Binary feature fusion for discriminative and secure multi-biometric cryptosystems

Mai, Guangcan; Lim, Meng Hui; Yuen, Pong C.

Published in:
Image and Vision Computing

DOI:
10.1016/j.imavis.2016.11.011

Published: 01/02/2017

Document Version:
Early version, also known as preprint

Link to publication

Citation for published version (APA):
Mai, G., Lim, M. H., & Yuen, P. C. (2017). Binary feature fusion for discriminative and secure multi-biometric cryptosystems. *Image and Vision Computing*, *58*, 254-265. https://doi.org/10.1016/j.imavis.2016.11.011

Fusing Binary Templates for Multi-biometric Cryptosystems

Guangcan Mai, Meng-Hui Lim, Pong C. Yuen*

Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong

Abstract

Biometric cryptosystem has been proven to be a promising approach for template protection. Cryptosystems such as fuzzy extractor and fuzzy commitment require discriminative and informative binary biometric input to offer accurate and secure recognition. In multimodal biometric recognition, binary features can be produced via fusing the real-valued unimodal features and binarizing the fused features. However, when the extracted features of certain modality are represented in binary and the extraction parameters are not known, real-valued features of other modalities need to be binarized and the feature fusion needs to be carried out at the binary level. In this paper, we propose a binary feature fusion method that extracts a set of fused binary features with high discriminability (small intra-user and large inter-user variations) and entropy (weak dependency among bits and high bit uniformity) from multiple sets of binary unimodal features. Unlike existing fusion methods that mainly focus on discriminability, the proposed method focuses on both feature discriminability and system security: The proposed method 1) extracts a set of weakly dependent feature groups from the multiple unimodal features; and 2) fuses each group to a bit using a mapping that minimize the intra-user variations and maximize the inter-user variations and uniformity of the fused bit. Experimental results on three multimodal databases show that fused binary feature of the proposed method has both higher discriminability and higher entropy compared to the unimodal features and the fused features generated from the state-of-the-art binary fusion approaches.

Keywords: Biometric, Binary Representation, Binary Feature, Multi-biometric, Feature Fusion, Template Protection, Cryptosystems

1. Introduction

Multimodal biometric systems, consolidating multiple traits (e.g., face, fingerprint, palmprint, voice, iris), address limitations of unimodal biometric systems in matching accuracy, spoofing difficulty, universality, and feasibility [1]. By leveraging information from multiple biometric sources for recognition, multi-biometric systems generally achieve better matching accuracy and are much harder to spoof. In addition, multi-biometric systems are able to recognize individuals using a subset of biometric traits via feature selection. This enables the systems to cover a wider range of population when some of the users cannot be identified by a certain trait.

Biometric template security is a critical issue because biometrics is unique and irrevocable once it is compromised. This security is especially crucial in multi-biometric systems because they store and process information about multiple biometric traits per user. Once the system storage is compromised, sensitive biometrics information could be revealed if biometric templates are not protected. An adversary can then create physical spoofs of the traits from the revealed templates to masquerade the target user in accessing the compromised system or other systems illegitimately [2–5]. Even worse, if the original biometric images corresponding to multiple traits of a user can all be reverse-engineered from the revealed biometric templates, it would cause permanent compromise of this user's biometrics.

To date, several template protection approaches have been proposed to ensure the security of the biometric templates. They can be categorized into feature transformation (e.g., cancellable biometric [6], RGHE [7], BioHash [8]), biometric cryptosystem (e.g., fuzzy extractor [9], fuzzy vault [10], fuzzy commitment [11]) and hybrid approach [12]. In the feature transformation approach, templates are transformed through a one-way transformation function using a user-specific random key. This approach provides cancellability, where a new transformation (based on a new key) can be used if any template is compromised. A biometric cryptosystem stores a sketch that is generated from the enrollment template, where an error correcting code (ECC) is employed to handle the intra-user variations. The security of the biometric cryptosystem is based on the randomness of the templates and the error correcting capability of the ECC. A hybrid approach combines the advantages of both feature transformation and biometric cryptosystem to provide stronger security and template cancellability.

Biometric cryptosystem takes a query sample and an earlier-generated sketch of the target user and produces a binary decision (accept/reject) in the verification stage. In a multi-biometric cryptosystem, the information of multiple traits could be fused at feature level or score/decision level:

*Corresponding author

Email addresses: csgcmai@comp.hkbu.edu.hk (Guangcan Mai), menghuilim@comp.hkbu.edu.hk (Meng-Hui Lim), pcyuen@comp.hkbu.edu.hk (Pong C. Yuen)

(a) [feature-level] features from different biometric traits are fused and then protected by a single biometric cryptosystem.

(b) [score/decision-level] features from each biometric trait are protected by a biometric cryptosystem and then the individual scores/decisions are fused.

The feature-level-fusion-based multi-biometric cryptosystems are arguably more secure than the score/decision-level-fusion-based systems [13]. In feature-level-fusion-based systems, a sketch generated from the multimodal template is stored, while in score/decision-level-fusion-based systems, multiple sketches corresponding to the unimodal templates are stored. As the adversarial effort for breaking a multimodal sketch is often much greater than the aggregate effort for breaking the unimodal sketches, feature-level-fusion-based systems are more secure. This has also been justified in a recent work [13] using hill-climbing analysis.

Biometric cryptosystems such as fuzzy extractor and fuzzy commitment mainly accept binary input. To produce a binary input for biometric cryptosystems, an integrated binary string needs to be extracted from the multimodal features. However, features of different modalities are usually represented differently, e.g., point-set for fingerprint [14], real-valued for face and binary for iris [15]. To extract the integrated binary string, one can either

(a) convert features of different types into point-set or real-valued features, fuse the converted features, and binarize them;

(b) convert point-set [16–18] and real-valued [12, 19–22] features into binary, then perform a binary feature fusion on these features.

When commercial black-box binary feature extractors such as IrisCode [15] and FingerCode [23] are employed for some biometric traits, the extraction parameters such as quantization and encoding information are not known. Hence, these binary features cannot be converted to other forms of representation appropriately. In this case, the second approach that is based on binary feature fusion is usually adopted.

In this paper, we focus on binary feature fusion for multi-biometric cryptosystems, where biometric features from multiple modalities are converted to a binary representation before being fused. Generally, in a multi-biometric cryptosystem, there are three criteria for its binary input (fused binary feature)

- **Discriminability:** The fused binary features have to be discriminative in order not to defeat the original purpose of recognizing users. The fused feature bits should have small intra-user variations and large inter-user variations.
- **Security:** The entropy of the fused binary features have to be adequately high in order to thwart guessing attacks, even if the stored auxiliary data is revealed. The fused feature bits should be highly uniform and weakly dependent among one another.
- **Privacy:** The stored auxiliary data for feature extraction and fusion should not leak substantial information on the raw biometrics of the target user.

A straightforward method to fuse binary features is to combine the multimodal features using a bitwise operator (e.g., OR, XOR). Concatenating unimodal binary features is another popular option for binary fusion [24, 25]. However, the fusion result of these methods is often suboptimal in terms of discriminability, because the redundant or unstable features cannot be removed. Selecting discriminative binary features is a better approach of obtaining discriminative binary representation. However, similar to bitwise fusion and concatenation, the inherent dependency among bits cannot be improved further. As a result, the entropy of the bit string could be limited, leading to weak security consequence.

To produce a bit string that offers accurate and secure recognition, we propose a binary fusion approach that can simultaneously maximize the discriminability and entropy of the fused binary output. As the properties for achieving both discriminability and security criteria can be divided into multiple-bit-based (i.e., dependency among bits) and individual-bit-based (i.e., intra-user variations, inter-user variations and bit uniformity). the proposed approach consists of two stages: (i) dependency-reductive bit-grouping and (ii) discriminative within-group fusion. In the first stage, we address the multiple-bit-based property: We extract a set of weakly dependent bit-groups from multiple sets of binary unimodal features, such that, if the bits in each group is fused into a single bit, these fused bits, upon concatenation, will be weakly interdependent. Then, in the second stage, we address the individual-bit-based properties: We fuse bits in each bit-group into a single bit with the objective of minimizing the intra-user variation, maximizing the inter-user variation and maximizing uniformity of the bits. As maximizing bit uniformity is equivalent to maximizing the inter-user variation of the corresponding bit, which will be discussed further in Section 3.3, the fusion function is designed to only maximize discriminability (minimize intra-user variations and maximize inter-user variations). The preliminary version of this work has been presented in [26].

The structure of this paper is organized as follows. In the next section, we review several existing binary feature fusion techniques. In Section 3, we describe the proposed two-stage binary feature fusion. We present the experimental results to justify the effectiveness of our fusion approach in Section 4. Finally, we draw a few concluding remarks in Section 5.

2. Related Work

To date, concatenation and bit selection are two typical binary fusion approaches. Sutcu et al. [27] concatenate binary representation of iris and face together to yield the fused binary string. Kanade et al. obtain the fused binary feature by concatenating the iris codes of both left and right iris [24] and concatenating the binary features of both iris and face [25]. Although concatenation of multiple binary features is computationally efficient, this approach treats features from multiple modalities equally and it could limit the discriminability of the fused feature if the multimodal features have different discrimination power.

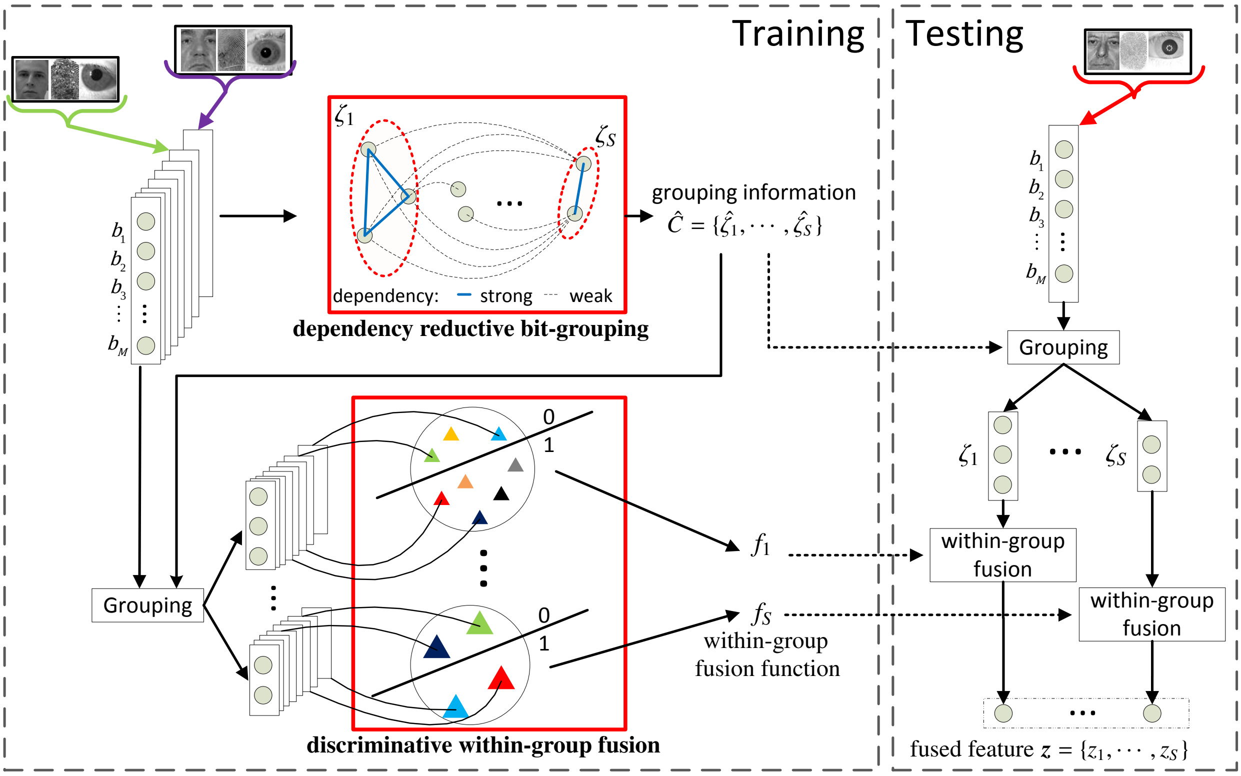

Training
Testing
b_1
b_2
b_3
b_M
ζ_1
ζ_S
dependency: strong weak
dependency reductive bit-grouping
grouping information
$\hat{C} = \{\hat{\zeta}_1, \cdots, \hat{\zeta}_S\}$
Grouping
0
1
f_1
f_S
within-group fusion function
discriminative within-group fusion
Grouping
ζ_1
ζ_S
within-group fusion
within-group fusion
fused feature $z = \{z_1, \cdots, z_S\}$

Figure 1: The proposed binary feature level fusion algorithm

Alternatively, bit selection can be adopted to generate a more discriminative fused binary feature by selecting bits with high discriminability from the multimodal features. Kelkboom et al. [21] select a subset of reliable features based on the estimated z-score of the features, which is the ratio between the distance of the estimated mean with respect to the quantization threshold and the estimated standard deviation. Nagar et al. [28] present a discriminability-based bit selection method to select a subset of bits from each biometric trait individually based on the genuine and impostor bit-error probability and concatenate the selected bits together. Bits with high discriminability are very likely to be mutually dependent because some of the discriminative information may be represented using multiple bits. It is rather difficult for the bit-selection approach to select discriminative bits with high entropy for multi-biometric cryptosystems.

Another possible approach for generating the fused binary features from multiple unimodal binary features is to apply a transformation such as PCA, LDA [29] and CCA [30] on the binary features, followed by a binarization on the transformed feature. However, this approach suffers from an unavoidable trade-off between dependency among feature components and discriminability. For instance, LDA and CCA features are highly discriminative but strongly interdependent; while PCA features are uncorrelated but less discriminative. With this approach, the discriminability and security criteria cannot be fulfilled simultaneously.

3. The proposed binary feature fusion

3.1. Overview of the proposed method

The proposed two-stage binary feature fusion approach generates an S-bit binary representation $\boldsymbol{z} = \{z_1, \cdots, z_s, \cdots, z_S\}$ from an input binary string $\boldsymbol{b} = \{b_1, \cdots, b_m, \cdots, b_M\}$, where typically $S \ll M$. The input binary string $\boldsymbol{b}$ consists of the concatenated multimodal binary features of a sample. The proposed approach can be divided into two stages: (i) dependency reductive bit-grouping and (ii) discriminative within-group fusion, where the block diagram is shown in Fig.1. The details of the two stages in testing phase are described as follows:

(1) **Dependency reductive bit-grouping**: Input bits of $\boldsymbol{b}$ are grouped into a set of weakly-dependent disjoint bit-groups $C = \{\zeta_1, \cdots, \zeta_s, \cdots, \zeta_S\}$ such that $\forall s_1, s_2 \in [1, S], \zeta_{s_1} \cap \zeta_{s_2} = \emptyset, \bigcup_{s=1}^{S} \zeta_s \subseteq \{b_1, \cdots, b_m, \cdots, b_M\}$.

(2) **Discriminative within-group fusion**: Bits in each group ζ_s are fused to a single bit z_s using a group-specific mapping function f_s that maximizes the discriminability of z_s.

The output bit z_s of all groups is concatenated to produce the final bit string $\boldsymbol{z}$. To realize these two stages, optimum grouping information in stage one and optimum within-group fusion functions in stage two need to be sought. In stage one, the grouping information $\hat{C} = \{\hat{\zeta}_1, \cdots, \hat{\zeta}_s, \cdots, \hat{\zeta}_S\}$ represents the S groups of bit indices, specifying which of the bits in $\boldsymbol{b}$ should be grouped together. Note that we use $'\hat{x}'$ to denote the index of the variable x throughout this paper unless stated otherwise. In stage two, the mapping function f_s specifies to which output bit value the bits in group ζ_s are mapped.

3.2. Dependency reductive bit-group search

To reduce the dependency among bits in the output binary string, a set of weakly-dependent bit-groups C need to be extracted from the input $\boldsymbol{b}$. One promising way to extract these weakly-dependent bit-groups is to adopt a proper clustering technique based on a dependency measure.

Existing clustering techniques can be categorized into partitional clustering (e.g., k-means) and hierarchical clustering [31]. The partitional clustering directly creates partitions of data and represents each partition using a representative (e.g., clustering center). However, the bit positions among which dependence needs to be measured cannot be effectively represented in a metric space because dependence does not satisfy the traingle inequality requirement of a metric space. As a result, partitional clustering is less feasible in our context. The hierarchical clustering, on the other hand, serves as a better option as it can operate efficiently based on a set of pairwise dependencies. In this proposed method, we adopt the agglomerative hierarchical clustering (AHC). The basic idea of AHC is as follows: we first create multiple singleton clusters where each cluster contains a single bit, and then we start to merge a cluster pair with the highest pairwise dependency iteratively, until the termination criterion is met.

To measure dependencies between two bits or two groups of bits, mutual information (MI) can be adopted [32, 33]. The MI of clusters ζ_{s_1} and ζ_{s_2} can be expressed as

$$I(\zeta_{s_1}, \zeta_{s_2}) = H(\zeta_{s_1}) + H(\zeta_{s_2}) - H(\zeta_{s_1}, \zeta_{s_2}) \tag{1}$$

where $H(\zeta_{s_1})$ and $H(\zeta_{s_2})$ denote the joint entropy of bits in an individual cluster ζ_{s_1} or ζ_{s_2}, respectively, and $H(\zeta_{s_1}, \zeta_{s_2})$ denotes the joint entropy of bits enclosed by both clusters. However, the above MI measurement is sensitive to the number of variables (bit positions) and is proportionate to the aggregate information of these variables. As a result, multiple MI measurements involving different number of bit positions cannot be fairly compared during the selection of cluster pair for cluster merging. That is, if MI is adopted for dependency measurement, the hierarchical clustering technique will always be inclined to select a cluster pair that involves the largest cluster for merging in every iteration, although this cluster pair may not be the pair with the highest average bit interdependency.

To obtain a better measure that precisely quantifies the bit interdependency irrespective of the size of the clusters, we normalize the MI using the size of clusters in the cluster pair. This normalized measure indicates how dependent on average a bit pair in a group is upon merging. We call this normalized measure as the average mutual information (AMI), such that

$$I_{avg}(\zeta_{s_1}, \zeta_{s_2}) = \frac{I(\zeta_{s_1}, \zeta_{s_2})}{|\zeta_{s_1}| \times |\zeta_{s_2}|} \tag{2}$$

With this AMI measure, we are able to identify cluster-pair with the strongest average bit-pair dependency for merging over cluster pairs of different sizes in each iteration. Our proposed AMI-based AHC algorithm is shown in Algorithm.1. As strongly-dependent cluster pairs will gradually be merged by the clustering algorithm, we will eventually be able to obtain a set of (remaining) weakly-dependent bit groups that were not selected for merging throughout the algorithm.

Algorithm 1 AMI-based agglomerative hierarchical clustering

```
Inputs:
    $N$ samples of all users' binary features
        $B = \{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n, \cdots, \boldsymbol{b}_N\}$,
    length of each binary feature $M$,
    number of clusters $S$,
    maximum cluster size $t_{size}$
Outputs:
    grouping information $\hat{C} = \{\hat{\zeta}_1, \cdots, \hat{\zeta}_s, \cdots, \hat{\zeta}_S\}$
Initialize:
    $\hat{C}_{tmp} = \{\hat{\zeta}_1, \cdots, \hat{\zeta}_m, \cdots, \hat{\zeta}_M\}$ where $\hat{\zeta}_m = \{m\}$
    compute entropy of each cluster $H(\zeta)$ in $\hat{C}_{tmp}$
    $h_{tmp} \leftarrow S$-th largest cluster entropy in $\hat{C}_{tmp}$
    $\hat{C} \leftarrow \hat{C}_{tmp}$
    $h \leftarrow 0$
    $D = \{d_{\alpha\beta}\}^{\alpha\neq\beta}_{\alpha,\beta\in[1,M]}$, where $d_{\alpha\beta} = I_{avg}(\zeta_\alpha, \zeta_\beta)$
while $|\hat{C}_{tmp}| > S$ do
    search for largest $d_{\alpha\beta}$
    if $|\hat{\zeta}_\alpha| + |\hat{\zeta}_\beta| > t_{size}$ then
        $d_{\alpha\beta} \leftarrow -1$
    else
        $\hat{\zeta}_\lambda \leftarrow \zeta_\alpha \cup \hat{\zeta}_\beta$
        $\hat{C}_{tmp} \leftarrow \hat{C}_{tmp} - \{\hat{\zeta}_\alpha\} - \{\hat{\zeta}_\beta\} + \{\hat{\zeta}_\lambda\}$
        compute entropy of each cluster $H(\zeta)$ in $\hat{C}_{tmp}$
        $h_{tmp} \leftarrow S$-th largest cluster entropy in $\hat{C}_{tmp}$
        if $h_{tmp} > \min(h, 1)$ then
            $\hat{C} \leftarrow \hat{C}_{tmp}$
            $h \leftarrow h_{tmp}$
        end if
        for each $\hat{\zeta}_\mu \in \hat{C}_{tmp}, \mu \neq \lambda$ do
            update $d_{\lambda\mu}$
        end for
    end if
end while
Discard the $(|\hat{C}| - S)$ lowest-entropy cluster in $\hat{C}$
```

{$H(\zeta)$ returns the entropy of cluster ζ, which is based on the observation of bit combination $\zeta^n = \{b_{nm}\}_{m\in\hat{\zeta}}$ that corresponds to cluster ζ and training sample $\boldsymbol{b}_n$. }

After the algorithm terminates, the grouping information $\hat{C}$ is obtained. It is noted that the size of each resulted group ζ specified in $\hat{C}$ determines the number of possible bit combinations (i.e., $2^{|\zeta|}$ bit-combinations for groups size $|\zeta|$). As we need to estimate the occurrence probabilities of these bit combinations from the training samples for within-group fusion search in the second stage described in 3.3, it is usual that one may not have arbitrarily large amount of training data in practice to ensure accurate estimation of these probabilities. To overcome this problem, we restrict the maximum group size to be t_{size} in order to ensure the feasibility of optimal within-group fusion search in the second stage.

The final set of S clusters is taken based on the entropy of the clusters. In the ideal scenario, every resulted bit group ζ specified in $\hat{C}$ should contain at least one bit entropy. According to our analysis in Section 3.4, optimal inter-user variation of the output bit of a group (during within-group fusion function search in the second stage) can only be achieved when the entropy of the corresponding group is not less than one bit. While this ideal scenario cannot be guaranteed all the time especially when the input bit string contains limited entropy, the entropy of the S clusters should be made as high as possible so that the possibility of obtaining high inter-user variation in the resulted fused bit from each cluster in the second stage can be heightened. Because the dependency (maximum AMI) of all cluster pairs is non-increasing as the iteration proceeds (see Appendix.Appendix A for the proof), the output grouping information $\hat{C}$ will be taken and updated whenever one of the following conditions is satisfied:

(a) The S-th largest cluster entropy in $\hat{C}_{tmp}$ is greater or equal to one bit;

(b) The S-th largest entropy of the clusters in $\hat{C}$ is less than one bit and less than that in $\hat{C}_{tmp}$.

3.3. Discriminative within-group fusion search

Suppose that we have obtained S groups of bits from the first stage. For each group, we seek for a discriminative fusion $f : \{0, 1\}^{|\zeta|} \rightarrow \{0, 1\}$ to fuse bits in group ζ to a single bit z. Here, the function f maps each combination of $|\zeta|$ bits to a bit value. The within-group fusion is analogous to a binary-label assignment process, where each bit combination is assigned a binary output label (a fused bit value). Since the dependency among fused bits has been reduced using AMI-based AHC in stage one, to obtain a discriminative bit string that contains high entropy, the fusion should minimize the intra-user variation, maximize the inter-user variation and uniformity of the output bit. Naturally, maximizing inter-user variations has an equivalent effect of maximizing bit uniformity. This is because a bit with maximum inter-user variation also indicates that the bit value would distribute uniformly among the population users. Thus, the fusion sought in the following need only to optimize the discriminability of the output bit, i.e., minimizing the intra-user variations and maximizing the inter-user variations.

The intra-user and inter-user variations of the fused bit z of group ζ could be measured using the genuine bit-error probability p_g^e and the impostor bit-error probability p_i^e, respectively. Genuine bit-error probability is defined as the probability where different samples of the same user are fused to different bit values, while the impostor bit-error probability is defined as the probability where samples of different users are fused to different bit values. Let x_t denotes the t-th bit-combination of group ζ, where $t = \{1, 2, \cdots, 2^{|\zeta|}\}$ and let $X^{(0)}$ and $X^{(1)}$ denote the sets of bit-combinations in group ζ that to be fused to '0' and '1', respectively. The genuine bit-error probability of fused bit z corresponding to group ζ can be expressed as

$$\begin{aligned} p_g^e &= \Pr(\zeta^{n_1} \in X^{(0)}, \zeta^{n_2} \in X^{(1)} | l_{n_1} = l_{n_2}) \\ &= \sum_{x_{t_1} \in X^{(0)}} \sum_{x_{t_2} \in X^{(1)}} \Pr(\zeta^{n_1} = x_{t_1}, \zeta^{n_2} = x_{t_2} | l_{n_1} = l_{n_2}) \end{aligned} \quad (3)$$

where l_{n_1} and l_{n_2} denote the label of n_1-th and n_2-th training sample, respectively, ζ^{n_1} and ζ^{n_2} denote the group ζ corresponding to the n_1-th and n_2-th training samples, $n_1 \neq n_2$ and $n_1, n_2 \in \{1, 2, \cdots, N\}$.

Similarly, the impostor bit-error probability can be expressed as

$$\begin{aligned} p_i^e &= \Pr(\zeta^{n_1} \in X^{(0)}, \zeta^{n_2} \in X^{(1)} | l_{n_1} \neq l_{n_2}) \\ &= \sum_{x_{t_1} \in X^{(0)}} \sum_{x_{t_2} \in X^{(1)}} \Pr(\zeta^{n_1} = x_{t_1}, \zeta^{n_2} = x_{t_2} | l_{n_1} \neq l_{n_2}) \end{aligned} \quad (4)$$

To seek the function f that minimizes genuine and maximizes impostor bit-error probability, we solve the following minimization problem using the integer genetic algorithm [34, 35],

$$\begin{aligned} &\min_f \left(p_g^e - p_i^e\right) \\ &= \sum_{x_{t_1} \in X^{(0)}} \sum_{x_{t_2} \in X^{(1)}} (\Pr(\zeta^{n_1} = x_{t_1}, \zeta^{n_2} = x_{t_2} | l_{n_1} = l_{n_2}) \\ &\qquad - \Pr(\zeta^{n_1} = x_{t_1}, \zeta^{n_2} = x_{t_2} | l_{n_1} \neq l_{n_2})) \end{aligned} \quad (5)$$

subject to

$$f(x_{t_1}) = 0, f(x_{t_2}) = 1$$

where $f(x_{t_1})$ and $f(x_{t_1})$ denote the fused bit value of bit-combination x_{t_1} and x_{t_2}, respectively. Note that this function f has to be sought for every bit group.

3.4. Discussion and analysis

An important requirement in Algorithm 1 is that each resulted bit group (joint entropy of bits in the group) should contain at least one-bit entropy to warrant the achievability of high inter-user variation. This is because when the group entropy is less than one bit, the probability of one of the fused bit values would become larger than 0.5, thus making the distribution of bit values less uniform among the population users. In the following, we analyze how group entropy that is less than one bit could negatively influence the impostor error probability of the fused bit.

Let p_t denotes the occurrence probability of a bit combination x_t in group ζ, where $t = \{1, 2, \cdots, 2^{|\zeta|}\}$. The corresponding joint entropy of bits in group ζ is expressed as

$$H(x) = -\sum_{t=1}^{2^{|\zeta|}} p_t \log_2 p_t \quad (6)$$

where $|\zeta|$ denotes group size and $\sum_{t=1}^{2^{|\zeta|}} p_t = 1$. If $H(x) < 1$,

(a) there exists a bit combination that has the highest occurrence probability $p_{max} = \max_t(p_t) > 0.5$; and

(b) the impostor bit-error probability p_i^e (the larger, the better) of the fused bit in stage two is upper bounded by

$$p_i^e \leq 2p_{max}(1 - p_{max}) < 0.5 \quad (7)$$

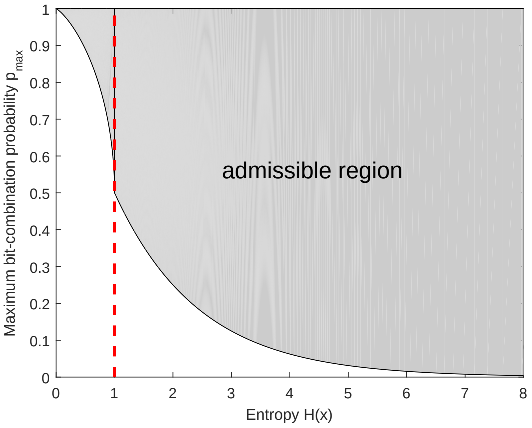


Figure 2: The lower bound of entropy $H_L(x)$, where the grey-shaded area depicts the admissible region of p_{max} given $H(x)$.

Proof. (a) To prove that there is an input bit combination that has the highest probability $p_{max} = \max_t(p_t) > 0.5$ when $H(x) < 1$, we construct a lower bound of entropy $H_L(x)$ w.r.t. p_{max} that is described as follows:

$$\begin{aligned} H_L(x) &= \max\left(H_{L1}(x), H_{L2}(x)\right) \\ &= \begin{cases} H_{L1}(x) = -\log_2 p_{max}, & 0 < p_{max} \leq 0.5 \\ H_{L2}(x) = H_b(p_{max}), & 0.5 \leq p_{max} \leq 1 \end{cases} \end{aligned} \tag{8}$$

where $H_{L1}(x)$ and $H_{L2}(x)$ are two lower bound functions and $H_b(p_{max})$ is the binary entropy function

$$H_b(p_{max}) = -p_{max}\log_2(p_{max}) - (1 - p_{max})\log_2(1 - p_{max})$$

The two lower bound functions $H_{L1}(x)$ and $H_{L2}(x)$ are derived as follows:

$$\begin{aligned} H(x) &= -\sum_{t=1}^{2^{|\zeta|}} p_t \log_2 p_t \\ &\geq -\sum_{t=1}^{2^{|\zeta|}} p_t \log_2 p_{max} = -\log_2 p_{max} = H_{L1}(x) \end{aligned} \tag{9}$$

$$\begin{aligned} H(x) &= -\sum_{t=1}^{2^{|\zeta|}} p_t \log_2 p_t \\ &\geq -\sum_{z=0}^{1}\left(\left(\sum_{t,f(x_t)=z} p_t\right)\log_2 \sum_{t,f(x_t)=z} p_t\right) \\ &\geq H_b(p_{max}) = H_{L2}(x) \end{aligned} \tag{10}$$

The inverse function of Eq.(8) is plotted as the solid curve in Fig.2, where the admissible region of p_{max} lies within the grey-shaded area, indicating the possible p_{max} values given an entropy value $H(x)$ of a bit group. Based on this plot, it can be observed that when group entropy $H(x) < 1$, all of the possible p_{max} values in the dark-grey-shaded area are greater than 0.5, which completes the proof. □

Proof. (b) The impostor bit-error probability p_i^e is the probability of getting a different fused bit value from that of the target genuine user. Hence, we obtain the following:

$$\begin{aligned} p_i^e &= \Pr(z = 0)\Pr(z = 1) + \Pr(z = 1)\Pr(z = 0) \\ &= 2\Pr(z = 0)\Pr(z = 1) \\ &\leq 2p_{max}(1 - p_{max}) \\ &< 0.5 \end{aligned} \tag{11}$$

□

With this, the lower $H(x) < 1$ is, the larger the p_{max}, and the smaller the impostor bit-error probability p_i^e will be. This completes the proof.

4. Experimental Results

4.1. Database and experiment setting

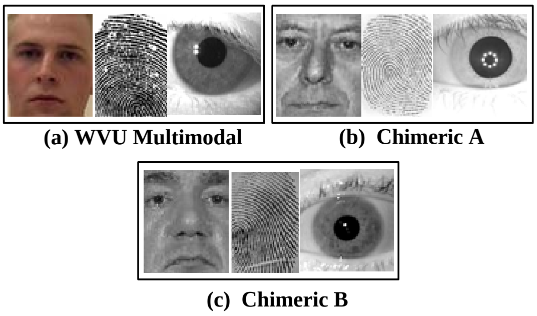
We evaluated the proposed fusion algorithm using a real and two chimeric multi-modal databases, involving three modalities: face, fingerprint and iris. The real multi-modal database, WVU [36], contains images of 106 subjects, where each subject has five multi-modal samples. The two chimeric multi-modal databases are obtained by randomly matching images from a face, a fingerprint and an iris database. The first chimeric multi-modal database named Chimeric A consists of faces from FERET [37], fingerprints from FVC2000-DB2 and irises from CASIA-Iris-Thousand [38]. The second database named Chimeric B consists of faces from FRGC [39], fingerprints from FVC2002-DB2 and irises from ICE2006 [40]. These chimeric databases contain 100 subjects with eight multi-modal samples per subject. Fig.3 shows the sample images from the three databases.

Table 1: Experimental settings

	WVU	Chimeric A	Chimeric B
Subjects	106	100	100
Samples per subject	5	8	8
Training Sample	3	4	4
Testing Sample	2	4	4
Genuine Attempts	106	300	300
impostor attempts	111,30	19,800	19,800

The training-testing partitions for each database is shown in Table.1. Our testing protocol is described as follows. For the genuine attempts, the first sample of each subject is matched against the remaining samples of the subject. For the impostor attempts, the i-th sample of each subject is matched against the i-th sample of the remaining subjects. Consequently, the number of genuine and impostor attempts in WVU multi-modal database are 106 ($106 \times (2-1)$) and 11,130 ($(106 \times 105)/2 \times 2$), respectively, while the number of genuine and impostor attempts in the two chimeric multi-modal databases are 300 ($100 \times (4-1)$) and 19,800 ($(100 \times 99)/2 \times 4$) respectively.

Prior to evaluating the binary fusion algorithms, we extract the binary features of face, fingerprint and iris from the



(a) WVU Multimodal

(b) Chimeric A

(c) Chimeric B

Figure 3: Sample face, fingerprint, and iris images from (a) WVU; (b) Chimeric A (FERET, FVC2000-DB2, CASIA-Iris-Thousand); and (c) Chimeric B (FRGC, FVC2002-DB2, ICE2006)

databases. The images of each modality are first processed as follows:

- **Face:** Proper face alignment is first applied based on the standard face landmark. To eliminate effect from variations such as hair style and background, the face region of each sample is cropped and resized to 61×73 pixels in FERET and FRGC databases, and 15×20 pixels in WVU database.
- **Fingerprint:** We first extract minutiae from each fingerprint using Verifinger SDK 4.2 [41]. The extracted minutiae are converted into an ordered binary feature using the method proposed in [16] without randomization. Following parameters in [16], each fingerprint image is represented by a vector with length 2^{24}.
- **Iris:** The weighted adaptive hough and ellipsopolar transform (WAHET) [42] is employed to segment the iris. Then, 480 real features are extracted from the segmented iris using Ko et al.'s extractor [43]. Both segmentation and extraction algorithms are implemented using the iris toolkit (USIT) [44].

After preprocessing, we apply PCA on face, and LDA on fingerprint and iris to reduce the feature dimensions to 50. Then, we encode each feature component with a 20-bit binary vector using LSSC [22] and obtain a 1000-bit binary feature for each modality.

In this comparative study, we compare the proposed method with the following existing methods:

- **single modality baselines:** face, fingerprint, iris
- **bit selection [28]**
- **concatenation [24, 25]**
- **bit-wise operation:** AND, OR, XOR
- **decision fusion:** AND, OR (denoted as 'and_d' and 'or_d' in the experimental results, respectively)

For the proposed method, the parameter of largest cluster size t_{size} in stage one is set to 8. Throughout the comparative study, features produced by the evaluated methods are made to be of the same length for comparison fairness purpose, except the concatenation method. For instance, the original length of the unimodal binary features is reduced to the evaluated length through discriminative selection using a discriminability criterion [28]. The features of the bit-wise operation and the results of decision-level fusion methods are obtained from these selected uni-biometric features.

4.2. Evaluation measures for discriminability and security

Discriminability. The discriminability of the fused feature is measured using the area under curve (AUC) of the receiver operating characteristic (ROC) curve. The higher the AUC, the better the matching accuracy would be.

Security. The security of the template is evaluated using quadratic Renyi entropy [45]. Specifically, the quadratic Renyi entropy measures the effort for searching an identified sample of the target template. Assuming that the average impostor Hamming distance (aIHD) or the impostor Hamming distance per bit obeys binomial distribution with expectation p and standard deviation σ, the entropy of the template can be estimated as

$$\begin{aligned} H &= -\log_2 \Pr(\text{aIHD} = 0) \\ &= -\log_2 p^0 (1-p)^{N_*} = -N_* \log_2(1-p) \end{aligned} \tag{12}$$

where p and σ denote the mean and standard deviation of the aIHD, resp., and $N_* = p(1-p)/\sigma^2$ denotes the estimated number of independent Bernoulli trials.

Trade-off analysis. The GAR-Security (G-S) analysis [28] is an integrated measure for template discriminability and security in biometric cryptosystems. It analyzes the trade-off between matching accuracy and security in a fuzzy commitment system by varying the error correcting capability. The G-S analysis is based on the decoding complexity of Nagar's ECC decoding algorithm [28], where a query is accepted only if the corresponding decoding complexity is less than a given threshold.

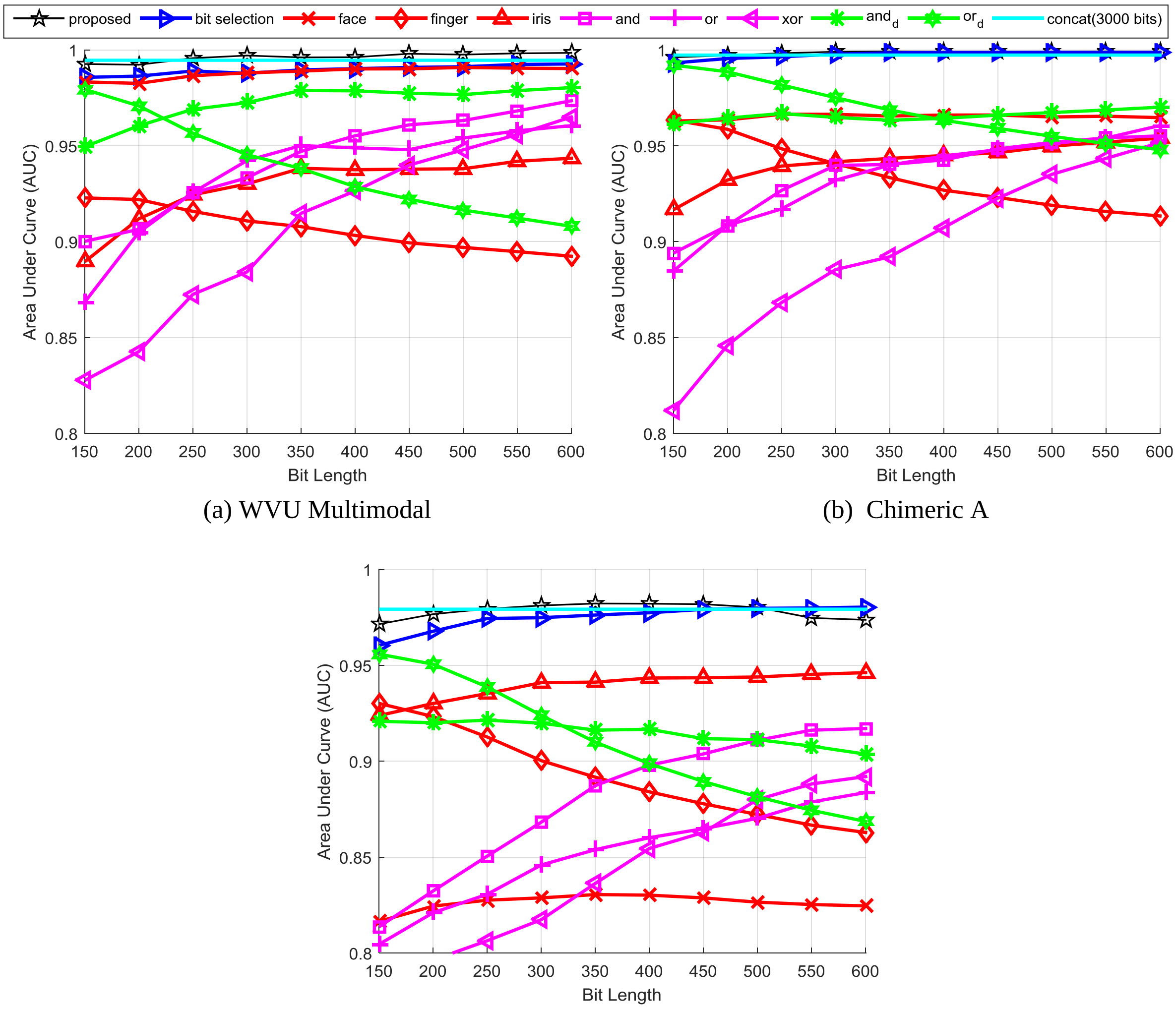
A G-S point is produced via computing the GAR and the minimum decoding complexity among all impostor attempts given an error correcting capability. More details of the decoding complexity can be found in [28]. We estimate the entropy of the binary feature using the quadratic Renyi entropy [45], which is a more accurate measure than the Daugman's DOF [46] that is only reliable as the aIHD expectation $p = 0.5$.

4.3. Discriminability evaluation

The AUC for fusion bit length from 150 to 600 is shown in Fig.4. It can be observed that the proposed method has comparable performance compared to bit selection and concatenation on all three databases and it outperforms the remaining methods in general. On WVU multi-modal database, the proposed method performs as good as the unimodal face baseline.

For the results on WVU multi-modal database in Fig.4(a), the proposed method outperforms the curves of bit selection, concatenation and face. When the bit length equals 350, the AUC of the proposed method is 0.9961, which is slightly higher than the AUC of bit selection (0.9896), concatenation (0.9946) and the best single modality: face (0.9890). Compared to face, the proposed method has a marginal improvement of 0.71%.

For the results on Chimeric A database shown in Fig.4(b), the proposed method performs equally well with bit selection and concatenation methods. The AUC of the proposed method, the



(a) WVU Multimodal

(b) Chimeric A

(c) Chimeric B

Figure 4: Comparison of area under ROC curve on (a) WVU multi-modal, (b) Chimeric A, (c) Chimeric B databases.

bit selection and the concatenation methods are 0.9992, 0.9985, and 0.9973 at 350-bit feature length, respectively. This shows a 3.4% improvement of the proposed method compared to the best-performing unimodality: face (AUC = 0.9656).

For the results of Chimeric B database in Fig.4(c), it can be observed that the AUC of the proposed method is slightly higher than the bit selection method when the bit length is less than 500. For this database, the proposed method, bit selection and concatenation methods outperform significantly the best-performing unimodality: iris. At 350-bit feature length, the AUC of the proposed method is 0.9823 compared to the concatenation (0.9793) and bit selection (0.9763) methods. The AUC improvement of the proposed method is approximately 3.5% compared to iris (AUC = 0.9413) at 350-bit feature length.

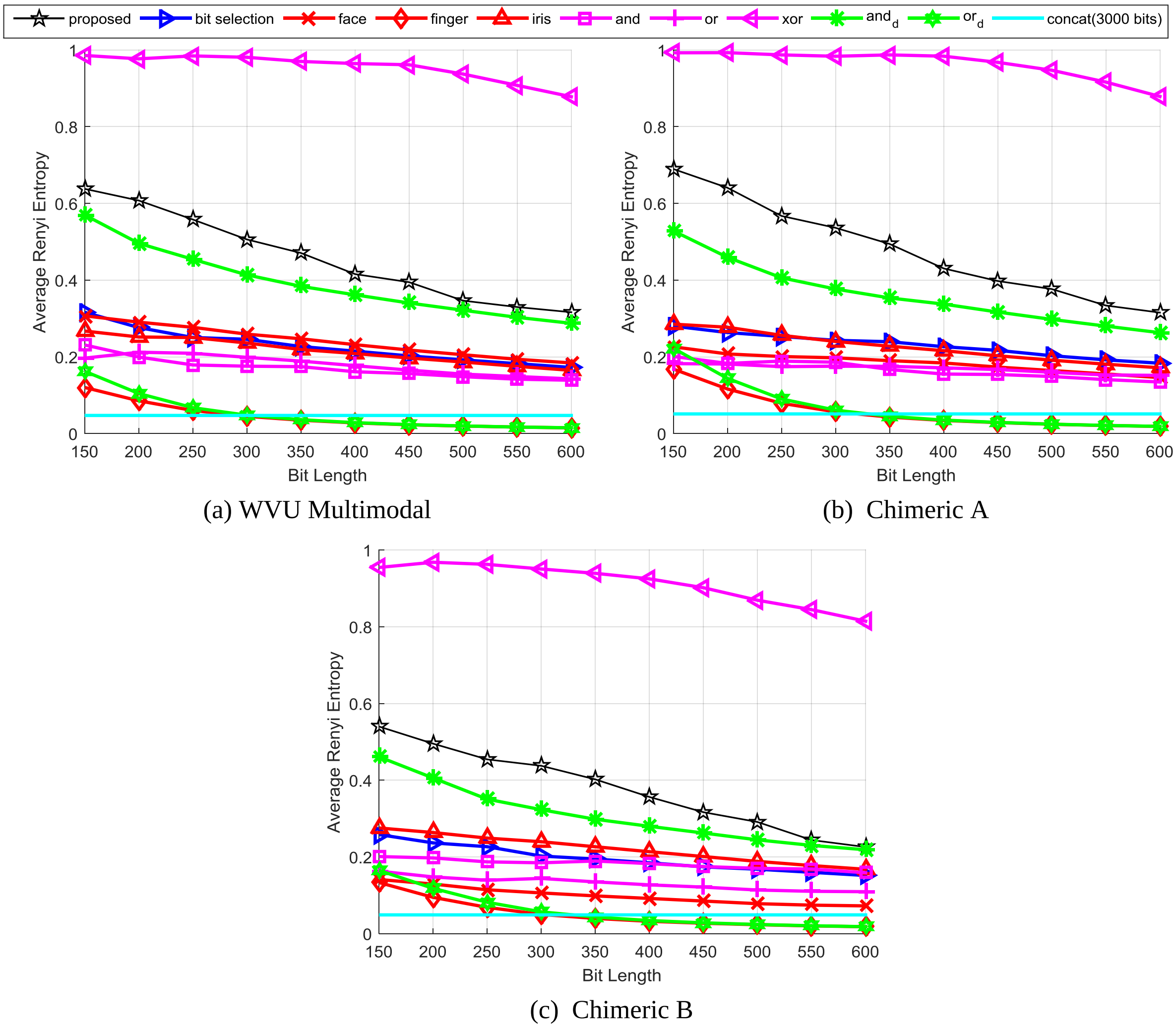
These results show that the proposed method could perform equally well, or even slightly better than bit selection and concatenation although the biometric modalities could vary significantly in quality. It is noted that the difference between the AUC of face and fingerprint is around 7 ~ 10% on WVU multimodal database and 2 ~ 5% on Chimeric A database; while the difference between the AUC of iris and face is around 10% on Chimeric B.

Additionally, it is observed that there is no guarantee on the performance of features produced based on AND-, OR- and XOR-feature fusion rule. The features produced by XOR rule are always the worst compared to AND and OR rules.

4.4. Security evaluation

In this section, the results on template security are shown, which is measured using quadratic Renyi entropy [45]. The average Renyi entropy of the binary feature fused using the evaluated schemes are plotted in Fig.5. Here, the average Renyi entropy is the Renyi entropy divided by the bit length of the fused features, thus ranging from 0 to 1. A higher average Renyi entropy implies stronger template security.

On all three databases, it can be observed that the proposed method ranks second in terms of entropy. The best-performing method turns out to be the XOR feature fusion because the features tends to be more uniform upon XOR fusion, despite its



(a) WVU Multimodal

(b) Chimeric A

(c) Chimeric B

Figure 5: Comparison of average Renyi entropy on (a) WVU multi-modal, (b) Chimeric A, (c) Chimeric B databases.

poor performance in the discriminability evaluation.

For the WVU multi-modal database shown in Fig.5(a), it is observed that at 350-bit feature length, the average entropy achieved by the proposed method is 0.4674 bit, while the XOR-feature fusion method achieves an average entropy of 0.9603 bit, which is nearly double of the proposed method. Besides that the 'and$_d$' method slightly underperforms the proposed method, the remaining methods could only achieve at most half of the average entropy of the proposed method.

Similar results can be seen on Chimeric A and B databases in Fig.5(b) and (c). When the bit length equals 350, the proposed method achieves an average entropy of 0.4896 bit in Fig.5(b) and 0.4021 bit in Fig.5(c), that is half of that of the XOR-feature fusion method but is at least double of that of the remaining methods.

4.5. *Trade-off analysis between discriminability and security*

Using the parameters suggested in [28], the G-S curves of the evaluated methods are plotted in Fig.6. The maximum acceptable decoding complexity is fixed as 15 bits and the minimum distance of the ECC ranges from 0.02 to 0.6 times the bit length S. It can be observed that the proposed method outperforms the bit selection method on all three databases. This implies that the proposed method achieves a better discriminability-security tradeoff than the bit selection method and the remaining methods.

For 40-bit security at 350-bit feature length, the proposed method performs the best, achieving 69% GAR. This is followed by the face (57% GAR) and bit selection method (38% GAR). For the same settings on Chimeric A database, the proposed method achieves 64% GAR, which is 13% higher than face modality and 26% higher than bit selection method. As for Chimeric B database, the proposed method achieves 20% GAR, which is 11% higher than the iris modality and 17% higher than bit selection method.

5. Conclusion

In this paper, we have proposed a binary feature fusion algorithm that can produce discriminative binary templates with high entropy for multi-biometric cryptosystems. The proposed binary feature fusion algorithm consists of two stages: dependency reductive bit grouping and discriminative and uniform within-group fusion. The first stage creates multiple weakly-interdependent bit groups using grouping information that is obtained from an average mutual information-based agglomerative hierarchical clustering; while the second stage fuses the bits in each group through a function that minimizes intra-user variation, and maximizes uniformity and inter-user variation of the output fused bit. We have conducted experiments on WVU multi-modal database and two chimeric databases and the results have justified the effectiveness of the proposed method in producing a highly discriminative fused template with high entropy per multimodal sample.

Appendix A. Proof of the Non-increasing of the Maximum AMI

Appendix A.1. *In the agglomerative clustering that merge cluster pairs with maximum AMI at each iteration, let* $\mathbf{MI}_{avg}^{iter}$ *and* $\mathbf{MI}_{avg}^{iter+1}$ *denotes maximum AMI among all cluster-pairs in the start of iter-th and* $(iter+1)$*-th iteration, resp., then* $\mathbf{MI}_{avg}^{iter} \geq \mathbf{MI}_{avg}^{iter+1}$.

Proof. (proof by contradiction) Suppose that the cluster-set $C_{iter} = \{\zeta_1, \zeta_2, \cdots, \zeta_S\}$ in the start of *iter*-th iteration contains L clusters, and the cluster-pair $(\zeta_{s_1}, \zeta_{s_2})$, where $s_1, s_1 = \{1, 2, \cdots, S\}$, is the cluster-pair with highest AMI among all possible cluster-pairs from C_{iter}, i.e., $\mathbf{MI}_{avg}^{iter} = I_{avg}(\zeta_{s_1}, \zeta_{s_2})$. In the start of $(iter+1)$-th (after *iter*-th) iteration, cluster-pair $(\zeta_{s_1}, \zeta_{s_2})$ is merged to cluster ζ_{s_3}, the corresponding cluster-set C_{iter+1} contains ζ_{s_3} and all the clusters in C_{iter} excluding ζ_{s_1} and ζ_{s_2}, i.e.,

$$C_{iter+1} = C_{iter} - \{\zeta_{s_1}\} - \{\zeta_{s_2}\} + \{\zeta_{s_3}\}$$

As $\mathbf{MI}_{avg}^{iter} = I_{avg}(\zeta_{s_1}, \zeta_{s_2})$, $I_{avg}(\zeta_{s_1}, \zeta_{s_2})$ greater than the AMI of all possible cluster-pair in C_{iter+1} excluding cluster ζ_{s_3}. Therefore, if $\mathbf{MI}_{avg}^{iter} < \mathbf{MI}_{avg}^{iter+1}$, there must exist a ζ_{s_4} in C_{iter+1}, such that $I_{avg}(\zeta_{s_1}, \zeta_{s_2}) < I_{avg}(\zeta_{s_3}, \zeta_{s_4})$. Since

$$\begin{aligned}I_{avg}(\zeta_{s_3}, \zeta_{s_4}) &= \frac{H(\zeta_{s_3}) + H(\zeta_{s_4}) - H(\zeta_{s_3}, \zeta_{s_4})}{|\zeta_{s_3}||\zeta_{s_4}|} \\ &= \frac{H(\zeta_{s_3}) + H(\zeta_{s_4}) - H(\zeta_{s_3}, \zeta_{s_4})}{(|\zeta_{s_1}| + |\zeta_{s_2}|)|\zeta_{s_4}|}\end{aligned}$$

Furthermore, we have

$$\begin{aligned}&H(\zeta_{s_3}) + H(\zeta_{s_4}) - H(\zeta_{s_3}, \zeta_{s_4}) \\ =&I_{avg}(\zeta_{s_1}, \zeta_{s_4})|\zeta_{s_1}||\zeta_{s_4}| + I_{avg}(\zeta_{s_2}, \zeta_{s_4})|\zeta_{s_2}||\zeta_{s_4}| \\ &+ H(\zeta_{s_1}, \zeta_{s_4}) + H(\zeta_{s_2}, \zeta_{s_4}) \\ &- \Big(I_{avg}(\zeta_{s_1}, \zeta_{s_2}) + H(\zeta_{s_4}) + H(\zeta_{s_1}, \zeta_{s_2}, \zeta_{s_4})\Big) \\ \leq&I_{avg}(\zeta_{s_1}, \zeta_{s_4})|\zeta_{s_1}||\zeta_{s_4}| + I_{avg}(\zeta_{s_2}, \zeta_{s_4})|\zeta_{s_2}||\zeta_{s_4}| \\ \leq&\max\{I_{avg}(\zeta_{s_1}, \zeta_{s_4}), I_{avg}(\zeta_{s_2}, \zeta_{s_4})\}(|\zeta_{s_1}| + |\zeta_{s_2}|)|\zeta_{s_4}|\end{aligned}$$

Finally,

$$\begin{aligned}&I_{avg}(\zeta_{s_3}, \zeta_{s_4}) \\ &\leq \frac{\max\{I_{avg}(\zeta_{s_1}, \zeta_{s_4}), I_{avg}(\zeta_{s_2}, \zeta_{s_4})\}(|\zeta_{s_1}| + |\zeta_{s_2}|)|\zeta_{s_4}|}{(|\zeta_{s_1}| + |\zeta_{s_2}|)|\zeta_{s_4}|} \\ &\leq \max\{I_{avg}(\zeta_{s_1}, \zeta_{s_4}), I_{avg}(\zeta_{s_2}, \zeta_{s_4})\}(|\zeta_{s_1}| + |\zeta_{s_2}|)|\zeta_{s_4}| \\ &\leq I_{avg}(\zeta_{s_1}, \zeta_{s_2})\end{aligned}$$

Therefore, there is no cluster ζ_{s_4} that fulfill the condition $I_{avg}(\zeta_{s_1}, \zeta_{s_2}) < I_{avg}(\zeta_{s_3}, \zeta_{s_4})$, which means that $\mathbf{MI}_{avg}^{iter} \geq \mathbf{MI}_{avg}^{iter+1}$ always true. This completes the proof. □

References

[1] A. A. Ross, K. Nandakumar, A. K. Jain, Handbook of multibiometrics, Vol. 6, Springer, 2006.

[2] R. Cappelli, D. Maio, A. Lumini, D. Maltoni, Fingerprint image reconstruction from standard templates, Pattern Analysis and Machine Intelligence, IEEE Transactions on 29 (9) (2007) 1489–1503.

[3] Y. C. Feng, M.-H. Lim, P. C. Yuen, Masquerade attack on transform-based binary-template protection based on perceptron learning, Pattern Recognition 47 (9) (2014) 3019–3033.

[4] R. N. Rodrigues, N. Kamat, V. Govindaraju, Evaluation of biometric spoofing in a multimodal system, in: Biometrics: Theory Applications and Systems, Fourth IEEE International Conference on, 2010, pp. 1–5.

[5] A. Ross, J. Shah, A. K. Jain, From template to image: Reconstructing fingerprints from minutiae points, Pattern Analysis and Machine Intelligence, IEEE Transactions on 29 (4) (2007) 544–560.

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

[7] Z. Jin, M.-H. Lim, A. B. J. Teoh, B.-M. Goi, A non-invertible randomized graph-based hamming embedding for generating cancelable fingerprint template, Pattern Recognition Letters 42 (2014) 137–147.

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

[9] Y. Dodis, R. Ostrovsky, L. Reyzin, A. Smith, Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, SIAM journal on computing 38 (1) (2008) 97–139.

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

[11] A. Juels, M. Wattenberg, A fuzzy commitment scheme, in: Proceedings of the 6th ACM conference on Computer and communications security, 1999, pp. 28–36.

[12] Y. C. Feng, P. C. Yuen, A. K. Jain, A hybrid approach for generating secure and discriminating face template, Information Forensics and Security, IEEE Transactions on 5 (1) (2010) 103–117.

[13] E. Maiorana, G. Hine, P. Campisi, Hill-climbing attacks on multibiometrics recognition systems, Information Forensics and Security, IEEE Transactions on 10 (5) (2015) 900–915.

[14] ISO/IEC 19794-2:2011 Information technology – Biometric data interchange formats – Part 2: Finger minutiae data, 2011.

[15] J. Daugman, High confidence visual recognition of persons by a test of statistical independence, Pattern Analysis and Machine Intelligence, IEEE Transactions on 15 (11) (1993) 1148–1161.

[16] F. Farooq, R. M. Bolle, T.-Y. Jea, N. Ratha, Anonymous and revocable fingerprint recognition, in: Computer Vision and Pattern Recognition, IEEE Conference on, 2007, pp. 1–7.

[17] A. Vij, A. Namboodiri, Learning minutiae neighborhoods: A new binary representation for matching fingerprints, in: Computer Vision and Pattern Recognition Workshops, IEEE Conference on, 2014, pp. 64–69.

[18] H. Xu, R. Veldhuis, Binary representations of fingerprint spectral minutiae features, in: Pattern Recognition, 2010 20th International Conference on, IEEE, pp. 1212–1216.

[19] Y. C. Feng, P. C. Yuen, Class-distribution preserving transform for face biometric data security, in: Acoustics, Speech and Signal Processing, 2007. IEEE International Conference on, Vol. 2, pp. II–141.

[20] Y. C. Feng, P. C. Yuen, Binary discriminant analysis for generating binary face template, Information Forensics and Security, IEEE Transactions on 7 (2) (2012) 613–624.

[21] E. Kelkboom, X. Zhou, J. Breebaart, R. Veldhuis, C. Busch, Multi-algorithm fusion with template protection, in: Biometrics: Theory, Applications, and Systems, IEEE 3rd International Conference on, 2009, pp. 1–8.

[22] M.-H. Lim, A. B. J. Teoh, A novel encoding scheme for effective biometric discretization: Linearly separable subcode, Pattern Analysis and Machine Intelligence, IEEE Transactions on 35 (2) (2013) 300–313.

[23] A. K. Jain, S. Prabhakar, L. Hong, S. Pankanti, Fingercode: a filterbank for fingerprint representation and matching, in: Computer Vision and Pattern Recognition, 1999. EEE International Conference on, Vol. 2.

[24] S. Kanade, D. Petrovska-Delacretaz, B. Dorizzi, Multi-biometrics based cryptographic key regeneration scheme, in: Biometrics: Theory, Applications, and Systems. IEEE 3rd International Conference on, 2009, pp. 1–7.

[25] S. Kanade, D. Petrovska-Delacretaz, B. Dorizzi, Obtaining cryptographic keys using feature level fusion of iris and face biometrics for secure user authentication, in: Computer Vision and Pattern Recognition Workshops, IEEE Computer Society Conference on, 2010, pp. 138–145.

[26] G. Mai, M.-H. Lim, P. C. Yuen, Fusing binary templates for multi-biometric cryptosystems, in: Biometrics: Theory Applications and Systems, Seventh IEEE International Conference on, 2015.

[27] Y. Sutcu, Q. Li, N. Memon, Secure biometric templates from fingerprint-face features, in: Computer Vision and Pattern Recognition, IEEE Conference on, 2007, pp. 1–6.

[28] A. Nagar, K. Nandakumar, A. K. Jain, Multibiometric cryptosystems based on feature-level fusion, Information Forensics and Security, IEEE Transactions on 7 (1) (2012) 255–268.

[29] M. Prasad, M. Sukumar, A. Ramakrishnan, Orthogonal lda in pca transformed subspace, in: Frontiers in Handwriting Recognition, International Conference on, 2010, pp. 172–175.

[30] J. Yang, X. Zhang, Feature-level fusion of fingerprint and finger-vein for personal identification, Pattern Recognition Letters 33 (5) (2012) 623–628.

[31] R. Xu, D. Wunsch, et al., Survey of clustering algorithms, Neural Networks, IEEE Transactions on 16 (3) (2005) 645–678.

[32] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, Pattern Analysis and Machine Intelligence, IEEE Transactions on 27 (8) (2005) 1226–1238.

[33] A. Kraskov, P. Grassberger, Mic: mutual information based hierarchical clustering, in: Information theory and statistical learning, Springer, 2009, pp. 101–123.

[34] K. Deb, An efficient constraint handling method for genetic algorithms, Computer methods in applied mechanics and engineering 186 (2) (2000) 311–338.

[35] K. Deep, K. P. Singh, M. Kansal, C. Mohan, A real coded genetic algorithm for solving integer and mixed integer optimization problems, Applied Mathematics and Computation 212 (2) (2009) 505–518.

[36] L. Hornak, A. Ross, S. G. Crihalmeanu, S. A. Schuckers, A protocol for multibiometric data acquisition storage and dissemination, Tech. rep., West Virginia University, https://eidr. wvu. edu/esra/documentdata. eSRA (2007).

[37] P. J. Phillips, H. Moon, S. A. Rizvi, P. J. Rauss, The feret evaluation methodology for face-recognition algorithms, Pattern Analysis and Machine Intelligence, IEEE Transactions on 22 (10) (2000) 1090–1104.

[38] Casia iris image database.
URL http://biometrics.idealtest.org/

[39] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, in: Computer vision and pattern recognition, IEEE Conference on, Vol. 1, 2005, pp. 947–954.

[40] K. W. Bowyer, P. J. Flynn, The ND-IRIS-0405 iris image dataset (2009).

[41] Verifinger sdk.
URL http://www.neurotechnology.com/

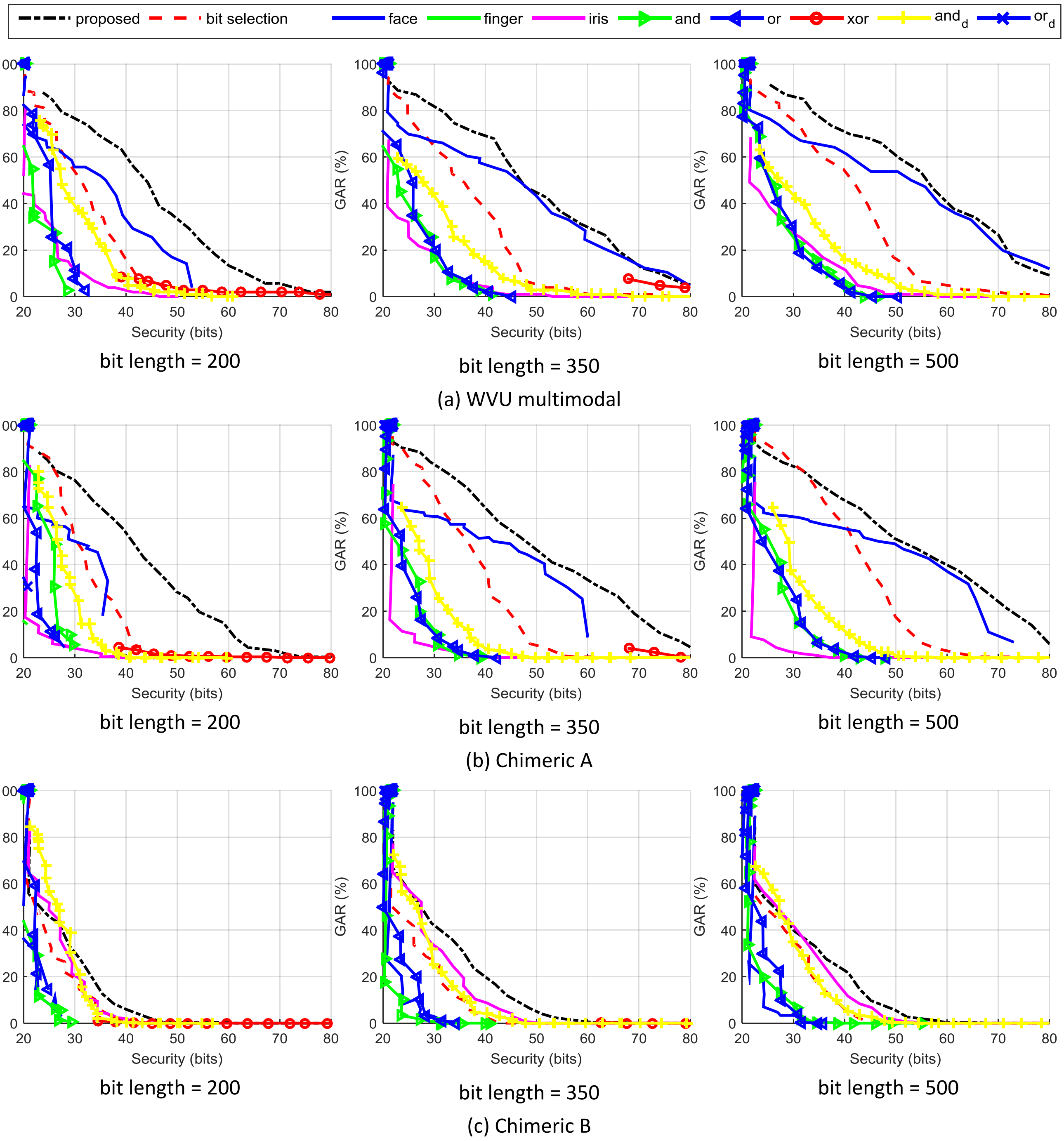
[42] A. Uhl, P. Wild, Weighted adaptive hough and ellipsopolar transforms for real-time iris segmentation, in: Biometrics (ICB), 5th IAPR International Conference on, IEEE, 2012, pp. 283–290.

[43] J.-G. Ko, Y.-H. Gil, J.-H. Yoo, K.-I. Chung, A novel and efficient feature extraction method for iris recognition, ETRI journal 29 (3) (2007) 399–401.

[44] C. Rathgeb, A. Uhl, P. Wild, Iris recognition: from segmentation to template security, Advances in Information Security 59.

[45] S. Hidano, T. Ohki, K. Takahashi, Evaluation of security for biometric guessing attacks in biometric cryptosystem using fuzzy commitment scheme, in: Conference of the Biometrics Special Interest Group (BIOSIG), IEEE, 2012, pp. 1–6.

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



bit length = 200

bit length = 350

bit length = 500

(a) WVU multimodal

bit length = 200

bit length = 350

bit length = 500

(b) Chimeric A

bit length = 200

bit length = 350

bit length = 500

(c) Chimeric B

Figure 6: G-S Trade-off Analysis on (a) WVU multi-modal, (b) Chimeric A, and (c) Chimeric B.