+Model 2	5.0				
LC+Model 1	9.5				
LC+Model 2	3.9				

(c)

the mean – no longer provide enough useful information for the DROBA principle. We could imagine that at high L, the bit assignment of DROBA+Model 3 tends to become 'random', so that it is even not competitive to FBA, which has a uniform bit assignment. DROBA+Model 4, however, does not show great advantages over FBA. Since both DROBA+Model 4 and FBA obtain global bit assignment, we could analyze it for every feature. In Fig. 3.11 we plot their bit assignment at $D = 50$, $L = 50$ and 100, for FRGCt. After PCA/LDA transformation, the features with lower index are generally more discriminative than those with higher index. We observe that DROBA+Model 4 consistently assigns more bits to more discriminative features than less discriminative ones. Contrarily, FBA assigns equal bits to every feature. This explains the better performances of DROBA+Model 4.

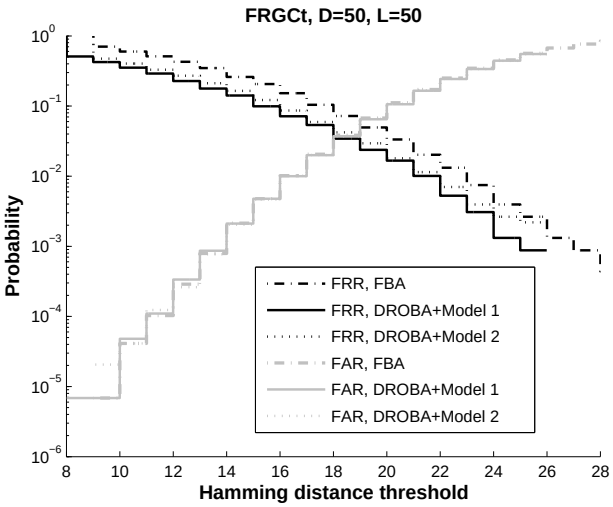


Figure 3.10: *The FAR and FRR performances of FBA and DROBA+Model 1/2, at $D = 50$, $L = 50$.*

Comparing the performances of DROBA to MC and LC in Fig. 3.9(a) (c) (e), we observe that at some lengths L, DROBA+Model 1/2/3 outperform MC+Model 1/2 and LC+Model 1/2, except for LC+Model 2 in FRGCt. Likewise, DROBA+Model 4 obtains better performances than MC+Model 1/2 and LC+Model 1 at some lengths L, but worse performances than LC+Model 2, for all three data sets.

In Experiment II, we investigated the verification performance with various numbers of features D, given a bit budget $L = 50$. Fig. 3.12(a) (c) (e) and Table 3.5 show the corresponding EER performances for FVC2000, FRGCt and FRGCs. We can imagine that more features give DROBA more freedom to choose the optimal bit assignment, which theoretically should give equal or better detection rate bound at a given string length L. On the other hand, we know that the PCA/LDA transformation yields less reliable feature components, as the dimensionality D increases. This means that at a high D, if the detection rate model we apply is not robust

Table 3.5: *Exp. II: the EER performances of DROBA+Model 1/2/3/4, FBA, MC+Model 1/2 and LC+Model 1/2, at $L = 50$, for (a) FVC2000, (b) FRGCt and (c) FRGCs.*

FVC2000	L=50 EER=(%)					
	D=20	30	40	50	60	79
MC+Model 1	7.2	7.3	7.3	8.0	8.2	8.7
MC+Model 2	5.4	5.4	5.3	5.2	5.2	5.4
LC+Model 1	7.3	6.9	7.1	7.4	7.5	7.9
LC+Model 2	4.8	4.6	4.7	4.3	4.3	3.8
DROBA+Model 1	8.4	5.2	4.5	3.6	3.5	**2.9**
DROBA+Model 2	8.3	5.4	4.0	3.2	3.1	**2.7**
DROBA+Model 3	8.5	6.2	4.7	3.8	3.4	**2.8**
DROBA+Model 4	8.2	6.5	5.5	5.4	**5.4**	5.4

(a)

FRGCt	L=50 EER=(%)				
	D=20	50	80	100	120
MC+Model 1	4.9	5.5	6.9	8.1	9.0
MC+Model 2	3.8	4.2	5.7	6.2	6.9
LC+Model 1	4.5	4.6	5.3	5.8	6.3
LC+Model 2	2.7	2.2	2.2	2.2	2.2
DROBA+Model 1	7.0	3.6	3.0	3.0	**3.0**
DROBA+Model 2	7.2	3.8	3.8	3.7	**3.6**
DROBA+Model 3	7.7	4.0	**3.8**	3.9	4.2
DROBA+Model 4	7.3	4.3	**4.3**	4.3	4.3

(b)

FRGCs	L=50 EER=(%)				
	D=20	50	80	100	120
MC+Model 1	8.1	10.3	12.1	13.9	14.8
MC+Model 2	4.3	5.0	6.1	6.6	7.2
LC+Model 1	7.7	9.5	11.4	12.6	13.0
LC+Model 2	3.9	3.9	3.9	3.9	3.7
DROBA+Model 1	6.5	3.0	3.0	2.7	**2.4**
DROBA+Model 2	6.7	2.7	2.5	2.2	**2.1**
DROBA+Model 3	7.5	2.7	2.7	**2.6**	2.8
DROBA+Model 4	6.7	**4.4**	4.4	4.4	4.4

(c)

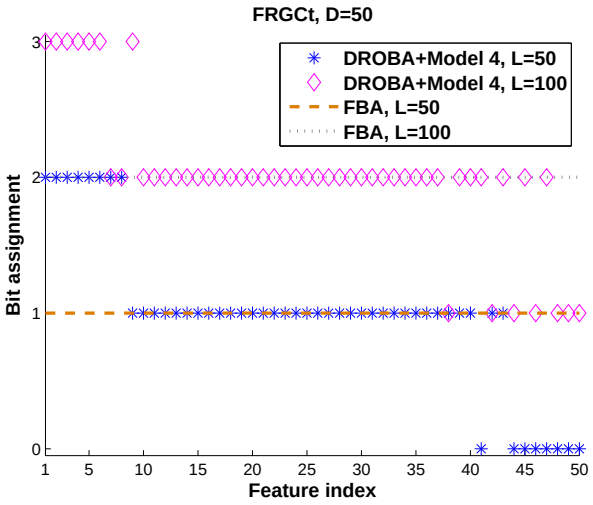


Figure 3.11: *The bit assignment of FBA and DROBA+Model 4, at* $D = 50$, $L = 50$ *and* 100*, for FRGCt.*

enough against the feature unreliability, the computed detection rate might not be accurate and consequently mislead the DROBA. Results show that the performances of DROBA+Model 1/2 on the three data sets consistently improve as D increases. This suggests that given a larger number of less reliable features, DROBA+Model 1/2 are still quite effective. Unlike DROBA+Model 1/2, DROBA+Model 3 starts to degrade at very high D, for FRGCt and FRGCs. This suggests that Model 3 is more susceptible to unreliable features. Since it only uses feature mean to predict the detection rate, when the dimensionality is high, the feature mean becomes unreliable, Model 3 no longer computes accurate detection rate. As a global implementation, DROBA+Model 4 gives relatively worse performances than DROBA+Model 1/2/3. However, we observe that when D is larger than a certain value (50 for FVC2000, 50 for FRGCt, and 20 for FRGCt), the bit assignment of DROBA+Model 4 does not change at all, leading to exactly the same performance. This result is consistent with the PCA/LDA transformation, proving that globally the features are becoming less discriminative as D increases, so that DROBA simply discards all the upcoming features. Therefore, by sacrificing the user specificity, DROBA+Model 4 is immune to unreliable features. Fig. 3.12(b), (d) and (f) plot the DET curves of their best performances.

Comparing the performances of DROBA to MC and LC in Fig. 3.12(a) (c) (e), we observe that for all three data sets, DROBA+Model 1/2/3 easily outperform MC+Model 1/2 and LC+Model 1 as D increases. Similar results are obtained when comparing DROBA+Model 1/2/3 to LC+Model 2 in the context of FVC2000 and FRGCs, whereas for FRGCt, DROBA+Model 1/2/3 do not outperform LC+Model 2. Additionally, DROBA+Model 4 outperforms MC+Model 1 and LC+Model 1, as

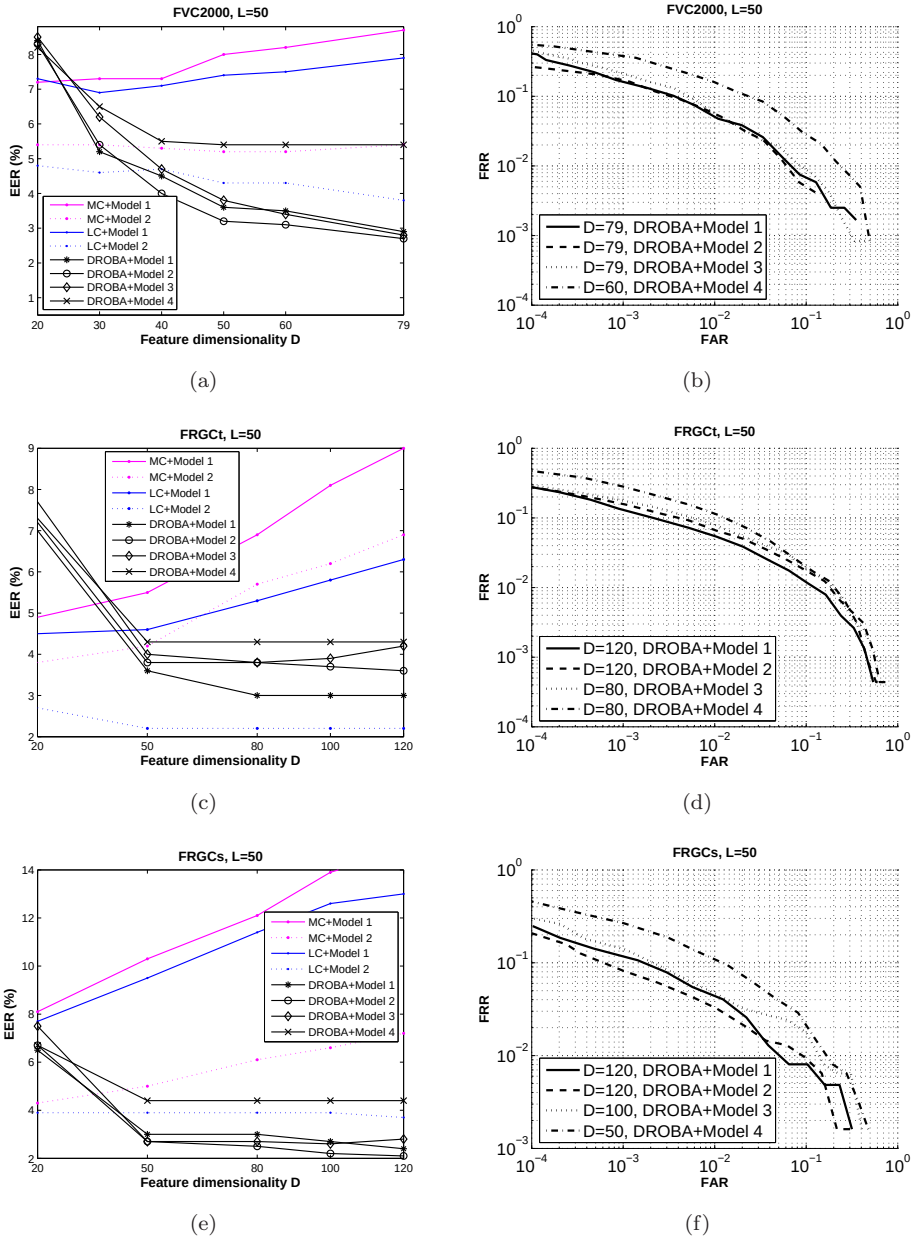


Figure 3.12: *Exp. II: the EER performances of the binary strings generated under DROBA and FBA principles, compared with the real-value feature based Mahalanobis distance classifier (MC) and likelihood-ratio classifier (LC), at $L = 50$, for (a) FVC2000, (c) FRGCt and (e) FRGCs, with the DET of their best performances in (b), (d) and (f), respectively.*

well as MC+Model 2, except for FVC2000. Unfortunately, for all three data sets, DROBA+Model 4 does not outperform LC+Model 2.

3.2.6 Discussion

Since DROBA decides the bit assignment according to the detection rate, determining the underlying genuine user PDF is crucial. However, in practice, it turns out to be difficult, due to the lack of samples. To solve this problem, we proposed three user-specific models: (1) Gaussian density (Model 1), (2) Gaussian density with approximated parameters (Model 2), and (3) heuristic model (Model 3). Experimental results suggest that FVC2000 and FRGCs obtain better performances from Model 2, while FRGCt obtains better performances from Model 1. Generally speaking, the genuine user PDF is associated with the biometric modality, as well as the feature extraction method, thus how to choose the right model (e.g. Gaussian) is important. Furthermore, how to accurately estimate the parameters (e.g. μ, σ) in the model is also a problem to solve. There is no gold standard, and choosing the right model and estimation method is a matter of how accurately it fits the features.

Apart from the user-specific models (Model 1/2/3), we also proposed a global model (Model 4). Our experimental results suggest that in a system with multiple enrollment samples per user, it is preferable to choose user-specific models. Nevertheless, Model 4 still has significant potentials: It is purely empirical and nonparametric, avoiding all problems related with model based estimation; It is robust to unreliable features; It is easily adaptable to all biometric systems.

Essentially, unlike the real-valued classifiers (e.g. MC and LC), which fully depend on or 'trust' the feature density model, DROBA only partially depends on such model. Thus we might see quantization under DROBA as a model oriented compression procedure, where the bit allocation is obtained according to the statistics of the model but the data variation within every quantization interval is ignored, leading to a binary string with compressed information. In fact, in Experiment I, we proved that Hamming distance classifier with binary strings may outperform the MC and LC with real-valued features: The applied density model (e.g. Model 1) is not accurate, so that a compressed binary representation might be less prone to overfitting. The compression can be optimized by carefully tuning the $D - L$ or even the $b_{\max}$ configurations in DROBA.

3.2.7 Conclusion

Generating binary strings from real-valued biometric measurements in fact acts as a data compression process. Thus, in biometric applications, we aim to generate binary strings that not only retain the discriminative information, but also are robust to intra-class variations, so that the performance of the classification is ensured, while the binary strings can be used in various applications. Basically, there are two factors that influence the performance of the binary string: (1) the quantizer design of every feature component; (2) the principle to compose the binary string from all the feature components. In this paper, independent of the quantizer design, we proposed a

detection rate optimized bit allocation principle (DROBA), which can be achieved by both a dynamic programming and a greedy search approach. Consequently DROBA assigns more bits to discriminative features and fewer bits to non-discriminative features. This process is driven by the statistics derived from the training and enrollment data, based on which we proposed four models. Experiments on the FVC2000 fingerprint and the FRGC face database show promising results.

The DROBA principle has the advantage that it is adaptable to arbitrary biometric modalities, such as fingerprint texture, iris, signature and face. Additionally, the binary strings can be used in any kind of binary string based classifiers, as well as crypto systems. The practical applications of the biometric binary strings are not only limited to the template protection systems, but also systems requiring fast matching or constrained storage capability. Furthermore, combined with various detection rate estimation methods, binary strings generated under DROBA can be a new promising biometric representation as opposed to the real-valued representation.

3.3 Chapter conclusion

In this chapter, a detection rate optimized bit allocation (DROBA) principle is presented. Regarding the research objectives, DROBA is able to allocate a user-dependent number of bits to every biometric feature, while maintaining a fixed total length of the binary string. The extracted binary string has *i.i.d.* bits. Independent of the quantizers that are used, DROBA optimizes the overall theoretical detection rate, when the Hamming distance threshold is zero. Compared to quantizing every feature into a prescribed fixed number of bits, as described in Chapter 2, combining quantizers with DROBA yields better recognition performance. Furthermore, because the bit allocation is user-dependent, the extracted bits are more reliable, which allows for a longer length of the random key, or equivalently, a longer secret length.

4

Area under the FRR Curve Optimized Bit Allocation

4.1 Chapter introduction

PURPOSE. In chapter 3, a detection rate optimized bit allocation (DROBA) principle was presented. Independent of the quantizers, DROBA aims to minimize the theoretical FRR performance at zero Hamming distance threshold for a Hamming distance classifier (HDC). As a result, a user-dependent number of bits is assigned to every biometric feature, while maintaining a fixed total length of the binary string. However, the theoretical FAR performance at zero Hamming distance threshold is far below the required FAR range of a biometric system. Therefore, an advanced bit allocation principle is required to optimize the overall FAR and FRR performances for a HDC. Minimizing the area under the FRR curve over all possible Hamming distance thresholds is such a principle. Furthermore, similar to DROBA, this new bit allocation principle should be independent of the quantizers. Given independent features and quantizers that can extract statistically independent and identically distributed (*i.i.d.*) bits, the new bit allocation principle should preserve the *i.i.d.* property.

CONTENTS. In this chapter, we first present a theoretical model of the HDC, based on the bit error probabilities of independent biometric features. For an enrolled user, the k-bit error probability of a feature is defined as the probability of having k erroneous bits between the quantized features at enrollment and verification. Thus, given a quantizer and the number of quantization bits per feature, the corresponding bit error probabilities for every feature can be computed for both the genuine user and the

imposters. Assuming independent features, the overall FAR and FRR performances at a Hamming distance threshold are then predictable as a combination of these bit error probabilities.

Given the theoretical model of the HDC, its theoretical FRR performance over all possible Hamming distance thresholds can be computed. Therefore, we propose the area under the FRR curve optimized bit allocation (AUF-OBA) principle. Given any type of quantizer, for every feature of an enrolled user, the error probabilities with all possible number of erroneous bits are computed at a range of allowed bits. AUF-OBA then aims to minimize the theoretical area under the FRR curve over all Hamming distance thresholds, subject to a fixed total number of bits. A dynamic programming approach is then applied to search for the optimal solution. AUF-OBA can be applied in both one- and two-dimensional quantization schemes. In this chapter, AUF-OBA is presented in combination with the one-dimensional quantizers, as illustrated in Fig. 4.1. Figure 4.2 shows the contribution of this chapter in the context of the thesis.

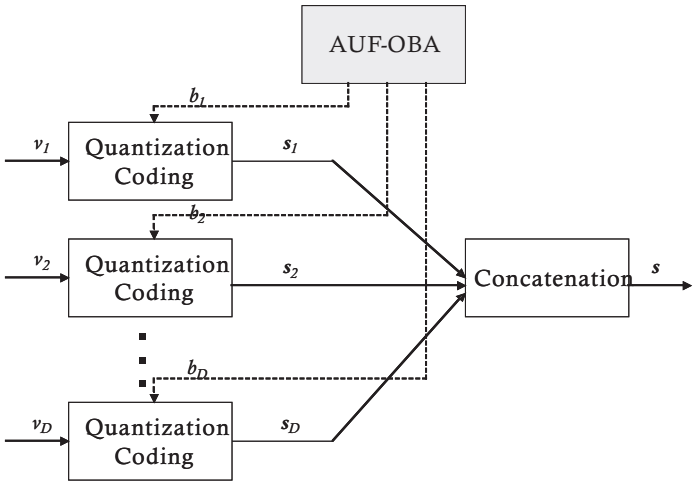


Figure 4.1: *Block diagram of the one-dimensional quantization and coding scheme, highlighted in AUF-OBA design. The $v_i, i = 1 \dots D$ denote D independent biometric features, and b_i denotes the assigned number of bits to the i^{th} feature. The quantized bits $s_i, i = 1 \dots D$ from all D features are then concatenated into the binary string s.*

PUBLICATION(S). The content of Section 4.2 has been published in [50].

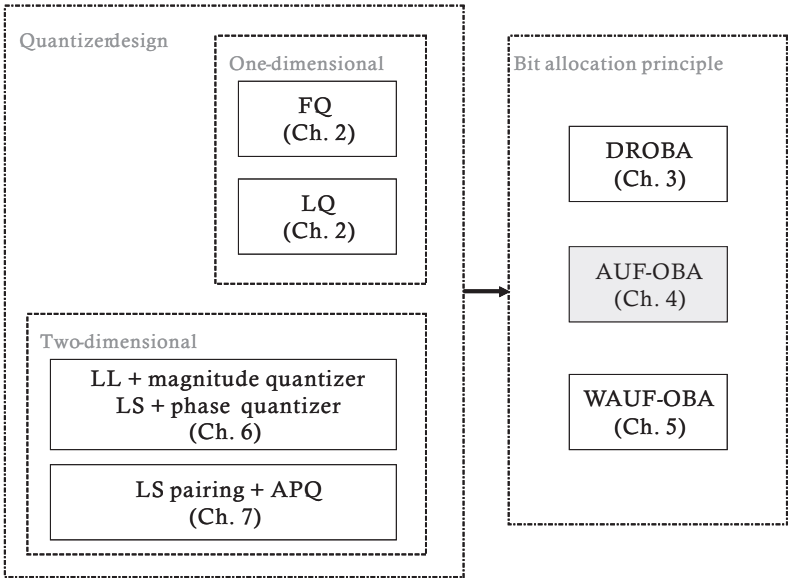


Figure 4.2: *Block diagram of the main contributions, highlighted in chapter* 4.

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

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 is 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.

4.2.1 Introduction

Binary representations for biometrics have drawn considerable interest for their merits in template compression, and particularly template protection [40], [4]. Unprotected storage and transfer of biometric information allows 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 [14], [15], [16], [17], [18], cancelable biometrics [19], [20], biometric key generation [13], [5], [6], [7], [8], [9], [10], and biometric key binding [21], [32], [24], [22], [23], [25], [26], [33], [27], [28], [29], [30]. 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 [22]. Thus, before being used for template protection purpose, the biometric features need to be transformed into a binary string. Therefore, as shown in Fig. 4.3, a template protected biometric verification system with binary representations can be generalized into three modules.

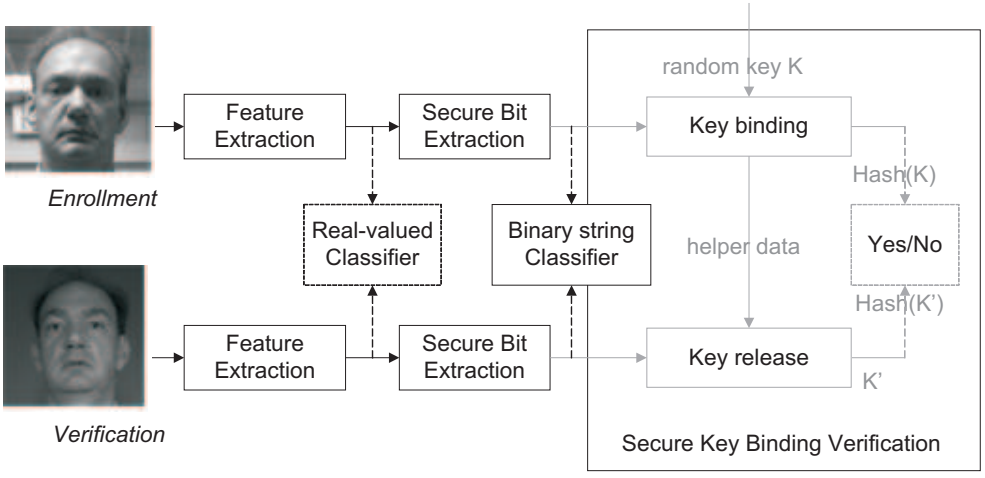


Figure 4.3: *The scheme of a template protected biometric verification system with binary representations.*

Feature extraction: This module aims to extract independent, reliable and discriminative real-valued features from raw measurements. Independent features are highly 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 Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [34] 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.

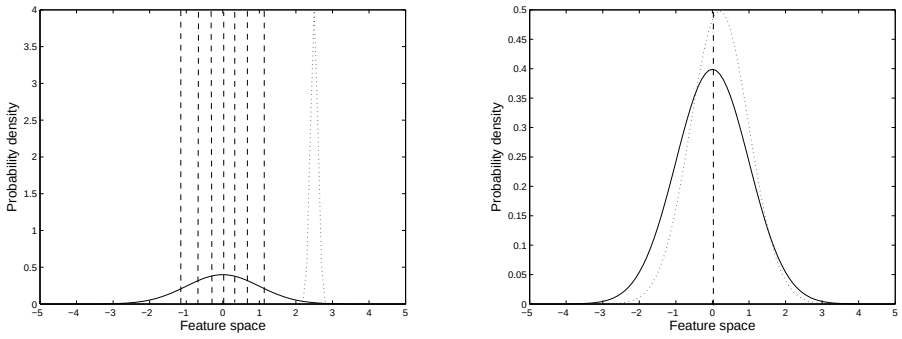
Secure bit extraction: This module aims to transform the real-valued features into a fixed-length binary string, which is used to bind a crypto key. Biometric information is well-known for its uniqueness. Unfortunately, due to sensor and user behavior, it is inevitably noisy, which leads to intra-class 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 [22], 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. 4.4, 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 [24], [22], [23], [8], [46], [9], [39]. Quantizers in [24], [22], [23] are user-independent, constructed merely from the background PDF, whereas quantizers in [8], [46], [9], [39]

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 [39], which is optimal in the Neyman-Pearson sense. Quantizers in [24], [8], [46] and [9] 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 [22], [23] and [39] have equal-probability intervals, which meets the *i.i.d.* bit requirements mentioned above.



(a) (b)

Figure 4.4: *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.*

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 [22], [23], [39] and a detection rate optimized bit allocation (DROBA) principle [51] have been proposed. The FBA principle assigns a fixed number of bits to every feature. As seen in Fig. 4.4, 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 [48]. The DROBA principle solves this problem by assigning a variable number of bits based on 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 [52], [53] 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. 4.5 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} to 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.

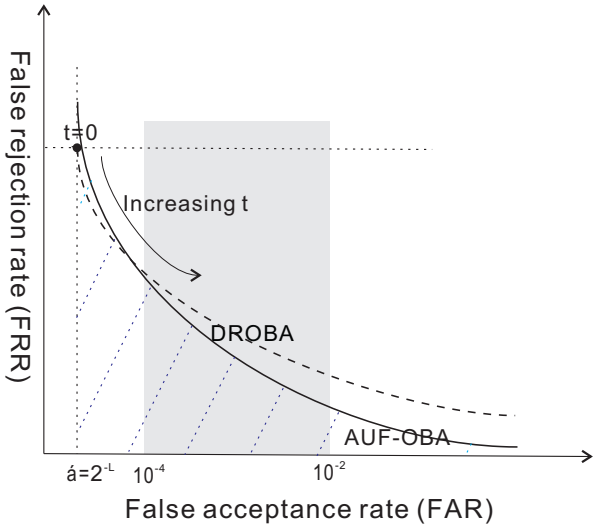


Figure 4.5: *Illustration of DROBA and AUF-OBA principles.*

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 4.2.2 we give the analytical performance of a HDC, given the features' bit error probability. In Section 4.2.3 we present the AUF-OBA principle. Simulation results are illustrated in Section 4.2.4. In Section 4.2.5, we give some experimental results of AUF-OBA on the FVC2000 fingerprint database and the FRGC face database. In Section 4.2.6 the results are discussed and conclusions are drawn in Section 4.2.7.

4.2.2 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 j^{th} feature.

During the enrollment, let $s_{\text{g},j}$ denote the string of b_j bits generated by the genuine user for the j^{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. $\boldsymbol{s}_{\text{g}} = s_{\text{g},1} \ldots s_{\text{g},D}$. Similarly, during the verification, let $s'_{\text{g},j}$ and $s'_{\text{i},j}$ be the bits generated by the genuine user and the imposters, respectively, for the j^{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 intra-class variation, the genuine user might not extract the same string as the enrollment template, i.e. $s'_{\text{g},j} \neq s_{\text{g},j}$. Contrarily, the imposter might end up with the same string as that of the genuine user in the enrollment, i.e. $s'_{\text{i},j} = s_{\text{g},j}$. Therefore, we introduce the following definitions.

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

$$P_{\text{g},j}(k_j; b_j) = P\{d_H(s_{\text{g},j}, s'_{\text{g},j}) = k_j\}, k_j \in 0, \ldots, b_j \ , \tag{4.1}$$

$$P_{\text{i},j}(k_j; b_j) = P\{d_H(s_{\text{g},j}, s'_{\text{i},j}) = k_j\}, k_j \in 0, \ldots, b_j \ , \tag{4.2}$$

where d_H is the Hamming distance between two input bit strings. Hence $P_{\text{g},j}$ and $P_{\text{i},j}$ 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 j^{th} feature during the verification.

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

$$\phi_{\text{g}}(k; \{b_j\}_{j=1}^{D}) = P\{d_H(\boldsymbol{s}_{\text{g}}, \boldsymbol{s}'_{\text{g}}) = k\}, k \in 0, \ldots, L \ , \tag{4.3}$$

$$\phi_{\text{i}}(k; \{b_j\}_{j=1}^{D}) = P\{d_H(\boldsymbol{s}_{\text{g}}, \boldsymbol{s}'_{\text{i}}) = k\}, k \in 0, \ldots, L \ , \tag{4.4}$$

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 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. $P_{\text{g},j}$, $P_{\text{i},j}$, 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 [54], 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 $P_{\text{g},j}$ and $P_{\text{i},j}$:

$$\phi_{\text{g}}(k; \{b_j\}_{j=1}^{D}) = (P_{\text{g},1} * P_{\text{g},2} * \ldots * P_{\text{g},D})(k; \{b_j\}_{j=1}^{D}) \ , \tag{4.5}$$

$$\phi_{\text{i}}(k; \{b_j\}_{j=1}^{D}) = (P_{\text{i},1} * P_{\text{i},2} * \ldots * P_{\text{i},D})(k; \{b_j\}_{j=1}^{D}) \ . \tag{4.6}$$

Expressions in (4.5) and (4.6) 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, ($0 \leq t \leq L$), is defined as:*

$$\alpha(t; \{b_j\}_{j=1}^{D}) \quad = \quad P\{d_H(\boldsymbol{s}_{\mathrm{g}}, \boldsymbol{s}'_{\mathrm{i}}) \leq t\} \ . \tag{4.7}$$

Given (4.4), we have

$$\alpha(t; \{b_j\}_{j=1}^{D}) \quad = \quad \sum_{k=0}^{t} \phi_{\mathrm{i}}(k; \{b_j\}_{j=1}^{D}) \ . \tag{4.8}$$

Furthermore, to obtain *i.i.d.* bits, an equal-probability quantizer ([22], [23], [39]), with 2^{-b_j} probability mass for every interval, is required for the quantization of every feature. Thus, for the j^{th} feature, when assigned with 2^{b_j} code words, the $P_{\mathrm{i},j}(k_j; b_j)$, as defined in (4.2), becomes:

$$P_{\mathrm{i},j}(k_j; b_j) = 2^{-b_j} \left(\begin{array}{c} b_j \\ k_j \end{array} \right) \ . \tag{4.9}$$

Subject to $\sum_{j=1}^{D} b_j = L$, the FAR in (4.7) becomes:

$$\begin{aligned} \alpha(t; \{b_j\}_{j=1}^{D}) \quad &= \quad \sum_{k=0}^{t} \phi_{\mathrm{i}}(k; \{b_j\}_{j=1}^{D}) \ , \\ &= \quad 2^{-L} \sum_{k=0}^{t} \left(\begin{array}{c} L \\ k \end{array} \right) \ . \end{aligned} \tag{4.10}$$

The proof of (4.10) is given in Appendix B. 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, ($0 \leq t \leq L$), as:*

$$\beta(t; \{b_j\}_{j=1}^{D}) \quad = \quad P\{d_H(\boldsymbol{s}_{\mathrm{g}}, \boldsymbol{s}'_{\mathrm{g}}) > t\} \ . \tag{4.11}$$

Given (4.3), we have

$$\beta(t; \{b_j\}_{j=1}^{D}) \quad = \quad \sum_{k=t+1}^{L} \phi_{\mathrm{g}}(k; \{b_j\}_{j=1}^{D}) \ . \tag{4.12}$$

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

Given the analytical FRR performance in (4.11), 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 $P_{\mathrm{g},j}(k_j; b_j)$ 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.

4.2.3.1 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 j^{th} 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:

$$\begin{aligned} \{b_j^*\}_{i=1}^{D} &= \arg \min_{\sum_{j=1}^{D} b_j = L} A_{\text{FRR}} \ , \\ &= \arg \min_{\sum_{j=1}^{D} b_j = L} \sum_{t=0}^{L} \beta(t; \{b_j\}_{j=1}^{D}) \ , \end{aligned} \tag{4.13}$$

4.2.3.2 AUF-OBA Solution

We first reformulate the FRR in (4.12) into the following expression:

$$\beta(t; \{b_j\}_{j=1}^{D}) = \sum_{l=0}^{L} \mathrm{u}(l - (t+1)) \phi_{\mathrm{g}}(l; \{b_j\}_{j=1}^{D}) \ , \tag{4.14}$$

with

$$\mathrm{u}(l) = \begin{cases} 1, & l \geq 0 \ , \\ 0, & l < 0 \ . \end{cases} \tag{4.15}$$

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:

$$\begin{aligned} A_{\mathrm{FRR}} &= \sum_{t=0}^{L} \beta(t; \{b_j\}_{j=1}^{D}) \ , \\ &= \sum_{t=0}^{L} \sum_{l=0}^{L} \left[\mathrm{u}(l-(t+1)) \phi_{\mathrm{g}}(l; \{b_j\}_{j=1}^{D}) \right] \ , \\ &= \sum_{l=0}^{L} \left[\phi_{\mathrm{g}}(l; \{b_j\}_{j=1}^{D}) \sum_{t=0}^{L} \mathrm{u}(l-(t+1)) \right] \ , \\ &= \sum_{l=0}^{L} l \phi_{\mathrm{g}}(l; \{b_j\}_{j=1}^{D}) \ . \end{aligned} \tag{4.16}$$

Expression (4.16) is the expected value of the number of bit errors k, which we denote by $E[k; \{b_j\}_{j=1}^{D}]$. Hence, for a certain bit assignment $\{b_j\}_{j=1}^{D}$, A_{FRR} equals $E[k; \{b_j\}_{j=1}^{D}]$.

$$A_{\mathrm{FRR}} = E[k; \{b_j\}_{j=1}^{D}] \ . \tag{4.17}$$

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,

$$\begin{aligned} A_{\mathrm{FRR}} &= E[k; \{b_j\}_{j=1}^{D}] \ , \\ &= \sum_{j=1}^{D} E[k_j; b_j] \ , \end{aligned} \tag{4.18}$$

where $E[k_j; b_j]$ is the expected value of the number of errors k_j for the j^{th} feature:

$$E[k_j; b_j] = \sum_{l=0}^{b_j} l P_{\mathrm{g},j}(l; b_j) \ . \tag{4.19}$$

We can now reformulate the AUF-OBA problem as:

$$\{b_j^*\}_{j=1}^{D} = \arg \min_{\sum_{j=1}^{D} b_j = L} \sum_{j=1}^{D} E[k_j; b_j] \ . \tag{4.20}$$

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

$$G_j(b_j) = -E[k_j; b_j] \ . \tag{4.21}$$

The AUF-OBA then becomes a maximization problem:

$$\begin{aligned} \{b_j^*\}_{j=1}^D &= \arg \min_{\sum_{j=1}^D b_j = L} \sum_{j=1}^D E[k_j; b_j] \ , \\ &= \arg \max_{\sum_{j=1}^D b_j = L} \sum_{j=1}^D G_j(b_j) \ . \end{aligned} \tag{4.22}$$

With the gain factor defined in (4.21), the problem in (4.22) has the same form as the DROBA optimization problem presented in [51]. Therefore, solving (4.22) 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 [51].

4.2.3.3 Computing $G_j(b_j)$

To compute $G_j(b_j)$, the genuine user bit error probability $P_{\mathrm{g},j}(k_j; b_j)$ is required. As defined in (4.1), given the feature's genuine user PDF $p_{\mathrm{g},j}$, the quantizer and the number of quantization bits b_j, we can compute $P_{\mathrm{g},j}(k_j; b_j)$ as:

$$P_{\mathrm{g},j}(k_j; b_j) = \int_{Q(k_j;b_j)} p_{\mathrm{g},j}(v) dv \ , \tag{4.23}$$

where $Q(k_j; b_j)$ indicates the quantization intervals with k_j-bit error as compared to the genuine code $s_{\mathrm{g},j}$. An example of these intervals encoded by a Gray code [44] is illustrated in Fig. 4.6.

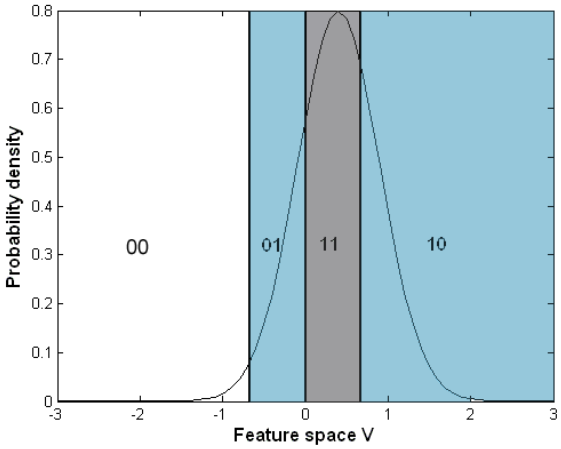


Figure 4.6: *An example of computing $P_{\mathrm{g},j}(k_j; b_j)$ for the j^{th} feature, assigned with $b_j = 2$ bits Gray code. The genuine user PDF $p_{\mathrm{g},j}$ (black curve); $Q(0; 2)$ with the genuine code '11' (grey); $Q(1; 2)$ with 1-bit error (blue); and $Q(2; 2)$ with 2-bit error (white).*

4.2.3.4 Dynamic programming approach

The optimization problem in (4.22) has the same form as DROBA [51]. Therefore, once the $G_j(b_j)$ is computed, (4.22) can be solved by a common recursive dynamic programming approach, as described in Appendix C. As explained in [51], the essential concept is that the optimal bits assignment for j features can be computed directly from 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.

4.2.4 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 (4.23) is the user-independent equal-probability quantizer [22], [23], [39], defined as :

$$B_0 = -\infty\ , \tag{4.24}$$

$$B_m = \arg_B\left[\int_{B_{m-1}}^{B_m} p_{\mathrm{b},j}dv = 2^{-b_j}\right], m=1,\ldots,2^{b_j}\ , \tag{4.25}$$

where $(B_{m-1},B_m]$ represents the m^{th} 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 C. Afterwards, we compute the corresponding FAR and FRR performances for HDC according to (4.10) and (4.11).

Figure 4.7 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.

Figure 4.8 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.

In Fig. 4.9, 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 4.2.5.2 we tested this Gaussian assumption on real data.

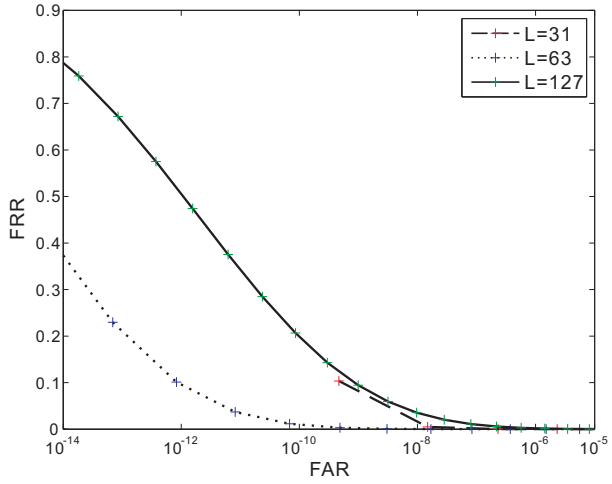


Figure 4.7: *The FAR vs. FRR performances of AUF-OBA on the synthetic features, when the output L=31, 63 and 127, at D=50.*

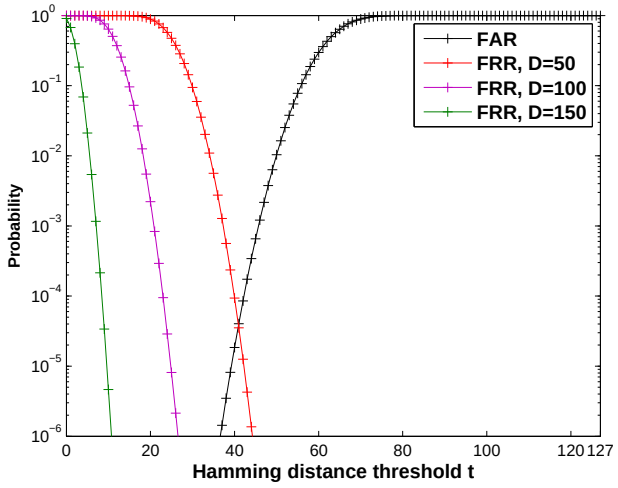


Figure 4.8: *The FAR vs. FRR performances of AUF-OBA on the synthetic features, when the input D=50, 100 and 150, at L=127.*

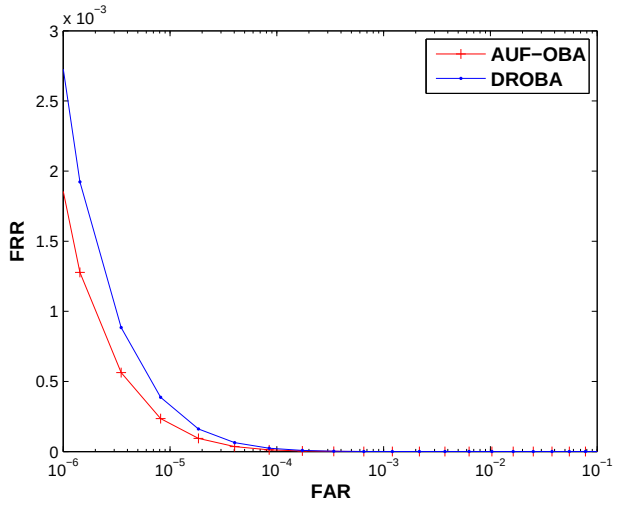


Figure 4.9: *The FAR vs. FRR performances of AUF-OBA on the synthetic features, compared with DROBA, at D=50, L=127.*

4.2.5 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.

4.2.5.1 Experimental Setup

We tested the AUF-OBA on three data sets, derived from the FVC2000(DB2) fingerprint database [35] and the FRGC(version 1) face database [37]. 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 data set, containing 8 images of 110 users. Images are aligned to an automatically detected standard core point position through translation. As illustrated in Fig. 4.10, the raw measurements contain two categories: the squared directional field in both x and y directions, and the Gabor response in 4 orientations (0, $\pi/4$, $\pi/2$, $3\pi/4$). Determined by a regular grid of 16 by 16 points with spacing of 8 pixels, measurements are taken at 256 positions, leading to a total of 1536 elements [22].

- **FRGC$_{\mathbf{H}}$**: 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. 4.11, 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$_{\mathbf{L}}$**: 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$_{\mathrm{H}}$.

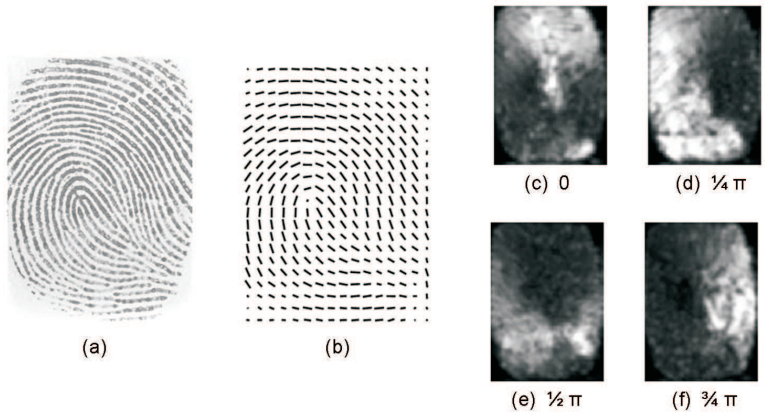


Figure 4.10: *(a) Fingerprint image, (b) directional field, (c)-(f) the absolute values of Gabor responses for different orientations θ.*

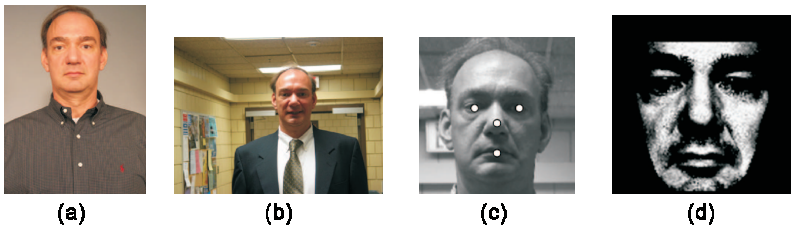


Figure 4.11: *(a) Controlled image, (b) uncontrolled image, (c) landmarks and (d) the region of interest (ROI).*

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 4.1.

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 features can be applied.

Table 4.1: *Data division: number of users×number of samples per user(n), and the number of trials for FVC2000,* $\mathrm{FRGC_H}$ *and* $\mathrm{FRGC_L}$.

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

During the training step in our experiment, we applied a common PCA/LDA [42] 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 j^{th} feature, we first have to estimate both the background PDF $p_{\mathrm{b},j}$ and the genuine user PDF $p_{\mathrm{g},j}$. In [51], it is shown that modeling every feature as Gaussian density gives reasonably good performances. Therefore, we model both PDFs as Gaussian density $p_{\mathrm{b},j} = N(v, 0, 1)$, $p_{\mathrm{g},j} = N(v, \mu_j, \sigma_j)$. Additionally, we set $b_{\max} = 3$, and the gain factor G_j was computed from the fixed quantizer in (4.25). 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.

4.2.5.2 Experimental Results

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, $\mathrm{FRGC_H}$ and $\mathrm{FRGC_L}$ are shown in Fig. 4.12, Fig. 4.13, Fig. 4.14, Fig. 4.15, Fig. 4.16 and Fig. 4.17, 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. Figure 4.14 and 4.15 suggests that for the high quality data $\mathrm{FRGC_H}$, 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. 4.8 and prove that AUF-OBA can effectively extract distinctive bits when the feature dimensionality is high. Contrarily, Fig. 4.16 and 4.17 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. 4.16(a), 4.16(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 data sets 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. 4.7. 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 4.2. 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.

Bit Capacity of Features

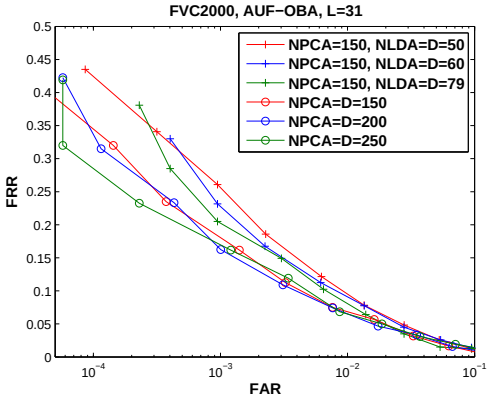
Since AUF-OBA enables more quantization bits for distinctive features than for non-distinctive feature, the bit assignment to some extent indicates the feature distinctiveness. Therefore, we take the best D-L settings in Table 4.2, and in Fig. 4.18 we plot the bit assignment histogram for the features, averaged over all genuine users. All three data sets show consistent results. A large proportion of features are assigned with 0 bits or discarded, which means these features are not distinctive. However, only few features are distinctive enough to extract 2 or 3 bits.

Comparison with DROBA

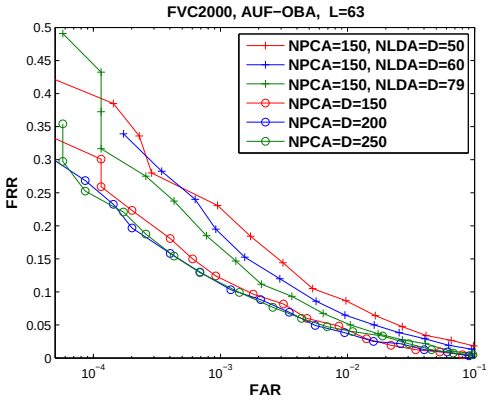
In Fig. 4.9 we showed that theoretically AUF-OBA is superior to DROBA concerning the performances at the operational points. Now we further compare their performances on the real data. In Fig. 4.19 we illustrate their performances at the same *D-L* settings. Results show that AUF-OBA is indeed slightly better than DROBA.

Considerations about the independent Gaussian assumption

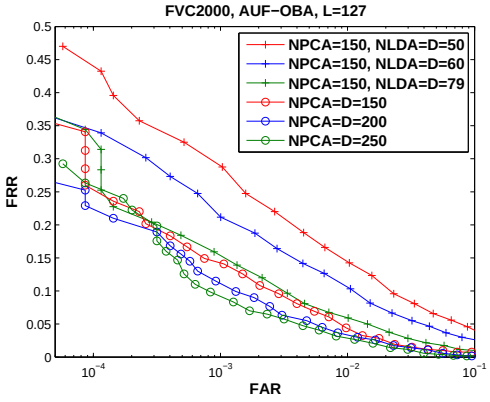
One important assumption in AUF-OBA is – for both the imposters and the genuine user – the independency among the features. In our experiments, we assume that the measurements are with Gaussian density, thus after the PCA transforma-



(a)

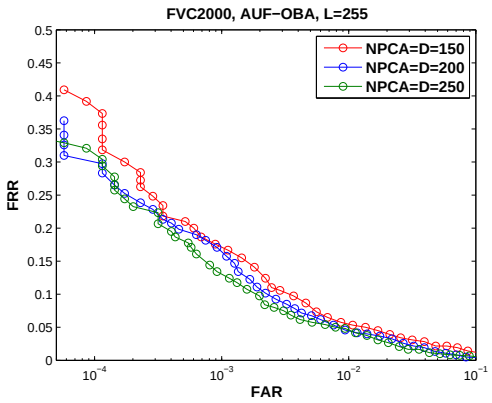


(b)



(c)

Figure 4.12: *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) $L = 127$.*



(a)

Figure 4.13: *The FAR vs. FRR performances for FVC2000 extracted with AUF-OBA, from various numbers of features D, at L = 255.*

Table 4.2: *The FAR vs. FRR performances for (a) FVC2000, (b)* $\mathrm{FRGC_H}$ *and (c)* $\mathrm{FRGC_L}$.

FVC2000	FRR (%)	FAR	FRR (%)	FAR	FRR (%)	FAR
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

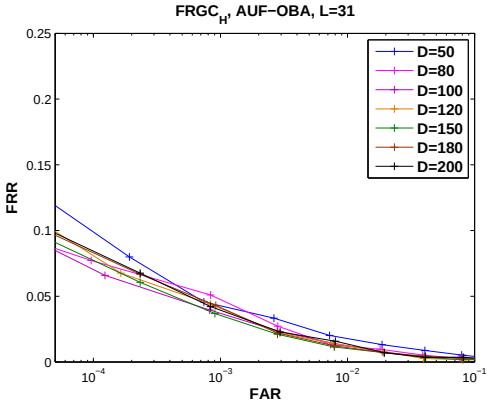
(a)

$\mathrm{FRGC_H}$	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**
D=200, L=255	6.4	0.01	2.6	0.1	1.4	1.0

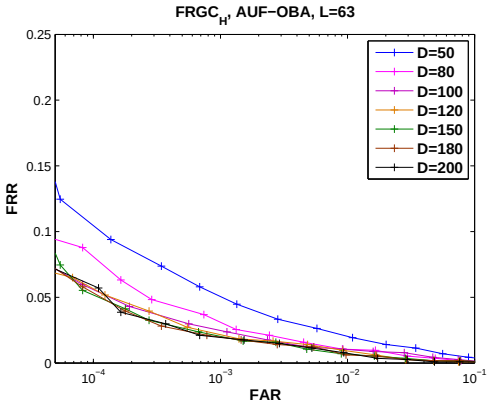
(b)

$\mathrm{FRGC_L}$	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**
D=149, L=255	18	0.01	10	0.1	5	1.0

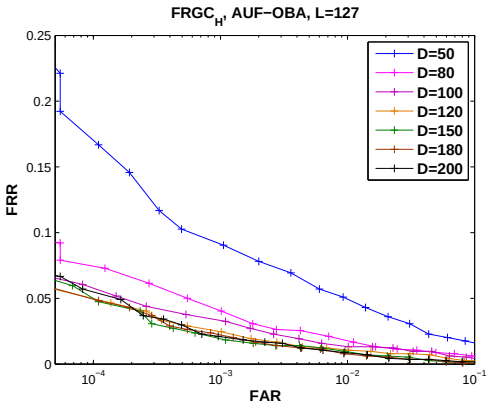
(c)



(a)

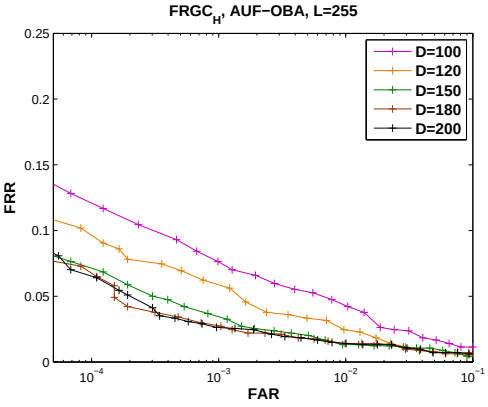


(b)



(c)

Figure 4.14: *The FAR vs. FRR performances for* $\mathrm{FRGC_H}$*, with varying* D *(*$NPCA = 250$*,* $NLDA = D$*), at (a)* $L = 31$*; (b)* $L = 63$*; (c)* $L = 127$*.*



(a)

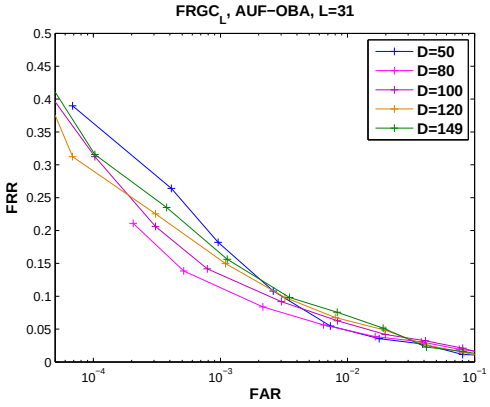
Figure 4.15: *The FAR vs. FRR performances for* $\mathrm{FRGC_H}$*, with varying* D *(*$NPCA = 250$*,* $NLDA = D$*), at* $L = 255$*.*

tion, 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.

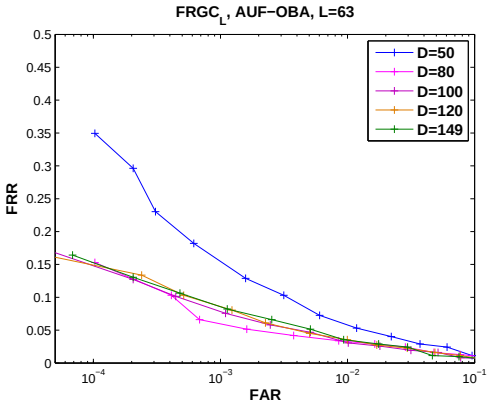
As in the previous experiments, computing the $\{b_j^*\}_{j=1}^D$ output of AUF-OBA is based on the independent Gaussian density $p_{\mathrm{g},j}$, $p_{\mathrm{b},j}$. Then, according to (4.10) and (4.11), we can compute the theoretical FAR as well as the theoretical averaged FRR performances over all 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 data sets. 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. 4.20 we give an example of the performances for $\mathrm{FRGC_H}$, 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.

4.2.6 Discussion

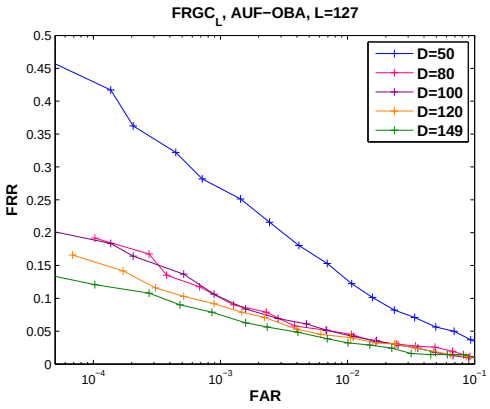
An important assumption in AUF-OBA is that after feature extraction (e.g. PCA/LDA), the features are independent among both the entire populations and the genuine user. In Section 4.2.5.2 we proved the independency among the entire



(a)

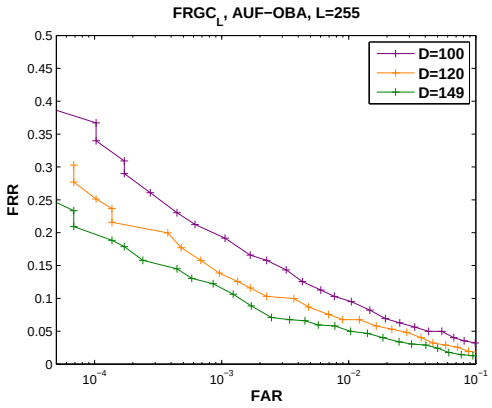


(b)



(c)

Figure 4.16: *The FAR vs. FRR performances for* $\mathrm{FRGC_L}$*, with varying* D *(*$NPCA = 250$*,* $NLDA = D$*), at (a)* $L = 31$*; (b)* $L = 63$*; (c)* $L = 127$*.*



(a)

Figure 4.17: *The FAR vs. FRR performances for* $\mathrm{FRGC_L}$*, with varying* D *(*$NPCA = 250$*,* $NLDA = D$*), at* $L = 255$*.*

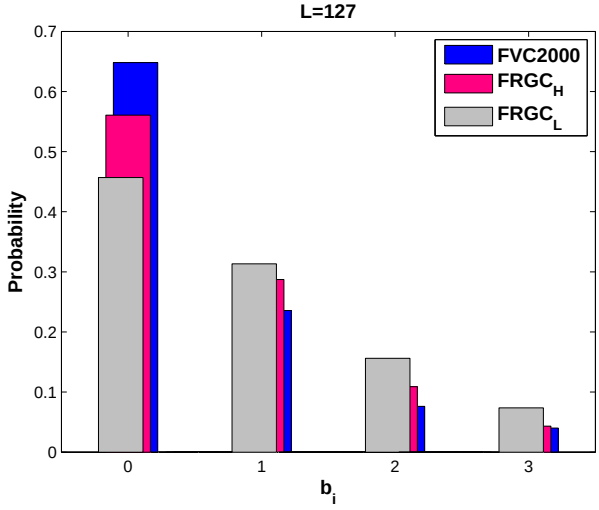
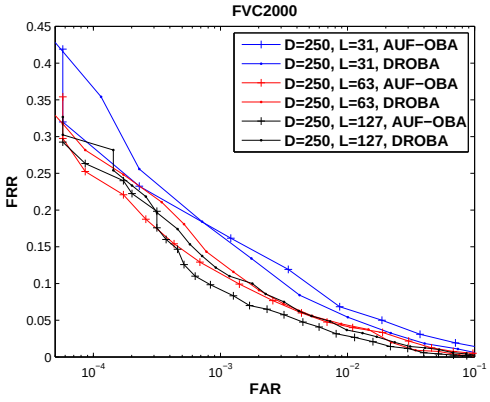
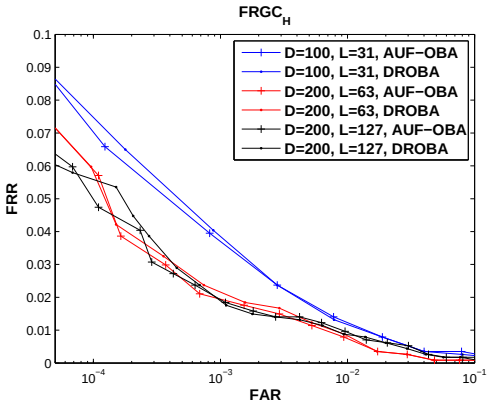


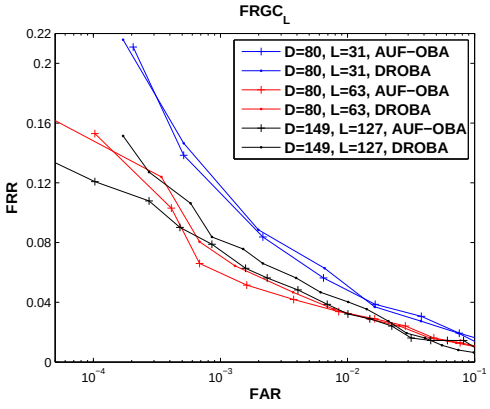
Figure 4.18: *An example of the bit assignment histogram for the features, averaged over all genuine users, for FVC2000,* $\mathrm{FRGC_H}$ *and* $\mathrm{FRGC_L}$*.*



(a)



(b)



(c)

Figure 4.19: *The FAR vs. FRR performances of AUF-OBA, compared with DROBA, for (a) FVC2000, (b)* $\mathrm{FRGC_H}$ *and (c)* $\mathrm{FRGC_L}$.

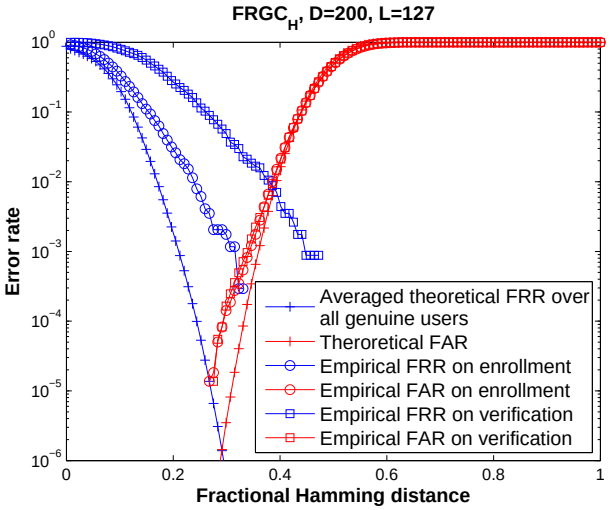


Figure 4.20: *Comparing the theoretical FAR vs. FRR performances with the FAR vs. FRR performances on the enrollment and verification data.*

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 provide 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.

4.2.7 Conclusion

Binary biometric representations are becoming popular for its benefits in template compression and protection. Quantization and coding is 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 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.

4.3 Chapter conclusion

In this chapter, a bit allocation principle AUF-OBA is presented. Regarding the research objectives, AUF-OBA is able to allocate a user-dependent number of bits to every biometric feature, while maintaining a fixed total length of the binary string. The extracted binary string has *i.i.d.* bits. Superior to DROBA, which optimizes the theoretical FRR performance at zero Hamming distance threshold, AUF-OBA optimizes the overall FRR over all Hamming distance thresholds. As a result, independent of quantizers, AUF-OBA yields better overall FAR and FRR performances. Furthermore, the extracted bits are more reliable, which allows for a longer length of the random key, or equivalently, the length of the secret.

5

Weighted Area under the FRR Curve Optimized Bit Allocation

5.1 Chapter introduction

PURPOSE. In Chapter 3 and 4, two bit allocation principles were presented to assign various numbers of bits according to a Hamming distance classifier (HDC): the detection rate optimized bit allocation (DROBA) and the area under the FRR curve optimized bit allocation (AUF-OBA). Both principles aim to optimize the recognition performances for a HDC: DROBA optimizes the theoretical FRR performance when the Hamming distance threshold is zero, while AUF-OBA optimizes the area under the FRR curve over all the Hamming distance thresholds. Although AUF-OBA is superior to DROBA by considering a wider range of Hamming distance thresholds, often in biometric applications, only a partial, rather than the entire range of the Hamming distance thresholds is important. Therefore, the purpose of this chapter is to provide a bit allocation principle that optimizes the area under the FRR curve with an emphasis on a certain range of Hamming distance thresholds. Similar to DROBA and AUF-OBA, this new bit allocation principle should be independent of the quantizers. Given independent features and quantizers that can extract statistically independent and identically distributed (*i.i.d.*) bits, the new bit allocation principle should preserve the *i.i.d.* property.

CONTENTS. In this chapter, we present a weighted area under the FRR curve optimized bit allocation (WAUF-OBA) principle, where the area is emphasized with an exponential weight function. Based on the bit error probabilities of the biometric features, a theoretical HDC model predicts the FAR and FRR performance

at the Hamming distance threshold t. WAUF-OBA then aims to optimize the area of the FRR performances over all possible Hamming distance thresholds, with an exponential weight function z^{-t} for each threshold t. Parameter $0 < z < 1$ emphasizes the FRR performances in the range of large Hamming distance threshold t, whereas $z > 1$ emphasizes the FRR performances in the range of small t. A dynamic programming approach is then applied to search for the optimal solution. WAUF-OBA is a generalization of the DROBA when $z = 1$, or AUF-OBA when $z \to \infty$. WAUF-OBA can be applied in both one- and two-dimensional quantization schemes. In this chapter, WAUF-OBA is presented in combination with the one-dimensional quantizers, as illustrated in Fig. 5.1. Figure 5.2 shows the contribution of this chapter in the context of the thesis.

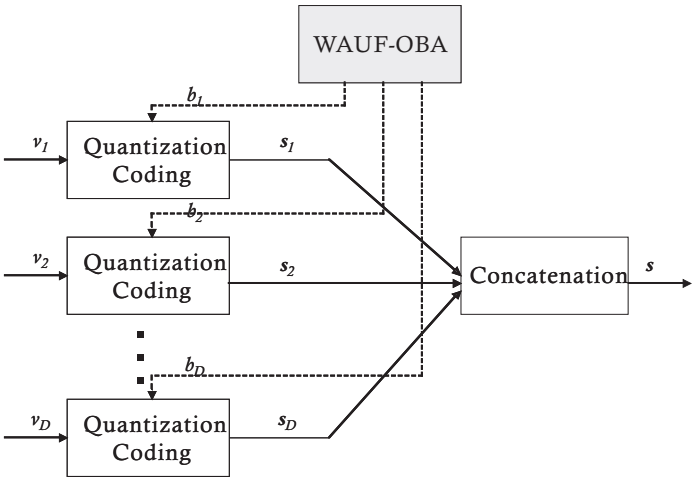


Figure 5.1: *Block diagram of the one-dimensional quantization and coding scheme, highlighted in WAUF-OBA design. The $v_i, i = 1 \dots D$ denote D independent biometric features, and b_i denotes the assigned number of bits to the i^{th} feature. The quantized bits $s_i, i = 1 \dots D$ from all D features are then concatenated into the binary string s.*

PUBLICATION(S). The content of Section 5.2 has been published in [55].

5.2 Extracting biometric binary strings with optimal weighted area under the FRR curve for the Hamming distance classifier

Abstract

Binary biometric representations are becoming popular for their benefits in template

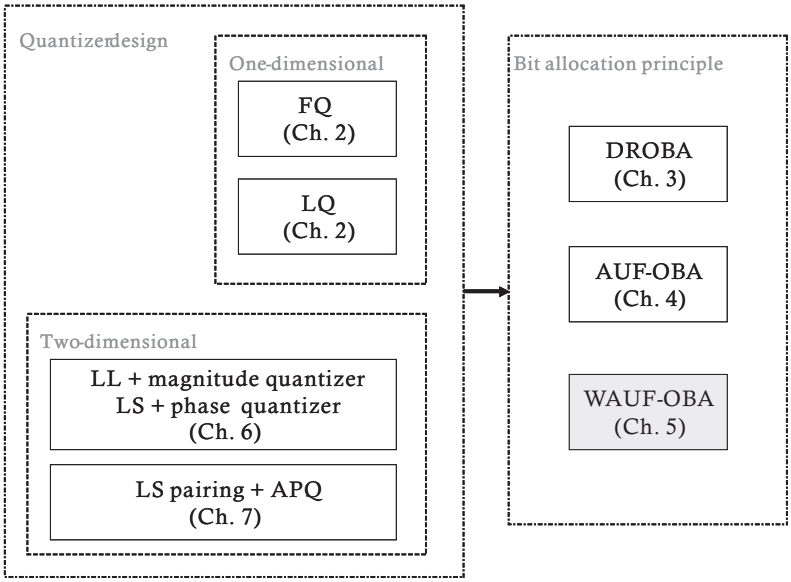


Figure 5.2: *Block diagram of the main contributions, highlighted in chapter* 5.

compression and protection. The straightforward method to extract the binary biometric strings is by quantization and coding the real-valued features. Afterwards, the binary strings are compared by means of a Hamming distance classifier (HDC). In this paper, we first give expressions for the theoretical false acceptance rate (FAR) and false-rejection rate (FRR) of the HDC, based on the features' bit error probability after the quantization. One of the problems of binary biometric representations is the allocation of quantization bits to the biometric features. Therefore, we propose a bit allocation principle (WAUF-OBA) that minimizes the exponentially weighted area under the FRR curve. We show that this method is a generalization of the bit allocation principles that minimize the area under the FRR curve (AUF-OBA), or the FRR at zero Hamming distance threshold (DROBA).

5.2.1 Introduction

Binary biometric representations are used in template compression and protection [4]. The binary strings should result in a low false acceptance rate (FAR) and false rejection rate (FRR). Additionally, the bits should be independent and identically distributed (*i.i.d*), in order to maximize the efforts of guessing the genuine template.

The common way to extract binary strings is quantizing and coding the real-valued features: Firstly, by means of the PCA/LDA transformation, independent features are extracted from the raw measurements. Afterwards, features are quantized individually. The final binary string is then the concatenation of the bits extracted from every feature. To obtain *i.i.d* bits, equal-probability quantizers have been proposed

[22], [23], [39]. Furthermore, independent of the quantizer design, several bit allocation principles have been proposed to determine the number of quantization bits for every feature. Based on the feature density distribution, a detection rate optimized bit allocation principle (DROBA) [56] was proposed to minimize the theoretical FRR at zero Hamming distance threshold. Unfortunately, it does not correspond to the operational points with a FAR between 10^{-4} and 10^{-2}. Therefore, an area under the FRR curve optimized bit allocation principle (AUF-OBA) was proposed to minimize the theoretical area under the FRR curve for the Hamming distance classifier (HDC).

In this paper, we propose a weighted area under the FRR curve optimized bit allocation principle (WAUF-OBA), where the area is emphasized with an exponential weight function. We first show in Section 5.2.2 that given the features' bit error probabilities after the quantization, we can predict the theoretical FAR/FRR as well as the weighted area under the FRR curve for the HDC. Then, in Section 5.2.3, we present the bit allocation solution that minimizes this area, and we prove that WAUF-OBA is a generalization of DROBA and AUF-OBA. In Section 5.2.4, we give some simulation results on the synthetic data and conclusions are drawn in Section 5.2.5.

5.2.2 Hamming Distance Classifier (HDC)

Suppose a sequence of L bits is extracted from D independent features, i.e. $\sum_{i=1}^{D} b_i = L$, where b_i bits are extracted from the i^{th} feature. 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 $\boldsymbol{s}_{\text{g}} = s_{\text{g},1} \dots s_{\text{g},D}$. 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 strings. Then we can define the bit error probabilities for $\boldsymbol{s}'_{\text{g}}$ and $\boldsymbol{s}'_{\text{i}}$ for the i^{th} feature:

$$P_{\text{g},i}(k_i; b_i) \stackrel{\text{def}}{=} \mathcal{P}\{d_H(s_{\text{g},i}, s'_{\text{g},i}) = k_i\} \ , \tag{5.1}$$

$$P_{\text{i},i}(k_i; b_i) \stackrel{\text{def}}{=} \mathcal{P}\{d_H(s_{\text{g},i}, s'_{\text{i},i}) = k_i\}, k_i \in 0, \dots, b_i \ , \tag{5.2}$$

where d_H is 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 in the b_i bits extracted for the i^{th} feature during the verification.

Regarding a total of D features, we define:

$$\phi_{\text{g}}(k; \{b_i\}) \stackrel{\text{def}}{=} \mathcal{P}\{d_H(\boldsymbol{s}_{\text{g}}, \boldsymbol{s}'_{\text{g}}) = k\} \ , \tag{5.3}$$

$$\phi_{\text{i}}(k; \{b_i\}) \stackrel{\text{def}}{=} \mathcal{P}\{d_H(\boldsymbol{s}_{\text{g}}, \boldsymbol{s}'_{\text{i}}) = k\}, k \in 0, \dots, L \ , \tag{5.4}$$

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 in the entire L bits extracted during the verification. Assuming independency among the features, the error probability of the whole feature set equals the convolution of the individual probabilities. Thus:

$$\phi_{\text{g}}(k; \{b_i\}) = (P_{\text{g},1} * P_{\text{g},2} * \dots * P_{\text{g},D})(k; \{b_i\}) \ , \tag{5.5}$$

$$\phi_{\text{i}}(k; \{b_i\}) = (P_{\text{i},1} * P_{\text{i},2} * \dots * P_{\text{i},D})(k; \{b_i\}) \ . \tag{5.6}$$

Expressions in (5.5) and (5.6) are defined as the bit error probabilities of the binary string for the genuine user and the imposters. Based on these, we can compute the analytical FAR and FRR performances of the HDC. Thus, the FAR ($\alpha(t;\{b_i\})$ at the Hamming distance threshold t is:

$$\begin{aligned}\alpha(t;\{b_i\}) &\stackrel{\text{def}}{=} \mathcal{P}\{d_H(\boldsymbol{s}_{\text{g}},\boldsymbol{s}'_{\text{i}}) \le t\}\ , \\ &= \sum_{k=0}^{t}\phi_{\text{i}}(k;\{b_i\})\ , \qquad i=1,\ldots,D\ . \end{aligned} \tag{5.7}$$

Furthermore, to obtain *i.i.d* bits, an equal-probability quantizer [22], [23], [39], with 2^{-b_i} probability mass for every interval, is required for the quantization of every feature. Thus, for the i^{th} feature, when assigned with 2^{b_i} code words, the $P_{\text{i},i}(k_i;b_i)$, as defined in (5.2), becomes:

$$P_{\text{i},i}(k_i;b_i) = 2^{-b_i}\left(\begin{array}{c} b_i \\ k_i \end{array}\right)\ . \tag{5.8}$$

Subject to $\sum_{i=1}^{D} b_i = L$, the FAR in (5.7) becomes:

$$\alpha(t;\{b_i\}) \;=\; 2^{-L}\sum_{k=0}^{t}\left(\begin{array}{c} L \\ k \end{array}\right)\ . \tag{5.9}$$

This expression proves that when quantized by an equal-probability quantizer, the FAR depends on the string length L and becomes independent of the bit assignment $\{b_i\}$.

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

$$\begin{aligned}\beta(t;\{b_i\}) &\stackrel{\text{def}}{=} \mathcal{P}\{d_H(\boldsymbol{s}_{\text{g}},\boldsymbol{s}'_{\text{g}}) > t\}\ , \\ &= \sum_{k=t+1}^{L}\phi_{\text{g}}(k;\{b_i\})\ , \qquad i=1,\ldots,D\ . \end{aligned} \tag{5.10}$$

5.2.3 Weighted area under the FRR curve optimized bit allocation (WAUF-OBA)

Given the analytical FRR performances in (5.10), we compute the weighted area under the FRR curve as a criterion for the overall HDC performance. Furthermore, the area depends on the bit assignment $\{b_i\}$. Therefore, in this section, we give a $\{b_i\}$ solution that minimizes the weighted area under the FRR curve.

5.2.3.1 Problem Formulation

The optimization problem is user dependent. 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 are known. The i^{th}

feature is quantized into b_i bits, $i = 1, \ldots, D$, $b_i \in \{0, \ldots, b_{\max}\}$. The problem is then formulated as:

$$\{b_i^*\} \quad = \quad \arg \min_{\sum_{i=1}^{D} b_i = L} \sum_{t=0}^{L} z^{-t} \beta(t; \{b_i\}), \quad z > 0 \ , \tag{5.11}$$

where z^{-t} is an exponential weight function. $0 < z < 1$ emphasizes the FRR performances in the range of large Hamming distance threshold t. Contrarily, $z > 1$ emphasizes the FRR performances in the range of small t.

In Appendix D, we give the solution of (5.11), with respect to the value of z. In the following part of this section, we present the simplified problem, which can by solved by a dynamic programming approach.

5.2.3.2 WAUF-OBA: $0 < z < 1$ or $z > 1$

Let G_i denote a gain factor for the i^{th} feature when quantized into b_i bits:

$$G_i(b_i) = -\left| \log \left(\sum_{k_i=0}^{b_i} z^{-k_i} P_{\text{g},i}(k_i, b_i) \right) \right| . \tag{5.12}$$

The $G_i(b_i)$ is the logarithm of a weighted sum over the probability $P_{\text{g},i}(k_i, b_i)$. Therefore, we reformulate (5.11) into:

$$\{b_i^*\} = \arg \max_{\sum_{i=1}^{D} b_i = L} \sum_{i=1}^{D} G_i(b_i) \ . \tag{5.13}$$

By maximizing the sum of G_i over all the D features subject to L bits, we in fact minimize a weighted overall probability of producing bit errors for the genuine user.

Computing the gain factor G_i relies on the $P_{\text{g},i}(k_i; b_i)$, as defined in (5.1). Given the real-valued genuine user PDF $p_{\text{g},i}$ as well as a quantizer, we can compute $P_{\text{g},i}(k_i; b_i)$ as:

$$\begin{aligned} P_{\text{g},i}(k_i; b_i) \quad &\stackrel{\text{def}}{=} \quad \mathcal{P}\{d_H(s_{\text{g},i}, s'_{\text{g},i}) = k_i\} \ , \\ &= \quad \int_{Q(k_i; b_i)} p_{\text{g},i}(v) dv \ , \end{aligned} \tag{5.14}$$

where $Q(k_i; b_i)$ indicates the quantization intervals with k_i-bit error as compared to the genuine code $s_{\text{g},i}$. After G_i is determined, a common dynamic programming procedure, as described in [56], is applied to search for the optimal $\{b_i\}$ solution.

5.2.3.3 WAUF-OBA: $z = 1$

AUF-OBA is a particular case of WAUF-OBA when $z = 1$. Thus the points on the FRR curve are treated equally. We define the gain factor $G_i(b_i)$ as:

$$G_i(b_i) = - \sum_{k_i=0}^{b_i} k_i P_{\text{g},i}(k_i, b_i) \ . \tag{5.15}$$

With this definition, we reformulate the problem into (5.13). Note that WAUF-OBA ($z = 1$) can also be interpreted as to minimize the expected value of the total number of error bits.

5.2.3.4 WAUF-OBA: $\mathbf{z \to \infty}$

DROBA is an extreme case of WAUF-OBA when $z \to \infty$. Thus the FRR performance is only minimized at zero Hamming distance threshold ($t = 0$). We define the gain factor G_i as:

$$G_i(b_i) \quad = \quad \log P_{\mathrm{g},i}(0, b_i) \ . \tag{5.16}$$

Only the $P_{\mathrm{g},i}(k_i; b_i)$ with $k_i = 0$ determines the value of the gain factor G_i, and with this definition we reformulate the problem into (5.13).

5.2.4 Evaluation on synthetic data

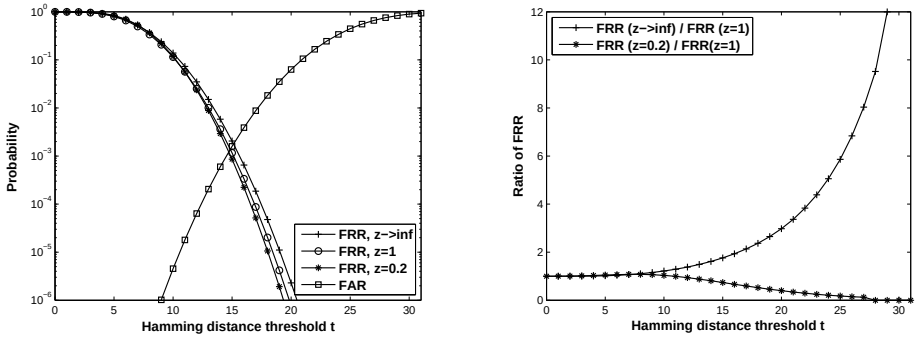
In this section we test the HDC performances of the bit strings extracted with WAUF-OBA, on randomly generated synthetic features. The background PDF of every feature is generated as a Gaussian density with zero-mean and unit-variance, i.e. $p_{\mathrm{b},i} = N(v, 0, 1)$. Additionally, the genuine user PDF of every feature is generated as a Gaussian density with user-specific mean and standard deviation, i.e. $p_{\mathrm{g},i} = N(v, \mu_i, \sigma_i)$. The quantizer that we employed to compute $P_{\mathrm{g},i}(k_i; b_i)$ is the user-independent equal-probability quantizer [22], [23], [39], defined as :

$$B_0 \quad = \quad -\infty \ , \tag{5.17}$$

$$B_j \quad = \quad \arg_B \left[\int_{B_{j-1}}^{B_j} p_{\mathrm{b},i} dv = 2^{-b_i} \right], j = 1, \ldots, 2^{b_i} \ , \tag{5.18}$$

where $(B_{j-1}, B_j]$ represents the j^{th} 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 $\{b_i\}$ through the dynamic programming process in [56]. Afterwards, we compute the theoretical FAR/FRR performances for HDC from (5.9) and (5.10).

In the earlier work of DROBA ($z \to \infty$) [56], the FAR/FRR performances with respect to the D/L settings have been analyzed: There exists a good choice at a moderate length L when D is fixed. Furthermore, when L is fixed, increasing D will not decrease the performance. Such properties are consistent in the general case ($z > 0$). Therefore, in this paper, we only present the properties in terms of z values at a fixed D/L setting, e.g. $D = 50$, $L = 50$, as seen in Fig. 5.3. The minimized area under the FRR curve are 0.240 for $z \to \infty$, 0.236 for $z = 1$, and 0.261 for $z = 0.2$, respectively. In general, $z = 1$ gives the best overall FRR curve. Additionally, Fig. 5.3(b) shows that $z = 0.2$ gives lower FRR performance at larger t. Unfortunately, $z \to \infty$ does not present a considerably lower FRR performance at smaller t.



(a) (b)

Figure 5.3: *(a) The FAR/FRR performances with different z values, and (b) the ratio of their FRR performances, at* $D = 50$*,* $L = 50$*.*

5.2.5 Conclusion

In this paper, we first give a theoretical model of the Hamming distance classifier (HDC), based on the features' bit error probabilities after the quantization. This model predicts the FAR and the FRR as a function of the Hamming distance threshold. Additionally, we introduce an exponential weight function to formalize the analytical weighted area under the FRR curve. One of the problems in binary biometric representations is the allocation of quantization bits to the features. Therefore, we propose a bit-allocation principle (WAUF-OBA) that minimizes the exponentially weighted area under the FRR curve. We show that this method is a generalization of the bit allocation principles that minimize the area under the FRR curve (AUF-OBA), or the FRR at Hamming distance threshold zero (DROBA).

5.3 Chapter conclusion

In this chapter, a bit allocation principle WAUF-OBA is presented. Regarding the research objectives, WAUF-OBA is able to allocate a user-dependent number of bits to every biometric feature, while maintaining a fixed total length of the binary string. The extracted binary string has *i.i.d.* bits. Superior to DROBA and AUF-OBA, WAUF-OBA optimizes the overall FRR performances with an exponential weighting function. As a result, independent of quantizers, WAUF-OBA yields better overall FAR and FRR performances at the emphasized range of Hamming distance thresholds. Furthermore, the extracted bits are more reliable, which allows for a longer length of the random key.

6

Two-dimensional Polar Quantizer

6.1 Chapter introduction

PURPOSE. In Chapter 2, one-dimensional quantizers, the fixed quantizer (FQ) and the likelihood ratio based quantizer (LQ), are presented. Both quantizers are capable of extracting multiple statistically independent and identically distributed (*i.i.d.*) bits. Superior to FQ, LQ determines quantization intervals with the maximized theoretical detection rate, given a prescribed fixed number of quantization bits per feature. Although one-dimensional quantizers yield reasonably good FAR and FRR performances, quantizing every feature independently ends up with inflexible quantization intervals, for instance, orthogonal boundaries in the two-dimensional feature space. Therefore, the purpose of this chapter is to design two-dimensional quantizers. The two-dimensional quantizers should be capable of extracting multiple *i.i.d.* bits. Furthermore, after every feature pair is quantized into a prescribed number of bits, the concatenated binary strings should result in good recognition performance, when applied to a Hamming distance classifier (HDC).

CONTENTS. In this chapter, we propose quantizing the phase and the magnitude of pairwise features in polar coordinates. As shown in Fig. 6.1, the phase and the magnitude quantization are applied mutually exclusively, and for each a strategy is designed to compose the feature pairs. Given a feature pair, both the phase and the magnitude quantizers are user-independent. Thus, the quantization intervals are merely determined by equally dividing the probability mass of the background probability density function (PDF), in the phase or the magnitude domain. The Gray codes, with only one single bit Hamming distance between any two adjacent codewords, are then assigned to the quantization intervals. This reduces the number

of erroneous bits due to the intra-class variation. Since the quantization intervals have equal background probability, the bits assigned to each feature pair are *i.i.d.*. Moreover, if the biometric features are statistically independent, the composed feature pairs are independent as well, the bits in the entire binary string are then *i.i.d.*.

Furthermore, quantizing two-dimensional features allows user-dependent configuration of the feature pairs, which can further optimize the overall recognition performances of the binary strings. We present this procedure in two steps. First, the optimization problem is formulated as follow: We know that the performance of a HDC depends on how a Hamming distance threshold could separate two densities: the genuine Hamming distance (GHD) density and the imposter Hamming distance (IHD) density. Therefore, optimizing the performance of a HDC is equivalent to optimizing the ratio between their intra- and inter-class scatters. Although it is not feasible to find an analytical pairing solution for this problem, we find that the problem can be approximated as an optimization of the overall distance between the feature pairs and the origin, each for phase and magnitude. In the second step, to solve this simplified problem, we develop two heuristic pairing strategies: A long-short (LS) pairing strategy, which combines feature pairs with a large mean and a small mean, is designed for phase. Alternatively, a long-long (LL) pairing strategy, which selects features that either both have large means or both have small means, is designed for magnitude. The pairing is applied to every enrolled user. As a result, the pairing configurations are user-dependent. Figure 6.2 shows the contribution of this chapter in the context of the thesis.

PUBLICATION(S). The content of Section 6.2 has been published in [57].

6.2 Binary biometric representation through pairwise polar quantization

Abstract

Binary biometric representations have great significance for data compression and template protection. In this paper, we introduce pairwise polar quantization. Furthermore, aiming to optimize the discrimination between the genuine Hamming distance (GHD) and the imposter Hamming distance (IHD), we propose two feature pairing strategies: the long-short (LS) strategy for phase quantization, as well as the long-long (LL) strategy for magnitude quantization. Experimental results for the FRGC face database and the FVC2000 fingerprint database show that phase bits provide reasonably good performance, whereas magnitude bits obtain poor performance.

6.2.1 Introduction

Binary biometric representations have great significance for data compression and template protection [4]. A common way to extract binary strings is by quantizing

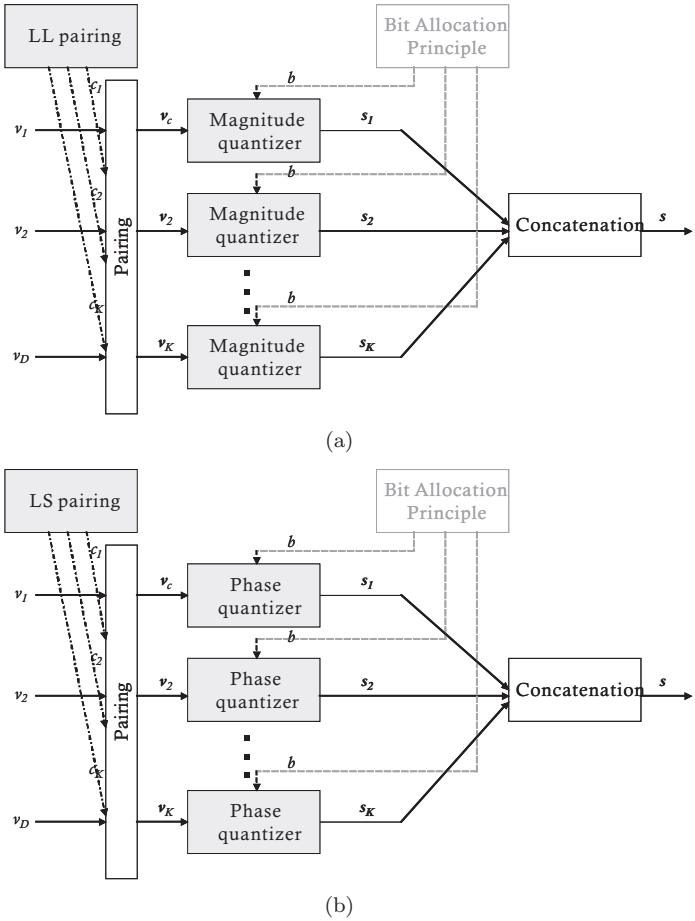


Figure 6.1: *Block diagram of the two-dimensional polar quantization: (a) the magnitude quantization with LL pairing strategy and (b) the phase quantization with LS pairing strategy. The $v_i, i = 1 \dots D$ denote D independent biometric features, which are composed into K feature pairs. The $c_i, i = 1 \dots K$ indicates the configuration for the i^{th} feature pair. Since bit allocation (in gray) is not in scope of this chapter, every feature is prescribe to a fixed length of b-bit. The quantized bits $s_i, i = 1 \dots K$ from all K feature pairs are then concatenated into the binary string s.*

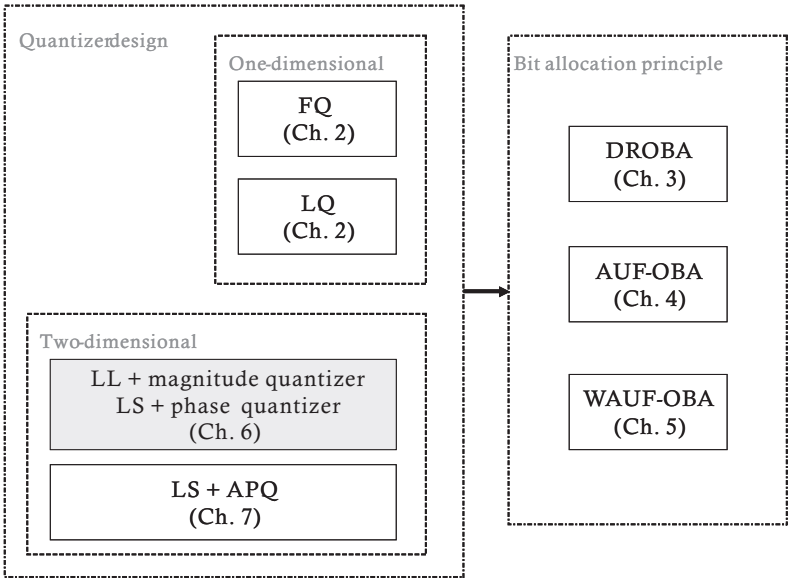


Figure 6.2: *Block diagram of the main contributions, highlighted in chapter* 6.

and coding the real-valued biometric templates. The binary string has to suffice the following two requirements:

1. To obtain a verification performance with low false acceptance rate (FAR) and false rejection rate (FRR), the binary strings are desired to be not only discriminative, but also robust to intra-class variation;

2. Considering template protection, the bits generated by the imposters should be independent and identically distributed, in order to maximize the efforts of guessing the genuine template.

To achieve the requirements, many work focus on designing one-dimensional quantizers, relying on the feature's statistical properties, e.g. the genuine user probability density function (PDF) p_{g} and the background PDF p_{b} [8], [46], [9], [22], [23], [39]. Among them the fixed quantizer [22], [23] is global, constructed merely from the background PDF, whereas quantizers in [8], [46], [9], [39] are user-specific, constructed from both the genuine user PDF and the background PDF. Quantizers in [8], [46] and [9] have equal-width-intervals. Alternatively, quantizers in [22], [23], [39] have equal-probability-intervals. Furthermore, independent of the one-dimensional qantizers, the DROBA principle [56] was proposed to assign various numbers of quantization bits to every feature. In this paper we concentrate on the quantizer design. Although one-dimensional quantizers yield reasonably good performances, quantizing every feature independently ends up with inflexible quantization intervals, for instance, orthogonal boundaries in the two-dimensional feature space. Therefore, two-dimensional quantization might bring more flexible quantizer structures.

In this paper, we propose quantization in polar coordinates, including phase and magnitude. Unlike in audio and image coding [58], where polar quantization is directly applied on predetermined complex variables, quantizing biometric features allows freedom to compose the pairwise features. Furthermore, we define a measure of the discrimination between the quantized bits, by computing the inter-class and intra-class scatters between the genuine and the imposter Hamming distances. To optimize such discrimination, we develop the long-short (LS) and the long-long (LL) pairing strategies for phase and magnitude, respectively.

In Section 6.2.2 the polar quantization is introduced. In Section 6.2.3 we propose the long-short and the long-long pairing strategies, to optimize the discrimination between the genuine and the imposter Hamming distances. In Section 6.2.4, some experimental results are given for the FRGC face database and the FVC2000 fingerprint database, and conclusions are drawn in Section 6.2.5.

6.2.2 Polar quantization

Let $\mathbf{v} = \{v_1, v_2\}$ denote a two-dimensional feature vector. In polar coordinates, the phase θ and magnitude r are:

$$\theta = \text{angle}(v_1, v_2) \,, \tag{6.1}$$

$$r = \sqrt{v_1^2 + v_2^2} \,, \tag{6.2}$$

where θ is the counterclockwise angle from the v_1-axis, and r is the radial distance from the origin. We assume that biometric features have circularly symmetric background PDF, feasible for polar quantization. A n-bit phase quantizer is then constructed as:

$$\xi = \frac{2\pi}{2^n} \,, \tag{6.3}$$

$$Q_{\theta,i} = [(i-1)\xi \quad i\xi),\ i = 1, \ldots 2^n \,, \tag{6.4}$$

where $Q_{\theta,i}$ represents the i^{th} quantization interval within boundaries $[(i-1)\xi \quad i\xi)$. When the background PDF is circularly symmetric, θ is uniformly distributed, leading to both equal-ξ-width and equal-2^{-n}-probability intervals.

A n-bit magnitude quantizer is constructed as:

$$B_0 = 0 \,, \tag{6.5}$$

$$B_i = \arg_B \left[\int_{B_{i-1}}^{B} \int_0^{2\pi} p_{\text{b}}(\theta, r) d\theta dr = 2^{-n} \right],\ i = 1, \ldots 2^n \,, \tag{6.6}$$

$$Q_{\text{r},i} = [B_{i-1} \quad B_i),\ i = 1, \ldots 2^n \,, \tag{6.7}$$

where $Q_{\text{r},i}$ represents the i^{th} quantization interval within boundaries $[B_{i-1} \quad B_i)$. Determining these intervals depends on the background PDF p_{b}. The expression in (6.6) ensures equal-2^{-n}-probability intervals.

To summarize, both phase and magnitude quantization obtain equal background probability intervals. Thus, the imposters obtain independent and identically distributed bits.

6.2.3 Feature pairing

6.2.3.1 Hamming Distance Discriminant Analysis

Often, binary biometric strings are matched via their Hamming distances. To design a verification system, a genuine Hamming distance (GHD) is computed when the query and the target share the same identity, otherwise an imposter Hamming distance (IHD) is computed. The decision is then made by applying a threshold T to both distances, as illustrated in Fig. 6.3. In this paper, we aim to optimize the discrimination between the GHD and IHD densities.

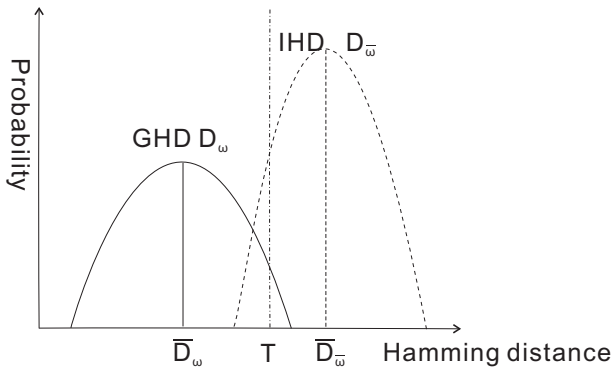


Figure 6.3: *The genuine Hamming distance (GHD) density and the imposter Hamming distance (IHD) density in a biometric verification system.*

We begin by defining the discrimination between GHD and IHD. Suppose we have P feature pairs. Each pair is quantized as a code $x_i, i = 1, \ldots, P$, and subsequently concatenated into the binary string $X = x_1, \ldots, x_P$. Considering a genuine user ω, with $\widehat{X}_{\omega} = \hat{x}_{\omega,1}, \ldots, \hat{x}_{\omega,P}$ as the enrollment template. Let $d_{\omega,i}$ and $d_{\bar{\omega},i}$ be the GHD and IHD for the i^{th} feature pair, defined as:

$$d_{\omega,i} = h(x_i, \hat{x}_{\omega,i}),\ x_i \in \text{user } \omega\ ; \tag{6.8}$$

$$d_{\bar{\omega},i} = h(x_i, \hat{x}_{\omega,i}),\ x_i \notin \text{user } \omega\ , \tag{6.9}$$

where function h computes the Hamming distance between the two inputs. Then the GHD (D_{ω}) and the IHD ($D_{\bar{\omega}}$) for the entire binary string are:

$$D_{\omega} = \sum_{i=1}^{P} d_{\omega,i}\ ; \tag{6.10}$$

$$D_{\bar{\omega}} = \sum_{i=1}^{P} d_{\bar{\omega},i}\ . \tag{6.11}$$

Furthermore, for the i^{th} feature pair, we define the expectation of the GHD ($\bar{d}_{\omega,i}$)

and the IHD ($\bar{d}_{\bar{\omega},i}$) as:

$$\bar{d}_{\omega,i} = \mathcal{E}[d_{\omega,i}],\ x_i \in \text{user } \omega\ ; \tag{6.12}$$

$$\bar{d}_{\bar{\omega},i} = \mathcal{E}[d_{\bar{\omega},i}],\ x_i \notin \text{user } \omega\ . \tag{6.13}$$

Thus, for the entire binary string, the expectation of the GHD ($\overline{D}_{\omega}$) and the IHD ($\overline{D}_{\bar{\omega}}$) become:

$$\overline{D}_{\omega} = \sum_{i=1}^{P} \bar{d}_{\omega,i}\ ; \tag{6.14}$$

$$\overline{D}_{\bar{\omega}} = \sum_{i=1}^{P} \bar{d}_{\bar{\omega},i}\ . \tag{6.15}$$

A measure of separation between GHD and IHD densities is their intra-class scatter and the inter-class scatter. Thus, we would like to minimize the intra-class scatter $S_{\text{intra},\omega}$, defined as:

$$S_{\text{intra},\omega} = \mathcal{E}[(D_{\omega} - \overline{D}_{\omega})^2] + \mathcal{E}[(D_{\bar{\omega}} - \overline{D}_{\bar{\omega}})^2]\ . \tag{6.16}$$

In the mean time, we want to maximize the inter-class scatter $S_{\text{inter},\omega}$:

$$S_{\text{inter},\omega} = (\overline{D}_{\bar{\omega}} - \overline{D}_{\omega})^2\ . \tag{6.17}$$

Substituting (6.10), (6.11), (6.14), (6.15) into (6.16) and (6.17), we have:

$$S_{\text{intra},\omega} = \mathcal{E}\left[\left[\sum_{i=1}^{P}(d_{\omega,i} - \bar{d}_{\omega,i})\right]^2\right] + \mathcal{E}\left[\left[\sum_{i=1}^{P}(d_{\bar{\omega},i} - \bar{d}_{\bar{\omega},i})\right]^2\right]\ ;$$

$$S_{\text{inter},\omega} = \left[\sum_{i=1}^{P}(\bar{d}_{\bar{\omega},i} - \bar{d}_{\omega,i})\right]^2\ .$$

Assuming that the P feature pairs are independent, $S_{\text{intra},\omega}$ and $S_{\text{inter},\omega}$ can be approximated as:

$$S_{\text{intra},\omega} = \sum_{i=1}^{P}\left[\mathcal{E}\left[(d_{\omega,i} - \bar{d}_{\omega,i})^2\right] + \mathcal{E}\left[(d_{\bar{\omega},i} - \bar{d}_{\bar{\omega},i})^2\right]\right] = \sum_{i=1}^{P} S_{\text{intra},\omega,i}\ ; \tag{6.18}$$

$$S_{\text{inter},\omega} = \sum_{i=1}^{P}(\bar{d}_{\bar{\omega},i} - \bar{d}_{\omega,i})^2 = \sum_{i=1}^{P} S_{\text{inter},\omega,i}\ . \tag{6.19}$$

Hence, the intra-/inter-class scatter for the entire binary string is simplified as the sum of the intra-/inter-class scatter over all the feature pairs. Usually biometric features are not presented in pairs, allowing the freedom to compose the pairwise features to optimize $S_{\text{intra},\omega}$ and $S_{\text{inter},\omega}$. Therefore, we formulate the problem as: for a genuine

user ω, to find a pairing configuration $\{C^*_{\omega,i}\}, i = 1, \ldots, P$, so that the discrimination between GHD and IHD is maximized:

$$\{C^*_{\omega,i}\} = \arg\max_{\{C_{\omega,i}\}} \frac{S_{\text{inter},\omega}}{S_{\text{intra},\omega}} , \tag{6.20}$$

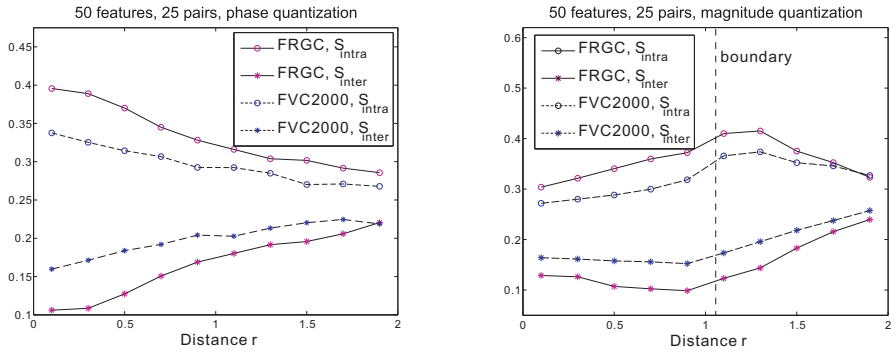
$$= \arg\max_{\{C_{\omega,i}\}} \frac{\sum_{i=1}^{P} S_{\text{inter},\omega,i}}{\sum_{i=1}^{P} S_{\text{intra},\omega,i}} . \tag{6.21}$$

6.2.3.2 Pairing solutions

In order to optimize (6.21), we first have to analyze how the genuine user PDF relates to $S_{\text{inter},\omega,i}$ and $S_{\text{intra},\omega,i}$ in case of the polar quantization. Afterwards, we could decide a strategy to pair features with specific p_{g} to optimize $S_{\text{inter},\omega,i}$ and $S_{\text{intra},\omega,i}$. However, it is difficult to analytically find an expression for the relation, due to lack of samples and complex integral calculation in polar coordinates. Therefore, we employ an empirical method to simplify the relation of $S_{\text{inter},\omega,i}$, $S_{\text{intra},\omega,i}$ and the genuine user PDF. We take two data sets: FRGC(version 1) [37] face database and FVC2000(DB2) fingerprint database [35].

- **FRGC**: It contains 275 users with various numbers of images, taken under both controlled and uncontrolled conditions. A set of standard landmarks, i.e. eyes, nose and mouth, are used to align the faces. The raw measurements are the gray pixel values, leading to a total of 8762 elements.
- **FVC2000**: It contains 8 images of 110 different users. Images are aligned according to a standard core point position. The raw measurements contain two categories: the squared directional field in both x and y directions, and the Gabor response in 4 orientations (0, $\pi/4$, $\pi/2$, $3\pi/4$). Determined by a regular grid of 16 by 16 points with spacing of 8 pixels, measurements are taken at 256 positions, leading to a total of 1536 elements [22].

We first apply PCA/LDA [42] to reduce both data sets into 50 features. Afterwards, for every genuine user, we randomly pair the features into 25 pairs. Following this pairing configuration, the entire data set, including the genuine user samples and the imposter samples are quantized via a 1-bit phase quantizer and a 1-bit magnitude quantizer, where the magnitude quantizer boundary is determined by a two-dimensional Gaussian density with zero mean and unit variance $p_{\text{b}}(\mathbf{v}) = N(\mathbf{v}, 0, 1)$. The $S_{\text{inter},\omega,i}$ and $S_{\text{intra},\omega,i}$ for every feature pair is then computed based on the quantized bits. We repeat this process for all the genuine users in the data set. Eventually, we average $S_{\text{inter},\omega,i}$ and $S_{\text{intra},\omega,i}$ over all features as well as all genuine users, so that the averaged $\overline{S}_{\text{inter},\omega,i}$ and $\overline{S}_{\text{intra},\omega,i}$ are neither user nor feature biased. Intuitively, we speculate that the distance $r_{\omega,i}$, – distance between the feature pair mean and the origin – dominates the inter- and intra-class scatter. To analyze, in Fig. 6.4, we plot the value of $\overline{S}_{\text{inter},\omega,i}$, $\overline{S}_{\text{intra},\omega,i}$ as sorted by $r_{\omega,i}$, for both phase bits and magnitude bits. Both data sets reveal the same relations: Fig. 6.4(a) suggests that for phase quantization, when $r_{\omega,i}$ increases, the inter-class scatter increases and the intra-class scatter



(a) (b)

Figure 6.4: *The averaged* $\overline{S}_{\text{inter},\omega,i}$ *and* $\overline{S}_{\text{intra},\omega,i}$ *for (a) phase and (b) magnitude, as sorted by* $r_{\omega,i}$.

decreases; Fig. 6.4(b) suggests that for magnitude quantization, when the distance $\mid r_{\omega,i} - B \mid$ between the feature pair mean and the magnitude boundary increases, the inter-class scatter increases and the intra-class scatter decreases. Therefore, we simplify the problem (6.21) as:

$$\{C^*_{\omega,i}\} = \arg\max_{\{C_{\omega,i}\}} \sum_{i=1}^{P} r_{\omega,i}, \text{ for phase }, \tag{6.22}$$

$$\{C^*_{\omega,i}\} = \arg\max_{\{C_{\omega,i}\}} \sum_{i=1}^{P} \mid r_{\omega,i} - B \mid, \text{ for magnitude }. \tag{6.23}$$

Optimizing the inter- and intra-class scatter is now simplified as optimizing the overall distance of the feature pairs. To solve (6.22) and (6.23), the straightforward way is to conduct a brute force search of all possible pairing configurations and pick the one with the maximum overall distance. Unfortunately, the computational complexity is too high. Therefore, we propose the following two heuristic pairing strategies: Given $2P$ features, we first sort the mean of the $2P$ features $\{\text{abs}(\mu_k)\}$, $k = 1, \ldots, 2P$ from the smallest to the largest into a sequence of ordered feature indexes $\{I_1, I_2, \ldots, I_{2P}\}$.

1. **long-short (LS) strategy**: The index of the i^{th} feature pair is:

$$C^*_{\omega,i} = (I_i, I_{2P+1-i}),\ i = 1, \ldots, P\ ; \tag{6.24}$$

2. **long-long (LL) strategy**: The index of the i^{th} feature pair is:

$$C^*_{\omega,i} = (I_{2\times i-1}, I_{2\times i}),\ i = 1, \ldots, P\ . \tag{6.25}$$

Apparently, the long-short strategy selects features with a large mean and a small mean as a pair, keeping their distance $r_{\omega,i}$ large; thus, somehow maximizes the overall distances in (6.22) for phase quantization. Contrarily, long-long strategy selects

features that either both have large means or both have small means, keeping their distance $r_{\omega,i}$ far away from the boundary; thus, maximizes the overall distances in (6.23) for magnitude quantization. The advantage of these two pairing strategies is that the computational complexity is low $O(P)$.

6.2.4 Experiments

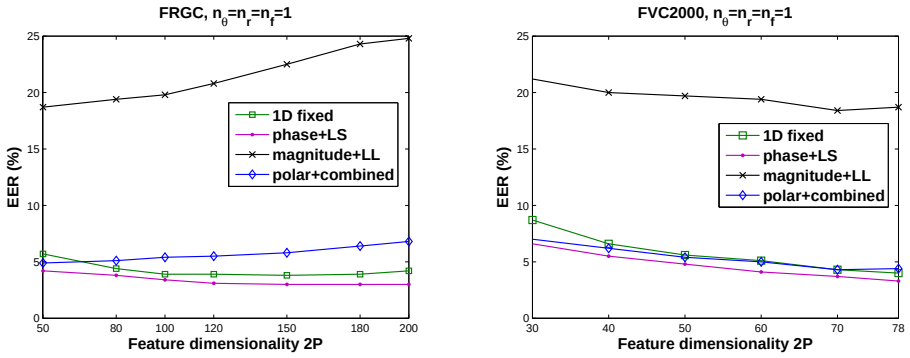
We tested the polar quantization on the FRGC and the FVC2000 database, as described in Section 6.2.3. To first reduce the feature dimensionality, we applied PCA/LDA [42] on a training set, consisting of independent users from the enrollment and verification. The obtained transformation was then applied to both the enrollment and verification samples. In the enrollment, for every target user, assigned with Gray codes, the phase bits (phase+LS) were generated following the LS pairing strategy, while the magnitude bits (magnitude+LL) were generated following LL pairing strategy. By concatenating both the phase and the magnitude bits we obtained the total polar bits (polar+combined). The quantized codes, together with the pairing configuration $\{C^*_{\omega,i}\}$, was stored for each target user. During the verification, features of the query user were quantized according to the $\{C^*_{\omega,i}\}$ of the claimed identity, resulting in a query binary string. Eventually the verification performance was evaluated by a Hamming distance classifier. With, in total, n samples per user ($n = 8$ for FVC2000, and n ranges from 6 to 48 for FRGC), the division of the data is indicated in Table 6.1.

Table 6.1: *Training, enrollment and verification data (number of users×number of samples per user) and the number of partitionings for FRGC and FVC2000.*

	Training	Enrollment	Verification	Partitionings
FRGC	$210 \times n$	$65 \times 2n/3$	$65 \times n/3$	5
FVC2000	$80 \times n$	$30 \times 3n/4$	$30 \times n/4$	20

Since both the phase and the magnitude have fixed equal-probability-intervals, we compared their performances with the one-dimensional fixed quantizer (1D fixed) [22], [23], which has the same property. We first investigated the 1-bit quantization ($n_\theta = n_\mathrm{r} = n_\mathrm{f} = 1$) performances of phase+LS, magnitude+LL and polar+combined. The EER results for the FRGC and the FVC2000 at various feature dimensions are shown in Fig. 6.5. In general, the magnitude bits give poor performances, whereas the phase bits consistently yield good performances and outperform 1D fixed quantization. Furthermore, since the magnitude bits are so poor, combining both phase and magnitude bits, as seen with polar+combined, does not show good performance.

In Fig. 6.6 we further illustrate the GHD and IHD densities of the phase and the magnitude bits, at $P = 60$ for FRGC, as compared to the 1D fixed quantization. We observe that for the three types of bits, the mean of their IHD densities are all around 0.5, demonstrating the equal-probability-intervals. The IHD density of the 1D fixed quantizer is relatively narrow, compared to those of the phase and magnitude.



(a) (b)

Figure 6.5: *The EER performances of phase+LS, magnitude+LL and polar+combined, as compared to the 1D fixed quantization, for (a) FRGC and (b) FVC2000.*

Additionally, the GHD density of the phase bits are located farther away from the IHD. Unfortunately, the GHD densities of the three types of bits are all quite wide.

The discrimination property explained above can be quantified as the inter- and intra-class scatter. Some examples computed on the fractional Hamming distances are given in Table 6.2. Consistent to what we observed in Fig. 6.6, for phase+LS, the S_{inter} is on average two times the 1D fixed. In the mean time, its S_{intra} does not increase much. For this reason, phase+LS yields better performance. On the other hand, magnitude+LL obtains smaller S_{inter} and larger S_{intra}, leading to poor performance. Based on these analysis, we could conclude that even with carefully paired features, the magnitude does not provide discriminative bits for classification. Contrarily, the phase provides reasonably discriminative bits for classification.

In fact, the 1-bit phase quantizer, with boundary at π, quantizes only the feature that has a larger mean in the pairs, leaving the other feature discarded. Thus it also acts as a feature selection procedure. Now we test the phase quantizer with more quantization bits at $n_\theta = 1, 2, 3, 4$, compared to the 1D fixed quantizer at $n_{\text{f}} = 1, 2$. Their EER results for the FRGC and the FVC2000 are shown in Fig. 6.7. Note that when $n_\theta = 2$, in the two-dimensional feature space, the phase quantizer has the same orthogonal boundaries as the 1D fixed quantizer at $n_{\text{f}} = 1$, leading to the same performances. Results show that at a given feature dimensionality, phase bits at $n_\theta = 1$ always give the best performances, while $n_\theta = 2, 3$ also yield reasonably good performances. Unfortunately, when $n_\theta = 4$, the performances turn poor. Generally, compared to the 1D fixed quantizer, the phase quantizer gives better performances at a lower bit length.

To summarize, although the magnitude $r_{\omega,i}$ itself does not provide discriminative bits, it in fact facilitates generating better phase bits. Additionally, the phase quantization has the following properties: (1) The LS pairing strategy is universal and simple, without modeling the specific genuine feature PDF; (2) The phase quantizer boundaries are not necessarily orthogonal, allowing correlations between the two fea-

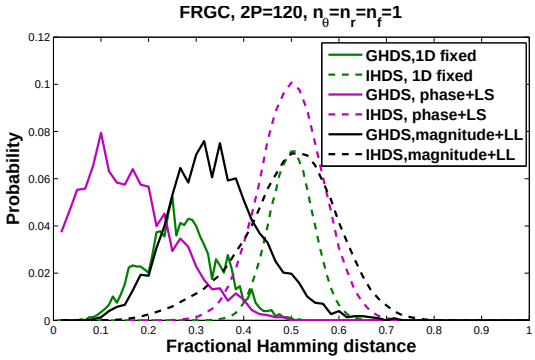


Figure 6.6: *The GHD and the IHD densities of phase+LS, magnitude+LL, and the 1D fixed quantization, at* $2P = 120$ *for FRGC.*

Table 6.2: *The inter-class and intra-class scatter of phase+LS, magnitude+LL and the 1D fixed, for (a-b) FRGC and (c-d) FVC2000.*

FRGC	2P=50		
	1D fixed	phase+LS	magnitude+LL
S_{inter} $(\times 10^{-2})$	7.2	14.9	4.4
S_{intra} $(\times 10^{-2})$	1.3	2.0	2.7

(a)

FRGC	2P=120		
	1D fixed	phase+LS	magnitude+LL
S_{inter} $(\times 10^{-2})$	5.2	11.5	2.3
S_{intra} $(\times 10^{-2})$	0.8	1.4	2.0

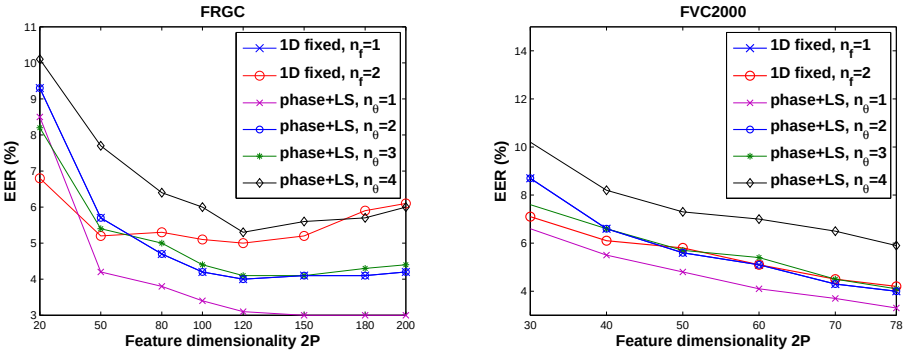
(b)

FVC2000	2P=50		
	1D fixed	phase+LS	magnitude+LL
S_{inter} $(\times 10^{-2})$	7.2	14.6	4.3
S_{intra} $(\times 10^{-2})$	1.2	2.1	2.8

(c)

FVC2000	2P=78		
	1D fixed	phase+LS	magnitude+LL
S_{inter} $(\times 10^{-2})$	6.7	13.5	3.7
S_{intra} $(\times 10^{-2})$	1.0	1.6	2.2

(d)



(a) (b)

Figure 6.7: *The EER performances of phase+LS and 1D fixed quantization at various feature dimensionalities with various quantization bits , for (a) FRGC and (b) FVC2000.*

tures. Furthermore, the phase is uniformly distributed as long as the background PDF is circularly symmetric, which easily fits many feature modalities.

6.2.5 Conclusions

In this paper, we propose extracting binary biometric bits through polar quantization. In addition to the polar quantizer structure, quantizing features in pairs allows feature pair configuration. Therefore, we propose the long-long and the long-short pairing strategies to optimize the discrimination between the genuine Hamming distance and the imposter Hamming distance. Experimental results on the FVC2000 and the FRGC database show that magnitude yields poor classification performances, whereas phase provides reasonably good performances.

6.3 Chapter conclusion

In this chapter, we present a pairwise polar quantization, including the phase and the magnitude. Furthermore, aiming to optimize the discrimination between the IHD and the GHD densities, LL and LS pairing strategies are designed for the magnitude and the phase quantization, respectively. Regarding the research objectives, both quantizers extract multiple *i.i.d.* bits. Compared to the one-dimensional quantizers, two-dimensional quantizers may construct less error-prone quantization intervals. Although both quantizers are theoretically reasonable, the experiments on the real data show that the magnitude quantization combined with LL pairing fails, while the phase quantizer combined with LS pairing gives better recognition performance. With more reliable bits extracted from the user-dependent feature pairs, the length of the random key K can be increased.

7

Two-dimensional Adaptive Phase Quantizer

7.1 Chapter introduction

PURPOSE. In Chapter 6, a user-independent pairwise phase quantizer is proposed. Moreover, a long-short (LS) pairing strategy that selects feature pairs each with a large mean and a small mean showed a promising recognition performance for phase quantization. However, the intervals of this phase quantizer are determined merely by the background probability density function (PDF) of the paired features. Thus, without considering the genuine user PDF, the FAR and FRR performances are not optimal. Therefore, the purpose of this chapter is to adapt the phase quantization intervals to obtain better recognition performances. The new phase quantizer should be capable of extracting multiple statistically independent and identically distributed (*i.i.d.*) bits. Furthermore, after every feature pair is quantized into a prescribed number of bits, the concatenated binary strings should result in good recognition performance, when applied to a Hamming distance classifier (HDC).

CONTENTS. In this chapter, we propose a user-dependent adaptive phase quantizer (APQ) with an improved LS pairing strategy, as illustrated in Fig. 7.1. For every feature pair of an enrolled user, given a uniform background PDF in the phase domain, equal-width quantization intervals also gives equal background probability mass. The APQ then adjusts the phase quantization intervals with an offset, according to the genuine user PDF and the background PDF of the feature pair, so that the theoretical detection rate at Hamming distance zero is maximized. However, in practice, computing the offset based on the two PDFs is difficult. Therefore, later in this chapter, we propose a simplified APQ, in which the offset is computed based on the phase of the feature pair's mean. We show that without loosing much performance, the simplified

APQ can be a good approximation of APQ. The Gray codes, with only one single bit Hamming distance between any two adjacent codewords, are then assigned to the quantization intervals. This reduces the number of erroneous bits due to the intra-class variation. Since the quantization intervals have equal background probability, the bits assigned to each feature pair are *i.i.d.*. Moreover, if the biometric features are statistically independent, the composed feature pairs are independent as well, the bits in the entire binary string are then *i.i.d.*.

To compose feature pairs for APQ, we apply a heuristic long-short (LS) pairing strategy, which composes feature pairs each with a large with a large mean and a small mean, based on the Mahalanobis distance. The pairing is applied to every enrolled user, which makes it user-dependent. The LS pairing proposed in this chapter is in fact similar to the LS pairing in Chapter 6, where the Euclidian distances are used.

Theoretically a bit allocation principle can be applied after LS pairing and APQ. However, since the LS pairing strategy already optimizes the overal binary string performances. There is not much room for improvement to apply the bit allocation. This has been shown in the paper, by comparing LS+APQ with one-dimensional quantizer FQ+DROBA. Therefore, in this chapter, every feature is prescribe to a fixed length of b-bit. Figure 7.2 shows the contribution of this chapter in the context of the thesis.

PUBLICATION(S). The content of Section 7.2 has been published in [59].

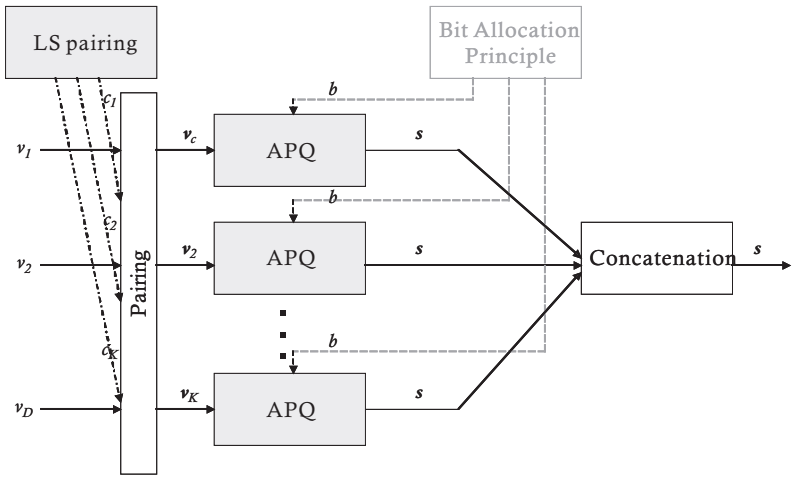


Figure 7.1: *Block diagram of the two-dimensional APQ with LS pairing strategy. The $v_i, i = 1 \dots D$ denote D independent biometric features, which are composed into K feature pairs. The $c_i, i = 1 \dots K$ indicates the configuration for the i^{th} feature pair. The quantized bits $s_i, i = 1 \dots K$ from all K feature pairs are then concatenated into the binary string s.*

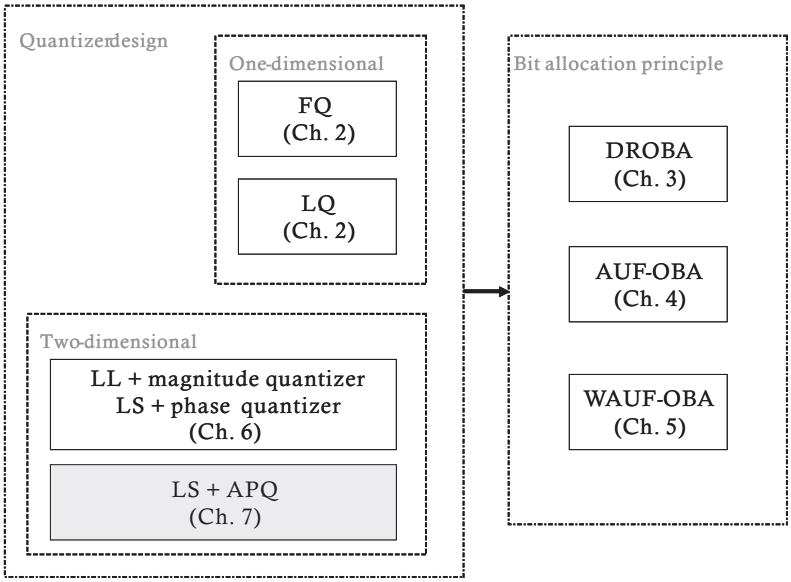


Figure 7.2: *Block diagram of the main contributions, highlighted in chapter* 7.

7.2 Binary biometric representation through pairwise adaptive phase quantization

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 is the straightforward way to extract binary representations from arbitrary real-valued biometric modalities. In this paper, we propose a pairwise adaptive phase quantization (APQ) method, together with a long-short (LS) pairing strategy, which aims to maximize the overall detection rate. Experimental results on the FVC2000 fingerprint and the FRGC face database show reasonably good verification performances.

7.2.1 Introduction

Extracting binary biometric strings is a fundamental step in template compression and protection [4]. It is well-known that biometric information is unique, yet inevitably noisy, leading to intra-class variations. Therefore, the binary strings are desired to be not only discriminative, but also have low intra-class variations. Such requirements translate to both low false acceptance rate (FAR) and low false rejection rate (FRR). Additionally, from the template protection perspective, we know that general biometric information is always public, thus any person has some knowledge of

the distribution of biometric features. Furthermore, the biometric bits in the binary string should be independent and identically distributed (*i.i.d.*), in order to maximize the attacker's efforts in guessing the target template.

Several biometric template protection concepts have been published. Cancelable biometrics [19], [20] distort the image of a face or a fingerprint by using a one-way geometric distortion function. The fuzzy vault method [32], [33] is a cryptographic construction allowing to store a secret in a vault that can be locked using an possibly unordered set of features, e.g. fingerprint minutiae. A third group of techniques, containing fuzzy commitment [21], fuzzy extractor [11], secure sketch [26] and helper data system [24], [22], [23], [25], [15], derive a binary string from a biometric measurement and store an irreversibly hashed version of the string with or without binding a crypto key. In this paper, we adopt the third group of techniques.

The straightforward way to extract binary strings is quantization and coding of the real-valued features. So far, many works [24], [22], [23], [8], [46], [9], [39], [56], [45], [51] have adopted the bit extraction framework shown in Fig. 7.3, involving two tasks: (1) designing a one-dimensional quantizer and (2) determining the number of quantization bits for every feature. The final binary string is then the concatenation of the output bits from all the individual features.

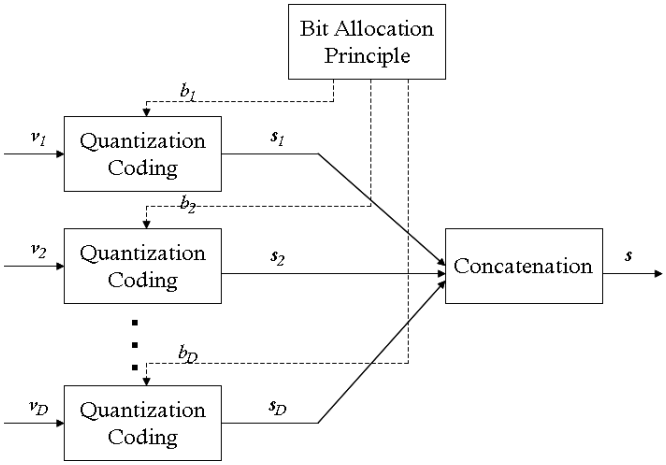


Figure 7.3: *The bit extraction framework based on the one-dimensional quantization and coding, where D denotes the number of features; b_i denotes the number of quantization bits for the i^{th} feature ($i = 1, \ldots, D$), and s_i denotes the output bits. The final binary string is $\boldsymbol{s} = s_1 s_2 \ldots s_D$.*

Designing a one-dimensional quantizer relies on two probability density functions (PDFs): the background PDF and the genuine user PDF, representing the probability density of the entire population and the genuine user, respectively. Based on the two PDFs, quantization intervals are determined to maximize the detection rate, subject to a given FAR, according to the Neyman-Pearson criterion. So far, a number

of one-dimensional quantizers have been proposed [24], [22], [23], [8], [46], [9], [39], as categorized in Table 7.1. Quantizers in [24], [22], [23] are user-independent, constructed merely from the background PDF, whereas quantizers in [8], [46], [9], [39] are user-specific, constructed from both the genuine user PDF and the background PDF. Theoretically, user-specific quantizers provide better verification performances. Particularly, the likelihood-ratio based quantizer [39], among all the quantizers, is optimal in the Neyman-Pearson sense. Quantizers in [24], [8], [46] and [9] have equal-width intervals. Unfortunately, this leads to potential threats: Features obtain higher probabilities in certain quantization intervals than in others, and thus attackers can easily find the genuine interval by continuously guessing the one with the highest probability. To avoid this problem, quantizers in [22], [23] and [39] have equal-probability intervals, ensuring *i.i.d.* bits.

Table 7.1: *The categorized one-dimensional quantizers.*

user-independent	**user-specific**
Linnartz et al. [24]	Vielhauer et al. [8]
Tuyls et al. [22]	Hao and Wah [46]
Kevenaar et al. [23]	Chang et al. [9]
	Chen et al. [39]
equal-width	**equal-probability**
Linnartz et al. [24]	Tuyls et al. [22]
Vielhauer et al. [8]	Kevenaar et al. [23]
Hao and Wah [46]	Chen et al. [39]
Chang et al. [9]	

Apart from the one-dimensional quantizer design, some papers focus on assigning a varying number of quantization bits to each feature. So far, several bit allocation principles have been proposed: Fixed bit allocation (FBA) [22], [23], [39] simply assigns a fixed number of bits to each feature. On the contrary, the detection rate optimized bit allocation (DROBA) [45] and the area under the FRR curve optimized bit allocation (AUF-OBA) [51], assign a variable number of bits to each feature, according to the features' distinctiveness. Generally, AUF-OBA and DROBA outperform FBA.

In this paper, we deal with quantizer design rather than assigning the quantization bits to features. Although one-dimensional quantizers yield reasonably good performances, a problem remains: Independency between all feature dimensions is usually difficult to achieve. Furthermore, one-dimensional quantization leads to inflexible quantization intervals, for instance, the orthogonal boundaries in the two-dimensional feature space, as illustrated in Fig. 7.4a. Contrarily, two-dimensional quantizers, with an extra degree of freedom, bring more flexible quantizer structures. Therefore, a user-independent pairwise polar quantization was proposed in [57]. The polar quantizer is illustrated in Fig. 7.4b, where both the magnitude and the phase intervals are determined merely by the background PDF. In principle, polar quantization is less prone to outliers and less strict on independency of the features, when the

genuine user PDF is located far from the origin. Therefore, in [57], two pairing strategies, the long-long and the long-short pairing, were proposed for the magnitude and the phase, respectively. Both pairing strategies use the Euclidean distances between each feature's mean and the origin. Results showed that the magnitude yields a poor verification performance, whereas the phase yields a good performance. The two-dimensional quantization based bit extraction framework, including an extra feature pairing step, is illustrated in Fig. 7.5.

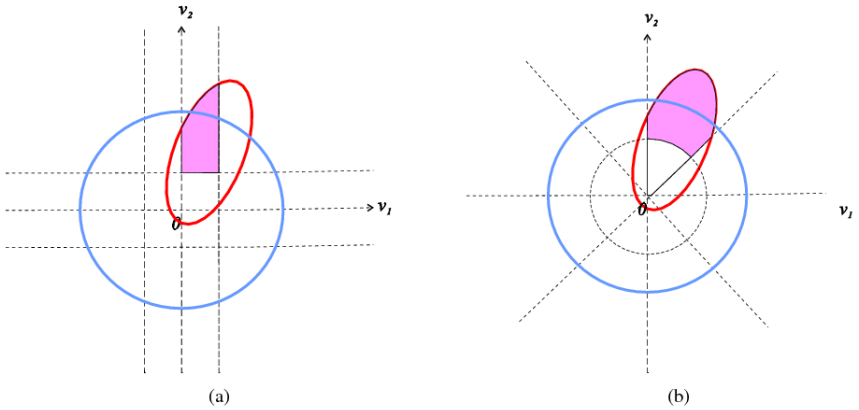


Figure 7.4: *The two-dimensional illustration of (a) the one-dimensional quantizer boundaries (dash line) and (b) the user-independent polar quantization boundaries (dash line). The genuine user PDF is in red and the background PDF is in blue. The detection rate and the FAR are the integral of both PDFs in the pink area.*

Since the phase quantization has shown in [57] to yield a good performance, in this paper, we propose a user-specific adaptive phase quantizer (APQ). Furthermore, we introduce a Mahalanobis distance based long-short (LS) pairing strategy that by good approximation maximizes the theoretical overall detection rate at zero Hamming distance threshold.

In Section 7.2.2 we introduce the adaptive phase quantizer (APQ), with simulations in a particular case with independent Gaussian densities. In Section 7.2.3 the long-short (LS) pairing strategy is introduced to compose pairwise features. In Section 7.2.4, we give some experimental results on the FVC2000 fingerprint database and the FRGC face database. In Section 7.2.5 the results are discussed and conclusions are drawn in Section 7.2.6.

7.2.2 Adaptive Phase Quantizer (APQ)

In this section, we first introduce the APQ. Afterwards, we discuss its performance in a particular case where the feature pairs have independent Gaussian densities.

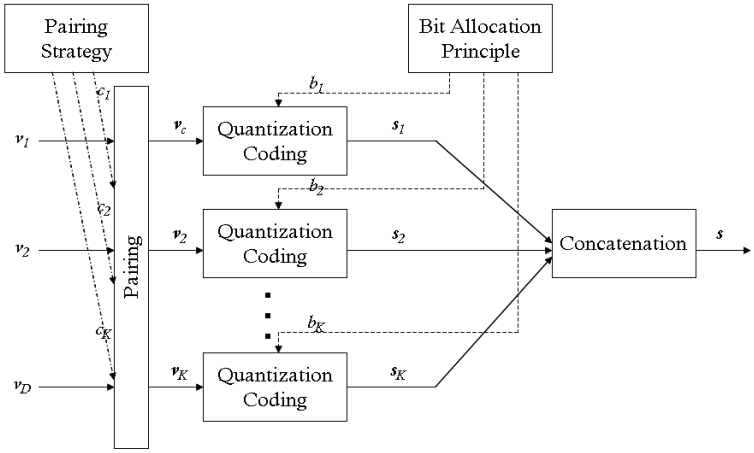


Figure 7.5: *The bits extraction framework based on two-dimensional quantization and coding, where D denotes the number of features; K denotes the number of feature pairs; c_k denotes the feature index for the k^{th} feature pair ($k = 1, \ldots, K$) and s_i denotes the corresponding quantized bits. The final output binary string is $S = s_1 s_2 \ldots s_K$.*

7.3.2.1 Adaptive Phase Quantizer (APQ)

The adaptive phase quantization can be applied to a two-dimensional feature vector if its background PDF is circularly symmetric about the origin. Let $\mathbf{v} = \{v_1, v_2\}$ denote a two-dimensional feature vector. The phase $\theta = \text{angle}(v_1, v_2)$, ranging from $[0, 2\pi)$, is defined as its counterclockwise angle from the v_1-axis. For a genuine user ω, a b-bit APQ is then constructed as:

$$\xi = \frac{2\pi}{2^b} , \tag{7.1}$$

$$Q_{\omega,j} = \left(\varphi^*_\omega + (j-1)\xi \bmod 2\pi,\ \varphi^*_\omega + j\xi \bmod 2\pi \right], \quad j = 1, \ldots, 2^b , \tag{7.2}$$

where $Q_{\omega,j}$ represents the j^{th} quantization interval, determined by the quantization step ξ and an offset angle φ^*_ω. Every quantization interval is uniquely encoded using b bits. Let $\boldsymbol{\mu}_\omega$ be the mean of the genuine feature vector $\mathbf{v}$, then among the intervals, the genuine interval $Q_{\omega,\text{genuine}}$, which is assigned for the genuine user ω, is referred to as:

$$Q_{\omega,j} = Q_{\omega,\text{genuine}} \Longleftrightarrow \boldsymbol{\mu}_\omega \in Q_{\omega,j} , \tag{7.3}$$

that is, $Q_{\omega,\text{genuine}}$ is the interval where the mean $\boldsymbol{\mu}_\omega$ is located. In Fig. 7.6 we give an illustration of a b-bit APQ.

The adaptive offset φ^*_ω in (7.2) is determined by the background PDF $p_{\bar{\omega}}(\mathbf{v})$ as well as the genuine user PDF $p_\omega(\mathbf{v})$: Given both PDFs and an arbitrary offset φ, the

Figure 7.6: *An illustration of a b-bit APQ in the phase domain, where $Q_{\omega,j}$, $j = 1, \ldots, 2^b$ denotes the j^{th} quantization interval with width ξ, and offset angle φ_ω^*. The first interval $Q_{\omega,1}$ is wrapped.*

theoretical detection rate δ and the FAR α at zero Hamming distance threshold are:

$$\delta_\omega(Q_{\omega,\text{genuine}}) = \int_{Q_{\omega,\text{genuine}}(b,\varphi)} p_\omega(\mathbf{v})d\mathbf{v} \ , \tag{7.4}$$

$$\alpha_\omega(Q_{\omega,\text{genuine}}) = \int_{Q_{\omega,\text{genuine}}(b,\varphi)} p_{\bar{\omega}}(\mathbf{v})d\mathbf{v} \ . \tag{7.5}$$

Given that the background PDF is circularly symmetric, (7.5) is independent of φ. Thus, (7.5) becomes:

$$\alpha_\omega = 2^{-b} \ . \tag{7.6}$$

Therefore, the optimal φ_ω^* is determined by maximizing the detection rate in (7.4):

$$\varphi_\omega^* = \arg\max_{\varphi} \delta_\omega \ . \tag{7.7}$$

After the φ_ω^* is determined, the quantization intervals are constructed from (7.2). Additionally, the detection rate of the APQ is

$$\delta_\omega(Q_{\omega,\text{genuine}}) = \int_{Q_{\omega,\text{genuine}}(b,\varphi_\omega^*)} p_\omega(\mathbf{v})d\mathbf{v} \ . \tag{7.8}$$

Essentially, APQ has both equal-width and equal-probability intervals, with rotation offset φ_ω^* that maximizes the detection rate.

7.3.2.2 Simulations on Independent Gaussian Densities

We investigate the APQ performances on synthetic data, in a particular case where the feature pairs have independent Gaussian densities. That is, the background PDF of both features are normalized as zero-mean and unit-variance, i.e. $p_{\bar{\omega},1} = p_{\bar{\omega},2} = N(v, 0, 1)$. Similarly, the genuine user PDFs are $p_{\omega,1}(v) = N(v, \mu_{\omega,1}, \sigma_{\omega,1})$ and $p_{\omega,2}(v) = N(v, \mu_{\omega,2}, \sigma_{\omega,2})$. Since the two features are independent, the two-dimensional joint background PDF $p_{\bar{\omega}}(\mathbf{v})$ and the joint genuine user PDF $p_\omega(\mathbf{v})$ are:

$$p_{\bar{\omega}}(\mathbf{v}) = p_{\bar{\omega},1} \cdot p_{\bar{\omega},2} \ , \tag{7.9}$$

$$p_\omega(\mathbf{v}) = p_{\omega,1} \cdot p_{\omega,2} \ . \tag{7.10}$$

According to (7.6), the FAR for a b-bit APQ is fixed to 2^{-b}. Therefore, we only have to investigate the detection rate in (7.8) regarding the genuine user PDF p_ω,

defined by the μ and σ values. In Fig. 7.7, we show the detection rate δ_ω of the b-bit APQ ($b = 1, 2, 3, 4$), when $p_\omega(\mathbf{v})$ is modeled as $\sigma_{\omega,1} = \sigma_{\omega,2} = 0.2$; $\sigma_{\omega,1} = \sigma_{\omega,2} = 0.8$; $\sigma_{\omega,1} = 0.8$, $\sigma_{\omega,2} = 0.2$, at various $\{\mu_{\omega,1}, \mu_{\omega,2}\}$ locations for optimal φ^*_ω. The white pixels represent high values of the detection rate whilst the black pixels represent low values. The δ_ω appears to depend more on how far the features are from the origin than on the direction of the features. This is due to the rotation adaptive property. In general, the δ_ω is higher when the genuine user PDF has smaller σ_ω and larger μ_ω for both features. Either decreasing the μ_ω or increasing the σ_ω deteriorates the performance.

To generalize such property, we define a Mahalanobis distance $d_{\omega,i}$ for feature i as:

$$d_{\omega,i} = \mathrm{abs}(\mu_{\omega,i}/\sigma_{\omega,i}) \ . \tag{7.11}$$

Given the Mahalanobis distances $d_{\omega,1}$, $d_{\omega,2}$ of two features, we define $\bar{d}_\omega$ for this feature pair as:

$$\bar{d}_\omega = \sqrt{d^2_{\omega,1} + d^2_{\omega,2}} \ . \tag{7.12}$$

In Fig. 7.8 we give some simulation results for the relation between $\bar{d}_\omega$ and δ_ω. The parameters μ and σ for the genuine user PDF p_ω are modeled as four σ combinations at various μ locations. For every μ-σ setting, we plot its $\bar{d}_\omega$ and δ_ω. We observe that the detection rate δ_ω tends to increase when the feature pair Mahalanobis distance $\bar{d}_\omega$ increases, although not always monotonically.

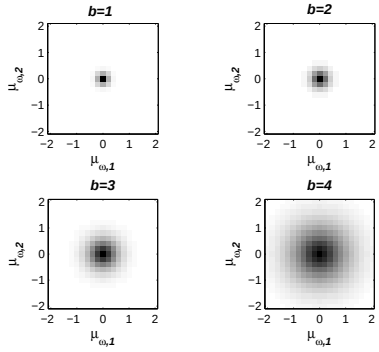
We further compare the detection rate of APQ to that of the one-dimensional fixed quantizer (FQ) [39]. In order to compare with the 2-bit APQ at the same FAR, we choose a 1-bit FQ ($b = 1$) for every feature dimension. In Fig. 7.9 we show the ratio of their detection rates ($\delta_{\mathrm{APQ}}/\delta_{\mathrm{FQ}}$) at various μ-σ values. The white pixels represent high values whilst the black pixels represent low values. It is observed that APQ consistently outperforms FQ, especially when the mean of the genuine user PDF is located far away from the origin and close to the FQ boundary, namely the v_1-axis and v_2-axis. In fact, the two 1-bit FQ works as a special case of the 2-bit APQ, with $\varphi^*_\omega = 0$.

7.2.3 Biometric Binary String Extraction

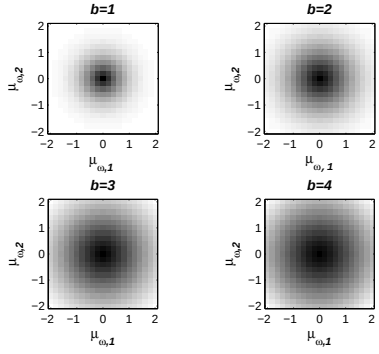
The APQ can be directly applied to two-dimensional features, such as Iris [53], while for arbitrary features, we have the freedom to pair the features. In this section, we first formulate the pairing problem, which in practice is difficult to solve. Therefore, we simplify this problem and then propose a long-short pairing strategy (LS) with low computational complexity.

7.3.3.1 Problem Formulation

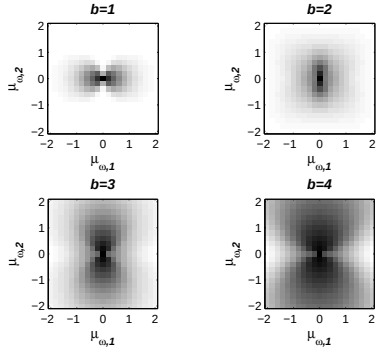
The aim for extracting biometric binary string is: for a genuine user ω who has D features, we need to determine a strategy to pair these D features into $D/2$ pairs, in such way that the entire L-bit binary string ($L = b \times D/2$) obtains optimal classification



(a)

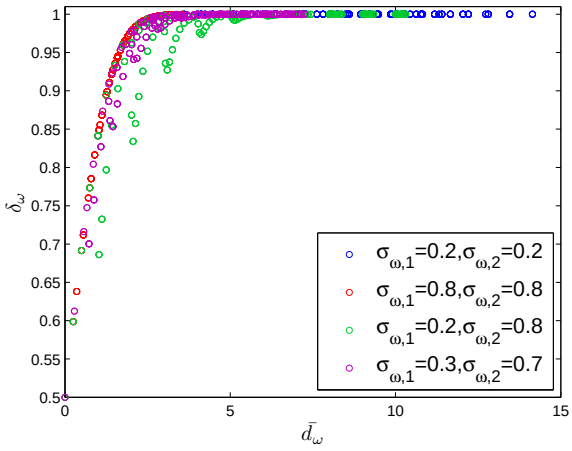


(b)

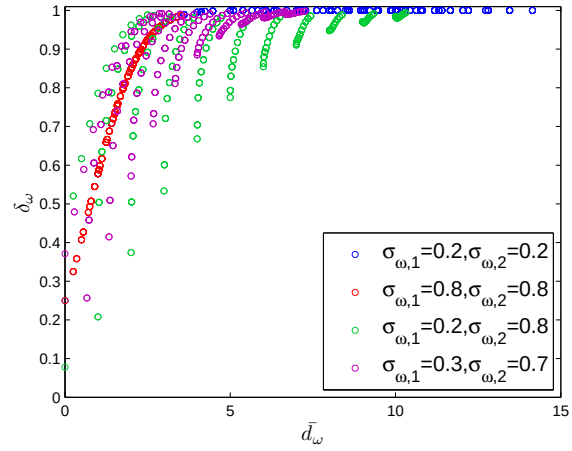


(c)

Figure 7.7: *The detection rate of the b-bit APQ ($b = 1, 2, 3, 4$), when $p_\omega(\mathbf{v})$ is modeled as (a) $\sigma_{\omega,1} = \sigma_{\omega,2} = 0.2$; (b) $\sigma_{\omega,1} = \sigma_{\omega,2} = 0.8$; (c) $\sigma_{\omega,1} = 0.8$, $\sigma_{\omega,2} = 0.2$, at various $\{\mu_{\omega,1}, \mu_{\omega,2}\}$ locations: $\mu_{\omega,1}, \mu_{\omega,2} \in [-2\ 2]$. The detection rate ranges from 0 (black) to 1 (white).*

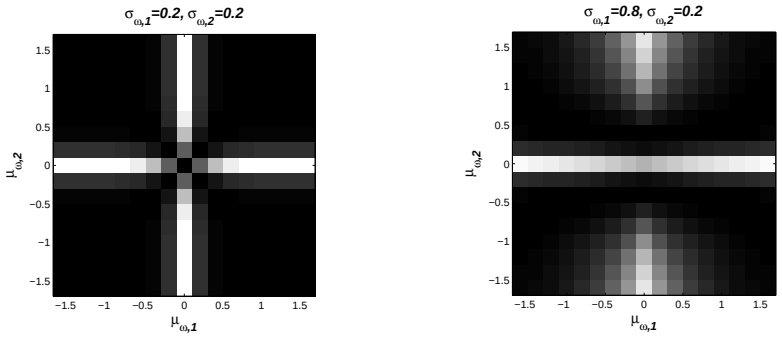


(a)



(b)

Figure 7.8: *The relations between $\bar{d}_\omega$ and δ_ω when the genuine user PDF p_ω is modeled as with $\mu_{\omega,1}, \mu_{\omega,2} \in [-2\ 2]$ and four $\sigma_{\omega,1}$, $\sigma_{\omega,2}$ settings. The result is shown as (a) 1-bit APQ; (b) 2-bit APQ.*



(a) (b)

Figure 7.9: *The detection rate ratio $\delta_{\mathrm{APQ}}/\delta_{\mathrm{FQ}}$ of the 2-bit APQ to the 1-bit FQ ($b = 1$), when $p_\omega(\mathbf{v})$ is modeled as (a) $\sigma_{\omega,1} = \sigma_{\omega,2} = 0.2$; (b) $\sigma_{\omega,1} = 0.8$, $\sigma_{\omega,2} = 0.2$, with various $\mu_{\omega,1}$, $\mu_{\omega,2}$ locations: $\mu_{\omega,1}, \mu_{\omega,2} \in [-1.6\ 1.6]$. The detection rate ratio ranges from 1 (black) to 2 (white).*

performance, when every feature pair is quantized by a b-bit APQ. Assuming that the $D/2$ feature pairs are statistically independent, we know from [45] that when applying a Hamming distance classifier, zero Hamming distance threshold gives a lower bound for both the detection rate and the FAR. Therefore, we decide to optimize this lower bound classification performance.

Let $c_{\omega,k}$, $(k = 1, \ldots, D/2)$ be the k^{th} pair of feature indices, and $\{c_{\omega,k}\}$ a valid pairing configuration containing $D/2$ feature index pairs such that every feature index only appears once. For instance, $c_{\omega,k} = (1, 1)$ is not valid because it contains the same feature and therefore cannot be included in $\{c_{\omega,k}\}$. Also, $\{c_{\omega,k}\} = \{(1, 2), (1, 3)\}$ is not a valid pairing configuration because the index value '1' appears twice. The overall FAR (α_ω) and the overall detection rate (δ_ω), at zero Hamming distance threshold are:

$$\alpha_\omega\big(\{c_{\omega,k}\}\big) = \prod_{k=1}^{D/2} \alpha_{\omega,k}\big(c_{\omega,k}\big), \tag{7.13}$$

$$\delta_\omega\big(\{c_{\omega,k}\}\big) = \prod_{k=1}^{D/2} \delta_{\omega,k}\big(c_{\omega,k}\big), \tag{7.14}$$

where $\alpha_{\omega,k}$ and $\delta_{\omega,k}$ are the FAR and the detection rate for the k^{th} feature pair, computed from (7.6) and (7.8). Furthermore, according to (7.6), α_ω becomes:

$$\alpha_\omega = 2^{-L}\ , \tag{7.15}$$

which is independent of $\{c_{\omega,k}\}$. Therefore, we only need to search for a user-specific pairing configuration $\{c^*_{\omega,k}\}$, that maximizes the overall detection rate in (7.14). Solv-

ing the optimization problem is formulated as:

$$\{c^*_{\omega,k}\} = \arg\max_{\{c_{\omega,k}\}} \prod_{k=1}^{D/2} \delta_\omega\big(c_{\omega,k}\big) \ . \tag{7.16}$$

The detection rate δ_ω given a feature pair $c_{\omega,k}$ is computed from (7.8). Considering that the performance at zero Hamming distance threshold indeed pinpoints the minimum FAR and detection rate value on the receiver operating characteristic curve (ROC), optimizing such point in (7.16) essentially provides a maximum lower bound for the ROC curve.

7.3.3.2 Long-short Pairing

There are two problems in solving (7.16): First, it is often not possible to compute $\delta_{c_{\omega,k}}$ in (7.8), due to the difficulties in estimating the genuine user PDF p_ω. Additionally, even if the $\delta_{c_{\omega,k}}$ can be accurately estimated, a brute-force search would involve $2^{-D/2}\frac{D!}{(D/2)!}$ evaluations of the overall detection rate, which renders a brute-force search unfeasible for realistic values of D. Therefore, we propose to simplify the problem definition in (7.16) as well as the optimization searching approach.

Simplified problem definition: In Section 7.2.2 we observed a useful relation between $\bar{d}$ and δ for the APQ: A feature pair with a higher $\bar{d}$ would approximately also obtain a higher detection rate δ_ω for APQ. Therefore, we simplify (7.16) into:

$$\{c^*_{\omega,k}\} = \arg\max_{\{c_{\omega,k}\}} \prod_{k=1}^{D/2} \bar{d}_\omega(c_{\omega,k}) \ , \tag{7.17}$$

with $\bar{d}_\omega(c_{\omega,k})$ defined in (7.12). Furthermore, instead of brute force searching, we propose a simplified optimization searching approach: the long-short (LS) pairing strategy.

Long-short (LS) pairing: For the genuine user ω, sort the set $\{d_{\omega,i} = \mathrm{abs}(\mu_{\omega,i}/\sigma_{\omega,i}) : i = 1,\ldots,D\}$ from largest to smallest into a sequence of ordered feature indices $\{I_{\omega,1}, I_{\omega,2}, \ldots, I_{\omega,D}\}$. The index for the k^{th} feature pair is then:

$$c_{\omega,k} = (I_{\omega,k}, I_{\omega,D+1-k}), k = 1,\ldots,D/2 \ . \tag{7.18}$$

The computational complexity of the LS pairing is only $O(D)$. Additionally, it is applicable to arbitrary feature types and independent of the number of quantization bits b. Note that this LS pairing is similar to the pairing strategy proposed in [57], where Euclidean distances are used. In fact, there are other alternative pairing strategies, for instance greedy or long-long pairing [57]. However, in terms of the entire binary string performance, these methods are not as good as the approach presented in this paper, especially when D is large. Therefore, in this paper, we choose the long-short pairing strategy, providing a compromise between the classification performance and computational complexity.

7.2.4 Experiments

In this section we test the pairwise phase quantization (LS+APQ) on real data. First we present a simplified APQ, which is employed in all the experiments. Afterwards, we verify the relation between $\bar{d}$ and δ for real data. We also show some examples of LS pairing results. Then we investigate the verification performances while varying the input feature dimensions (D) and the number of quantization bits per feature pair (b). The results are further compared to the one-dimensional fixed quantization (1D FQ) [39] as well as the the FQ in combined with the DROBA bit allocation principle (FQ+DROBA).

7.3.4.1 Experimental Setup

We tested the pairwise phase quantization on two real data sets: the FVC2000(DB2) fingerprint database [35] and the FRGC(version 1) face database [37].

- **FVC2000**: The FVC2000(DB2) fingerprint data set contains 8 images of 110 users. The features were extracted in a fingerprint recognition system that was used in [22]. As illustrated in Fig. 7.10, the raw features contain two types of information: the squared directional field in both x and y directions, and the Gabor response in 4 orientations (0, $\pi/4$, $\pi/2$, $3\pi/4$). Determined by a regular grid of 16 by 16 points with spacing of 8 pixels, measurements are taken at 256 positions, leading to a total of 1536 elements.

- **FRGC**: The FRGC(version 1) face data set contains 275 users with a different number of images per user, taken under both controlled and uncontrolled conditions. The number of samples s per user ranges from 4 to 36. The image size was 128×128. From that a region of interest (ROI) with 8762 pixels was taken as illustrated in Fig. 7.11.

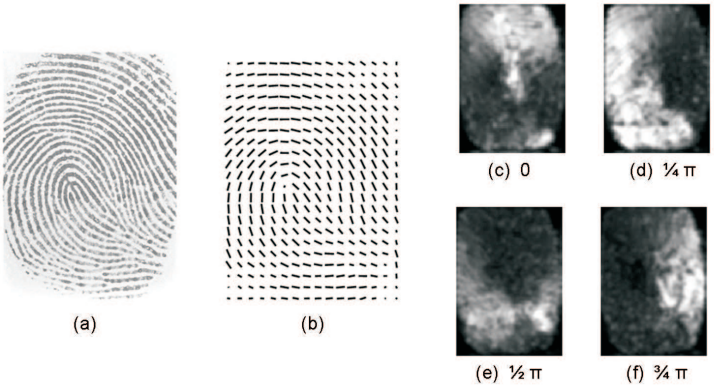


Figure 7.10: *(a) Fingerprint image, (b) directional field, (c)-(f) the absolute values of Gabor responses for different orientations θ.*

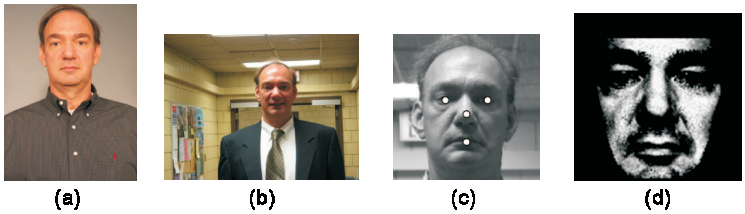


Figure 7.11: *(a) Controlled image, (b) uncontrolled image, (c) landmarks and (d) the region of interest (ROI).*

A limitation of biometric compression or protection is that it is not possible to conduct the user-specific image alignment, because the image or other alignment information can not be stored. Therefore, in this paper, we applied basic absolute alignment methods: The fingerprint images are aligned according to a standard core point position; The face images are aligned according to a set of four standard landmarks, i.e. eyes, nose and mouth.

We randomly selected different users for training and testing and repeated our experiments with a number of trials. The data division is described in Table 7.2, where s is the number of samples per user that varies in the experiments.

Table 7.2: *Data division: number of users×number of samples per user(s), and the number of trials for FVC2000 and FRGC. The s is a parameter that varies in the experiments.*

	Training	Enrollment	Verification	Trials
FVC2000	80×8	30×6	30×2	20
FRGC	$210 \times s$	$65 \times 2s/3$	$65 \times s/3$	5

Our experiments involved three steps: training, enrollment and verification. (1) In the training step, we first applied a combined PCA/LDA method [42] on a training set. The obtained transformation was then applied to both the enrollment and verification sets. We assume that the measurements have a Gaussian density, thus after the PCA transformation, the extracted features are assumed to be statistically independent. The goal of applying PCA/LDA in the training step is to extract independent features so that by pairing them we could subsequently obtain independent feature pairs, which meet our problem requirements. Note that for FVC2000, since we have only 80 users in the training set, applying LDA results in very limited number of features (e.g. $D \leq 79$). Therefore, we relax the independency requirement for the genuine user by applying only the PCA transformation. (2) In the enrollment step, for every genuine user ω, the LS pairing was first applied, resulting in the user-specific pairing configuration $\{c^*_{\omega,k}\}$. The pairwise features were further quantized through a b-bit APQ with the adaptive angle $\{\varphi^*_{\omega,k}\}$, and assigned with a Gray code [44]. The concatenation of the codes from $D/2$ feature pairs formed the L-bit target binary string S_ω. Both S_ω and the quantization information $(\{c^*_{\omega,k}\}, \{\varphi^*_{\omega,k}\})$ were stored for each genuine user. (3) In the verification step, the features of the query user were

quantized and coded according to the quantization information $(\{c^*_{\omega,k}\}, \{\varphi^*_{\omega,k}\})$ of the claimed identity, leading to a query binary string S'. Finally, the decision was made by comparing the Hamming distance between the query and the target string.

7.3.4.2 Simplified APQ

In practice, computing the optimal offset angle φ^*_ω for APQ in (7.7) is difficult, because it is hard to find a closed-form solution φ^*_ω. Besides, it is often impossible to accurately estimate the underlying genuine user PDF p_ω, due to the limited number of available samples per user. Therefore, instead of φ^*_ω, we propose an approximate solution φ'_ω. For genuine user ω, let the mean of the two-dimensional feature vector be $\{\mu_{\omega,1}, \mu_{\omega,2}\}$, and its phase be $\bar{\theta}_\omega = \text{angle}(\mu_{\omega,1}, \mu_{\omega,2})$, the approximate offset angle φ'_ω is then computed as:

$$\varphi'_\omega = \bar{\theta}_\omega - \frac{\xi}{2}\ , \tag{7.19}$$

where $\xi = 2\pi/2^b$. We give an illustration of computing φ'_ω in Fig. 7.12. The approximate solution φ'_ω in fact maximizes the product of two Euclidean distances, namely, the distance of the mean vector $\{\mu_{\omega,1}, \mu_{\omega,2}\}$ to both the lower and the higher genuine interval boundaries.

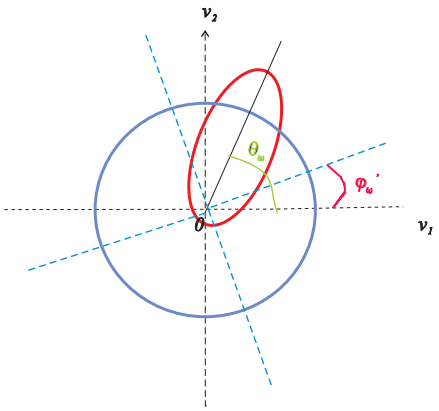


Figure 7.12: *An example of a 2-bit simplified APQ, with the background PDF (blue) and the genuine user PDF (red). The dashed lines are the quantization boundaries.*

Note that when the two features have independent Gaussian density with equal standard deviation, $\varphi^*_\omega = \varphi'_\omega$. Thus, in that case, the simplified APQ equals the original APQ. In Fig. 7.13, we illustrates an example of the detection rate ratio between the simplified and the original APQ, where both features are modeled as Gaussian with different standard deviations, e.g. $\sigma_{\omega,1} = 0.2$, $\sigma_{\omega,2} = 0.8$. The white pixels represent high values whilst the black pixels represent low values. Results show that the simplified APQ is only slightly worse than the original APQ when the mean of the two-dimensional feature $\{\mu_{\omega,1}, \mu_{\omega,2}\}$ is close to the origin. However, if we apply

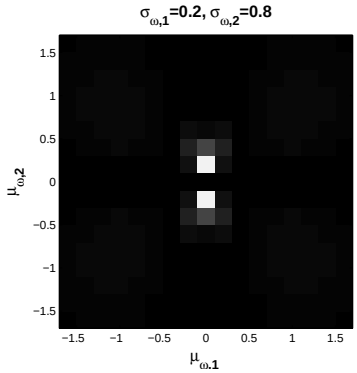


Figure 7.13: *The detection rate ratio bwteeen the original* 2*-bit APQ and the simplified APQ, when* $p_\omega(\mathbf{v})$ *is modeled as* $\sigma_{\omega,1} = 0.2$*,* $\sigma_{\omega,2} = 0.8$*, with various* $\mu_{\omega,1}$*,* $\mu_{\omega,2}$ *locations:* $\mu_{\omega,1}, \mu_{\omega,2} \in [-1.6\ 1.6]$*. The detection rate ratio scale is* $[1\ 2.2]$*.*

APQ after the LS pairing, we would expect that the overall selected pairwise features are located farther away from the origin. In such cases, the simplified APQ works almost the same as the original APQ. In Fig. 7.14 we illustrate the differences of the rotation angle between the original APQ and the simplified APQ, computed from (7.7) and (7.19) respectively. These differences are computed from 50 feature pairs for both FVC2000 and FRGC. The results show that there is no much differences between the rotation angle. Additionally, the simplified APQ is much simpler, avoiding the problem of estimating the underlying genuine user PDF p_ω. For these reasons, we employed this simplified APQ in all the following experiments (Section 7.2.4 to 7.2.4).

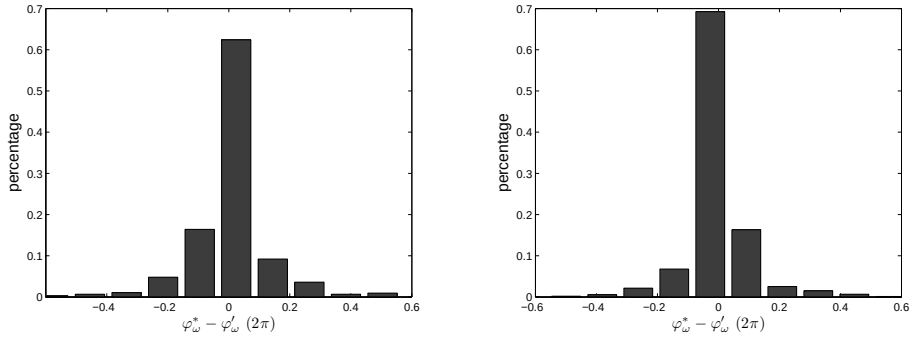


Figure 7.14: *The differences of the rotation angle between the original APQ and the simplified APQ* ($\varphi^*_\omega - \varphi'_\omega$)*, computed from* 50 *feature pairs, for (a) FVC2000 and (b) FRGC.*

7.3.4.3 APQ $\bar{d}$-δ Property

In this section we test the relation between the APQ detection rate δ_ω and the pairwise feature's distance $\bar{d}_\omega$ on both data sets. The goal is to see whether the real data exhibit the same $\bar{d}_\omega - \delta_\omega$ property as we found with synthetic data in Section 7.2.2: The feature pairs with higher $\bar{d}_\omega$ obtains higher detection rate δ_ω.

During the enrollment, for every genuine user, we conducted a random pairing. For every feature pair, we computed their $\bar{d}_\omega$ value according to (7.12). Afterwards, we applied the b-bit APQ quantizer to every feature pair. In the verification, for every feature pair, we computed the Hamming distance between the b-bits from the genuine user and the b-bits from the imposters. That is, we count as a detection if the b-bit genuine query string obtains zero Hamming distance as compared to the target string. Similarly, we count as a false acceptance if the b-bit imposter query string obtains zero Hamming distance as compared to the target string. We then repeated this process over all feature pairs as well as all genuine users, in order to ensure that the results we obtain are neither user or feature biased. Finally, in Fig. 7.15, we plot the relations between the $\bar{d}_\omega$ and the δ_ω. The points we plot are averaged according to the bins of $\bar{d}_\omega$, when $b = 2$. Results show that for the real data, the larger $\bar{d}_\omega$ is, consistently the higher detection rate we obtain. Additionally, the FAR performance is indeed independent of pairing, and equals the theoretical value 2^{-b}.

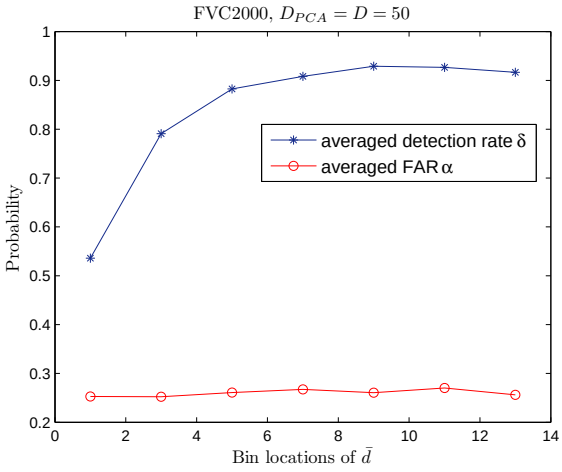
7.3.4.4 LS Pairing Performance

In this section we test the LS pairing performances. We give an example of FVC2000 at $D = 50$. Figure 7.16(a) shows the histogram of d for all single features over all the genuine users. Around 70% of them are close to zero, suggesting low quality features. After LS pairing, the histogram of the pairwise $\bar{d}$ values are shown in Fig. 7.16(b), as compared with the random pairing. In Fig. 7.16(c), we illustrate the 25 pairwise features in terms of independent Gaussian densities, for one specific genuine user. Fig. 7.16(b) and 7.16(c) shows that after LS pairing, a large proportion of feature pairs have relatively moderate 'size' densities and moderate $\bar{d}$ values. Thus it avoids small $\bar{d}$ values and effectively maximizes (7.17).

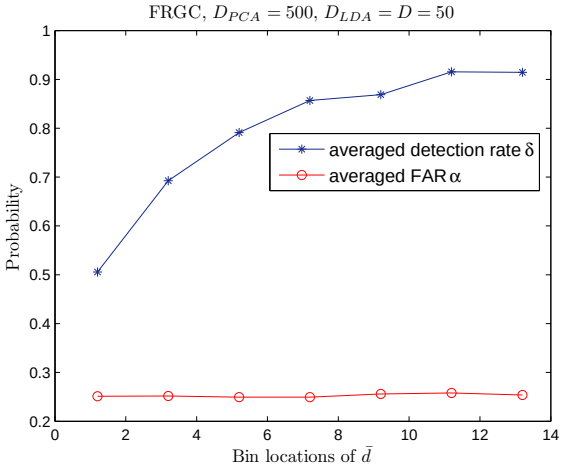
7.3.4.5 Verification Performance

We test the performances of LS+APQ at various numbers of input features D, as well as various numbers of quantization bits $b \in \{1, \ldots, 6\}$. The performances are further compared with the one-dimensional fixed quantization (1D FQ) [39]. The EER results for FVC2000 and FRGC are shown in Table 7.3 and Fig. 7.17.

Both data sets show that by increasing the number of features D at a fixed b-bit quantization per feature pair, the performances of LS+APQ improves and becomes stable. Additionally, given D features, the overall performances of LS+APQ are relatively good only when $b \leq 3$. However, when $b \geq 4$, the performances become poor. For FVC2000, an average of 1-bit per feature pair gives the lowest EER, while for FRGC, the lowest EER allows 2-bit per feature pair. In Fig. 7.18, we give

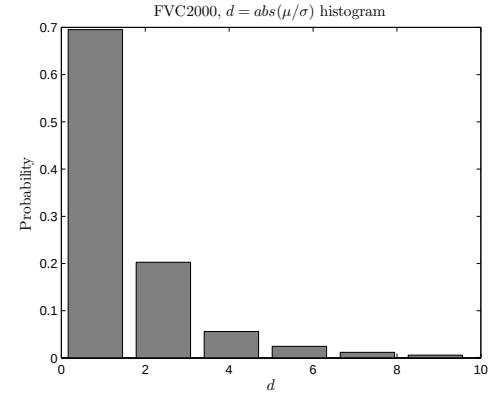


(a)

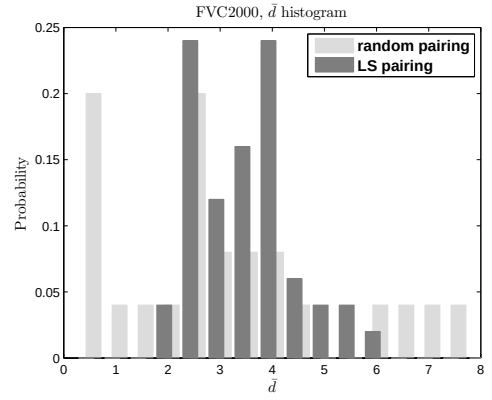


(b)

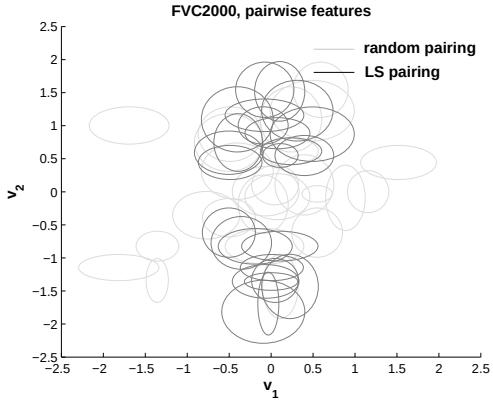
Figure 7.15: *The averaged value of the detection rate and the FAR that correspond to the bins of $\bar{d}$, derived from the random pairing and the 2-bit APQ, for (a) FVC2000 and (b) FRGC.*



(a)



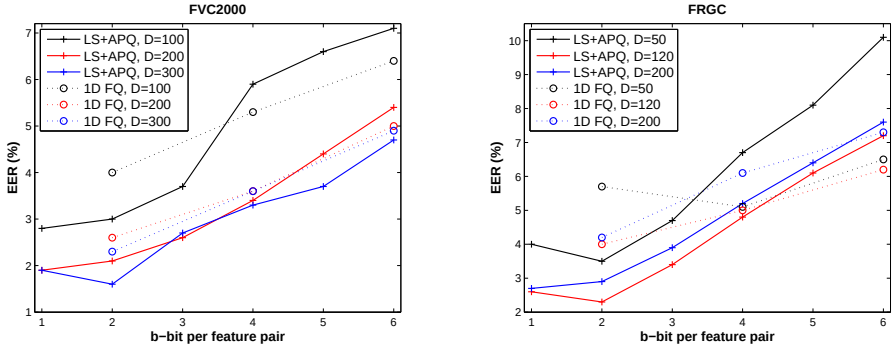
(b)



(c)

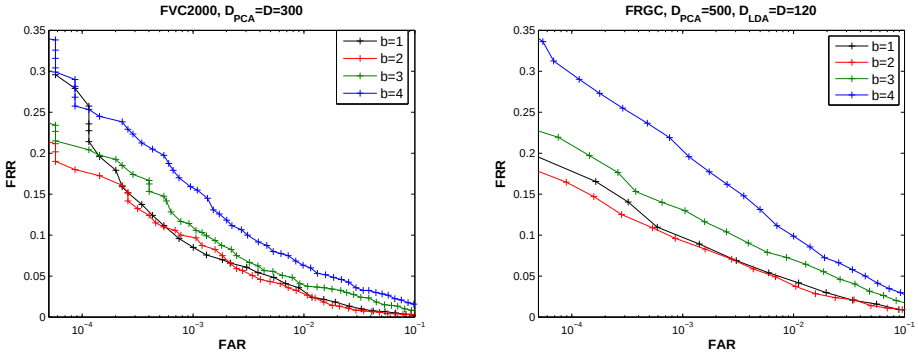
Figure 7.16: *An example of the LS pairing performance on FVC2000, at $D = 50$. (a) the histogram of $d = \text{abs}(\mu/\sigma)$; (b) the histogram of $\bar{d}$ for pairwise features and (c) an illustration of the pairwise features as independent Gaussian density, from both LS and random pairing.*

their FAR/FRR performances at the best D, with b from 1 to 4, and the FAR/FRR performances at the best b are given in Table 7.4.



(a) (b)

Figure 7.17: *The EER performances of b-bit ($b \in [1\ 6]$) LS+APQ at various feature dimensionality D, as compared with the $b/2$-bit 1D FQ (b-bit per feature pair), for (a) FVC2000, and (b) FRGC.*



(a) (b)

Figure 7.18: *An example of the FAR/FRR performances (FAR in logarithm) of LS+APQ, with b from 1 to 4, for (a) FVC2000 and (b) FRGC.*

We further compare the LS+APQ with the 1D FQ. In order to compare at the same string length, we compare the $b/2$-bit 1D FQ with the b-bit LS+APQ. The EER performances in Fig. 7.17 show that in general when $b \leq 3$, LS+APQ outperforms 1D FQ. However, when $b \geq 4$, LS+APQ is no longer competitive to 1D FQ. In Fig. 7.19, we give an example of comparing the FAR/FRR performances of LS+APQ and 1D FQ, on FRGC. Since both APQ and FQ provide equal-probability intervals, they yield almost the same FAR performance. On the other hand, LS+APQ obtains lower FRR as compared with 1D FQ.

Table 7.3: *The EER performances of LS+APQ and 1D FQ, at various feature dimensionality D and various numbers of quantization bits b, for (a) FVC2000 and (b) FRGC.*

FVC2000		$D_{\mathrm{PCA}} = D$, EER=(%)					
		$D = 50$	100	150	200	250	300
LS+APQ	$b = 1$	**4.4**	**2.8**	**2.0**	**1.9**	1.8	1.9
	$b = 2$	4.6	3.0	2.0	2.1	**1.7**	**1.6**
	$b = 3$	6.4	3.7	2.8	2.6	2.5	2.7
	$b = 4$	8.2	5.9	4.6	3.4	3.2	3.3
	$b = 5$	10.0	6.6	5.9	4.4	4.0	3.7
	$b = 6$	11.4	7.1	6.6	5.4	4.7	4.7
1D FQ	$b = 1$	6.7	4.0	2.9	2.6	2.7	2.3
	$b = 2$	7.5	5.3	4.2	3.6	3.6	3.6
	$b = 3$	9.2	6.4	5.5	5.0	5.2	4.9

(a)

FRGC		$D_{\mathrm{PCA}} = 500$, $D_{\mathrm{LDA}} = D$, EER=(%)						
		$D = 50$	80	100	120	150	180	200
LS+APQ	$b = 1$	4.0	3.4	3.0	2.6	2.9	2.7	**2.7**
	$b = 2$	**3.5**	**3.0**	**2.8**	**2.3**	**2.8**	**2.7**	2.9
	$b = 3$	4.7	4.1	3.7	3.4	3.3	3.6	3.9
	$b = 4$	6.7	5.9	5.0	4.8	4.7	5.0	5.2
	$b = 5$	8.1	7.0	6.3	6.1	6.5	6.6	6.4
	$b = 6$	10.1	8.6	7.5	7.2	7.2	7.4	7.6
1D FQ	$b = 1$	5.7	4.7	4.2	4.0	4.1	4.1	4.2
	$b = 2$	5.1	5.4	5.1	5.0	5.2	5.9	6.1
	$b = 3$	6.5	6.5	6.4	6.2	6.5	6.9	7.3

(b)

Table 7.4: *The FAR/FRR performances for FVC2000 and FRGC at the best D-L setting.*

FRR (%)	FAR = 10^{-4}	10^{-3}	10^{-2}
FVC2000, $D = 300$, $L = 300$	17.2	9.6	2.6
FRGC, $D = 120$, $L = 120$	14.7	8.2	3.7

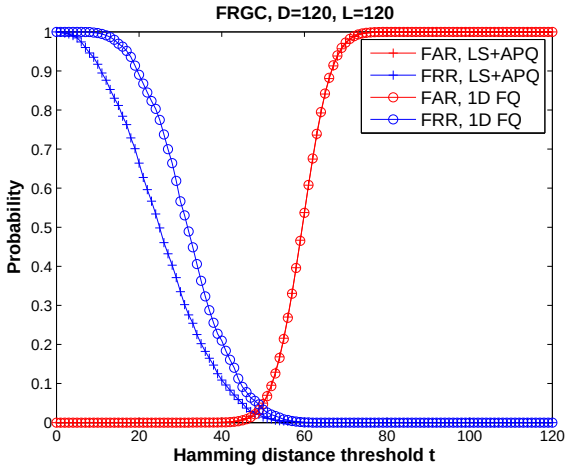


Figure 7.19: *An example of the FAR/FRR performances of LS+APQ and 1D FQ, at* $D = 120$, $L = 120$ *for FRGC.*

In [45], it was shown that FQ in combination with the DROBA adaptive bit allocation principle (FQ+DROBA) provides considerably good performances. Therefore, we compare the LS+APQ with the FQ+DROBA. In order to compare both methods at the same D-L setting, for LS+APQ, we extract only $2K$ features from the D features, thus K pairs from the LS pairing. Afterwards, we apply the 2-bit APQ for every feature pair (see Fig. 7.5). In this case, $K = L/2$. Table 7.5 shows the EER performances of LS+APQ and FQ+DROBA at several different D-L settings. Results show that LS+APQ obtains slightly better performances than FQ+DROBA.

Table 7.5: *The EER performances of LS+APQ and FQ+DROBA, at at several D-L settings, for (a) FVC2000 and (b) FRGC.*

FVC2000	$D = 250$, EER=(%)		
	$L = 50$	$L = 100$	$L = 150$
LS+APQ	2.3	1.7	1.9
FQ+DROBA	2.4	2.1	2.2

(a)

FRGC	$D = 120$, EER=(%)		
	$L = 60$	$L = 90$	$L = 120$
LS+APQ	2.3	2.4	2.3
FQ+DROBA	2.4	2.6	2.8

(b)

7.2.5 Discussion

Essentially, the pairwise phase quantization involves two user-specific adaptation steps: the long-short (LS) pairing, as well as the adaptive phase quantization (APQ). From the pairing's perspective, although we only quantize the phase, the magnitude information (i.e. the feature mean) is not discarded. Instead, it is employed in the LS pairing strategy to facilitate extracting distinctive phase bits. Additionally, although with low computational complexity, the LS pairing strategy is effective for arbitrary feature densities. From the quantizer's perspective, quantizing in phase domain has the advantage that a circularly symmetric two-dimensional feature density results in a simple uniform phase density. Additionally, we apply user-specific phase adaptation. As a result, the extracted phase bits are not only distinctive but also robust to over-fitting. However, the experimental results imply that such advantages only exist when $b \leq 3$. To summarize, as illustrated in Fig. 7.20, the LS pairing is a user-specific resampling procedure that provides simple unform but distinctive phase densities. The APQ further enhances the feature distinctiveness by adjusting the user-specific phase quantization intervals.

7.2.6 Conclusion

Extracting binary biometric strings is a fundamental step in biometric compression and template protection. Unlike many previous work which quantize features individually, in this paper, we propose a pairwise adaptive phase quantization (APQ), together with a long-short (LS) pairing strategy, which aims to maximize the overall detection rate. Experimental results on the FVC2000 and the FRGC database show reasonably good verification performances.

7.3 Chapter conclusion

In this chapter, we present a pairwise adaptive phase quantizer (APQ), together with a long-short (LS) pairing strategy. Regarding the research objectives, APQ is capable of extracting multiple *i.i.d.* bits. With adjusted quantization intervals, APQ extracts more reliable bits than the user-independent phase quantizer as described in Chapter 6. As a result, APQ and LS pairing gives better FAR and FRR performances. With more reliable bits extracted from the user-dependent feature pairs, the length of the random key K can be increased.

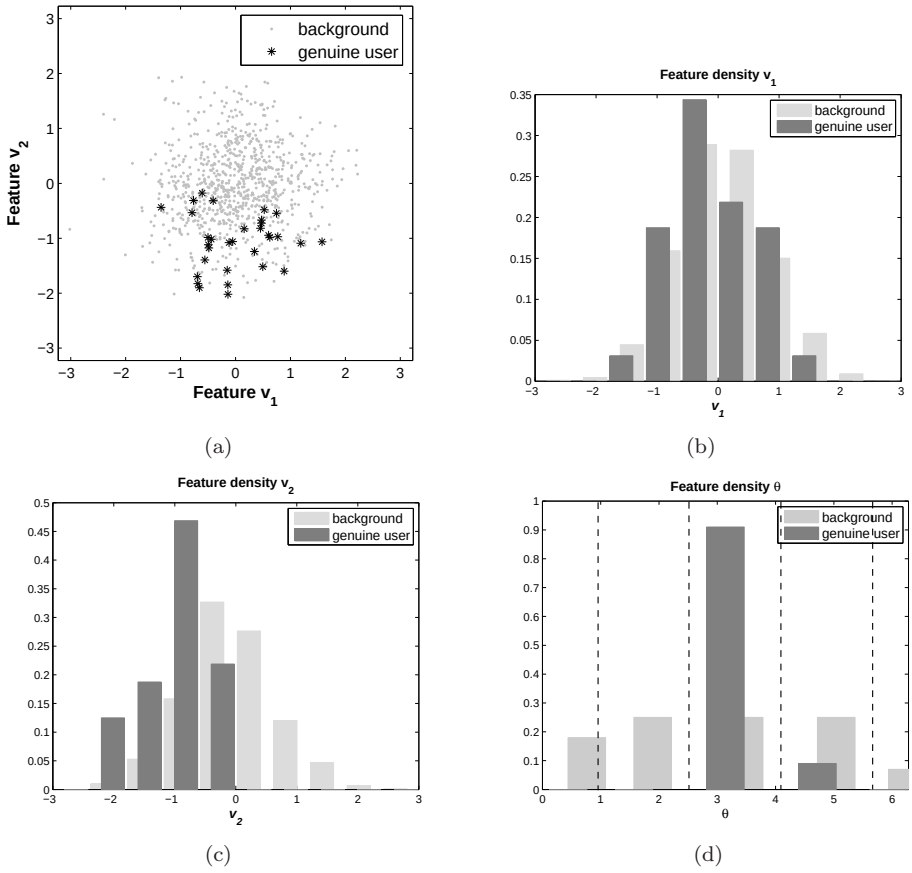


Figure 7.20: *An example of the feature density based on LS pairing and APQ. (a) the two-dimensional feature density; (b) the density of v_1; (c) the density of v_2; (d) the pairwise phase density of $\{v_1\ v_2\}$, with the adaptive quantization boundaries (dashed line).*

8 Conclusions

8.1 Research objectives

The context of this research is the development of a generic template protection scheme for biometric verification applications. The Helper Data scheme is chosen as a vehicle in this research. The whole template protection system is divided into three functional modules: feature extraction, reliable bit extraction and secure key binding verification. Reliable bit extraction is crucial for the template protection performance. Therefore, it is the main purpose of this research.

The research question addressed in this thesis is:
How can real-valued biometric features, in a Helper Data scheme based template protection system, be converted to a binary string, with the following requirements?

I. Since we adopt the Helper Data scheme, the binary strings extracted from the real-valued biometric features should be of fixed-length.

II. 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.*).

III. In order to maximize the length of the random key, the extracted bits should be as reliable as possible, i.e. for a given user the probability of bit errors should be as low as possible.

IV. The verification via binary strings should not degrade the FAR and the FRR performances.

Each of these requirements is translated into a corresponding research objective.

8.2 Contributions

The main contributions of the thesis involve two aspects: (1) how to optimize the quantization intervals for the biometric features and (2) how to allocate the bits to the features. Chapters 2 to 7 present methods regarding these two aspects.

In Chapter 2, two one-dimensional quantizers, the fixed quantizer (FQ) and the likelihood ratio based quantizer (LQ), are presented. Both quantizers are able to extract multiple bits per biometric feature. The FQ determines the quantization intervals merely by equally dividing the probability mass of the background probability density function (PDF). The LQ determines the quantization intervals from the likelihood ratio between the genuine user PDF and the background PDF of the feature. As a result, both quantizers are able to extract *i.i.d.* bits. Superior to FQ, LQ optimizes the theoretical FRR of a feature, given a prescribed number of quantization bits.

In Chapter 3, the detection rate optimized bit allocation (DROBA) principle is presented. Subject to a prescribed total length of the binary string, DROBA assigns user-dependent numbers of bits to every feature, in such way that the theoretical overall detection rate at zero Hamming distance threshold for a HDC is optimized. Both a dynamic programming and a greedy approach are then proposed to search for the optimal solution. Compared to quantizing every feature into a prescribed fixed number of bits, combining quantizers with DROBA yields better FAR and FRR performances of the entire binary strings.

In Chapter 4, the area under the FRR curve optimized bit allocation (AUF-OBA) principle is presented. Given the bit error probabilities of the biometric features, AUF-OBA assigns user-dependent numbers of bits to every feature, in such way that the theoretical area under the FRR curve for a HDC is minimized. A dynamic programming approach is then proposed to search for the optimal solution. Superior to DROBA, AUF-OBA optimizes the overall FRR performances, rather than the FRR at zero Hamming distance threshold.

In Chapter 5, the weighted area under the FRR curve optimized bit allocation (WAUF-OBA) principle is presented. Given the bit error probabilities of the biometric features, WAUF-OBA assigns user-dependent numbers of bits to every feature, in such way that the theoretical weighted area under the FRR curve for a HDC is minimized. Depending on the value of the parameter in the weighting function, different ranges of the Hamming distance thresholds are emphasized, which makes WAUF-OBA a generalization of DROBA and AUF-OBA. Superior to DROBA or AUF-OBA, WAUF-OBA optimizes the overall FRR performances in the emphasized range of Hamming distance thresholds.

In Chapter 6, a two-dimensional pairwise polar quantizer that quantizes the magnitude or the phase is introduced. Quantization intervals are dependent on the background PDFs of the pairwise features in either domain. Furthermore, aiming to optimize the discrimination between the genuine Hamming distance (GHD) distribution and the imposter Hamming distance (IHD) distribution, two heuristic feature pairing strategies are proposed: the long-short (LS) strategy for the phase quantization, as well as the long-long (LL) strategy for the magnitude quantization. The phase quantizer combined with the LS pairing gives low FAR and FRR performances.

In Chapter 7, a two-dimensional pairwise adaptive phase quantizer (APQ), together with an improved long-short (LS) pairing strategy, is presented. The APQ adjust the phase quantization intervals in order to maximize the theoretical detection rate of a given feature pair. The LS pairing strategy composes feature pairs in order to maximize the overall detection rate, for the total binary strings, at zero Hamming distance threshold. With APQ and LS pairing, the extracted binary strings obtain better FAR and FRR performances than the phase quantizer without adjustment in Chapter 6.

8.3 Discussion of achievements

To summarize, in this thesis, we present solutions to extract bits from biometric features. These solutions range from quantizers that extract bits from a feature or feature pair, to the bit allocation principles that assign user-dependent numbers of bits to every feature. The properties of these solutions regarding the defined research objectives are summarized in Table 8.1 and 8.2.

Regarding **research objective I**, both the one-dimensional quantizers (FQ, LQ) and the two-dimensional quantizers (polar quantizer, APQ) are capable of extracting a fixed number of bits per feature or feature pair. While the length of the total binary string is fixed, all three bit allocation principles (DROBA, AUF-OBA, WAUF-OBA) are capable of assigning user-dependent numbers of bits to every feature.

To extract *i.i.d.* bits, as required in **research objective II**, all the quantizers determine quantization intervals of equal background probability. The bit allocation principles, on the other hand, are independent of the quantization intervals. Hence, the bit allocation principles preserve the *i.i.d.* bits property of the binary strings. Furthermore, by giving equal-probability quantization intervals, the theoretical FAR for a HDC, as required in the research objective IV, is then fixed.

Optimizing the theoretical FRR at the fixed FAR, or equivalently extracting reliable bits, is the main task for **research objective III and IV**. For one-dimensional quantizer FQ, FRR optimization is not considered for every feature. Superior to FQ, LQ minimizes the FRR at zero Hamming distance per feature. In case of the two-dimensional phase or magnitude quantizer, although the quantization intervals do not optimize the FRR per feature pair, the configurations of the feature pairs (LS and LL pairing) optimize the overall FRR binary string performance of its quantizer. On real data experiments, however, only the phase quantizer combined with LS pairing gives good recognition performances. Therefore, in the next step, APQ improves the phase quantizer by optimizing the FRR per feature pair. Compared to one-dimensional quantizers (e.g. FQ and LQ), APQ with LS pairing has the advantage that adaptive quantization intervals in the phase domain might extract more reliable bits, and the pairing strategy compose feature pairs to optimize the overall FRR.

Instead of pairing, a solution to optimize the overall FRR in case of the one-dimensional quantization is to apply bit allocation principles: DROBA optimizes the overall FRR at zero Hamming distance threshold. Superior to DROBA, AUF-OBA optimizes the overall FRR over all Hamming distance thresholds, or equivalently,

Table 8.1: *The solutions and their properties to the defined research objectives. d_{H} refers to the Hamming distance.*

Objective	Solutions	Properties
I. Fixed-length	FQ and LQ Polar quantizer APQ	Fixed-length per feature or feature pair
	DROBA AUF-OBA WAUF-OBA	Fixed-length of the total binary string
II. *i.i.d.* bits	FQ and LQ Polar quantizer APQ	*i.i.d.* bits per feature or feature pair
	DROBA AUF-OBA WAUF-OBA	Preserve the *i.i.d.* bits extracted from the quantizers
III. Reliable bits	FQ	Not considered
	LQ	Minimize the FRR at $d_{\mathrm{H}} = 0$ per feature
	Polar quantizer	Minimize the FRR of the binary string via pairing
	APQ	Minimize the FRR at $d_{\mathrm{H}} = 0$ of the binary string via APQ and pairing
	DROBA	Minimize the FRR at $d_{\mathrm{H}} = 0$ of the binary string
	AUF-OBA	Minimize the area under the FRR curve of the binary string
	WAUF-OBA	Minimize the weighted area under the FRR curve of the binary string

Table 8.2: *(*cont'd*) The solutions and their properties to the defined research objectives.* d_{H} *refers to the Hamming distance.*

Objective	Solutions	Properties
IV. FAR	FQ and LQ Polar quantizer APQ	Fixed FAR due to *i.i.d.* bits
	DROBA AUF-OBA WAUF-OBA	Preserve the fixed FAR
IV. FRR	FQ	Not considered
	LQ	Minimize the FRR at $d_{\mathrm{H}} = 0$ per feature
	Polar quantizer	Minimize the FRR of the total binary string via pairing
	APQ	Minimize the FRR at $d_{\mathrm{H}} = 0$ of the binary string via APQ and pairing
	DROBA	Minimize the FRR at $d_{\mathrm{H}} = 0$ of the binary string
	AUF-OBA	Minimize the area under the FRR curve of the binary string
	WAUF-OBA	Minimize the weighted area under the FRR curve of the binary string

the area under the FRR curve. WAUF-OBA then optimizes the overall FRR with emphasis on a range of Hamming distance thresholds. WAUF-OBA in fact is a generalization of DROBA and AUF-OBA. Combined with one-dimensional quantizers, the bit allocation principles improves the overall FRR of the total binary strings. Although bit allocation principles can also be applied with the two-dimensional phase quantizer, we show that LS pairing strategy has achieved equivalently good performances. Therefore, in the thesis, bit allocation principles are not presented in the two-dimensional quantization scheme.

In practical applications, the choice of the appropriate bit extraction solutions depends on the properties of the data as well as the application context. The more optimal solutions, e.g. LQ, AUF-OBA, WAUF-OBA, also require more accurate modeling of the feature distributions and more computational complexity, which are often difficult to achieve in practical applications. Considering such trade-off, we recommend FQ and DROBA for the one-dimensional quantization scheme, which gives a roughly accurate and optimal solution. In case of the two-dimensional quantization scheme, we recommend APQ combined with LS pairing.

8.4 Future work

The main objective of the thesis is to extract biometric binary strings from real-valued features, leaving independent, discriminative and reliable biometric features an important assumption in the research. Unfortunately, in practice, most of the biometric features retain large intra-class variations. Consequently, the extracted bits are still error-prone, even though the quantization and coding procedure is well-designed. If the bits are less reliable, an advanced ECC is required, otherwise the number of secrets dramatically decreases. Another weak point is that the bit extraction procedure relies on the user-dependent feature distributions. However, in most of the current systems, only a few samples are captured for every enrolled user, making it difficult to accurately estimate the feature distributions. For these reasons, the future work will focus on the following aspects:

- **Improving feature quality** Higher feature quality directly gives more reliable features and therefore benefits in reliable bits. The straightforward solution to improve the feature quality is to optimize the biometric capture procedure. For instance, to provide a more controlled verification environment, or to improve the quality of the features that are captured, by enhancing both the hardware and the software of the biometric sensors. An alternative solution is to introduce a quality control step. Finally, despite the common PCA or LDA feature reduction methods, it is possible to employ some other methods to extract more reliable features.

- **Improving the modeling of the feature distributions** Optimal quantization intervals and bit allocation are dependent on the biometric feature distributions, which are estimated from the samples of the enrolled users. However,

in practice, it is impossible to obtain a sufficiently large number of samples during the enrollment. Further work would be to define, for every user, a proper number of samples required for a good estimation. As a result, the number of samples required could vary from user to user. Furthermore, an adaptive system could be employed to gradually enrich the modeling of the feature distributions.

- **Designing ECC** An ECC that can recover more erroneous bits will give better recognition performance as well as more secret bits. Therefore, according to the bit error probabilities of the biometric features, an advanced ECC is desirable.

- **Comparing or employing multiple biometric modalities** In addition to the template protection purpose, extracting biometric binary strings enables the opportunity to compute and compare the capacities of different biometric modalities. For instance, how many secret bits does this biometric fingerprint application really have? as compared to the other application? Furthermore, for a single biometric modality, the number of secret bits is relatively low. Thus, employing multiple biometric modalities might increase the number of secret bits.

A

Proving Optimal of the Dynamic Programming Approach

The question has to be answered whether the dynamic programming approach presented above will lead to the optimal bit assignment. The proof is as follows. Denote the optimal bit allocation over D' features by

$$\{\hat{b}_i(l)\} = \arg \max_{b_i | \sum b_i = l,\ b_i \in \{0,\ldots,b_{\max}\}} \prod_{i=1}^{D'} \delta_i(b_i),$$

and denote the maximum obtained by $\delta_{\max}$. Assume that we have a partitioning of the D' features into two arbitrary sets. The sets are fully characterized by the indices of the features, so we can speak of the index sets as well. Let $\mathcal{M}$ and $\mathcal{N}$ denote the index sets, such that $\mathcal{M} \cap \mathcal{N} = \emptyset$ and $\mathcal{M} \cup \mathcal{N} = \{1, \ldots, D'\}$. Define

$$\delta^{\mathcal{M}}(l) = \max_{b_i | \sum b_i = l,\ b_i \in \{0,\ldots,b_{\max}\}} \prod_{i \in \mathcal{M}} \delta_i(b_i),\ l = 0, \ldots, |\mathcal{M}| b_{\max},$$

and

$$\delta^{\mathcal{N}}(l) = \max_{b_i | \sum b_i = l,\ b_i \in \{0,\ldots,b_{\max}\}} \prod_{i \in \mathcal{N}} \delta_i(b_i),\ l = 0, \ldots, |\mathcal{N}| b_{\max},$$

Define

$$\hat{l}^{\mathcal{M}} = \sum_{i \in \mathcal{M}} \hat{b}_i(l)$$

and

$$\hat{l}^{\mathcal{N}} = \sum_{i \in \mathcal{N}} \hat{b}_i(l)$$

Now

$$\begin{aligned}
\max_{l',l''|l'+l''=l,l'\in\mathcal{M},l''\in\mathcal{N}} \delta^{\mathcal{M}}(l')\delta^{\mathcal{N}}(l'') &\geq \delta^{\mathcal{M}}(\hat{l}^{\mathcal{M}})\delta^{\mathcal{N}}(\hat{l}^{\mathcal{N}}) \\
&\geq \prod_{i\in\mathcal{M}} \delta_i(\hat{b}_i(l)) \prod_{i\in\mathcal{N}} \delta_i(\hat{b}_i(l)) \\
&= \prod_{i=1}^{D'} \delta_i(\hat{b}_i(l)) \\
&= \delta_{\max}.
\end{aligned}$$

The left-hand side of this inequality is a product of the form

$$\prod_{i=1}^{D'} \delta_i(b_i),$$

with the b_i constrained by $\sum b_i = l,\ b_i \in \{0,\ldots,b_{\max}\}$. This cannot, by definition, be greater than $\delta_{\max}$. Therefore, it must be identical to $\delta_{\max}$.

Note that the partitioning into index sets $\mathcal{M}$ and $\mathcal{N}$ was arbitrary. If we take $D' = j$, $\mathcal{M} = \{1,\ldots,j-1\}$, and $\mathcal{N} = \{j\}$ then we have proved that the jth recursion step of the above algorithm is optimal.

B

Derivation of the FAR for HDC

In order to prove (4.10), we only need to prove

$$\phi_{\mathrm{i}}(k;\{b_j\}_{j=1}^{D}) = 2^{-L}\left(\begin{array}{c} L \\ k \end{array}\right) . \tag{B.1}$$

Proof. Note that for binomial coefficients $\left(\begin{array}{c} m \\ q \end{array}\right)$ and $\left(\begin{array}{c} n \\ p-q \end{array}\right)$ Vandermonde's identity states that

$$\sum_{q=0}^{p}\left(\begin{array}{c} m \\ q \end{array}\right)\left(\begin{array}{c} n \\ p-q \end{array}\right) = \left(\begin{array}{c} m+n \\ p \end{array}\right) . \tag{B.2}$$

Thus, for instance, by using (4.9) we obtain

$$\begin{aligned}\sum_{l=0}^{k} P_{\mathrm{i},1}(l;b_1)P_{\mathrm{i},2}(k-l;b_2) &= \sum_{l=0}^{k} 2^{-b_1}\left(\begin{array}{c} b_1 \\ l \end{array}\right) 2^{-b_2}\left(\begin{array}{c} b_2 \\ k-l \end{array}\right) \\ &= 2^{-(b_1+b_2)}\left(\begin{array}{c} b_1+b_2 \\ k \end{array}\right) . \end{aligned} \tag{B.3}$$

Expression (B.3) in fact computes the convolution of the bit error probabilities of two features. In the case of D features, as in (4.6), ϕ_{i} is the convolution from all the D features. Therefore, we can apply (B.3) repetitively to all the D features. For

instance, to convolve with the third feature, we have

$$\begin{aligned}
&\sum_{m=0}^{k}\left[\sum_{l=0}^{m} P_{\mathrm{i},1}(l;b_1)P_{\mathrm{i},2}(m-l;b_2)\right]P_{\mathrm{i},3}(k-m;b_3)\\
&\quad=\sum_{m=0}^{k} 2^{-(b_1+b_2)}\begin{pmatrix} b_1+b_2 \\ m \end{pmatrix} 2^{-b_3}\begin{pmatrix} b_2 \\ k-m \end{pmatrix}\\
&\qquad=2^{-(b_1+b_2+b_3)}\begin{pmatrix} b_1+b_2+b_3 \\ k \end{pmatrix}. \qquad \text{(B.4)}
\end{aligned}$$

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

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 $\begin{pmatrix} L \\ k \end{pmatrix}$ possibilities to select k bits. □

C

Dynamic Programming Approach for AUF-OBA

Algorithm 3 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:

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

while $n \neq D$ **do**

$$\begin{aligned} n &= n + 1\ , \\ \hat{b}', \hat{b}'' &= \arg\max G^{(n-1)}(b') + G_n(b'')\ , \\ &\qquad \begin{array}{l} b' + b'' = l, \\ b' \in \{0, \ldots, (n-1) \times b_{\max}\}, \\ b'' \in \{0, \ldots, b_{\max}\} \end{array} \\ &\quad l = 0, \ldots, n \times b_{\max}\ , \\ G^{(n)}(l) &= G^{(n-1)}(\hat{b}') + G_n(\hat{b}'')\ , \\ b_j(l) &= b_j(\hat{b}'), j = 1, \ldots, n-1\ , \\ b_n(l) &= \hat{b}''\ , \end{aligned}$$

end while

Output:

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

D

Derivation of the Optimization Problem for WAUF-OBA

We first reformulate $\beta(t;\{b_i\})$ in (5.10) into the following expression:

$$\beta(t;\{b_i\}) \quad = \quad \sum_{k=0}^{L} \mathrm{u}(k-(t+1))\phi_{\mathrm{g}}(k;\{b_i\}) \ ,$$

with

$$\mathrm{u}(k) \stackrel{\mathrm{def}}{=} \begin{cases} 1, & k \geq 0 \ , \\ 0, & k < 0 \ . \end{cases} \tag{D.1}$$

Therefore the weighted area under the FRR curve becomes:

$$\begin{aligned} A_{\mathrm{FRR}} &= \sum_{t=0}^{L} z^{-t} \sum_{k=t+1}^{L} \phi_{\mathrm{g}}(k;\{b_i\}) \\ &= \sum_{t=0}^{L} z^{-t} \sum_{k=0}^{L} \Big[\mathrm{u}(k-(t+1))\phi_{\mathrm{g}}(k;\{b_i\}) \Big] \\ &= \sum_{t=0}^{L} \sum_{k=0}^{L} \Big[z^{-t}\mathrm{u}(k-(t+1))\phi_{\mathrm{g}}(k;\{b_i\}) \Big] \\ &= \sum_{k=0}^{L} \Big[\phi_{\mathrm{g}}(k;\{b_i\}) \sum_{t=0}^{L} z^{-t}\mathrm{u}(k-(t+1)) \Big] \\ &= \sum_{k=0}^{L} \Big[\phi_{\mathrm{g}}(k;\{b_i\}) \sum_{t=0}^{k-1} z^{-t} \Big] \ . \end{aligned} \tag{D.2}$$

D.1 $z \neq 1$, $z > 0$

When $z \neq 1$, $z > 0$, Eq. (D.2) becomes:

$$\begin{aligned} A_{\text{FRR}} &= \sum_{k=0}^{L} \left[\phi_{\text{g}}(k; \{b_i\}) \frac{1 - z^{-k}}{1 - z^{-1}} \right] \\ &= \frac{1}{1 - z^{-1}} \sum_{k=0}^{L} \left[\phi_{\text{g}}(k; \{b_i\})(1 - z^{-k}) \right] . \end{aligned} \tag{D.3}$$

We know that:

$$\sum_{k=0}^{L} \phi_{\text{g}}(k; \{b_i\}) = 1 . \tag{D.4}$$

Therefore, A_{FRR} equals to:

$$A_{\text{FRR}} = \frac{1}{1 - z^{-1}} \left[1 - \sum_{k=0}^{L} (\phi_{\text{g}}(k; \{b_i\}) z^{-k}) \right] . \tag{D.5}$$

Hence, the optimization problem in (5.11) is reformulated as:

$$\begin{cases} \max \sum_{k=0}^{L} -z^{-k} \phi_{\text{g}}(k; \{b_i\}), & 0 < z < 1 , \\ \max \sum_{k=0}^{L} z^{-k} \phi_{\text{g}}(k; \{b_i\}), & z > 1 , \end{cases} \tag{D.6}$$

The expression in (D.6) can be seen as the following Z-transform:

$$X(z) = \mathcal{Z}\{x[k]\} = \sum_{k=0}^{L} x[k] z^{-k} , \tag{D.7}$$

with

$$x[k] = \phi_{\text{g}}(k; \{b_i\}) . \tag{D.8}$$

Furthermore, we recall that in (5.5) we have:

$$\phi_{\text{g}}(k) = (P_{\text{g},1} * P_{\text{g},2} * \ldots * P_{\text{g},D})(k) . \tag{D.9}$$

According to the convolution property of Z-transform, (D.7) can be written as:

$$\begin{aligned} X(z) &= \mathcal{Z}\{\phi_{\text{g}}(k; \{b_i\})\} \\ &= \mathcal{Z}\{(P_{\text{g},1} * P_{\text{g},2} * \ldots * P_{\text{g},D})(k)\} \\ &= \mathcal{Z}\{P_{\text{g},1}(k)\} \cdot \mathcal{Z}\{P_{\text{g},2}(k)\} \cdot \ldots \cdot \mathcal{Z}\{P_{\text{g},D}(k)\} \\ &= \prod_{i=1}^{D} \left[\sum_{k_i=0}^{b_i} P_{\text{g},i}(k_i, b_i) z^{-k_i} \right] . \end{aligned} \tag{D.10}$$

By taking the logarithm of $X(z)$, and let $G_i(b_i)$ be a gain factor:

$$G_i(b_i) \quad = \quad \begin{cases} -\log\left(\sum_{k_i=0}^{b_i} z^{-k_i} P_{\mathrm{g},i}(k_i, b_i)\right) , & 0 < z < 1 , \\ \log\left(\sum_{k_i=0}^{b_i} z^{-k_i} P_{\mathrm{g},i}(k_i, b_i)\right) , & z > 1 , \end{cases} \tag{D.11}$$

the optimization problem then becomes:

$$\{b_i^*\} = \arg \max_{\sum_{i=1}^{D} b_i = L} \sum_{i=1}^{D} G_i(b_i) . \tag{D.12}$$

D.2 $\mathbf{z = 1}$

When $z = 1$, Eq. (D.2) becomes:

$$A_{\mathrm{FRR}} \quad = \quad \sum_{k=0}^{L} k \phi_{\mathrm{g}}(k; \{b_i\}) . \tag{D.13}$$

Hence, the area under the false-reject rate is the expected value of the number of bit errors . Furthermore, we know that the k bits errors of a L-bit binary string are supposed to come from D real-valued features. Thus with k_i $(i = 1, \ldots, D)$ bits error per feature. Of course, we have that the expected value of a sum equals the sum of the expected values. Therefore,

$$A_{\mathrm{FRR}} \quad = \quad \sum_{i=1}^{D} \sum_{k_i=0}^{b_i} k_i P_{\mathrm{g},i}(k_i, b_i) . \tag{D.14}$$

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

$$G_i(b_i) = - \sum_{k_i=0}^{b_i} k_i P_{\mathrm{g},i}(k_i, b_i) , \tag{D.15}$$

the optimization problem then becomes (D.12).

D.3 $\mathbf{z \to \infty}$

In an extreme case when $z \to \infty$, Let $G_i(b_i)$ be a gain factor:

$$\begin{aligned} G_i(b_i) \quad &= \quad \lim_{z \to \infty} \log\left(\sum_{k_i=0}^{b_i} z^{-k_i} P_{\mathrm{g},i}(k_i, b_i)\right) \\ &= \quad \log P_{\mathrm{g},i}(0, b_i) . \end{aligned} \tag{D.16}$$

Therefore, the optimization problem then becomes (D.12).

Bibliography

[1] A. Jain, P. Flynn, and A. Ross, *Handbook of Biometrics*. USA: Springer, 2008.

[2] A. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics*, vol. 14, no. 1, pp. 4–20, 2004.

[3] A. Adler, "Sample images can be independently restored from face recognition templates," in *Proc. IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2003)*, (Montral, Canada), 2003.

[4] A. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," *EURASIP Journal on Advances in Signal Processing*, no. 113, 2008.

[5] F. Monrose, M. Reiter, and S. Wetzel, "Password hardening based on keystroke dynamics," in *Proc. Sixth ACM conference on Computer and communications security*, (Singapore), pp. 73–82, 1999.

[6] F. Monrose, M. Reiter, Q. Li, and S. Wetzel, "Cryptographic key generation from voice," in *Proc. IEEE Symposium on Security and Privacy (S&P 2001)*, (CA, USA), pp. 202–213, May 2001.

[7] H. Feng and C. Wah, "Private key generation from on-line handwritten signatures," *Information Management and Computer Security*, vol. 10, no. 4, pp. 159–164, 2002.

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

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

[10] W. Zhang and T. Chen, "Generalized optimal thresholding for biometric key generation using face images," in *Proc. IEEE International Conference on Image Processing (ICIP 2005)*, vol. 3, (PA, USA), pp. 784–787, 2005.

[11] Y. Dodis, M. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. International Conference*

on the Theory and Applications of Cryptographic Techniques, vol. 3027, (Switzerland), pp. 523–540, May 2004.

[12] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith, "Secure remote authentication using biometric data," in *EUROCRYPT*, pp. 147–163, 2005.

[13] I. Buhan, J. Doumen, P. Hartel, and R. Veldhuis, "Fuzzy extractors for continuous distributions," in *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS), Singapore*, pp. 353–355, ACM, 2007.

[14] T. Connie, A. Teoh, M. Goh, and D. Ngo, "Palmhashing: a novel approach for dual-factor authentication," *Pattern Analysis and Applications*, vol. 7, no. 3, pp. 255–268, 2004.

[15] A. Teoh, A. Goh, and D. Ngo, "Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs," vol. 28, no. 12, pp. 1892–1901, 2006.

[16] A. Kong, K. Cheung, D. Zhang, M. Kamel, and J. You, "An analysis of biohashing and its variants," *Pattern Recognition*, vol. 39, no. 7, pp. 1359–1368, 2006.

[17] A. Lumini and L. Nanni, "An improved biohashing for human authentication," *Pattern Recognition*, vol. 40, no. 3, pp. 1057–1065, 2007.

[18] L. Nanni and A. Lumini, "Random subspace for an improved biohashing for face authentication," *Pattern Recognition Letters*, vol. 29, no. 3, pp. 295–300, 2008.

[19] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614–634, 2001.

[20] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle, "Generating cancelable fingerprint templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 561–572, 2007.

[21] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proc. Sixth ACM Conference on Computer and Communications Security*, pp. 28–36, 1999.

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

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

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

[25] F. Hao, R. Anderson, and J. Daugman, "Combining cryptography with biometrics effectively," vol. 55, no. 9, pp. 1081–1088, 2006.

[26] E.-C. Chang and S. Roy, "Robust extraction of secret bits from minutiae," in *Proc. The 2nd International Conference on Biometrics, ICB*, pp. 750–759, 2007.

[27] Q. Li and E.-C. Chang, "Robust, short and sensitive authentication tags using secure sketch," in *Proceedings of the 8th workshop on Multimedia and security*, (New York, NY, USA), pp. 56–61, ACM, 2006.

[28] Q. Li, Y. Sutcu, and N. Memon, "Secure sketch for biometric templates," in *Advances in Cryptology Asiacrypt*, pp. 99–113, Springer-Verlag, 2006.

[29] Y. Sutcu, Q. Li, and N. D. Memon, "Protecting biometric templates with sketch: Theory and practice," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3-2, pp. 503–512, 2007.

[30] Y. Sutcu, Q. Li, and N. D. Memon, "Secure biometric templates from fingerprint-face features," in *CVPR*, 2007.

[31] U. Uludag, S. Pankanti, and A. K. Jain, "Fuzzy vault for fingerprints.," in *Fifth Int. Conf. on Audio- and Video-Based Biometric Person Authentication (AVBPA 2005)* (T. Kanade, A. K. Jain, and N. K. Ratha, eds.), vol. 3546 of *Lecture Notes in Computer Science*, pp. 310–319, Springer-Verlag, 2005.

[32] A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 237–257, 2006.

[33] K. Nandakumar, A. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," *Information Forensics and Security, IEEE Transactions on*, vol. 2, pp. 744–757, Dec. 2007.

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

[35] D. Maio, D. Maltoni, R. Cappelli, J.L., and A. Jain, "FVC2000: Fingerprint verification competition," vol. 24, no. 3, pp. 402–412, 2002.

[36] "FVC2000 fingerprint verification competition." `http://bias.csr.unibo.it/fvc2000/default.asp`.

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

[38] "NIST: FRGC face db." `http://www.frvt.org/frgc/`.

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

[40] U. Uludag, S. Pankanti, S. Prabhakar, and A. Jain, "Biometric cryptosystems: Issues and challenges," vol. 92, no. 6, pp. 948–960, 2004.

[41] A. Webb, *Statistical Pattern Recognition*. England: John Wiley & Sons, LTD, second ed., 2002.

[42] R. 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)*, vol. 5306, (CA, USA), pp. 634–641, 2004.

[43] A. M. Bazen and R. N. J. Veldhuis, "Likelihood-ratio-based biometric verification," vol. 14, no. 1, pp. 86–94, 2004.

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

[45] C. Chen, R. Veldhuis, T. Kevenaar, and A. Akkermans, "Biometric quantization through detection rate optimized bit allocation," *EURASIP Journal on Advances in Signal Processing*, 2009.

[46] F. Hao and C. Wah, "Private key generation from on-line handwritten signatures," *Information Management & Computer Security*, vol. 10, no. 4, pp. 159–164, 2002.

[47] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. England: The MIT Press, second ed., 2001.

[48] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1445–1453, 1988.

[49] V. Perlibakas, "Distance measures for pca-based face recognition," *Pattern Recognition Letters*, vol. 25, no. 6, pp. 711–724, 2004.

[50] C. Chen and R. Veldhuis, "Extracting biometric binary strings with minimal area under the frr curve for the hamming distance classifier," *Journal Signal Processing*, vol. 91, pp. 906–918.

[51] C. Chen and 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.

[52] J. Daugman, "Biometric decision landscapes," *Technical Report No. TR482, University of Cambridge Computer Laboratory*, 2000.

[53] J. Daugman, "The importance of being random: Statistical principles of iris recognition," *Pattern Recognition*, vol. 36, no. 2, pp. 279–291, 2003.

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

[55] C. Chen and R. Veldhuis, "Extracting biometric binary strings with optimal weighted area under the frr curve for the hamming distance classifer," in *The 30th Symposium on Information Theory in the Benelux (WIC2009)*, 2009.

[56] C. Chen, R. Veldhuis, T. Kevenaar, and A. Akkermans, "Biometric binary string generation with detection rate optimized bit allocation," in *Computer Vision and Pattern Recognition Workshops (CVPR 2008). IEEE Computer Society Conference on*, 2008.

[57] C. Chen and R. Veldhuis, "Binary biometric representation through pairwise polar quantization," in *The 3rd IAPR/IEEE International Conference on Biometrics (ICB09)*, 2009.

[58] H. Pobloth, R. Vafin, and W. Kleijn, "Multivariate block polar quantization," vol. 53, no. 12, pp. 2043–2053, 2005.

[59] C. Chen and R. Veldhuis, "Binary biometric representation through pairwise adaptive phase quantization," *EURASIP Journal on Information Security*, 2011.

List of publications

[60] R. N.J. Veldhuis, C. Chen, T. A.M. Kevenaar, and T. H.M. Akkermans, Method for Transforming a Feature Vector, WO patent No. 2009034498 (A1) (published March 19, 2009)

[61] C. Chen, R. N.J. Veldhuis, T. A.M. Kevenaar, and T. H.M. Akkermans, "Multi-bits biometric string generation based on the likelihood ratio," in *Proceedings of the 1st IEEE Conference on Biometrics: Theory, Applications and Systems (BTAS'07)*, (Crystal City, Va, USA), pp. 1–6, September 2007.

[62] C. Chen, R. N.J. Veldhuis, T. A.M. Kevenaar, and T. H.M. Akkermans, "Biometric binary string generation with detection rate optimized bit allocation," in *Computer Vision and Pattern Recognition Workshops (CVPR 2008). IEEE Computer Society Conference on*, (Anchorage, Alaska, USA), June 2008.

[63] C. Chen, R. N.J. Veldhuis, T. A.M. Kevenaar, and T. H.M. Akkermans, "Performances of the likelihood-ratio classifier based on different data modelings," in *Control, Automation, Robotics and Vision, 2008. (ICARCV 2008). 10th International Conference on*, (Hanoi, Vietnam), December 2008.

[64] C. Chen, R. N.J. Veldhuis, T. A.M. Kevenaar, and T. H.M. Akkermans, "Biometric quantization through detection rate optimized bit allocation," *EURASIP Journal on Advances in Signal Processing*, 2009.

[65] C. Chen, and R. N.J. Veldhuis, "Extracting biometric binary strings with optimal weighted area under the FRR curve for the Hamming distance classifer," in *The 30th Symposium on Information Theory in the Benelux (WIC2009)*, (Eindhoven, Netherlands), May, 2009.

[66] C. Chen, and R. N.J. Veldhuis, "Binary biometric representation through pairwise polar quantization," in *The 3rd IAPR/IEEE International Conference on Biometrics (ICB09)*, (Sassari, Italy), June, 2009.

[67] C. Chen, and R. N.J. Veldhuis, "Extracting biometric binary strings with minimal area under the FRR curve for the Hamming distance classifier," in *The 17th European Signal Processing Conference (EUSIPCO 2009)*, (Glasgow, Scotland), August, 2009.

[68] E. J.C. Kelkboom, K. T.J. Groot de, C. Chen, J. Breebaart, and R. N.J. Veldhuis, "Pitfall of the detection rate optimized bit allocation within template protection and a remedy," in *Proceedings of the 3rd IEEE Conference on Biometrics: Theory, Applications and Systems (BTAS'09)*, (Washington DC, USA), September, 2009.

[69] C. Chen, and R. N.J. Veldhuis, "Extracting biometric binary strings with minimal area under the FRR curve for the hamming distance classifier," *Signal Processing*, vol. 91, pp. 906–918, October, 2010.

[70] C. Chen, and R. N.J. Veldhuis, "Binary biometric representation through pairwise adaptive phase quantization," *EURASIP Journal on Information Security*, 2011.

Acknowledgement

As I write this last section of my thesis, I am looking back at the years I spent in this beautiful tiny country. It was, I would say without doubt, a unique adventure for me! (Chun in Nederland) An adventure filled with joy, passion, curiosity, but also accompanied with anger, sorrow and despair; An adventure to find who I am; An adventure to understand lifeAn adventure that I could not accomplish without all the people that I have encountered! I would like to thank all of you, for a word, a smile, a moment, that you have shared with me!

Thank you prof. Kees Slump for supporting my PhD project. It was a great pleasure working in your group.

Truly, madly, deeply many gratitude to my supervisor Raymond Veldhuis! I appreciate your passion. You are the button of my inspiration machine, a crazy professor who is still hunting for time to do research on his own. Many times, you jumped in front of my office door and said to me in excitement: "I just had an idea, do you want to hear?" As a way of communication, I am also allowed to jump in front of your office door and shout: "Raymond, I was thinking ..." Looking back to our discussions, it was full of joy and inspirations; it does not feel like we were discussing a problem, instead, we were expressing our ways of thinking. I appreciate your patience. Whenever I got puzzled, you were always ready to explain (in ≥ 1 times), every time in a different way. I appreciate your generosity. I was allowed to say "you are wrong" and make decisions on my own, although most of the times it turned out that you were right. Making my 'own' mistakes helps me to learn faster! I appreciate your honesty. You are brave enough to say "I don't know" when you don't know. Finally, I appreciate your kindness. You were willing to buy my childish behavior but always pay me back with your lifetime proverbs.

Many thanks to my "old" friend Tom Kevenaar. I remembered that the first day we met in Philips, you figured out that you are ** years older than me, but it does not become any obstacle in our friendship at all. I admire your discipline!! You are almost never late in any kinds of appointments, which is almost impossible! The same as me, you are a sensitive person; I enjoyed sharing every tiny piece of feeling about life with you, no matter good or bad. Thank you and your wife Chieko for bringing a lot of fun in my life.

And of course I owe many thanks to my sisterhood in Enschede: Xiaoxin Shang and Shuo Kang. We together were "three musketeers". One for all, all for one. We laughed for each other's happiness; we cried for each other's sorrow; we fought for each other's injustice; we made decisions together. Thank you for always standing by me. I miss a lot the days that we laughed and cooked and ate and laughed and ate

. . .

Also many thanks to my supervisors from my master study: Letty Weger, Berend Stoel, Emile Hendriks and Bob Duin. It was my first time to work with so many people in different expertise and gender. But still it turned out to be very successful! It was also the first time that I realized that as a supervisor, you are not only sharing your knowledge, but also life experience with me.

Thank you Bob Duin. You inspired me to find who I am, and what I want to be, to handle my feelings about life. Your wisdom and encouragement saved me when I was in darkness. You are a real thinker with a kind heart!

Many thanks to my friends Daniela Deiana, Walter Rhee, Ying Zhao, Yu Pan and Chunyan Jia.

I would also like to thank Geert Jan Laanstra, Berk Gokberk, Bas Boom and all the other PhDs/post-docs in the SAS group for valuable discussions, tea/coffees and helps. Thanks to Emile Kelkboom, Ton Akkermans, Ileana Buhan from Philips for the fun moments in Philips. Thanks to Hua Zhang for designing such a sophisticated cover for my thesis.

Finally, I would like to thank my family. Thank you my mum and dad for your endless love and support for me.

Netherlands is a tiny country, but with you all, I have found a great treasure in this land - my free mind!

Chun Chen
Amsterdam

Curriculum Vitae

Chun Chen was born in Nanjing, China, 1981. She received her M.Sc. degree in Electrical Engineering from the Delft University of Technology, The Netherlands, in 2005. For her graduation project, she carried out a study on automated recognition of allergic pollen: grass, birch, mugwort. It was performed under the supervision of dr. E.A. Hendriks, dr. R.P.W. Duin from Delft University of Technology and dr. L.A. de Weger, dr. B.C. Stoel from Leiden University Medical Center.

From September 2005 until September 2009, she was pursuing her Ph.D. degree in the department of Electrical Engineering, University of Twente, under the supervision of Prof. dr. C.H. Slump and dr. R.N.J. Veldhuis. Her research topic was on binary representations for biometric template protection. In 2008, she received the best student paper award in CVPR Workshop for her paper on detection rate optimized bit allocation. In 2009, she received the best student paper award in ICB for her paper on pairwise polar quantization. Her research on a method for transforming a feature vector into a binary string was patented in 2009, together with dr. R.N.J. Veldhuis, dr. T.A.M. Kevenaar and A.H.M. Akkermans.

Since September 2009, she is employed as post-doc at the radiotherapy department of The Netherlands Cancer Institute (NKI), under the supervision of Prof. dr.M. van Herk and dr. J.-J. Sonke. Her research topic is datamining in clinical cancer treatment.