Binary Feature Vector Fingerprint Representation From Minutiae Vicinities

Julien Bringer
Morpho
julien.bringer@morpho.com

Vincent Despiegel
Morpho
vincent.despiegel@morpho.com

***Abstract*— Today, when comparing classical fingerprint matching and more constrained algorithms, like after binary quantization for biometric privacy protection purpose, there is an important gap in terms of performances. Performances of the latter solutions still need to be improved to decrease this gap. The main difficulty comes from the fact that fingerprint captures of the same trait give very different minutiae sets with possibly small overlaps and very different distortions among the different part of the images; and comparison of a stored reference with the fresh captured fingerprint data has to take into account those local variabilities. In this paper, we study a new approach to this problem by exhibiting a way to transform a minutiae set into a quantized feature vector by local comparisons. The encoding of the original fingerprint template is made by matching small minutiae vicinities with a set of representative vicinities. Moreover, the representation achieves the interesting property of self-alignment of the vectors.**

***Index Terms*— Fingerprint, Feature Vector, Binarization, Self-alignment.**

I. Introduction

With the growing use of biometric systems comes the need for privacy protection of biometric data in order to prevent someone to be able to track back the users of the system. Maintaining privacy of users has received a lot of attention during the last decade. To cope with the variability of biometric data, error correcting code methods have been proposed first, with the introduction of secure sketches and related solutions [15], [22], [26]. But it is now known [3], [32] that the security is not well established in practice. Nonetheless, this has opened the path toward quantization of biometric data, which is a useful preliminary step before applying protection techniques. Moreover the simplest the underlying comparison of quantized biometric data is, the easiest the integration into cryptographic techniques will be. For instance, when comparison is simplified to a bit per bit exclusive-or (XOR) between two fixed length binary vectors, [2], [5], [9], [33] suggested to combine XOR or secure sketch operations with homomorphic encryption; solutions which lead to high security properties thanks to the use of provably cryptographic schemes. The need for quantization algorithms holds for identification purpose as well where the user has only to provide a new capture of his biometric trait to be identified. Indeed when dealing with binary vectors, comparison speed can be greatly increased [18], [19]. Even embedding identification into protocols with privacy protection has been suggested [1], [6], [8] thanks to the quantization of biometrics.

This work was sponsored in part by the EU project TURBINE, which is funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nb. ICT-2007-216339.

A. Related Works

Although highlighted by the privacy issues and the suggested solutions, quantization of biometric data is not new. The use of binary vectors with a quite simple comparison, namely a relative Hamming distance (a Hamming distance corresponds to the number of coordinates in which two vectors differ) has been introduced for iris recognition technology in [14] with the concept of iriscodes where the features of iris are represented as binary vectors of length 2048. Below a given threshold for this distance, two iriscodes are assumed as matching ones, otherwise this is a non-matching pair. Various attempts have been made for obtaining similar representation and comparison with other modalities, like for faces and fingerprints, e.g. [4], [7], [13], [21], [24], [34], [37]. For face biometrics for which templates can be already represented as feature vectors, the matter is generally to transform a vector of floats into a vector of bits through some quantization techniques [12], [23], [24] which use multiple samples at enrollment to obtain reliable bits.

However concerning fingerprint, classical representation is based on minutiae set which is an unordered set of characteristic points (ridge endings and bifurcations) with variable length and fingerprint matching is based on geometric methods. More precisely, classical fingerprint comparators are minutiae oriented and their goal is to find the optimal translation and rotation which best superimpose two clouds of oriented points (the minutiae of the search and of the reference fingerprint). Then a score is estimated based on correlation between the two minutiae sets [30]. Hence, for quantization, numerous difficulties come with this, e.g. how to deal with the insertion and deletion of minutiae, and also with misalignment and overlapping issues between two fingerprints.

Various methods for transformation of a fingerprint into a quantized feature vector have been studied in previous works. [21] introduced the concept of fingercode which consists to encode the ridge local direction field around the core of a fingerprint. [34] also suggested a quantization algorithm based on local orientation of ridges. This has been later enhanced in several research papers – for instance [7], [13] – but they are still based on fingerprint patterns and need several samples per user at the enrollment to extract stable binary vectors thanks to a reliable component quantization

978-1-4244-7580-3/10/$26.00 ©2010 IEEE

principle. [18] uses representative fingerprint patterns and constitutes a feature vector by concatenating the match scores between the fresh fingerprint pattern and the representative patterns.

Several techniques based on minutiae have also been investigated. In particular, [36] explained the construction of a feature vector of floats via the spectral representation of a minutiae set. Starting with a grid of minutiae, [16] designed specific encoding and decoding algorithms to deal with displacement, erasure and insertion of minutiae. In [29], numbers of minutiae into local cuboids are used for binary representation. Unfortunately, these methods suffer from misalignment problem and need either exhaustive search on the orientation/translation (only orientation for spectral minutiae) or a preliminary registration step before comparison. [17] suggested the use of histogram of minutiae triplets to overcome this issue, but adaptation of the technique to n-tuples ($n > 3$) to increase the discriminative power is too costly (length of the vector equals the number of possible n-tuples).

To sum up, some techniques are built on Ridge Flow Matrix recognition (globally or only around some points) which is less discriminative than the minutiae of a fingerprint. For better performances, we will look in this paper for local comparisons with representative minutiae vicinities. Dealing only with minutiae has also the advantage to be compliant with most of existing fingerprint suppliers and databases. Based or not on minutiae, the methods are very often not resilient to misalignment and need a registration step that may lead to private information leakage. Multi-sample enrollment is also one usual constraint for binary feature vectors that should be made as flexible as possible. And finally existing performances – while dealing with simple matching of quantized fingerprint feature vectors – still need to be improved to come closer to classical minutiae-based matching.

B. Our Work

To overcome these issues, we suggest a new feature vector construction for fingerprints when encoded as standard [20] minutiae sets. Our method is neighborhood oriented. The goal is to encode a fingerprint according to its distances to several representative minutiae neighborhoods. Thanks to these distances and to statistics learned on the representative neighborhoods, a fixed size binary feature vector is computed. One main advantage is that all the computation efforts are concentrated on the feature extraction process whereas the matching process is almost reduced to a simple AND between two fixed size binary feature vectors. This is of great importance for future integration into cryptographic protocols.

The paper is organized as follows. The vicinity definition, the representation, and the algorithm to compare two vicinities are described in Section II. To underline the interest of our strategy we also introduce an algorithm for matching two sets of vicinities and evaluate the performances. Section III explains how representative vicinities are selected and the way to transform a set of vicinities into a binary feature vector. Section IV illustrates the performances of our construction on several public datasets and Section V details the extension of our technique to multi-samples scenario. Section VI concludes.

II. Vicinities of Minutiae

Although our construction of minutiae vicinities is inspired from previous works on the subject, which are numerous (see for instance the recent paper [10] that designs a lightweight representation for fast comparison), we design our vicinities and the related matching algorithm from scratch to be fully compliant with our constraints (in particular to reach good performances with a simple comparison between binary feature vectors). The description follows.

A. Vicinities Construction

A fingerprint will be characterized in all the paper by its minutiae set according to ISO standard [20]. Given a set of minutiae extracted from a fingerprint image, i.e. a set of oriented points (x, y, θ), we consider local neighborhoods around each minutia. These neighborhoods, called vicinities in the sequel, are in fact the core elements of all our feature vector construction. One such vicinity around one minutia point m is defined as the set of all the minutiae which are within a disk of some radius ρ in $\mathbb{R}^2$ and with center the point m.

Moreover the minutia m, called the central minutia of the vicinity, is used to defined a new coordinate system for the position and for the orientation (cf. Figure 1). The position of m gives the new center of the coordinate system and the orientation of m gives the direction of the x-axis. The coordinates of the vicinity points are expressed in this new system. Thanks to this absolute representation, two vicinities are directly comparable without any prerequisite registration for relative realignment.

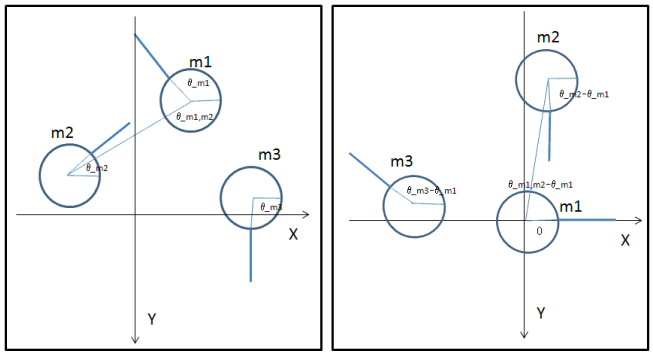

Fig. 1. Vicinity coordinate system

Definition 1: Let $\mathcal{M} = (m_1, \ldots, m_n)$ be a list of n minutiae obtained from a fingerprint, with $m_i = (x_{m_i}, y_{m_i}, \theta_{m_i})$.

The vicinity of radius ρ centered around the minutia m_i is denoted $V_i(\mathcal{M})$ and defined as

$$V_i(\mathcal{M}) = \Big\{ \big(d(m_i, m_j)\cos(\alpha_{m_i,m_j}), -d(m_i, m_j)\sin(\alpha_{m_i,m_j}), \beta_{m_i,m_j}\big) \mid \forall m_j \in \mathcal{M} - \{m_i\} \text{ s.t. } d(m_i, m_j) < \rho \Big\}$$

where $\alpha_{m_i,m_j} = \theta_{m_i,m_j} - \theta_{m_i}$, $\beta_{m_i,m_j} = \theta_{m_j} - \theta_{m_i}$, $d(m_i, m_j)$ is the euclidean distance in $\mathbb{R}^2$ between the minutiae m_i and m_j and θ_{m_i,m_j} the angle between the vector $(m_i, \vec{m_j})$ and the x-axis.

The formula above corresponds to the change of coordinates to the coordinate system defined by the central minutia. The position of the central minutia is the origin of the new coordinate system and the orientation of the minutia defines the orientation of the x-axis.

Instead of characterizing a fingerprint by a global set of minutiae, the approach chosen here is to represent it by several local sets of minutiae. For a set of n minutiae, as many vicinities are constructed. With two sets of vicinities coming from two fingerprints to be compared, the matching step can consist in locally comparing vicinities pairwise and to estimate then the score via the local scores of the best pairings. Although the representation is equivalent to a set of minutiae at the encoding step, the above rough matching procedure is yet less complex than usual minutiae comparison. In fact, our goal is to simplify at most the matching phase thanks to the use of our vicinities (cf. Section III for further simplifications).

In this rough matching procedure, we also see that the global coherency in the minutiae set is not used. Nevertheless the local approach has the advantage of limiting one crucial problem in fingerprint matching: the elastic distortion. Many authors have proposed methods to cope with the elastic distortion of the skin [11], [31]. In the local area of the vicinity, the distortion due to the elasticity of the skin can be considered as negligible. Of course there is a trade-off on the radius of a vicinity so that it contains several minutiae in order to be sufficiently discriminative while staying small enough to be considered as a local area.

Moreover as already mentioned, as vicinities are self-aligned through their central minutia, this solves the misalignment problem between two fingerprints.

B. Comparison of Vicinities

Comparing two vicinities is simplified as the two minutiae clouds are already in the same coordinate system, so the translation and rotation which best superimpose the two neighborhoods have no need to be found.

Let $V = \{p_1, \ldots, p_{l_V}\}$ and $V' = \{q_1, \ldots, q_{l_{V'}}\}$ be two vicinities of minutiae (either from the same fingerprint or from different fingerprints) to be compared. Each point p_i (resp. q_j) is given with coordinates (x, y, θ) in $\mathbb{R}^3$ with respect to the coordinate system associated to V (resp. V'). We proceed as follows.

- As the neighborhoods are aligned, we defined directly a pairing score between minutiae:

$$\mu(p_i, q_j) = e^{-\frac{(x_{p_i} - x_{q_j})^2}{\sigma_X^2}} e^{-\frac{(y_{p_i} - y_{q_j})^2}{\sigma_X^2}} e^{-\frac{(\theta_{p_i} - \theta_{q_j})^2}{\sigma_\theta^2}}$$

  with σ_X the parameter chosen for the variance on the position, and σ_θ on the orientation. Examples of matching vicinities are used to determine the standard deviation of the minutiae in order to tune the parameters σ_X and σ_θ.
- By computing pairing scores $\mu(p_i, q_j)$ between each possible pair of points, we obtain a pairing matrix M of size $l_V \times l_{V'}$ with all possible combinations.
- Now, we can use a closest neighbor search algorithm, which outputs a list of potential minutiae pairings, combined with a post-processing algorithm to avoid multiple pairing for the same minutia. An alternative approach is the classical hungarian algorithm [25] to directly output the best associations. Let $f : V \to V'$ be the related association function defined by $f(p_i) = \{q_j\}$, if p_i is associated with q_j, $f(p_i) = \emptyset$, otherwise (no association to p_i).

Note that in Section IV, we present performances based on the use of the hungarian algorithm.

The matching score between the two vicinities is computed as follows. Let NA_S be the number of non associated minutiae of the vicinity V' and NA_R be the number of non associated minutiae of the vicinity V. We have, $NA_R(V, V', f) = \sharp(V \cap f^{-1}(\emptyset))$ and $NA_S(V, V', f) = \sharp V' - \sharp f(V)$. The global matching score between V and V' is defined by:

$$Score(V, V') = \sum_{\substack{p_i \in V, q_j \in V', \\ f(p_i) = \{q_j\}}} \mu(p_i, q_j) - (NA_R(V, V', f) + NA_S(V, V', f)) K_{NA}$$

where K_{NA} is a penalty coefficient for non associated minutiae. The formula takes into account the number of non associated minutiae to normalize the behavior of neighborhood with various number of minutiae. It avoids the awkward situation where a vicinity with a large number of minutiae matches with too many vicinities or when 2 almost empty neighborhoods have a low score even if they match perfectly.

C. Fingerprint Matching Through Vicinities

We now describe how to use comparison of vicinities in order to match two fingerprints which are represented by a set of vicinities. The principle we use is similar to the basic algorithm of [10]. We construct a pairing matrix containing scores between vicinities of the first fingerprint FP and the vicinities of the second one FP'. These scores are computed using the method described in the previous section. Based on the pairing matrix, we apply again an hungarian algorithm to select the best associations of vicinities. And a final score between the fingerprints is computed with similar formula as previously.

Our goal here is to measure the discriminative power of our vicinity construction compared to classical minutiae matching. The performances are evaluated on the FVC2002 database 2 [27] which is an optical sensor database with 800 images (8 acquisitions times 100 subjects). Note that ρ is chosen such that a vicinity covers about 5% of the whole fingerprint image. On FVC2002 DB2, the Equal Error Rate (EER) measured for the above matching algorithm is 2.4%. For a False Accept Rate (FAR) of 10^{-3}, the False Reject Rate (FRR) is 5.8%. The EER is of course not at the state

of the art performances [27], but not so far, which confirms the quality of the vicinities.

III. From Vicinities to Binary Feature Vectors

To transform a fingerprint represented as a set of vicinities into a feature vector of a given length N, we build a projecting space containing N representative vicinities. The vicinities of the fingerprint would be then compared to each representative vicinities to obtain the feature vector.

A. Representative Database of Vicinities

Starting with a large external fingerprint database, we extract all the vicinities of the fingerprints in the database and determine a subset of representative vicinities. For this, various criteria are used to shorten the list of vicinities:

1) sparse neighborhoods (vicinity with less than a number l_{min} of minutiae) are discarded;
2) similar neighborhoods are discarded (a neighborhood is not kept if it has a matching score greater than $score_{max}$ with a selected neighborhood).

After shortening the list of vicinities, we obtain a representative dataset DB_R with N representative vicinities, denoted R_i.

The radius ρ of the vicinities (cf. Definition 1) represents a trade-off between performances and the number N of needed vicinities to densely span the space. If ρ is small, one does not need many representative neighborhoods to span the space of all existing vicinities. On the opposite, if ρ is big, a vicinity is very discriminative and a huge number of representative vicinities would be needed.

B. Discriminative Power of Representative Vicinities

As explained, a fingerprint will be compared to all representative vicinities to fill the feature vector, so it is necessary to normalize the score computed with respect to each representative vicinity. This normalization is not limited to the number of non associated minutiae, that is already taken in account (cf. Section II-B). There are lots of differences between the scores dynamic of the R_i. Some neighborhoods extracted in low curvature area are commonly found in fingerprints and statistically match well with many vicinities. On the contrary, some areas of high curvature are rare in fingerprints and have very few matching vicinities.

To determine whether an R_i is rare or common in the vicinities world, we extract another set of vicinities from a fingerprint dataset DB'. Those vicinities are compared to each R_i using the matching procedure presented in Section II-B to build a histogram of scores. Its dynamic indicates if the minutiae configuration of R_i is common or rare in fingerprints.

Let p be a probability value. Based on the histograms, we are able to estimate the value $\tau_i(p)$ such that the probability for a random vicinity V of DB' to obtain a score $Score(V, R_i)$ greater than τ_i is approximately equal to p. This is used to set binarization thresholds in next section.

Figure 2 illustrates the histograms for the four representative vicinities of Figure 3. We see that the vicinities have different matching score repartitions. Vicinity 1 is the most discriminative one among the four, it corresponds to a high curvature area with many minutiae. Vicinity 4 is the least discriminative: the curvature is very low.

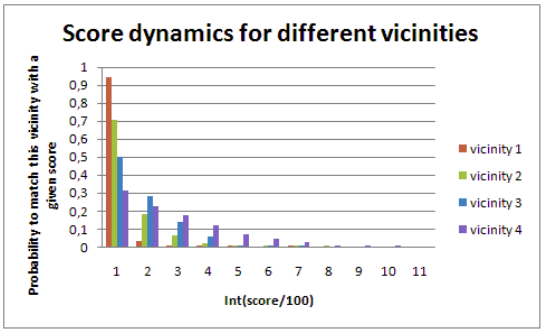

Fig. 2. Vicinities Matching Score Repartition

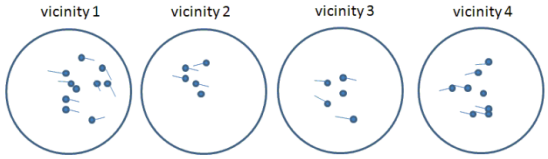

Fig. 3. Example of Vicinities

C. Binary Feature Vector

Let $\mathcal{M}$ be a list of n minutiae obtained from a fingerprint. The binary feature vector $FV(\mathcal{M})$ is elaborated as follows (see an overview in Figure 4):

- All the vicinities of the fingerprint are extracted: $\mathcal{V} = \{V_1(\mathcal{M}), \ldots, V_n(\mathcal{M})\}$.
- For $i \in \{1, \ldots, N\}$, we compute the matching scores of all the fingerprint vicinities with R_i:

$$Score(V_j(\mathcal{M}), R_i)\,(j \in \{1, \ldots, n\})$$

- Let $T(\mathcal{M})$ be the vector of length N with the i-th coordinate defined by $T(\mathcal{M})_i = Score(V_{a[i]}(\mathcal{M}), R_i)$ where $a[i] = argmax_{j \in \{1,\ldots,n\}} Score(V_j(\mathcal{M}), R_i)$.
- For a given probability p, and the related values $\tau_i(p)$ $(i \in \{1, \ldots, N\})$, the previous vector is binarized to obtain $FV(\mathcal{M})$:

$$FV(\mathcal{M})_i = \begin{cases} 1, & \text{if } T(\mathcal{M})_i > \tau_i(p) \\ 0, & \text{otherwise} \end{cases}$$

The value $\tau_i(p)$ are used above as binarization thresholds. The probability p should be small (for instance 1%) to ensure that a bit set to 1 actually means that the associate representative vicinity R_i is close to one of the fingerprint vicinities. It is also necessary to avoid almost empty feature vectors which will be non-resilient to noise. As a rough estimation, the number of 1 in the binary feature vector would be in average $p \times N \times n$.

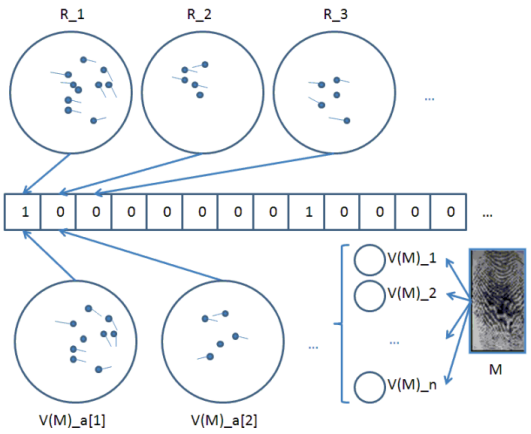

Fig. 4. Overview of the Binarization

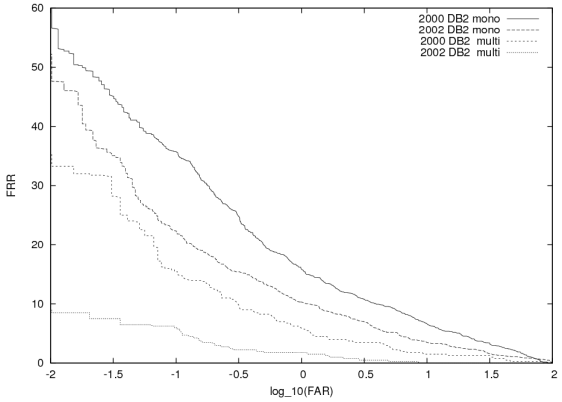

Fig. 5. performances on FVC2000 DB2, FVC2002 DB2

D. Binary Feature Vectors Matching

The score between two binary feature vectors $FV(\mathcal{M})$ and $FV(\mathcal{M}')$ is computed by counting the number of 1 at the same coordinate, normalized by the minimum Hamming weight of the vectors and some penalty constant K:

$$Sc(FV(\mathcal{M}), FV(\mathcal{M}')) = \frac{\sum_{i=1}^{N}(FV(\mathcal{M})_i \text{ AND } FV(\mathcal{M}')_i)}{K+\min(\sum_{i=1}^{N} FV(\mathcal{M})_i, \sum_{i=1}^{N} FV(\mathcal{M})_i)}$$

IV. Performances

We evaluated the performances of our construction on the two following public databases: 1/ FVC2000 DB2 [28], a low-cost capacitive sensor database with 800 images (8 acquisitions times 100 subjects); 2/ FVC2002 DB2 [27], an optical sensor database with 800 images (8 acquisitions times 100 subjects).

Using several different databases acquired with different type of sensors increase the confidence on the results of the binary feature vector representation. The vicinity comparison algorithm is the hungarian version and the size of the feature vectors N is 50 000 bits (for this, as explained in Section III, we use N=50 000 representative vicinities chosen thanks to a large external fingerprint database with an empty intersection with the FVC datasets). The Detection Error Trade-off curves are given in Figure 5. For instance, on FVC2002 DB2, the EER is of 5.3%[1] and a FRR of 22% at 10^{-3} FAR. As the templates are already self aligned, the results are invariant to rotation and translation of the fingerprints. The difference of performances on the databases can be explained: The neighborhood approach works well on good quality images even if those images are small. On the contrary, bad quality images mean spurious minutae which degrade the vicinities matching performances.

V. Multi-Enrollment

Multi-samples enrollment scenarios are quite common: during the enrollment process, 3 or 4 acquisitions are often captured by the operator. With our techniques, these acquisitions can be easily consolidate to build one simple binary feature vector.

Let $(T(\mathcal{M}^1), \ldots, T(\mathcal{M}^k))$ the k feature vectors calculated – before binarization (cf. Section III-C) – from k acquisitions of the same fingerprint. To fuse those vectors and to be as conservative as possible in regards to the possible few overlap of the different acquisitions, we defined the fused feature vector $T(\mathcal{M}^1, \ldots, \mathcal{M}^k)$ by:

$$T(\mathcal{M}^1, \ldots, \mathcal{M}^k)_i = \max(T(\mathcal{M}^1)_i, \ldots, T(\mathcal{M}^k)_i)$$

for $i \in \{1, \ldots, N\}$.

The binarization of this vector is done similarly as in Section III-C. A probability p' possibly depending on k and different from the previous binarization probability is used to generate the binarized vector via:

$$FV(\mathcal{M}^1, \ldots, \mathcal{M}^k)_i = \begin{cases} 1, & \text{if } T(\mathcal{M}^1, \ldots, \mathcal{M}^k)_i > \tau_i(p') \\ 0, & \text{otherwise} \end{cases}$$

The size of the multi-acquisition template remains the same as for mono-acquisition. And, as for classical matching [38], the performances are improved in multi-acquisition scenario. The performances with 4 samples at enrollment are given on FVC2000 DB2 and FV2002 DB2 in Figure 5.

The Detection Error Trade-off curves are given in Figure 5. For instance, on FVC2002 DB2, the EER is of 1.7% and a FRR of 8% at 10^{-3} FAR. As for mono-acquisition scenarii, the bad quality of the images is the reason for the difference of performances between the databases.

VI. Conclusion

We have proposed in this article a new method to transform a minutiae set into a fixed size binary feature vector. The vicinity approach has the advantage to give the self-alignement property to the feature vectors for free. The results obtained both in mono and multi acquisition scenario are extremely promising. The fact that all the heavy computations are done during the binary feature vector construction process and that the matching is extremely simple is another huge advantage of the method proposed which should be maintained in further works.

[1]This is for instance close – although a bit higher – to the 4% of EER in [35] but which is obtained with non-binary fixed-length vectors.

To improve this construction, we can look on ways to incorporate a global coherency measure and to reduce the length of the feature vectors. Application to identification scenario for high speed filtering algorithm could also be investigated.

REFERENCES

[1] Michael Adjedj, Julien Bringer, Hervé Chabanne, and Bruno Kindarji. Biometric identification over encrypted data made feasible. In Atul Prakash and Indranil Gupta, editors, *ICISS*, volume 5905 of *Lecture Notes in Computer Science*, pages 86–100. Springer, 2009.

[2] Julien Bringer and Hervé Chabanne. An authentication protocol with encrypted biometric data. In Serge Vaudenay, editor, *AFRICACRYPT*, volume 5023 of *Lecture Notes in Computer Science*, pages 109–124. Springer, 2008.

[3] Julien Bringer, Hervé Chabanne, Gérard Cohen, Bruno Kindarji, and Gilles Zémor. Optimal iris fuzzy sketches. In *Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on*, Sept 2007.

[4] Julien Bringer, Hervé Chabanne, Gérard D. Cohen, Bruno Kindarji, and Gilles Zémor. Theoretical and practical boundaries of binary secure sketches. *IEEE Transactions on Information Forensics and Security*, 3(4):673–683, 2008.

[5] Julien Bringer, Hervé Chabanne, Malika Izabachène, David Pointcheval, Qiang Tang, and Sébastien Zimmer. An application of the goldwasser-micali cryptosystem to biometric authentication. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, *ACISP*, volume 4586 of *Lecture Notes in Computer Science*, pages 96–106. Springer, 2007.

[6] Julien Bringer, Hervé Chabanne, Tom A. M. Kevenaar, and Bruno Kindarji. Extending match-on-card to local biometric identification. In Julian Fiérrez-Aguilar, Javier Ortega-Garcia, Anna Esposito, Andrzej Drygajlo, and Marcos Faúndez-Zanuy, editors, *COST 2101/2102 Conference*, volume 5707 of *Lecture Notes in Computer Science*, pages 178–186. Springer, 2009.

[7] Julien Bringer, Hervé Chabanne, and Bruno Kindarji. The best of both worlds: Applying secure sketches to cancelable biometrics. *Sci. Comput. Program.*, 74(1-2):43–51, 2008.

[8] Julien Bringer, Hervé Chabanne, and Bruno Kindarji. Error-Tolerant Searchable Encryption. In *International Conference on Communications*, June 2009.

[9] Julien Bringer, Hervé Chabanne, David Pointcheval, and Qiang Tang. Extended private information retrieval and its application in biometrics authentications. In Feng Bao, San Ling, Tatsuaki Okamoto, Huaxiong Wang, and Chaoping Xing, editors, *CANS*, volume 4856 of *Lecture Notes in Computer Science*, pages 175–193. Springer, 2007.

[10] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(PrePrints), 2010.

[11] Raffaele Cappelli, Dario Maio, and Davide Maltoni. Modelling plastic distortion in fingerprint images. In *ICAPR '01: Proceedings of the Second International Conference on Advances in Pattern Recognition*, pages 369–376, London, UK, 2001. Springer-Verlag.

[12] Chun Chen, Raymond N. J. Veldhuis, Tom A. M. Kevenaar, and Anton H. M. Akkermans. Biometric binary string generation with detection rate optimized bit allocation. In *IEEE CVPR 2008, Workshop on Biometrics*, pages 1–7, June 2008.

[13] Chun Chen and Raymond N. J. Veldhuis. Binary biometric representation through pairwise polar quantization. In Massimo Tistarelli and Mark S. Nixon, editors, *ICB*, volume 5558 of *Lecture Notes in Computer Science*, pages 72–81. Springer, 2009.

[14] John Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11):1148–1161, 1993.

[15] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer, 2004.

[16] Stark Draper, Jonathan Yedidia, Ashish Khisti, Emin Martinian, Anthony Vetro. Using distributed source coding to secure fingerprint biometrics. In *in Int. Conf. Acoutics Speech Signal Proc*, pages 129–132, 2007.

[17] Faisal Farooq, Ruud M. Bolle, Tsai-Yang Jea, and Nalini K. Ratha. Anonymous and revocable fingerprint recognition. In *CVPR*. IEEE Computer Society, 2007.

[18] Aglika Gyaourova and Arun Ross. A novel coding scheme for indexing fingerprint patterns. In Niels da Vitoria Lobo, Takis Kasparis, Fabio Roli, James Tin-Yau Kwok, Michael Georgiopoulos, Georgios C. Anagnostopoulos, and Marco Loog, editors, *SSPR/SPR*, volume 5342 of *Lecture Notes in Computer Science*, pages 755–764. Springer, 2008.

[19] Feng Hao, John Daugman, and Piotr Zielinski. A fast search algorithm for a large fuzzy database. *IEEE Transactions on Information Forensics and Security*, 3(2):203–212, 2008.

[20] ISO/IEC 19794-2:2005. Information technology, biometric data interchange formats, part 2: Finger minutiae data. Technical report, ISO/IEC, 2005.

[21] Anil K. Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti. Fingercode: A filterbank for fingerprint representation and matching. In *CVPR*, pages 2187–. IEEE Computer Society, 1999.

[22] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security*, pages 28–36, 1999.

[23] Emile J.C. Kelkboom, Gary G. Molina, Tom A.M. Kevenaar, Raymond N.J. Veldhuis, and Willem Jonker. Binary biometrics: An analytic framework to estimate the bit error probability under gaussian assumption. In *IEEE BTAS 2008*, pages 1–6, 2008.

[24] Tom A. M. Kevenaar, Geert J. Schrijen, Michiel van der Veen, Anton H. M. Akkermans, and Fei Zuo. Face recognition with renewable and privacy preserving binary templates. *Automatic Identification Advanced Technologies, IEEE Workshop on*, 0:21–26, 2005.

[25] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[26] Jean-Paul M. G. Linnartz and Pim Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In Josef Kittler and Mark S. Nixon, editors, *AVBPA*, volume 2688 of *Lecture Notes in Computer Science*, pages 393–402. Springer, 2003.

[27] Dario Maio, Davide Maltoni, Raffaele Cappelli, James L. Wayman, and Anil K. Jain. FVC2002: Second fingerprint verification competition. *Pattern Recognition, International Conference on*, 3:30811, 2002.

[28] Dario Maio, Davide Maltoni, Raffaele Cappelli, James L. Wayman, and Anil K. Jain. FVC2000: Fingerprint verification competition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):402–412, 2002.

[29] Abhishek Nagar, Shantanu Rane, and Anthony Vetro. Alignment and bit extraction for secure fingerprint biometrics. In *SPIE Conference on Electronic Imaging 2010*, 2010.

[30] Salil Prabakhar, Anil K. Jain, Dario Maio, and Davide Maltoni. *Handbook of Fingerprint Recognition*. Springer-Verlag New York, Inc., 2003.

[31] Arun Ross, Sarat Dass, and Anil Jain. A deformable model for fingerprint matching. *Pattern Recogn.*, 38(1):95–103, 2005.

[32] Koen Simoens, Pim Tuyls, and Bart Preneel. Privacy weaknesses in biometric sketches. In *IEEE Symposium on Security and Privacy*, pages 188–203. IEEE Computer Society, 2009.

[33] Alex Stoianov. Cryptographically secure biometrics. In *SPIE Biometric Technology for Human Identification VII*, volume 7667, 2010.

[34] Pim Tuyls, Anton H. M. Akkermans, Tom A. M. Kevenaar, Geert Jan Schrijen, Asker M. Bazen, and Raymond N. J. Veldhuis. Practical biometric authentication with template protection. In Takeo Kanade, Anil K. Jain, and Nalini K. Ratha, editors, *AVBPA*, volume 3546 of *Lecture Notes in Computer Science*, pages 436–446. Springer, 2005.

[35] Haiyun Xu and Raymond N. J. Veldhuis. Spectral minutiae representations of fingerprints enhanced by quality data. In *Biometrics: Theory, Applications, and Systems, 2009. BTAS '09. IEEE 3rd International Conference on, Washington DC, USA*, pages 1–5, Washington DC, September 2009. IEEE Computer Society Press.

[36] Haiyun Xu, Raymond N.J. Veldhuis, Tom A.M. Kevenaar, Anton H.M. Akkermans, and Asker M. Bazen. Spectral minutiae: A fixed-length representation of a minutiae set. *Computer Vision and Pattern Recognition Workshop*, 0:1–6, 2008.

[37] Bian Yang, Christoph Busch, Patrick Bours, and Davrondzhon Gafurov. Robust minutiae hash for fingerprint template protection. volume 7541. SPIE, 2010.

[38] Chunyu Yang and Jie Zhou. A comparative study of combining multiple enrolled samples for fingerprint verification. *Pattern Recogn.*, 39(11):2115–2130, 2006.