Learning Minutiae Neighborhoods : A New Binary Representation for Matching Fingerprints

Akhil Vij and Anoop Namboodiri

International Institute of Information Technology

Hyderabad, 500032, India

`akhil.vij@research.iiit.ac.in, anoop@iiit.ac.in`

Abstract

Representation of fingerprints is one of the key factors that limits the accuracy and efficiency of matching algorithms. Most popular methods represent each fingerprint as an unordered set of minutiae with variable cardinality and the matching algorithms are left with the task of finding the best correspondence between the two sets of minutiae. While this makes the representation more flexible and matching more accurate, the task becomes computationally intensive. Fixed length representations with aligned features are highly efficient to match. However, creating an aligned representation without the knowledge of the sample to which it is to be matched, makes the problem of representation more complex. Some of the fixed-length representations only provide partial alignment, leaving the rest to the matching stage. In this paper, we propose a fixed length representation for fingerprints that provides exact alignment between the features, thus enabling high-speed matching with minimal computational effort. The representation extends the idea of object representation using bag of words into a bag of minutiae neighborhoods. The representation is provably invariant to affine transformations (rotation, translation and uniform scaling), and is shown to be highly discriminative for the task of verification. Experimental results on FVC 2002 and 2004 datasets clearly show the superiority of the representation with competing methods. As the proposed representation can be computed from the standard minutiae templates, the method is applicable to existing datasets, where the original fingerprint images are not available.

1. Introduction

The identification of people by measuring some physiological or behavioral traits has led to the emergence of biometrics as a prominent research field in recent years. Several biometric technologies have been developed and suc-cessfully deployed around the world : fingerprints, face, iris, signature etc. Out of all biometric traits, fingerprints are the most popular because of their ease of capture, distinctiveness and persistence over time[1], as well as cost and maturity of products. As fingerprint sensors are becoming cheaper and smaller, in addition to military applications, a wide range of civilian applications such as passport control, border crossings, national identity projects, driver licences, fingerprint based smart cards etc. are using fingerprints as a primary trait for identifying people.

Although what we get from a fingerprint sensor is usually a grayscale image of some resolution, only a few fingerprint recognition algorithms work directly on the grayscale image. Before the matching stage, most of the algorithms have a pre-processing or a feature extraction stage where useful information is extracted from the fingerprint. This information is then stored in databases and is known as the representation for fingerprints. A good quality fingerprint representation should be robust to distortions, have small storage size, should be able to handle noisy images, should be easy to extract automatically from images and it should be easy to match two representations. Most of the above fingerprint authentication systems would benefit from a fixed-length binary representation of a fingerprint that has the above qualities. Many effective representations have been proposed in the literature. Based on features extracted and stored, the traditional fingerprint representation schemes can be classified as :

- **Global Features based Representation** : These representations include global ridge-line frequency, core points, orientation images, singular points etc. These features represent the global pattern of the ridges in the fingerprint. One disadvantage of these representations is that they cannot be easily extracted from poor quality fingerprints. Also, these representations do not offer good individual discrimination and are not good at handling distortions. Further, such representations cannot handle small local non-linear distortions and

there is no standard definition for most of these features leading to compatibility issues with most of the existing fingerprint databases.

- **Local Features based Representation** : The local approach refers to representing the fingerprint in the terms of minutiae sets, local ridge orientations and local ridge frequency. These local representations are quite distinctive and generally outperform their global counterparts. Ross in his work [2], uses representative local fingerprint patterns to construct a feature vector. Tuyls in his work [3] proposed a novel quantization algorithm to get fixed length representation based on local orientation of ridges. Minutiae based representations are the most popular as they are compliant with most of the existing fingerprint suppliers and databases and have small template size. As per the ISO/IEC 19794-2 minutiae template [4], each minutia m is a triplet $m=\{x_m,y_m,\theta_m\}$ where x_m and y_m are the minutia location, θ_m is the minutia direction (in the range [0,2π]). A fingerprint is stored as Fp $= \{m_1,m_2,m_3....m_n\}$ a collection of minutiae points. The basic disadvantage is that two impressions from the same finger can have different number of minutiae leading to a variable length descriptions. Another disadvantage is that the representation suffers from misalignment problem and require a preliminary registration step. Typical approach is to align the unregistered minutiae sets (having different sizes) of the two fingerprints and then find the overlap and similarity. Also, a minutiae based representation due to its variable size is not suitable for recently proposed template protection schemes such as [3] and [5]. Bringer in his work [6], transforms a minutiae set into a fixed-length quantized feature vector by matching small minutiae vicinities (or neighborhoods) with a set of representative vicinities.

- **Combination of Local and Global** : These schemes combine the local and global information present in a fingerprint. Fingercode proposed by Jain[7] , utilizes both local and global ridge descriptors and texture information. It is a fixed *640 byte* representation that is extracted by tessellating the image around the core point. The feature vector consists of an ordered collection of texture descriptors from various sectors of the tessellation. The disadvantage of fingercode is that it requires the core point to be accurately located which in itself is a difficult problem. Sha [8] proposed an improved version of fingercode but the same problem still persists. Benhammadi [9] also proposed a new representation called oriented minutiae codes based on minutiae texture maps. They use the response of eight gabor filters to generate the codes. However, representations based on textures and gabor response are not discriminative enough and are not robust to small local non-linear distortions.

- **Transform based representations** : Tico [10] proposed a *48 byte* length representation using Digital Wavelet Transform (DWT) features. Amornraksa [11] proposed a *24 byte* representation using the Digital Cosine Transform (DCT) features. However, drawback of transform-based representations is that they are not rotation invariant and rotation has to be handled explicitly. This was handled by Xu in his work [12], in which he proposed a spectral minutiae representation based on Fourier-Melin transform. By representing minutiae as a magnitude spectrum, he transforms a minutiae set into a fixed length feature vector. But still the scheme is not very robust to non-linear distortions.

Most of the representations described above either cannot handle global transformations like rotation etc. or are not tolerant towards small local non-linear distortions or are variable in size. This implies that the accuracy of matching using the quantized feature vector representations still is very low as compared to classical minutiae based matching. We need a fixed length (binary prefered) representation that is tolerant towards these distortions, can handle missing/spurious minutiae, is suitable for template protection schemes, small enough to be stored on smart cards and has a minutiae-only construction so that it can be applied to existing databases. In the next section we propose a new local minutiae structure called an arrangement structure that captures the complete geometry of neighboring points around a central minutia. Given a fingerprint database, we extract all the arrangement structures to populate the high-dimensional structure space. We then use *k-means* clustering to cluster this high dimensional space of arrangement structures. From this we get *k* cluster centers, which correspond to the *k* most prominent neighborhood structures learned from the fingerprint database. Then every fingerprint in the database is expressed as a collection of these cluster centers to get a fixed-length (of length *k*) representation for a fingerprint.

2. Representing Local Neighborhoods

We need an affine invariant method of representing all the information in the locality of a minutia point. We believe that there is sufficient information present in the locality of a point that can help us get an aligned representation without any knowledge of the sample to which it is to be matched. Bhanu [13] proved that relative geometric features around the locality of a minutia point are invariant to affine distortions (rotation, translation and uniform scaling). We try to use such local features to come up with an affine invariant representation of each minutia that allows us to
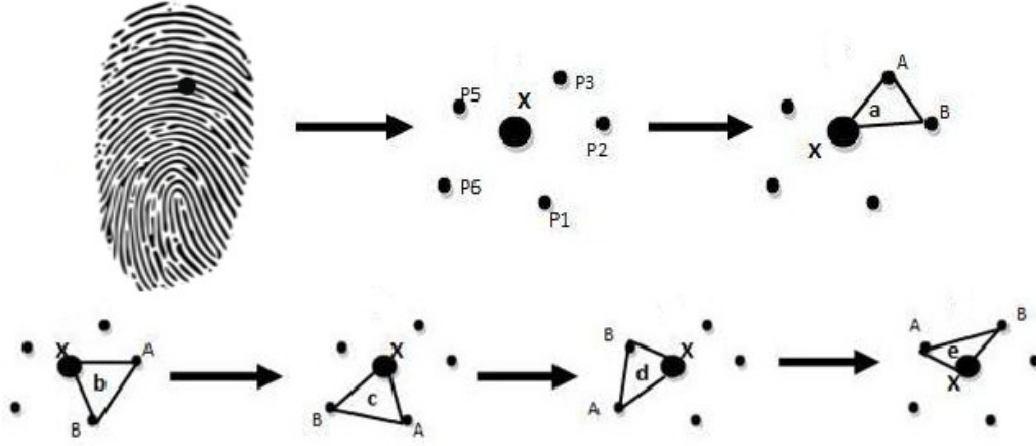
Figure 1. The process of creating an arrangement structure for minutia X. In step1, we find the nearest n minutiae of X. In subsequent steps, we take two points A,B and calculate invariants a,b,c,d,e (see Section 4.3). abcde is the required structure that describes the local neighborhood of central minutia X.
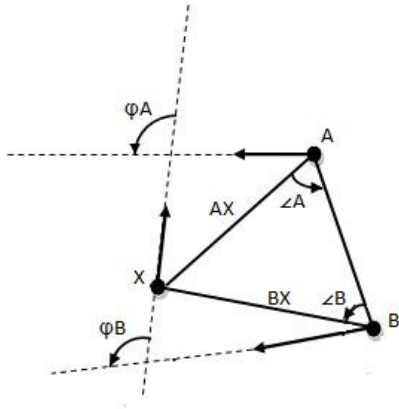

Figure 2. The geometric features computed from $\triangle$AXB. Relative distances AX and BX, Relative Orientations ϕ_A and ϕ_B and angles $\angle$B and $\angle$A

compare two minutiae points and determine their similarity irrespective of the global alignment.

2.1. The Arrangement Structure

Our local structure, called the arrangement structure, is a fixed-length descriptor for a minutia that captures the geometry formed by its neighboring points around that minutia. This distinctive representation of each minutiae allows us to compare two minutiae points and determine their similarity.

The process of calculating the arrangement structure for a minutia (X), shown in Figure 1, is as follows:

- We calculate the nearest n neighbors of minutia X based on their euclidean distances from X. In Figure 1, let $n = 5$, and the nearest minutiae are p1,p2,p3,p5 and p6.

- Starting with the nearest point, we arrange the n points in clockwise order. This is because the clockwise order of minutiae points remains unchanged even when the fingerprint image is rotated, translated, scaled or sheared.

- Now, we describe the local geometry of these n points around the minutia X. As shown in Figure 1, let n=5, and let p3, p2, p1, p6 and p5 be the n minutiae arranged in clockwise order. Now starting with the nearest point and with two points marked as A, B we calculate the following geometric features from $\triangle$AXB as shown in Figure 2:

  - **Relative Distances** : We calculate the euclidean distances between points X and A,B. The first feature is the ratio of these relative distances.

  - **Relative Orientation** : We calculate the orientations of points A,B with respect to the central minutia X (relative orientation of A is the ϕ_A - ϕ_X, where ϕ_A is the orientation of minutia A). The second feature is the ratio of these relative orientations.

  - **Angles of $\triangle$AXB** : The next features we use the angles $\angle$XBA and $\angle$XAB of the $\triangle$AXB. The third feature is the ratio of these angles.

- These features are provably invariant to geometric distortions [13] and remain unchanged even when the fingerprint is translated, rotated, scaled or sheared.
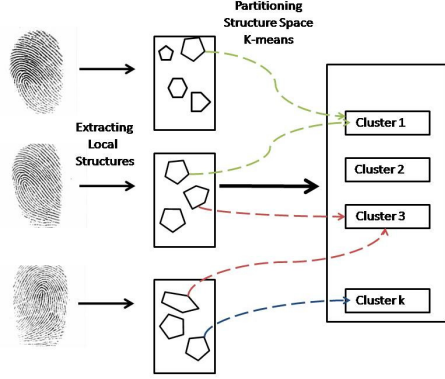
Figure 3. Populating the $n*3$-dimensional structure space. The arrangement structures are extracted from each fingerprint in the database. Then the structure space is partitioned into K clusters via the k-means algorithm.

We concatenate these three features to form a set of cardinality three ($a = [a1,a2,a3]$ as shown in figure 1) that describes contribution of $\triangle AXB$ in the arrangement of these n points around the minutia X. By sliding the points A, B in clockwise rotation, n such invariant sets are calculated (i.e a[a1,a2,a3], b[b1,b2,b3], c[c1,c2,c3], d[d1,d2,d3] and e[e1,e2,e3] in Figure 1). Thus *abcde* is the arrangement structure of length *n*3* that describes the geometric layout of these n points around our central minutia X. The structure *abcde* depends upon the initial choice of points A, B and is not invariant to rotations. To achieve rotation invariance, we use cyclic permutations of this structure. All n cyclic permutations of *abcde* (i.e *bcdea*, *cdeab*, *deabc*, *eabcd* and *abcde*) are calculated and stored in a list as shown in Figure 3. So we generate many $n*3$ dimensional arrangement structures in the learning phase as shown in Algorithm 1, where each structure represents a minutiae neighborhood. Now we use *k-means* to cluster this $n*3$-dimensional space as shown in Figure 3.

3. Representing a Fingerprint

Fingerprints can be seen as non-linearly distorted arrangement of neighborhoods. Our goal is to create an aligned fixed length representation for a fingerprint that is invariant to affine deformations. We use unsupervised clustering to achieve that. *K-means* results in K clusters $c_1,c_2,c_3......c_K$ where each cluster represents set of similar neighborhoods. The centroid of each cluster c_j, represented by m_j can be seen as the mean representative neighborhood for that set of neighborhoods that map to c_j. So, in essence, $m_1,m_2,m_3.......m_K$ are the most prominent neighborhoods learned by our algorithm. Any fingerprint now can be represented in terms of these representative neighborhoods. When a new fingerprint comes, we extract all the
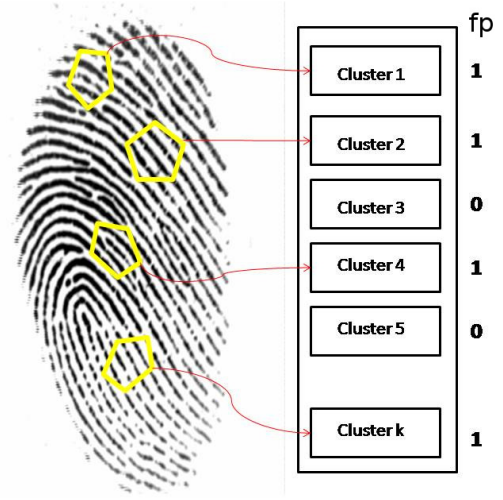

Figure 4. A fingerprint image represented in terms of representative neighborhoods. Given a image, we extract all the neighborhoods and map them to the nearest cluster. *fp* is the K length binary representation of the fingerprint.

neighborhoods from that and map each neighborhood feature vector to its nearest cluster center as shown in Figure 4. So, now each fingerprint is a binary feature vector **fp** of length K where fp_i tells whether a neighborhood similar to m_i is present in the fingerprint or not as shown in Algorithm 2. So, we now visualize fingerprints as a collection of neighborhoods rather than a grayscale image or a minutiae set.

4. Fingerprint Similarity Measure

Now given two binary vectors *fp1* and *fp2*, representing the two fingerprints, a formula based on simple bitwise operations on the two vectors will give a measure of number

---
Algorithm 2 Fixed Length Representation

INPUT → Entire Database db, List of clusters K from
learning phase, n

OUTPUT → Binary Vectors for each fingerprint template
fp in *db*

for all fingerprint template *fp* in *db* **do**

   B → Binary vector of length K initialized with zeros

   for all minutia p in *fp* **do**

      N → nearest n neighbors of minutia p

      find arrangement structure *a*bcde

      Let m_i → nearest cluster center for *a*bcde

      Set $B_i \to 1$

   end for

end for
---

of similar neighborhoods present in them. Thus, simple bit-oriented coding can now be used as a measure for fingerprint similarity. Similarity *s* between two binary vectors, *fp1* and *fp2* is calculated by using the *L2-norm* of the XOR of the two vectors . *L2-norm* is the square root of the number of one bits in the vector.

$$s(fp1, fp2) = 1 - (\|fp1 \; XOR \; fp2\|)/(\|fp1\| + \|fp2\|) \tag{1}$$

5. Experiments and Results

Experiments were conducted on FVC 2002 db1, db2, db3 and FVC 2004 db1 and db2 databases. Each database consists of 800 impressions from 100 different fingers, 8 impressions per finger. The minutiae were extracted using the standard NIST MINDTCT algorithm[14]. First six impressions per finger were used for learning the cluster centers. Then all templates in the databases were converted to their correspondening fixed length representations. The performance evaluation protocol used in FVC 2002 (same as in [15]) has been adopted. Experiments were done for different values of *k* and *n*. The best results were obtained for cluster size of 1000 (i.e k=1000) and neighborhood size of 5 (i.e n=5). A total of 14,000 genuine matches (2800 per database) and 24,750 imposter matches (4950 per database) were done. The ROC curves with different number of clusters have been plotted below. It was observed that the accuracy increased with increase in number of clusters upto an extend and then it started decreasing gradually after 1000 clusters as shown in Figure 8. If the number of clusters is less then the probablity of two different neighborhoods mapping to the same cluster increases lowering the accuracy. On the other hand, if the number of clusters are too high, then two similar neighborhoods can map to different clusters which again will lower the accuracy. So, there has to be an optimal value for number of clusters for which the accuracy is maximized, in our experiments we observed that

the accuracy was maximum for 1000 clusters. The results have been compared (see Figure 7) with spectral minutiae representation [12] and binary representation through minutiae vicinities [6]. These are the two major fixed-length quantized fingerprint representations in the literature. The ROC curves showing the accuracy on FVC 2002 databases (see Figure 6) and FVC 2004 databases (see Figure 5) have been plotted. To genuine-imposter class distribution for FVC 2002 db2 is shown in Figure 9.
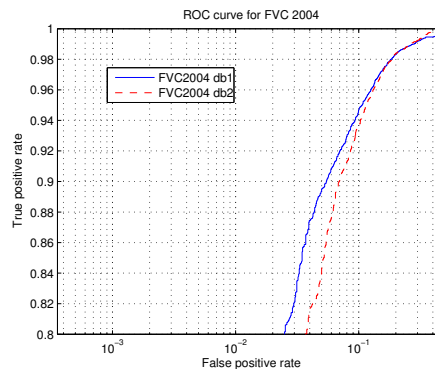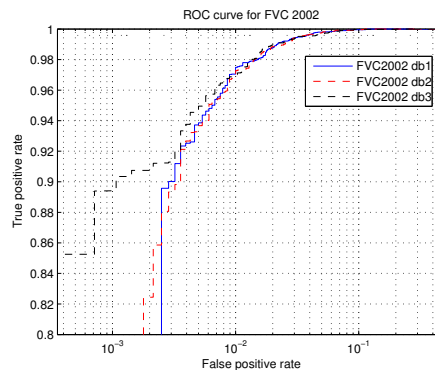

Figure 5. ROC curve for FVC2004 databases

Figure 6. ROC curve for FVC2002 databases

6. Conclusion

We proposed a novel binary fixed-length representation for a fingerprint constructed from minutiae-only features. We captured the local geometry around a minutia point into our local arrangement structure. We then applied unsupervised learning to learn prominent minutiae neighborhoods from the database. A fingerprint was then represented as a collection of neighborhoods resulting in a fixed 1000-length binary representation. The matching of two fingerprints is then reduced to a sequence of bitwise operation which is very quick. Experiments conducted of FVC 2002 and 2004 databases showed the effectiveness of our representation as compared with the major fingerprint representations exist-
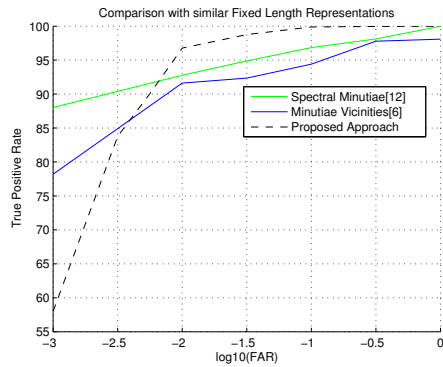
Figure 7. Comparison of the proposed approach with spectral representation[12] and minutiae vicinities [6]. The comparison is done with proposed representation based on 1000 clusters on FVC 2002 db2.
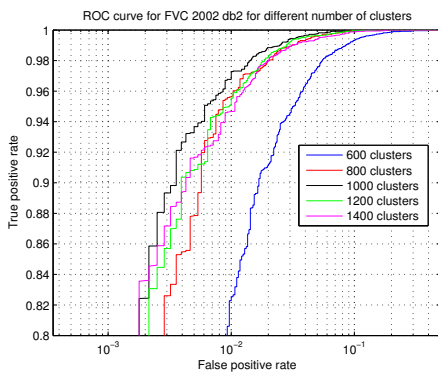

Figure 8. ROC curve for FVC2002 db3 database showing the increase in accuracy with the increase in number of clusters.
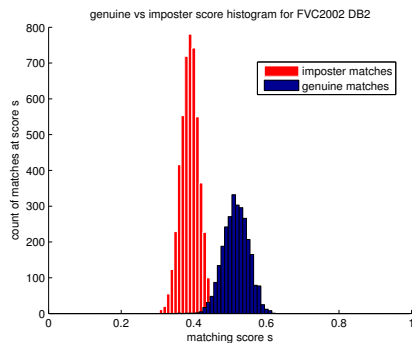

Figure 9. Class Distribution curve for FVC2002 db2 database

ing in the literature. Our representation is tolerant towards distortions, can be stored easily on light architectures such as smart cards and is suitable for biometric template protection schemes.

References

[1] D.Maltoni, D.Maio, A.K.Jain, and S.Prabhakar, *Handbook of Fingerprint Recognition*. Springer, 2009.

[2] A.Gyaourova and A.Ross, "A novel coding scheme for indexing fingerprint patterns," in *Proc. Seventh Int'l Workshop on Statistical Pattern Recognition*, Orlando, FL, 2008.

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

[4] I. 19794-2:2005, "Information technology–biometric data interchange formats–part2: Finger minutiae data," 2005.

[5] A.Jeuls, P.Tuyls, T.Kevenaar, and B.Skoric, "Fuzzy commitment," 2007, pp. 45–56.

[6] J.Bringer and V.Despiegel, "Binary feature vector fingerprint representation from minutiae vicinities," in *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2010, pp. 1–6.

[7] A. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Fingercode: a filterbank for fingerprint representation and matching," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 1999, pp. 187–193.

[8] L.Sha, F.Zhao, and X.Tang, "Improved fingercode for filterbank-based fingerprint matching," in *IEEE International Conference on Image Processing (ICIP)*, vol. 2, 2003, pp. II–895.

[9] F.Benhammadi, M.N.Amirouche, H.Hentous, K.Beghdad, and M.Aissani, "Fingerprint matching from minutiae texture maps," *Pattern Recognition*, vol. 40, no. 1, pp. 189–197, 2004.

[10] M.Tico, P.Kuosmanen, and J.Saarinen, "Wavelet domain features for fingerprint recognition," *Electronic Letters*, vol. 37, no. 1, pp. 21–22, 2001.

[11] T.Amornraksa and S.Tachaphetpiboon, "Fingerprint recognition using dct features," *Electronic Letters*, vol. 42, no. 9, pp. 522–523, 2006.

[12] H.Xu, R. Veldhuis, A.M.Bazen, T.Kevenaar, and T.Akkermans, "Fingerprint verification using spectral minutiae representations," in *IEEE Transactions On Information Forensics And Security*, vol. 4, 2009, pp. 397–409.

[13] B.Bhanu and X.Tan, "Fingerprint indexing based on novel features of minutiae triplet," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 25, 2003, pp. 616–622.

[14] M. Garris, C. Watson, R. McCabe, and C. Wilson, "User's guide to nist fingerprint image software(nfis)," *Technical Report NISTIR 6813. National Institute of Standards and Technology*, 2002.

[15] D.Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain, "Fvc 2000:fingerprint verification competition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 24, 2002, pp. 402–412.