FINGERPRINT RECOGNITION WITH RIDGE FEATURES AND MINUTIAE ON DISTORTION

Chu-Chiao Liao Ching-Te Chiu

Department of Computer Science, National Tsing Hua University, Taiwan , R.O.C

ABSTRACT

In this paper we present a new fingerprint matching method which combines different features, including minutiae and ridge features. The ridge features contain ridge count, ridge length, ridge curvature direction and ridge frequency. All ridge features are extracted in blocks around each minutia. The similarity scores of the features are summed with different weighting values as the final score of two fingerprints. Experiments are conducted on the FVC2002 database to compare the proposed method with other fingerprint methods on equal error rate (EER). The proposed method achieves better performance than other methods. The average EER value of the proposed method is 0.82 whereas the average EER value of the conventional matching method is 8.12.

Index Terms— Fingerprint Recognition, Minutiae, Ridge Features

1. INTRODUCTION

A fingerprint is the pattern of ridges and valleys on the surface of a fingertip. The uniqueness of a fingerprint is exclusively determined by local ridge characteristics and their relationships. Although fingerprint recognition has been significantly improved, there remain many challenging tasks like low quality or partial fingerprints. Several methods of automatic fingerprint identification have been proposed in the literature. The approaches of fingerprint identification are based on various features, e.g., singular points, ridges, pores, minutiae, etc.

Fingerprint matching approaches can be divided into three main families: minutiae-based, non-minutiae-based and hybrid matching. Minutiae based techniques are the most popular and widely used, which require an alignment between the template and input minutiae feature sets [1], [2], [3]. Non-minutiae based methods use features like local orientation, frequency, and ridge shape to recognize fingerprints [4], [5] [6]. The hybrid based matching algorithm uses both the features and the fingerprint image (pattern-based) for matching [7]. The pattern-based matching is a direct matching between the two fingerprint images.

In this paper, to increase the accuracy of conventional minutiae matching methods [8], we add other features to rec-

ognize the fingerprint images. The added features are ridge features as introduced by Choi *et al.* [9] and we modify it to enhance the performance. The ridge features contain ridge count, ridge length, ridge curvature direction and ridge frequency. By the ridge features, the method can overcome non-linear deformation of fingerprints.

The existing methods usually are affected by distortion. Regardless of the quality of the fingerprint is, it often has distortion problems, but poor quality has greater impact. Taking the methods Choi *et al.* [9] and Alilou [8] which are employed in our method as examples: the distortion may cause different minutiae types and relative positions. The problem leads to misjudge. Hence, we modify the referenced methods in order to conquer the distortion problem.

The rest of this article is organized as follows: Section II presents fingerprint matching using defined ridge features. Section III shows the performance evaluation compare to other algorithms. Section IV concludes the work of this paper.

2. PROPOSED RIDGE-BASED MATCHING

There is a ridge coordinate defined by every minutiae M_r as a center. A vertical axis which is extended from M_r and curved to be perpendicular to the ridge structures [9]. The curved vertical axis can overcome the distortion. An example of ridge vertical axis is shown in Fig.1.

In the ridge-based coordinate system, the ridge features describe the relation between the reference minutia and the other minutiae in the same fingerprint. The information of the ridge-based coordinate of a reference minutia is described as follows:

$$\overline{V} = (rc, rl, rcd, rf).$$
(1)

where rc, rl, rcd and rf represent the ridge count, ridge length, ridge curvature direction, and ridge frequency, respectively. All the ridge features are extracted in a 130×130 block with the referenced minutia as the center.

The ridge count (rc) is calculated by counting the number of ridges along the vertical axis until the axis meets the ridge attached to the neighboring minutia. The definition of the ridge count is not just the number of the ridges that vertical axis passed. If the ridges have been contained or the ridges



Fig. 1. Example of vertical axis. The red point refers to the minutia M_r . The curved line is vertical axis.

have no attached minutia, the ridge count would not be added. For each ridge, as long as it is attached by a minutia, its ID is set to be the same as the attached minutia's ID. If the ridge has no attached minutia, its ID is zero. Fig.2 shows the calculation of ridge count.



Fig. 2. Ridge count. The center minutia is referenced minutia M_r , blue numbers are the ID of the ridges, yellow line is the vertical axis of M_r and the red dots refer to the included ridge in ridge count.

The ridge length (rl) is the distance from the intersection of the vertical axis and the ridge to the attached minutia. Assuming there is a minutia M_c whose attached ridge is contained in ridge count. There are three methods that can generate the relation between M_r and M_c about ridge length: linear distance of referenced minutia M_r and M_c , vertical distance of the vertical axis and M_c , and the actual length from the intersection of the ridge which is attached by M_c and the vertical axis to M_c along the ridge. An example of the three ridge length is shown in Fig.3. The first type of calculating ridge length has no ridge identification. The second type of calculating ridge length is applicable to a straight line not a curved line. However, considering the computational complexity, the third type is not a good method. So, we record the coordination of the intersection of vertical axis and the ridge which attaches M_c , and use the distance of M_c and the recorded coordination as ridge length. Sometimes, the distance of M_c and the intersection is very close to the second type and the third type of ridge length.

Fig. 3. Three ridge length representation. The yellow line refers to the vertical axis of M_r . The orange one refers to linear distance of referenced minutia M_r and M_c . The green one refers to vertical distance of the vertical axis and M_c . And the blue one refers to the actual length from the intersection of the ridge which is attached by M_c and the vertical axis to M_c along the ridge.

Sometimes, even though the ridge count and ridge lengths are very similar, the shapes of the ridge patterns may be different. For example, there are two curves whose length are the same, but one is in concave shape, the other is in convex shape. In this case, we can tell they are different if we consider the curvature direction. The ridge curvature direction (rcd) is defined as follows:

$$rcd = sign\left(\sum_{i=1}^{N} \overrightarrow{v_i} \times \overrightarrow{v_{i-1}}\right).$$
 (2)

where v_i represents the *i*th vector between the sampling points along the attached ridge from the intersection of the vertical and attached ridge to the minutia on the ridge. N represents the number of sampling points. In the experiments, we set the sampling point every 10 pixels along the ridge. For each ridge, we take the sampling points no more than 15 points. Fig.4 shows how we take the sampling points to calculate ridge curvature direction.



Fig. 4. Example of sampling points when calculating *rcd*. It is a 130×130 block around M_r . The green line is the vertical axis. Red crosses refer to the sampling points. Yellow dots refer to the minutiae which are included in ridge count.

A fingerprint ridges frequency (rf) represents local average pixel distance between ridges. First, the test image is divided into a grid for each minutia as the center. In the experiment, we set the grid size as 36×36 . In the next step for each point of fingerprint image, the line L(i, j) orthogonal to local orientation is estimated. For each line L,

there are searched pixels which determine local minimum points B_r , $r = 1, 2, ..., \rho$ as fingerprint ridges, and local maximum points T_v , $v = 1, 2, ..., \sigma$ as fingerprint valleys between ridges. For the assumption, ρ refers to the number of fingerprint ridges in the sequence, and σ is the number of valleys in the grid. Fig.5 shows one grid of the fingerprint and its waveform. In Fig.5 (a), the green line is the line L which is orthogonal to the orientation of the referenced minutia.



Fig. 5. Fingerprint ridges as waveform and distance between ridges (B_r for green circles) and valleys(T_v for red triangles). (a) A 36×36 grid of fingerprint. The green line is the vertical line of the ridges. (b) Waveform of the sequence. The x-axis contains the pixel values which are on the L sequences(y-axis). The figure only contains the middle 50 pixels of the vertical line.

For each line *L*, there are two sequences of distances computed:

$$V_r = B_r - B_{r+1}, \quad r = 1, 2, ..., \rho - 1.$$
 (3)

$$R_v = T_v - T_{v+1}, \quad v = 1, 2, ..., \sigma - 1.$$
 (4)

The average local fingerprint ridge frequency for the line L centered at point (i, j) can be directly estimated by means of the formula:

$$ds(i,j) = \frac{\sum_{v=1}^{\rho} R_v + \sum_{r=1}^{\sigma} V_r}{\rho + \sigma + 2}$$
(5)

The measurement can be affected by noise. For this reason, the frequency smoothing stage is expected to reduce the noise and compute reliable frequency values.

Finally, the fingerprint image ridge frequency of the minutiae whose coordinate is (i, j) is calculated:

$$f(i,j) = \frac{\sum_{\mu=i-\frac{\beta}{2}}^{i+\frac{\beta}{2}} \sum_{\nu=j-\frac{\beta}{2}}^{j+\frac{\beta}{2}} ds(\mu,\nu)}{\beta^2}$$
(6)

where $\beta \times \beta$ is the size of smoothing window (8 × 8 in experiment).

When comparing every two sets of the ridge length, at first, we times each element of ridge length with corresponding ridge curvature direction as a new ridge length, then use a equation which is used to calculate the similarity between clusters:

$$D_{avg}(C_k, C_l) = \frac{1}{n_k n_l} \sum_{p_k \in C_k} \sum_{p_l \in C_l} |p_k - p_l|.$$
(7)

where C_k and C_l are the new ridge length sets of the minutia k in the input fingerprint image and minutia l in the query fingerprint image, respectively. n_k and n_l are the number of elements in C_k and C_l which are equal to the ridge count of the referenced minutia k and l. The smaller the result of D_{avg} is, the more similar these two sets are.

If the minutiae pair passes through three threshold TH_c , TH_f and TH_l ; one for ridge count, one for ridge frequency and another for D_{avg} , then we define the minutiae pair as corresponding minutiae pair. In this part, it returns the number of corresponding pair called $Corr_{ridge}$ of the two fingerprints.

There are some differences between the proposed method and the original Choi *et al.* [9] algorithm. Choi *et al.* is also vulnerable to fingerprint distortion and partial fingerprint even when the vertical axis is curved along with the ridge structure. In response to this problem, we use block information instead of whole information on the vertical axis and we ignore the ridge type as a feature. Additionally, the proposed method replaces the sign of ridge length with ridge curvature direction and changes the comparison algorithm of the ridge features. The sign of ridge length can be different when the vertical axis is shifted like Fig.6 shows. The replacement of ridge length sign with ridge curvature direction can avoid this error.



Fig. 6. Example of different signs of ridge length. There are two corresponding pairs called M_r and M_c located in input fingerprint and query fingerprint. In (a), the minutia M_c is on the right side of the vertical axis. But in (b), the corresponding minutia M_c is on the left side.

In Alilou [8] algorithm, if the minutiae pair has different minutiae types, then they are excluded from the possibility for being the corresponding pair. If distortion happens, some really corresponding pairs will be ignore. Fig.7 shows an example of false minutiae types of corresponding minutiae pairs. This situation may cause true negative. The proposed ridge features do not consider the ridge type as a matching feature. In this situation, Alilou's method and our proposed ridge features are complementary. So the combination of two methods can improve accuracy.

Methods	Туре	2002 DB1	2002 DB2	2002 DB3	2002 DB4
Proposed Ridge Features	R	5.57	1.64	2.86	4.18
Minutiae+Proposed Ridge Features	M+R	0.57	0.43	1.39	0.86
Conventional Minutiae Match [8]	М	5.53	5.1	14.13	7.74
MCC [1]	М	1	0.49	3.14	3
MCC_SDK [11]	М	17.7	18.52	19.02	17.88
Neibor Minutia Orientation [2]	М	1.41	1.44	4.32	2.12
Graphical Structures [3]	М	3.5	3.9	5.6	4.2
Ridge Coordinates [9]	M+R	1.8	0.8	3.5	-
Align Point Pattern [4]	M+R	2.95	2.45	7.76	3.44
Local Quadrangle Set [12]	M+R	1.38	1.33	4.96	2.12
Orientation Based Minutia Descriptor [13]	R	2.82	2.25	7.62	3.43
Line Similarity [6]	R	1.38	1.32	5.08	2.32
Core Phase Correlation [7]	Н	1.02	0.77	-	-

Table 1. EER(%) comparisons on FVC2002 databases

M: minutiae, R: ridges, H: hybrid



Fig. 7. Two fingerprint are from the same individual. The orange arrow is pointing to the case for corresponding minutiae pair, the minutia type in (a) is bifurcation, and in (b) is ending point. The blue arrow is pointing to the case for corresponding minutiae pair, the minutiae type in (a) is ending point, and in (b) is bifurcation.

The similarity score of two fingerprints using the proposed ridge-based matching is calculated by the number of corresponding pairs $Corr_{ridge}$ which is mentioned in Section II. The similarity score in the proposed ridge-based method is:

$$S_r = \frac{Corr_{ridge}}{|M_i||M_q|} \tag{8}$$

3. EXPERIMENTAL RESULTS

The FVC2002 [14] DB1_a, DB2_a, DB3_a and DB4_a databases are used to test the performance of our approach. There are 100 individuals in each database. Each individual has 8 different fingerprint images.

Equal error rate occurs when the decision threshold of a system is set so that the proportion of false rejections will be approximately equal to the proportion of false acceptances. To get equal error rate, we need score sets of genuine and impostor matches. For genuine matches, each impression of each finger is compared with other impressions of the same individuals. For imposter matches, each impression of each finger is compared with all impressions of the different fingers. There are four sets in the databases and in each set there are 100×8 images (8 fingerprints for each individual). The total number of genuine and impostor matching attempts are $((8 \times 7)/2) \times 100 = 2800$ and $(100 \times 99)/2 = 4950$, respectively.

Table 1 shows the comparisons of EER of each ridge feature that we use and EER with different fingerprint matching methods. In Table 1, our proposed method is superior to other methods.

In Table 1, the EER of other methods are higher than the proposed method in database FVC2002 DB3. The quality of the fingerprints in FVC2002 DB3 are bad. The average matching scores of genuine and impostor in conventional minutiae matches [8] are 0.47 and 0.28. In contrast, the average matching scores of genuine and impostor in proposed method are 0.45 and 0.03. With the similar scores of genuine matches, the proposed method has lower scores of impostor matches. Hence, the proposed method can reduce false positive.

4. CONCLUSIONS

This paper describes a fingerprint matching method that combines the information of minutiae, ridge features to enhance accuracy. We find some defects on the situation of distortion in the referenced method. And we improve the performance though modifying the referenced methods that can conquer distortion problem. The ridge features contain ridge count, ridge length, ridge curvature direction and ridge frequency. The ridge features are extracted in blocks around each minutia. With the ridge features, we calculate the similarity of the ridge features sets and the number of minutiae pairs that pass the thresholds. We transform each minutia and its neighbors based on the coordinates and orientation of the minutia. Then calculate the distance of the transformed minutia and its neighbors to get the similarity scores. The similarity scores of the three features are summed with different weighting value as the final score of the two fingerprints. The proposed method is superior to some existing methods on error equal rate (EER).

5. REFERENCES

- R. Cappelli, M. Perrara, and D. Maltoni, "Minutia cylinder-code: A new representation and marking technique for fingerpint recognition," *IEEE Transactions* on Pattern Analysis Machine Intelligence, no. 12, pp. 2128–2141, Dec. 2010.
- [2] H. Wei and D. Liu, "A multi-stage fingerprints matching algorithm," *Automation and Logistics, IEEE International Conference on*, pp. 197–199, August 2009.
- [3] Z. Gao, X. You, L. Zhou, and W. Zeng, "A novel matching technique for fingerprint recognition by graphical structures," *Wavelet Analysis and Pattern Recognition* (*ICWAPR*), pp. 77–82, July 2011.
- [4] A. K. Jain, H. Lin, and R. Bolle, "On-line fingerprint verification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, no. 4, pp. 302–313, April 1997.
- [5] A. Jain, Y. Chen, and M. Demirkus, "Pores and ridges: Fingerprint matching using level 3 features," *Pattern Recognition Letters*, pp. 2221–2224, 2004.
- [6] H. Wei and M. Yang, "Fingerprints matching based on line similarity," *Computer Science and Network Technology (ICCSNT)*, pp. 565–568, Oct. 2013.
- [7] P. Das, K. Karthik, and B. C. Garai, "An efficient hybrid fingerprint matching algorithm," *Computer Vision, Pattern Recognition, Image Processing and Graphics, 2011 Third National Conference on*, pp. 150–153, Dec. 2011.
- [8] V. K. Alilou, "Fingerprint matching: A simple approach," Nov. 2013. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/44369fingerprint-matching-a-simple-approach
- [9] H. Cho, K. Choi, and J. Kim, "Fingerprint matching incorporating ridge features with minutiae," *Information Forensics and Security, IEEE Transactions on*, pp. 338– 345, June 2011.
- [10] N. K. Rath, K. Karu, S. Chen, and A. K. Jain, "A realtime matching system for large fingerprint databases," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 799–813, August 1996.
- [11] X. Jiang and W. Yau, "Fingerprint minutiae matching based on local and global structures," *Int. Conf. Pattern Recognition*, pp. 6038–6041, 2000.
- [12] H. Wei, D. Liu, and C. Guo, "Fingerprints matching based on quadrangle," *Image and Signal Processing*, 2009. CISP '09. 2nd International Congress on, pp. 1–4, Oct. 2009.

- [13] M. Tico and P. Kuosmanen, "Fingerprint matching using an orientation-based minutia descriptor," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 8, pp. 1009–1014, August 2003.
- [14] D. Mario, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "Fvc 2002: Second fingerprint verification competition," *Proc. Int. Conf. on Pattern Recognition*, pp. 811–814, 2002.