Multi-Algorithm Fusion with Template Protection

E.J.C. Kelkboom, X. Zhou, J. Breebaart, R.N.J. Veldhuis, C. Busch

Abstract— **The popularity of biometrics and its widespread use introduces privacy risks. To mitigate these risks, solutions such as the helper-data system, fuzzy vault, fuzzy extractors, and cancelable biometrics were introduced, also known as the field of template protection. In parallel to these developments, fusion of multiple sources of biometric information have shown to improve the verification performance of the biometric system. In this work we analyze fusion of the protected template from two 3D recognition algorithms (multi-algorithm fusion) at feature-, score-, and decision-level. We show that fusion can be applied at the known fusion-levels with the template protection technique known as the Helper-Data System. We also illustrate the required changes of the Helper-Data System and its corresponding limitations. Furthermore, our experimental results, based on 3D face range images of the FRGC v2 dataset, show that indeed fusion improves the verification performance.**

I. Introduction

There is a growing popularity of using biometrics in applications ranging from simple home or business applications with a small and limited group of enrolled people (for example access control to buildings or rooms) to large-scale systems used by an entire nation or even the entire world (for example identity cards with biometrics or the electronic passport e-Passport). However, its widespread use increases the privacy risks such as identity theft or activity monitoring by cross-matching between biometric databases of different applications. The field of template protection provides the technology that mitigates these privacy risks by transforming the biometric template with a one-way function in order to guarantee the irreversibility property and by randomizing the biometric template in order to guarantee that multiple protected templates from the same biometric sample cannot be linked with each other. In the literature, multiple solutions have been presented to solve these problems. Some examples are the *Fuzzy Commitment Scheme* [1], *Helper-Data Systems* (HDS) [2], [3], [4], *Fuzzy Vaults* [5], [6], *Fuzzy Extractors* [7], [8], and *Cancelable Biometrics* [9].

In parallel to these developments, fusion of multiple sources of biometric information has shown to improve the recognition performance of the biometric system. As stated in [10], the basic principle of fusion is the reconciliation of evidence presented by multiple sources of biometric information in order to enhance the classification performance. As described in [10], multiple sources of biometric information can be extracted from the same biometric modality by (see Fig. 1 for the case of fingerprints): (i) capturing a sample of multiple instances (left and right index fingerprint or iris) with the same sensor, (ii) using different sensors to acquire a different type of biometric samples from the same instance, (iii) capturing multiple samples using the same sensor and instance, and (iv) extracting multiple feature representations of the same biometric sample using different algorithms. These cases are referred to as the multi-instance, multi-sensor, multi-sample, and multi-algorithm systems, respectively. Further more, the fifth type is the multi-modal system, which is the fusion of sources of biometric information from multiple modalities, for example fingerprint, face, iris, voice, palm or retina. To complete the summary from [10], the sixth type is referred to as the hybrid system, which consists of a combination of the aforementioned fusion types. Each multi-biometric fusion type can be implemented at feature-level, score-level, or decision-level of the biometric system.

In [11], multi-sample, multi-instance, and multi-modal fusion has been applied using the Fuzzy Vault as the template protection system. For multi-sample fusion a single mosaiced template is obtained from multiple fingerprint impressions from which the vault is constructed. For multi-instance fusion the union of the minutiae sets of the left and right index fingers is used to construct the vault. For multi-modal fusion, a fingerprint and an iris sample are combined by concatenating the unordered minutiae set with the transformed iriscode extracted from the fingerprint and iris samples, respectively. The concatenated unordered set is used to construct the vault. The recognition performance improved for all three cases as well as the claimed security.

Our Contribution: Our work consists of applying multi-algorithm fusion with the Helper-Data System. We show that fusion can be applied at feature-, score-, and decision-level and illustrate the required changes of the Helper-Data System and its corresponding limitations. We experimentally determine the performance of different fusion methods at each level. The experiments are based on 3D face range images of the FRGC v2 dataset [12], where we use two recognition algorithms from different vendors.

The outline of this paper is as follows. In Section II we briefly discuss the HDS system, while in Section III we discuss the application of multi-algorithm fusion at feature-, score-, and decision-level using the HDS system. The experimental setup and results are provided in Section IV. We finish with the conclusions in Section V.

E.J.C. Kelkboom and J. Breebaart are with Philips Research, The Netherlands {Emile.Kelkboom, Jeroen.Breebaart}@philips.com

R.N.J. Veldhuis is with the University of Twente, Fac. EEMCS, The Netherlands R.N.J.Veldhuis@utwente.nl

X. Zhou and C. Busch are with the Fraunhofer Institute for Computer Graphics Research IGD, Germany {Xuebing.Zhou, Christoph.Busch}@igd.fraunhofer.de

II. Template Protection Scheme

Many template protection schemes presented in the literature are based on the capability of generating a robust binary vector or key from biometric measurements of the same subject. The HDS system we consider is depicted in Fig. 2. For the sake of coherence we use the terminology *auxiliary data* (AD) and *pseudo identity* (PI) proposed in [13], which is in line with standardization activities in ISO. From the real-valued representation of the biometric sample, $\mathbf{f} \in \mathbb{R}^{N_\mathrm{F}}$, a binary vector $\mathbf{f}_\mathrm{B} \in \{0,1\}^{N_\mathrm{B}}$ is extracted within the *Bit Extraction* module. We use a single bit quantization scheme based on thresholding and the *reliable component selection* (RCS) algorithm. The N_B most reliable components are selected based on the estimated z-score for each component. With use of the multiple enrollment samples, the z-score is estimated as the ratio between the distance of the estimated mean with respect to the quantization threshold and the estimated standard deviation, see [2] for a more detailed description of the z-score estimation and the quantization scheme. The auxiliary data AD_1 contains the index information of the selected reliable components.

The binary vector $\mathbf{f}_\mathrm{B}^\mathrm{e}$ could be used as a key for any encryption purposes, however it is not considered as being practical because of the high probability that it is not exactly the same in both the enrollment and verification phase ($\mathbf{f}_\mathrm{B}^\mathrm{e} \neq \mathbf{f}_\mathrm{B}^\mathrm{v}$), due to measurement noise and biometric variability that lead to *bit errors*. The number of bit errors is also referred to as the Hamming distance $d_\mathrm{H}(\mathbf{f}_\mathrm{B}^\mathrm{e}, \mathbf{f}_\mathrm{B}^\mathrm{v})$. Therefore, error-correcting codes (ECC) are used to deal with the bit errors. Combining the ECC with a cryptographic hash function forms the scheme also known as the Fuzzy Commitment scheme [1]. In the enrollment phase, a binary secret or message vector $\mathbf{K}$ is randomly generated by the *Random-Number-Generator* (*RNG*) module. A codeword $\mathbf{C}$ of an error-correcting code is obtained by encoding $\mathbf{K}$ in the *ECC-Encoder* module. As the ECC we use the linear block type code "Bose, Ray-Chaudhuri, Hocquenghem" (BCH) [14], which is specified by the codeword length (n_c), message length (k_c), and the corresponding number of bits that can be corrected (t_c), in short $[n_\mathrm{c}, k_\mathrm{c}, t_\mathrm{c}]$. Some practical BCH settings are provided in Table I, where the bit error rate (BER) is the ratio $t_\mathrm{c}/n_\mathrm{c}$. The codeword is XOR-ed with $\mathbf{f}_\mathrm{B}^\mathrm{e}$ in order to obtain the auxiliary data AD_2. Hence, $\mathbf{f}_\mathrm{B}^\mathrm{e}$ should have the same dimension as $\mathbf{C}$ implying $N_\mathrm{B} = n_\mathrm{c}$. Furthermore, the hash of $\mathbf{K}$ is taken in order to obtain the pseudo identity PI. The larger the secret size the more difficult it is to guess $\mathbf{K}$ from PI.

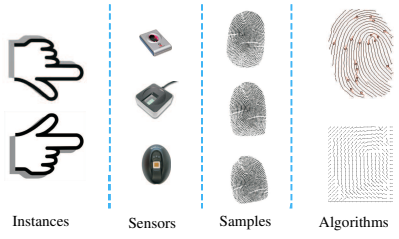


Fig. 1. Multiple sources of biometric information using fingerprints as the single modality.

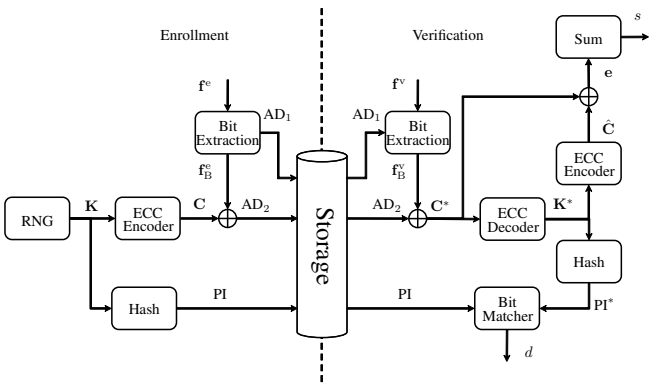


Fig. 2. The HDS template protection scheme.

In the verification phase, a new biometric sample is taken and transformed into its binary representation within the *Bit Extraction* module with help of auxiliary data AD_1. The new word $\mathbf{C}^*$ is computed by XOR-ing $\mathbf{f}_\mathrm{B}^\mathrm{v}$ with AD_2. The candidate secret $\mathbf{K}^*$ is obtained by decoding $\mathbf{C}^*$ in the *ECC-Decoder* module. Subsequently, the candidate pseudo identity PI^* is computed by hashing $\mathbf{K}^*$. The decision in the *Bit Matcher* module is based on whether PI and PI^* are bitwise identical.

The *Bit-Matcher* module yields identical PI and PI^* when the number of bit errors between the binary vectors $\mathbf{f}_\mathrm{B}^\mathrm{e}$ and $\mathbf{f}_\mathrm{B}^\mathrm{v}$ is smaller or equal to the error-correcting capability t_c of the ECC. Thus, there is an accept when the Hamming distance is smaller than t_c, $d_\mathrm{H}(\mathbf{f}_\mathrm{B}^\mathrm{e}, \mathbf{f}_\mathrm{B}^\mathrm{v}) = ||\mathbf{f}_\mathrm{B}^\mathrm{e} \oplus \mathbf{f}_\mathrm{B}^\mathrm{v}||_1 \leq t_\mathrm{c}$. Therefore, the fuzzy commitment scheme can be considered as a Hamming distance classifier with threshold t_c. Note, that the maximum number of bits that the BCH can correct t_c^* is close to 25% of the codeword length. In the remainder of the text, we indicate this limitation as the *ECC-limitation*.

As a distance score s we use the number of bits that had to be corrected by the ECC decoder. The candidate secret $\mathbf{K}^*$ is encoded to its corresponding codeword $\hat{\mathbf{C}}$ and is XOR-ed with $\mathbf{C}^*$ in order to obtain the error pattern $\mathbf{e}$. The error pattern is equal to the bit differences between the enrollment

TABLE I

Some examples of the BCH code given by the codeword (n_c and message (k_c) length, the corresponding number of correctable bits (t_c), and the bit error rate (BER) $t_\mathrm{c}/n_\mathrm{c}$.

n_c	k_c	t_c	BER $= t_\mathrm{c}/n_\mathrm{c}$
127	8	31	24.4%
	15	27	21.3%
255	9	63	24.7%
	21	55	21.6%
511	10	127	24.9%
	31	109	21.3%
1023	11	255	24.9%
	46	219	21.4%

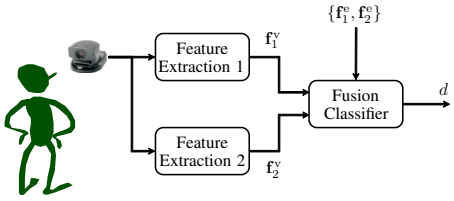


Fig. 3. A toy-example of a multi-algorithm fusion system.

and verification binary feature vectors ($\mathbf{f}_{\mathrm{B}}^{\mathrm{e}} \oplus \mathbf{f}_{\mathrm{B}}^{\mathrm{v}}$) as follows

$$\begin{aligned} \mathbf{e} &= \hat{\mathbf{C}} \oplus \mathbf{C}^* \\ &= \hat{\mathbf{C}} \oplus (\mathbf{f}_{\mathrm{B}}^{\mathrm{v}} \oplus \mathrm{AD}_2) \\ &= \hat{\mathbf{C}} \oplus (\mathbf{f}_{\mathrm{B}}^{\mathrm{v}} \oplus (\mathbf{f}_{\mathrm{B}}^{\mathrm{e}} \oplus \mathbf{C})) \\ &= (\hat{\mathbf{C}} \oplus \mathbf{C}) \oplus (\mathbf{f}_{\mathrm{B}}^{\mathrm{e}} \oplus \mathbf{f}_{\mathrm{B}}^{\mathrm{v}}) \\ &= (\mathbf{f}_{\mathrm{B}}^{\mathrm{e}} \oplus \mathbf{f}_{\mathrm{B}}^{\mathrm{v}}) \text{ if } \hat{\mathbf{C}} = \mathbf{C}, \end{aligned} \tag{1}$$

where $\hat{\mathbf{C}}$ is equal to $\mathbf{C}$ when there is an accept, i.e. $\mathbf{K}$ and $\mathbf{K}^*$ are equal. The distance score s is thus the sum of the error pattern, hence equal to $d_{\mathrm{H}}(\mathbf{f}_{\mathrm{B}}^{\mathrm{e}}, \mathbf{f}_{\mathrm{B}}^{\mathrm{v}})$ and only a valid score when there is an accept, i.e. $d_{\mathrm{H}}(\mathbf{f}_{\mathrm{B}}^{\mathrm{e}}, \mathbf{f}_{\mathrm{B}}^{\mathrm{v}}) \leq t_{\mathrm{c}}$. If the score is not valid we only know that $d_{\mathrm{H}}(\mathbf{f}_{\mathrm{B}}^{\mathrm{e}}, \mathbf{f}_{\mathrm{B}}^{\mathrm{v}}) > t_{\mathrm{c}}$.

III. Applying Template Protection at Different Fusion Levels

In this work we are interested in the multi-algorithm fusion system as depicted in Fig. 3, where a 3D image is taken of the face of the subject from which the feature vectors $\mathbf{f}_1^{\mathrm{v}}$ and $\mathbf{f}_2^{\mathrm{v}}$ are extracted using two different feature extraction algorithms. These features are compared with their enrolled version $\{\mathbf{f}_1^{\mathrm{e}}, \mathbf{f}_2^{\mathrm{e}}\}$ within the *Fusion Classifier* module and a decision d is made whether to accept or reject the identity claim of the subject.

The comparison within the *Fusion Classifier* module can occur at different levels, namely at feature-, score-, or decision-level. In the following sections we discuss the implementation of the template protection system at the different fusion levels.

A. Feature-Level Fusion

Applying the template protection scheme at feature-level fusion is straightforward, the two feature vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ are concatenated before entering the template protection scheme, thus $\mathbf{f} = [\mathbf{f}_1; \mathbf{f}_2]$. The fused feature vectors have more components and most likely more components that have discriminating and robust properties. Hence, it is expected that more robust and discriminating bits can be extracted, which allows the use of larger binary vectors $\mathbf{f}_{\mathrm{B}}$ and thus larger codewords. It is known from the BCH code that larger codewords are more efficient, they have a larger secret at the same bit error rate (BER), see Table I.

B. Decision-Level Fusion

At decision-level fusion there is a template protection system for each source of biometric information with an individual decision for each system. The two decisions can be fused into a single decision d_f using a AND-rule or OR-rule. For the AND-rule, there is a final accept if and only if both template protection systems lead to an accept, thus $d_{\mathrm{H}}(\mathbf{f}_{\mathrm{B},1}^{\mathrm{e}}, \mathbf{f}_{\mathrm{B},1}^{\mathrm{v}}) \leq t_{\mathrm{c},1}$ and $d_{\mathrm{H}}(\mathbf{f}_{\mathrm{B},2}^{\mathrm{e}}, \mathbf{f}_{\mathrm{B},2}^{\mathrm{v}}) \leq t_{\mathrm{c},2}$. The acceptance region is the intersection defined by the individual decision boundaries crossing the operating point $t_{\mathrm{op}} = \{t_{op,1},\ t_{op,2}\} = \{t_{\mathrm{c},1}, t_{\mathrm{c},2}\}$ as shown in Fig. 4(a). For the OR-rule, there is a final accept if at least a single template protection system gives an accept. Hence, the acceptance region is the union of both as portrayed in Fig. 4(b).

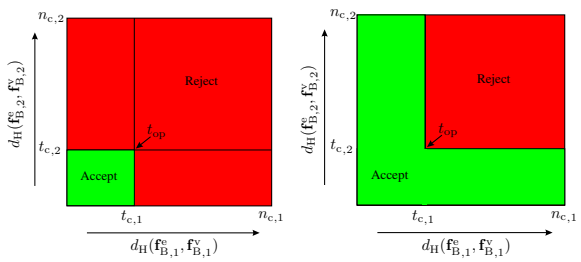


Fig. 4. Decision boundaries for the (a) AND and (b) OR decision fusion rule. The operating point t_{op} is at the intersection of the decision boundary given by $t_{\mathrm{c}1}$ and $t_{\mathrm{c}2}$.

Under the assumption that the binary vectors $\mathbf{f}_{\mathrm{B}}$ are randomly distributed in $\{0,1\}^{N_{\mathrm{B}}}$, it follows from the results in [15] that the maximum amount of privacy information that the HDS system can preserve is equal to the secret size $|\mathbf{K}| = k_{\mathrm{c}}$ from the ECC. The average number of attempts necessary for the adversary to randomly guess the secret $\mathbf{K}$ from its hashed version PI is equal to $\frac{1}{2}2^{k_{\mathrm{c}}}$. For the first source the secret size is $|\mathbf{K}_1| = k_{\mathrm{c},1}$ and $|\mathbf{K}_2| = k_{\mathrm{c},2}$ for the second source. For the OR-rule fusion, only one of the hash values has to be guessed correctly for a successful attack, hence the effective secret size in the fused setup is equal to the smallest secret size $|\mathbf{K}_f| = \min(k_{\mathrm{c},1}, k_{\mathrm{c},2})$. In case of the AND-rule fusion, both hash values have to be guessed correctly independently, thus the effective secret size is $|\mathbf{K}_f| = \log_2(2^{k_{\mathrm{c},1}} + 2^{k_{\mathrm{c},2}}) \leq \max(k_{\mathrm{c},1}, k_{\mathrm{c},2}) + 1$, where the equality holds only when $k_{\mathrm{c},1} = k_{\mathrm{c},2}$. This can be improved by combining or concatenating both secrets prior to hashing. In that case, the effective secret size is $|\mathbf{K}_f| = |\mathbf{K}_1| + |\mathbf{K}_2| = k_{\mathrm{c},1} + k_{\mathrm{c},2}$.

C. Score-Level Fusion

A general implementation of the template protection system at score-level fusion is depicted in Fig. 5. Each source of biometric information has a separate template protection system with a decision and score value as output. Note that we are using the number of corrected bits within the ECC as the distance score that is valid only when there is an accept, see Section II. Both scores (s_1 and s_2) and decisions (d_1 and d_2) are combined in the Score & Decision Fusion module into a single decision d_f. With the available scores, more flexible decision boundaries can be defined when compared to the AND-rule and OR-rule decision-level fusion cases that were presented in Fig. 4. Similar to the decision-level fusion case, an AND- or OR-rule can be used based on the decision d_i, which is now extended by incorporating the scores s_i

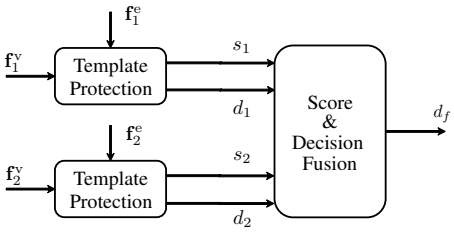
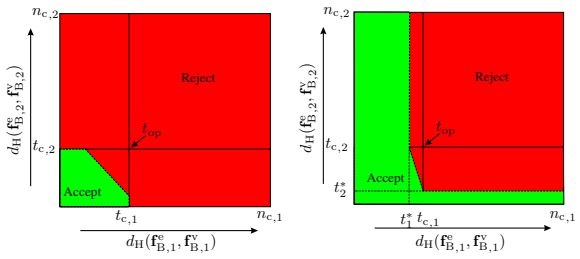


Fig. 5. Score fusion with template protection.



(a) AND-configuration (b) OR-configuration

Fig. 6. Examples of the decision boundaries for the score-level fusion case with (a) the AND- and (b) OR-configuration.

to determine the final decision d_f. Hence, there are two cases we refer to as the AND-configuration and the OR-configuration case.

For the AND-configuration case the initial acceptance region is similar to the AND-rule case as shown in Fig. 4(a). However, with use of the scores s_i a more refined decision boundary given by a function $f(s_1, s_2)$ can be defined. We mainly focus on the Sum-rule and Weighted-Sum-rule given as

$$s_f \quad = \quad w_1 s_1 + w_2 s_2, \text{ with } w_1 + w_2 = 1, \qquad (2)$$

where the Sum-rule is a degenerate case of the Weighted-Sum-rule by using weights equal to $\frac{1}{2}$. If there is an accept for both sources ($d_1 = d_2 = 1 =$ "Accept"), then there is only a final accept ($d_f = 1$) if the scores s_1 and s_2 are in the acceptance region defined by the function $f(s_1, s_2)$, see Fig. 6(a) for an example of the acceptance region using the Weighted-Sum-rule.

For the OR-configuration case, the same boundaries can be defined as for the AND-configuration case when there is an accept for both sources. However, if there is a single accept it is still possible to give a final accept if the single score s_i is smaller than a stricter threshold t_i^*. We use a stricter threshold because the final decision is now only based on a single source of biometric information. An example of the acceptance region is depicted in Fig. 6(b). Note that we define the stricter threshold t_1^* (t_2^*) as the intersection of the decision boundary function $f(s_1, s_2)$ with the $t_{c,2}$ ($t_{c,1}$).

IV. Experiments

In the previous section we presented the methods for multi-algorithm fusion at feature-, score-, and decision-level. In this section, we empirically validate the best performance achieved at each level by means of a biometric database and two feature extraction algorithms.

A. Experiment Setup

1) Biometric Databases: All the results in this work are obtained using the FRGC v2 dataset [12] containing a total of 4007 3D shape samples from 465 subjects.

However, one of the 3D shape recognizer we used could not successfully extract a feature vector out of each sample, hence reducing the dataset to 3507 samples from 454 subjects. As the template protection algorithm works best at multiple enrollment samples, the subset of subjects with at least 6 (5 as enrolment samples with at least one for the verification) samples or more is selected. This resulted into a subset of 261 subjects with in total 2970 samples.

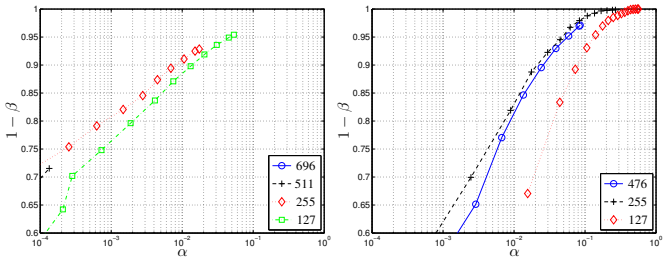
2) Feature Extraction Algorithms: The first algorithm is the shape-based 3D face recognizer from [16] and is referred to as Algo1. It has two main steps: 1) the alignment of faces, and 2) the extraction of surface features from 3D facial data. In the alignment step, each face is registered to a generic face model (GFM) and the central facial region is cropped. The GFM is computed by averaging correctly aligned images from a training set. After the alignment step, we can assume that all faces are transformed in such a way that they best fit the GFM, and have the same position in the common coordinate system.

After alignment, the facial surface is divided into 174 local regions. For each region, the maximum and minimum principal curvature direction are computed. Each of the two directions is presented by the azimuthal and the polar angle in the spherical coordinate system. Combining all the regions leads to a feature vector dimension $N_F = 174 \times 2 \times 2 = 696$.

The second algorithm, Algo2, is a histogram-based feature extraction method. After the pre-registration of the face data, a frontal view of the face model is obtained, where the tip of the nose is at the origin in the Cartesian coordinate system. The distribution of depth values of the normalized face model describes the characteristics of an individual facial surface. In order to obtain more detailed information about the local geometry, the 3D model is divided into several sub areas which are orthogonal to the symmetry plane of the face. The features are extracted from the depth value distribution in each sub-area. The feature vector dimension is $N_F = 476$. A full description of this algorithm is provided in [17].

For both feature extraction algorithms, the raw feature vectors they produce are used as input of the template protection system as described in Section II. Hence, no signal processing is performed.

3) Testing Protocols: The performance testing protocol consists of randomly selecting 50% (130) subjects as the training set and the other subjects as the test set, this is referred to as the training-test-set split. The template protection system parameters such as the quantization thresholds, used within the *Bit Extraction* module, are estimated on this training set. Hereafter, the test set is randomly split into an equally sized fusion-training and evaluation set containing around 65 subjects each. All the training needed for fusion is thus performed on the fusion-training set and the reported performance is obtained from the evaluation



(a) Algo1 (b) Algo2

Fig. 7. Individual ROC curves for algorithm (a) Algo1and (b) Algo2 at different settings of n_c.

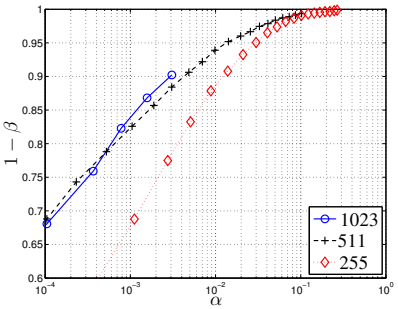


Fig. 8. ROC curves at feature-level fusion of Algo1 and Algo2 algorithm.

set. From the evaluation set, 5 samples of each subject are randomly selected as the enrollment samples while the remaining samples are considered as the verification samples. This split is referred to as the enrollment-verification split. The protected template is generated using all the enrollment samples and compared with each verification sample.

The training-test-set split is performed five times, while for each split the enrollment-verification split is performed five times. From each enrollment-verification split we measure the β_{tar} (the false rejection rate (FRR, β) at the targeted false acceptance rate (FAR, α) of $\alpha_{\text{tar}} = 0.25\%$) and the equal-error rate (EER), which is the error rate achieved at the operating point where both FRR and FAR are equal. With use of the 25 measurements we estimate the 95% confidence interval (ci) defined as $ci = 1.96\sigma_{EER}/\sqrt{(25)}$ for the EER case while using $\sigma_{\beta_{\text{tar}}}$ for the β_{tar} case, respectively. Note, that the splits are performed randomly, however the seed at the start of the protocol is always the same, hence all the splits are equal for the performance tests at feature-, score-, and decision-level fusion. Hence, the splitting process does not contribute to any performance differences.

B. Experiment Results

1) Individual Algorithm Performances: Before we start fusing the different biometric sources, we first determine their individual performance as given by the ROC curves in Fig. 7 for different codeword lengths n_c with the EER and β_{tar} details in Table II. The table provides the ci for both EER and β_{tar} and their operating point provided as the relative Hamming distance (RHD). The right column of the table provides the secret size $|\mathbf{K}|$ of the ECC corresponding to the t_c setting that leads to closest α but smaller than the target α_{tar}. This is the ECC setting with a BER just larger than the operating point in RHD corresponding to β_{tar}. Entries in the table indicated with quotes cannot be reached in practice because of the ECC-limitation, however we are able to estimate them because of the Hamming distance classifier assumption as discussed in Section II. Entries with "x" can neither be reached nor estimated.

Note that we used five enrollment samples ($N_e = 5$) from which the average is taken. Also note that the ROC curves are limited because of the ECC-limitation. In order to reach larger α and smaller β values it is required to tolerate and thus correct more bit errors. However, the error correcting capability of an ECC is limited. From the results we can conclude that both algorithms perform optimally at a codeword size of $n_c = 255$. These settings are used in the score- and decision-level fusion analysis. Compared to the Algo2 algorithm, Algo1 has a better performance but a smaller secret size (see Table II, right column).

TABLE II

THE EER AND β_{tar}, AND THEIR ci AND OPERATING POINT FOR THE INDIVIDUAL ALGORITHMS ALGO1 AND ALGO2 AT DIFFERENT SETTINGS OF n_c. THE LAST COLUMN IS THE SECRET SIZE $|\mathbf{K}|$ OF THE ECC AT THE OPERATING POINT t_c FOR ACHIEVING α_{tar}.

n_c	*EER* [%]	*RHD* [%]	β_{tar} [%]	*RHD* [%]	$\lvert\mathbf{K}\rvert$ [bits]
Algo1					
696	"3.75 ± 0.21"	"38.8"	"16.02 ± 1.61"	"33.6"	x
511	"3.69 ± 0.26"	"35.0"	"14.91 ± 1.63"	"29.0"	x
255	"3.99 ± 0.35"	"27.5"	15.33 ± 1.84	20.0	21
127	4.84 ± 0.42	23.6	19.18 ± 1.82	15.0	29
Algo2					
476	5.44 ± 0.35	22.1	37.69 ± 3.14	11.8	45
255	5.06 ± 0.30	10.2	30.25 ± 2.88	2.0	215
127	8.92 ± 0.33	3.9	89.57 ± 1.20	0	120

2) Multi-Algorithm Fusion at Feature-Level: At feature-level we concatenate both feature vectors together and consider it as a single feature vector. The new dimension of the feature vector is 1175. Because of the larger dimension it is possible to use larger codeword lengths as in the individual case in Section IV-B.1. The performances at different codeword lengths are shown in Fig. 8 with the EER and β_{tar} details in Table III. The best performance is achieved by using the largest codeword length of 1023 bits. It is just able to reach the targeted α_{tar} that leads to a $\beta_{\text{tar}} = 11.1\%$.

3) Multi-Algorithm Fusion at Decision-Level: At decision and score-level fusion, the scatter plot of the genuine and

TABLE III

PERFORMANCE RESULTS OF MULTI-ALGORITHM FUSION AT FEATURE-LEVEL.

n_c	*EER* [%]	*RHD* [%]	β_{tar} [%]	*RHD* [%]	$\lvert\mathbf{K}\rvert$ [bits]
1023	"2.45 ± 0.24"	"29.6"	11.10 ± 1.70	24.5	11
511	2.89 ± 0.34	18.6	12.88 ± 1.71	11.7	103
255	3.89 ± 0.32	11.8	22.79 ± 2.64	5.1	155

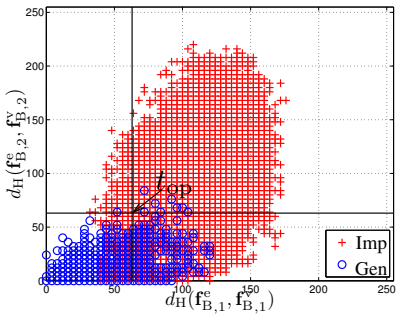
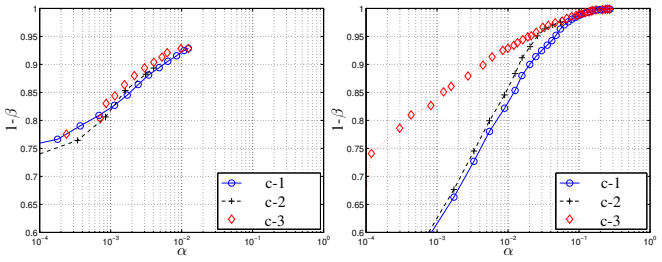


Fig. 9. Scatter plot of the genuine (Gen) and imposter (Imp) scores of the algorithms Algo1 and Algo2. The operating point $t_{\rm op}$ is at the intersection of the vertical and horizontal decision boundaries of Algo1 and Algo2, respectively.



(a) AND-rule (b) OR-rule

Fig. 10. Decision fusion results for (a) the AND-rule and (b) the OR-rule for the three cases (c-1, c-2, c-3).

imposter scores of both algorithms, as shown in Fig. 9, may indicate the possible gain when fusing at these levels. The scatter plot also depicts the decision boundary indicated by the operating point $t_{\rm op}$.

We will investigate both the AND-rule and OR-rule performance at different strategies of moving the operating point $t_{\rm op} = \{t_{op,1},\ t_{op,2}\}$ on the scatter plot, whose range is $t_{op,1} \in [0,\ t^*_{{\rm c},1}]$ and $t_{op,2} \in [0,\ t^*_{{\rm c},2}]$ for each axis respectively, with $t^*_{\rm c}$ being the maximum error-correcting capability of the ECC. In the first case (c-1) we consider $t_{op,2} = t_{op,1}$ and vary $t_{op,1}$ from 0 to $t^*_{{\rm c},1}$ considering that $t^*_{{\rm c},1} = t^*_{{\rm c},2}$ because the optimal individual performance is at the same codeword length as observed in Section IV-B.1. In the second case (c-2), the operating point crosses the EER operating point of the individual performances $\{t_{EER,1},\ t_{EER,2}\}$ linearly, hence the operating point is defined as $t_{\rm op} = \{t_{op,1}, \frac{t_{EER,2}}{t_{EER,1}} t_{op,1}\}$ with $t_{op,1} \in [0,\ \min(t^*_{{\rm c},1}, \frac{t_{EER,1}}{t_{EER,2}} t^*_{{\rm c},2})]$. In the third and final case (c-3) we use the optimal fusion method from [18], which estimates the performance in terms of α and β at each possible operating point in the scatter plot and takes the operating points on the envelope which leads to the best performance. This optimization process of finding the optimal operating points is in fact a training process and is thus performed on the fusion-training set. The final performance results are obtained by calculating the performance of the test set on the optimal operating points.

The performance results of the three cases are shown in Fig. 10(a) for the AND-rule and Fig. 10(b) for the OR-rule respectively with the performance details provided in Table IV. Because there are two template protection systems we provide the RHD of the operating point and the secret size for each system. From the results we can conclude that the optimal decision fusion method (c-3) leads to the best performance for both the AND-rule and OR-rule method. The performance differences between the three cases of moving the operating point is very small for the AND-rule method, while significant for the OR-rule method. This difference becomes more evident when analyzing the trajectory of the operating point as depicted in Fig. 11. The optimal operating points obtained by the optimal AND-rule method (c-3-AND) is between the operating points of cases c-1 and c-2. However, for the optimal OR-rule method (c-3-OR) the obtained operating points are significantly different than for case c-1 and c-2. For the first few points the operating points moves to the right, tangent to the x-axis ($t_{op,1}$ increases while $t_{op,2}$ stays relatively constant) and sharply moves up ($t_{op,2}$ increases) once $t_{op,1}$ reaches the limit of $t_{{\rm c},1}$. Because the optimal fusion method facilitates more flexibility of the operating points, it significantly improves the performance as is shown in Fig. 10(b).

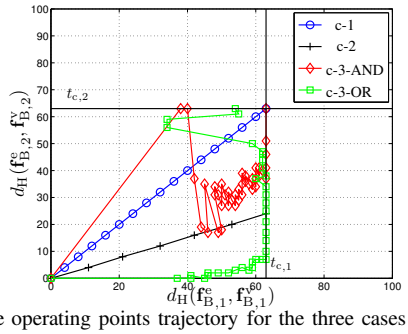


Fig. 11. The operating points trajectory for the three cases (c-1, c-2, c-3) for the AND-rule and OR-rule decision fusion methods.

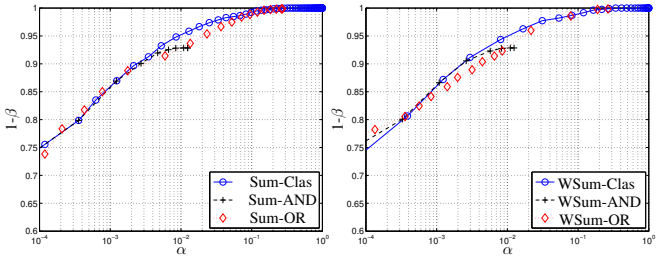
Observe that the OR-rule is able to obtain a greater part of the ROC curve than the AND-rule, as the OR-rule is able to reach the EER operating point while the AND-rule cannot, while both have the same ECC-limitation. The decision boundaries in Fig 4, 6, and 9 clearly show that at the same operating point the OR-rule has a larger Accept area than the AND-rule and can thus achieve a larger α and smaller β.

The effective secret size as discussed in Section III-C depends on the configuration being used. For the AND-configuration, the total secret size is the sum of the secret

TABLE IV

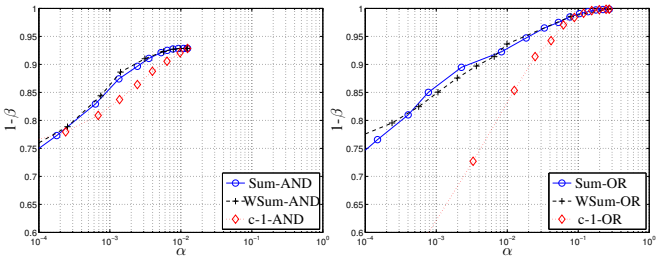
PERFORMANCE RESULTS OF MULTI-ALGORITHM FUSION AT DECISION-LEVEL. THE OPERATING POINTS AND SECRET SIZE ARE PROVIDED FOR BOTH TEMPLATE PROTECTION SYSTEMS.

$n_{\rm c}$	*EER* [%]	*RHD* [%]	$\beta_{\rm tar}$ [%]	*RHD* [%]	$\lvert\mathbf{K}\rvert$ [bits]
AND-rule					
c-1	x	x	13.45 ± 1.87	$[20.8, 20.8]$	$[21, 21]$
c-2	x	x	12.71 ± 2.59	$[23.9, 9.0]$	$[9, 99]$
c-3	x	x	11.34 ± 2.72	$[22.0, 13.7]$	$[13, 47]$
OR-rule					
c-1	4.78 ± 0.29	$[10.2, 10.2]$	29.83 ± 3.31	$[2.4, 2.4]$	$[207, 207]$
c-2	3.46 ± 0.34	$[21.2, 7.8]$	28.23 ± 3.50	$[6.7, 2.4]$	$[131, 207]$
c-3	3.27 ± 0.38	$[24.7, 5.9]$	12.58 ± 6.27	$[19.2, 0.8]$	$[21, 239]$



(a) Sum-rule (b) Weighted-Sum-rule

Fig. 12. ROC curves at score-level fusion using (a) the Sum-rule and (b) the Weighted-Sum-rule. In both cases we compare the classical performance (Clas) where there is no ECC-limitation with the AND- and OR-configuration with ECC-limitation.



(a) AND-rule (b) OR-rule

Fig. 13. Convergence of the score-level fusion ROC curves (Sum and Wsum) towards the decision-level curves (c-1) for the (a) OR-rule and (b) AND-rule cases.

size of each template protection system individually. For the OR-configuration case the effective secret size is the minimum of both.

4) Multi-Algorithm Fusion at Score-Level: The scatter plot indicates that using a Sum-rule or Weighted-Sum-rule score fusion method should improve the overall performance with respect to the individual performances. For the Weighted-Sum-rule method given by (2), the weighting coefficients are estimated from the disjunct fusion-training set as discussed in Section IV. The weights are iteratively varied and the values with the best performance in terms of the EER are selected. If the EER cannot be estimated, for example because of the ECC-limitation, we optimize using β_{tar} instead.

The score fusion algorithm can only be applied when the scores of both algorithm are available as portrayed by the accept region in Fig. 6(a) for the AND-configuration case. The accept region can be extend by using the OR-configuration given in Fig. 6(b). If only a single score s_1 (s_2) is available a stricter threshold t_1^* (t_2^*) is used. Note that the ECC settings are set to t_{c}^* for both template protection systems in order to have the largest acceptance region where both scores are available, hence fully benefitting from the score-fusion method. Thus, the threshold variable for the ROC curve becomes the weighted sum given by (2).

The results for the Sum-rule and Weighted-Sum-rule score fusion methods are depicted in Fig. 12(a) and (b), respectively. We investigate both the AND- and OR-configuration indicated as Sum-AND and Sum-OR for the Sum-rule and WSum-AND and WSum-OR for the Weighted-Sum-rule. As a comparison, the classical Sum-rule and Weighted-Sum-rule without the ECC limitation are included and referred to as Sum-Clas and WSum-Clas, respectively. The average weights $[\bar{w}_1, \bar{w}_2]$ found during the fusion training are $[0.59, 0.41]$ for the WSum-Clas case, $[0.7, 0.3]$ for the WSum-AND case, and $[0.8, 0.2]$ for the WSum-OR case. More performance details are provide in Table V. Because there are two template protection system we provide the RHD of the operating point and the secret size for each system. In terms of the β_{tar} values, the results indicate that the AND-configuration outperforms the OR-configuration but not the classical results without the ECC-limitations. Within the AND-configuration, the Weighted-Sum-rule has the best performances, while the Sum-rule has a better performance for the OR-configuration case. Note, that all the measured differences are within the estimated confidence intervals, hence the observed differences cannot be considered as being significant. The results also show that the Sum-AND (WSum-AND) curve follows the Sum-Clas (WSum-Clas) curve at smaller α values, but starts deviating at larger α values. At smaller α values the accept area for the Sum-AND case is not limited by the ECC-limitation and is thus equal to the accept area of the Sum-Clas case. This also holds for the WSum-AND and WSum-Clas scenario only if the weights are equal for both cases. However at larger α values the decision boundary is at a larger Hamming distances with the consequence that the accept area for the WSum-AND and Sum-AND cases are limited by the ECC-limitation as shown in Fig. 6(a) and approaches the accept area for the AND-rule c-1 decision-level fusion method case as depicted in Fig. 4(a). Under the same conditions this also holds for the OR-rule cases. The convergence of the score-level fusion ROC curves towards the decision-level curves are portrayed in Fig. 13.

TABLE V

PERFORMANCE RESULTS OF MULTI-ALGORITHM FUSION AT FEATURE-LEVEL.

case	*EER* [%]	β_{tar} [%]	$\lvert\mathbf{K}\rvert$ [bits]
Sum			
Clas	2.58 ± 0.30	9.83 ± 1.81	$[9, 9]$
AND	x	10.26 ± 1.80	$[9, 9]$
OR	3.45 ± 0.37	10.38 ± 1.56	$[9, 9]$
WSum			
Clas	2.57 ± 0.32	9.58 ± 1.74	$[9, 9]$
AND	x	9.63 ± 2.20	$[9, 9]$
OR	3.28 ± 0.39	11.68 ± 1.74	$[9, 9]$

Because we fixed the ECC correcting capability at t_{c}^* the secret size for each protected template is 9 bits at $n_{\text{c}} = 255$ and the effective secret size is the sum of 18 bits for the AND-configuration when both secrets are concatenated before hashing. For the OR-configuration case the effective secret size is the minimum of both, hence 9 bits.

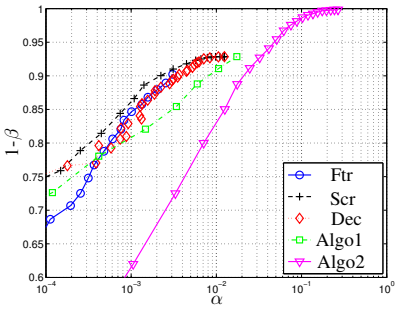


Fig. 14. Overview of the best ROC curves obtained at feature-, score-, and decision-level fusion, and the individual algorithms Algo1 and Algo2.

TABLE VI

SUMMARY OF EMPIRICAL RESULTS OF MULTI-ALGORITHM.

Type	*EER* [%]	*RHD* [%]	β_{tar} [%]	*RHD* [%]	k_c [bits]
Feature	x	x	11.16 ± 1.70	24.5	11
Score	x	x	9.63 ± 2.20	$[24.7, 24.7]$	2×9
Decision	x	x	11.34 ± 2.72	$[22.0, 13.7]$	$[13, 47]$
Algo1	x	x	15.84 ± 2.10	19.6	21
Algo2	x	x	29.97 ± 3.29	2.4	207

5) Summary and Discussions: As a summary we compare the performance of the individual algorithms with the best performances obtained at each fusion level, see Fig. 14 for the ROC curves with the details in Table VI. The best performance at feature-level fusion was with a codeword of 1023 bit. At score-level fusion, the best performance is obtained using the Weight-Sum-rule with the AND-configuration, while at decision-level fusion the optimal AND-rule method led to the best performance.

Compared to the individual performances, the performance improvement with fusion in terms of β_{tar} exceeds 6%. The difference can be considered as significant because the combined confidence interval is around 4%. The best performance is obtained at score-level fusion, however the differences with the feature- and decision-level fusion methods are not significant. The effective secret size at score-level fusion is close to the secret size of the best individual algorithm. Hence we can conclude that the performance has been improved while maintaining a similar effective secret size.

V. CONCLUSIONS

We have shown that it is possible to apply fusion with the Helper-Data System at feature-, score-, and decision-level. However, the Helper-Data System inherently has only a decision as the output, hence it had to be adapted in order to have a score as output for the score-level fusion. We took the number of the bits the ECC had to correct as the distance score measurement.

Furthermore, we have also shown that applying fusion with template protection at feature- or decision-level is straightforward and conventional. However, fusion at score-level is different due to the use of an ECC, which has a limited error correcting capability. Consequently, for each template protection system there is only a valid score when there is a match. Hence, this ECC-limitation limits the decision boundaries.

The performance at all fusion levels is significantly better than the performance of the individual biometric sources. The best performance is obtained at score-level fusion, with a β_{tar} improvement of 6% while maintaining a similar secret size.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the partners within the 3DFACE project, a European Integrated Project funded under the European Commission IST FP6 program.

REFERENCES

[1] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *6th ACM Conference on Computer and Communications Security*, November 1999, pp. 28–36.

[2] E. J. C. Kelkboom, B. Gökberk, T. A. M. Kevenaar, A. H. M. Akkermans, and M. van der Veen, ""3D face": Biometric template protection for 3d face recognition," in *Int. Conf. on Biometrics*, Seoul, Korea, August 2007, pp. 566–573.

[3] T. A. M. Kevenaar, G.-J. Schrijen, A. H. M. Akkermans, M. van der Veen, and F. Zuo, "Face recognition with renewable and privacy preserving binary templates," in *4th IEEE workshop on AutoID*, Buffalo, New York, USA, October 2005, pp. 21–26.

[4] P. Tuyls, A. H. M. Akkermans, T. A. M. Kevenaar, G.-J. Schrijnen, A. M. Bazen, and R. N. J. Veldhuis, "Pratical biometric authentication with template protection," in *5th International Conference, AVBPA*, Rye Brook, New York, July 2005.

[5] A. Juels and M. Sudan, "A fuzzy vault scheme," in *Proc. of the 2002 International Symposium on Information Theory (ISIT 2002)*, Lausanne, 2002.

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

[7] E.-C. Chang and S. Roy, "Robust extraction of secret bits from minutiae," in *Int. Conf. on Biometrics*, Seoul, South Korea, August 2007, pp. 750–759.

[8] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong secret keys from biometrics and other noisy data," in *Advances in Cryptology - Eurocrypt 2004, LNCS 3027*, 2004, pp. 532–540.

[9] 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.

[10] A. A. Ross, K. Nandakumar, and A. K. Jain, *Handbook of Multibiometrics*, D. D. Zhang and A. K. Jain, Eds. Springer, 2006.

[11] K. Nandakumar and A. K. Jain, "Multibiometric template security using fuzzy vault," in *International Conference on Biometrics: Theory, Applications and Systems*, 2008, pp. 1–6.

[12] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *IEEE CVPR*, vol. 2, June 2005, pp. 454–461.

[13] J. Breebaart, C. Busch, J. Grave, and E. Kindt, "A reference architecture for biometric template protection based on pseudo identities," in *BIOSIG*, Darmstadt, Germany, September 2008.

[14] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, no. 1, pp. 68–79, March 1960.

[15] P. Tuyls and J. Goseling, "Capacity and examples of template-protecting biometric authentication systems," in *Biometric Authentication Workshop ECCV2004*.

[16] B. Gökberk, M. O. Irfanoglu, and L. Akarun, "3D shape-based face representation and feature extraction for face recognition," *Image and Vision Computing*, vol. 24, no. 8, pp. 857–869, August 2006.

[17] X. Zhou, H. Seibert, C. Busch, and W. Funk, "A 3D face recognition algorithm using histogram-based features," in *Eurographics 2008 Workshop on 3D Object Retrieval*, Crete, Greece, April 2008, pp. 65–71.

[18] Q. Tao and R. N. Veldhuis, "Threshold-optimized decision-level fusion and its application to biometrics," *Pattern Recognition*, vol. 4, no. 5, pp. 823–836, May 2009.