Evaluation of Minutia Cylinder-Code on Fingerprint Cross-matching and Its Improvement with Scale

Yali Zang†, Xin Yang*, Xiaofei Jia, Ning Zhang, Jie Tian
Institute of Automation, Chinese Academy of Sciences, Beijing 100190 China
Jianmin Zhao†
College of Mathematics Physics and Information Technology, Zhejiang Normal University
†These authors contribute equally to this work. *Corresponding author: xin.yang@ia.ac.cn

Abstract

Recognition of fingerprints from multi-type capture sensors is a challenging work. While minutia cylinder-code, which is suggested to be invariant for translation and rotation and robust against skin distortion and small feature extraction errors, is an excellent feature proposed recently. In this paper, we conduct a series of experiments with minutia cylinder-code to evaluate its performance on fingerprints from multi-type capture sensors. Then a scale step is added to deal with the different resolutions and the creation of minutia cylinder-code is modified accordingly. The modified minutia cylinder-code with scale and two other matchers are tested. The experimental results proved that minutia cylinder-code is an effective and efficient feature for fingerprints from multi-type sensors, especially after modified with scale.

1. Introduction

Fingerprints have been increasingly used for individual identification in the governmental and civil application for the uniqueness, permanence and universality. Recently great improvement has been achieved in the fingerprint sensing technology and automatic recognition algorithms. The development of sensor technology allows us to acquire on-line fingerprints with various types of sensors. We define the matching of fingerprints from multi-type sensors as ”cross-matching” in contrast to ”regular matching”. The difference among multi-type sensors will significantly affect the characteristics of the raw data, the extracted features and the similarity score generated by matching algorithm. Only a few of researchers have studied the scale [6, 13] or non-linear distortion [4, 11] in cross-matching. But these algorithms are either with less satisfactory accurate or time-consuming. It is still a challenging work to develop proper features or matching algorithms for fingerprint cross matching.

Minutia Cylinder-Code is a local data structure: 1) invariant for translation and rotation, and 2) robust against skin distortion and small feature extraction errors. MCC was first proposed for fingerprint representation and regular matching by Cappelli et al in [2]. The accuracy and efficiency was both proved to be good and the application of MMC on palmprint recognition [1] also made a satisfactory performance.

In this paper, we conduct a series of experiments with MCC on a cross-device matching database to evaluate its performance on fingerprints from multi-type capture sensors. Then a scale step is added to deal with the different resolutions and the creation of MCC is modified accordingly. The modified MCC with scale and two other matchers are tested on the same database. The experimental results proved that MCC is an effective and efficient feature for fingerprints cross matching, especially after modified with scale.

The rest of the paper is organized as follows: Section 2 gives the preprocessing and feature extraction stages of the fingerprints. Section 3 describes original minutia cylinder-code and the proposed improvements with scale. Section 4 introduces the matching method of MCC. The experimental results and analysis are displayed in Section 5. Section 6 summarizes our researches.

2. Preprocessing and Feature Extraction

The original fingerprints need preprocessing to remove the noise and obtain clearer ridge information. Since minutia cylinder-code is related to minutiae local structure, the performance of preprocessing is important to the sequent feature extraction. The whole computational process mainly consists of the following steps as displayed in Fig .1:

- Segmentation: The original fingerprint is divided into

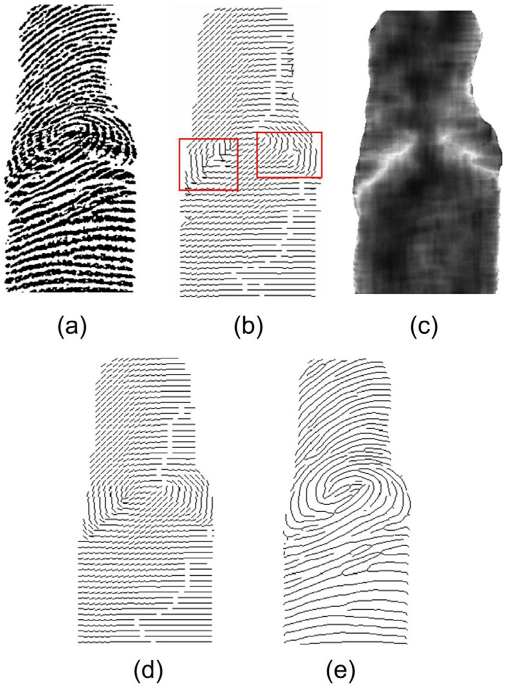

Figure 1. (a) Original fingerprint; (b) Coarse orientation field; (c) Reliability map with the burred field inside; (d) Modified orientation field; (f) Enhanced fingerprint skeleton.

many blocks of size $n*n$. For each block, we calculate three statistic features, the mean value, the variance of the gray levels and the clusters degree [3]. Using a linear classifier, we can decide whether a certain block belongs to the background or the valid region.

- Orientation field estimation: We adopt global orientation model to modify the orientation feature in the low-quality fingerprints. First the gradient-based approach [5] is used to compute the coarse orientation. Then we measure the reliability of orientation at each point using the normalized coherence of the gradient within the neighborhood area. The blurred region is detected composing of those points with a lower reliability value than a certain threshold. The orientation in these regions is considered incorrectly computed, which will be corrected by the 2D Fourier-based orientation model [16].

- Enhancement: The Gabor filter is applied to enhance the fingerprint with the advantages of its robust ridge enhancement and efficient noise removing. Then the ridge-line skeleton is obtained by thinning the binary ridges. The valley skeleton is also obtained by the same way. A series of postprocessing steps are produced to cut short branches, remove holes and delete spurious minutiae [9].

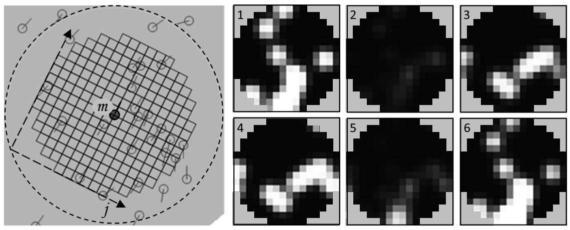

Figure 2. An example of the original MCC [2].

After preprocessing, minutiae are extracted, from both skeletons, using the the crossing number [10]. Then some simple rules are applied to remove spurious minutiae. Finally, a quality map is calculated starting from the consistencies of local orientations; minutiae located in regions where the quality is below a fixed threshold are discarded.

The result of the minutiae extraction procedure is a set of minutiae $M = \{m|m = (x_m, y_m, \theta_m)\}$, where for each minutia m, x_m and y_m represent its location, θ_m is its direction (in the range $[0, 2\pi)$).

3. Feature Representation

In this section, the original minutia cylinder-code and the proposed improvements with scale will be introduced.

3.1. The Original Minutia Cylinder-Code

Minutiae cylinder-code representation associates a local structure to each minutia. For each minutia m in M with sufficient number of neighbors, a cylinder with radius R and height 2π is created, whose base and height are related to the spatial and directional information, respectively. The cylinder is enclosed inside a cuboid whose base is aligned according to minutia m, the cuboid is discretized into $N_S \times N_S \times N_D$ cells. A numerical values is calculated for each cell by accumulating the spatial and directional contributions from minutiae in a neighborhood around the cell. The values of a cell can be a float number or invalid, depending on whether it is in the valid zone of both the fingerprint foreground and the certain neighborhood of a minutiae. Fig. 2 illustrates the original minutia cylinder-code of a minutia.

3.2. Minutia Cylinder-Code with Scale

Though MCC is suggested to be robust against distortion and invariant for translation and rotation, it could not directly tolerate the scale difference which is common for fingerprints from multi-type capture sensors. The scale parameter need be calculated before the creation of MCC.

Ridge distance can effectively characterize the difference of scales in the multi-type fingerprints. We calculate the global ridge distance map by the spatial approach [8] and utilize the polynomial approximation to smooth away the acuity over the map [15]. The modeled map is proved more robust to noise and distortion than the original one. Based on the experimental results, we choose the 4-order polynomial model as a tradeoff in balancing the computational cost and estimative accuracy. Then we can calculate the global ridge distance of the fingerprints as the average of all the values in the modeled map. For two compared fingerprints, the scale parameter sl can be obtained from the quotient of their ridge distance.

With the scale parameter, the creation of MCC for the second fingerprint should be changed as follows: 1) the radius R of the cylinder should be enlarged to sl times; 2) for each cell (i, j, k) in the cuboid, where $i, j, k \in \mathbb{N}$ and $0 \leqslant i, j \leqslant N_S$, $0 \leqslant k \leqslant N_D$, its two-dimensional location $p^m_{i,j}$ are recomputed as Eq. 1; 3) σ_S used to measure the neighborhood of $p^m_{i,j}$ should be enlarge to sl times, too.

$$p^m_{i.j} = \begin{bmatrix} x_m \\ y_m \end{bmatrix} + sl \cdot \Delta_S \cdot \begin{bmatrix} \cos(\theta_m) & \sin(\theta_m) \\ -\sin(\theta_m) & \cos(\theta_m) \end{bmatrix} \cdot \begin{bmatrix} i - \frac{N_S+1}{2} \\ j - \frac{N_S+1}{2} \end{bmatrix} \tag{1}$$

where (x_m, y_m) and θ_m is the location and direction of the center minutia, separately; $\delta_S = \frac{2 \cdot R}{N_S}$.

4. Matching

Given the MCC descriptors of two fingerprints, the similarity between each minutia descriptor in the first fingerprints and each descriptor in the second one is computed first, then a global score, denoting the overall similarity of the two fingerprints, is obtained from the local similarities.

In local matching step, the similarity between two minutiae is estimated by applying simple bitwise operations between the two corresponding binary vectors, as described in [2]. All parameters used for MCC are the same with MCC16 in [2].

An improved global matching step proposed in [1] is applied to calculate the overall similarity of the two fingerprints.

5. Experimental Results

The evaluation experiment of MCC on fingerprint cross-matching is conduct on FingerPass databases [7]. FingerPass database consists of nine sub-databases captured with nine representative sensors in the market including URU4000B optical press sensor, UPEK TCRU2C capacitive press sensor and Authentec AES2501 sweep sensor etc. Each sub-database contains 720*12 impressions (720 fingers, 12 impressions per finger).

Four sub-databases are chosen for the evaluation. Tab. 1 describes the characteristics of these four sub-databases, where we can see that the resolution of URU4000B sensor is 700dpi, different from the other three sensors (about 500dpi). Fig. 3 gives an example including four impressions from the same finger. Fingerprints from URU sensor have obvious fine pore features on their ridges due to the higher resolution.

Table 1. Characteristics of the chosen sub-databases.

	Capture Sensor	Image Size	DPI
DB1	URU 4000B	500*550	700
DB2	UPEK TCRU2C	208*288	508
DB3	Authentec AES2501	Unfixed	500
DB4	Symwave SW6888	288*384	500

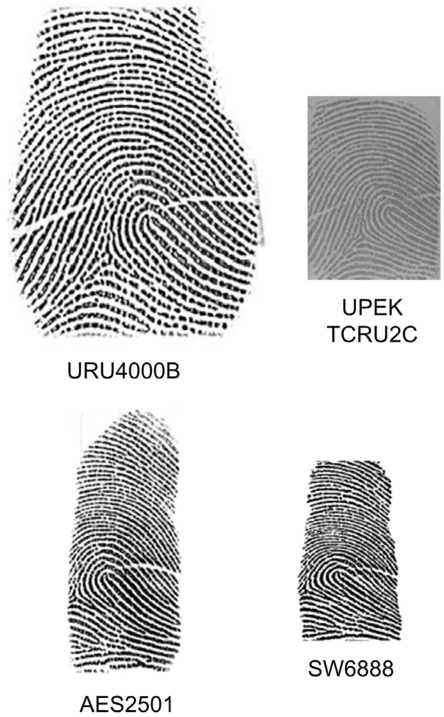

Figure 3. Four impressions from the same finger in the chosen sub-databases.

The MCC with scale method is also tested on the four sub-databases as well as a thin plate spline(TPS) model [12] and a state-of-the-art commercial matcher VeriFinger 6.1 SDK [14] for comparison. Since TPS model is proved robust to describe the non-linear transformation and can somewhat deal with the cross-matching issue, we suggest its performance as the advance level of the cross-matching algorithms. while VeriFinger 6.1 SDK is with mature commercial technology for regular matching and is considered as a baseline.

The sequence of the compared fingerprints in cross-matching is arranged as following: In genuine match each impression is matched against all the impressions of the same finger to compute the False Non-Match Rate(FNMR); In imposter match the first impression of each finger is

Table 2. Global ridge distance of fingerprints from the first finger in each sub-databases.

Fingerprint	URU	UPEK	AES	SW
1_1	10.30	8.49	7.67	7.95
1_2	10.22	8.44	7.70	7.97
1_3	10.26	8.57	7.88	8.06
1_4	10.24	8.57	7.85	8.02
1_5	10.21	8.60	6.96	8.03
1_6	10.28	8.51	7.39	7.86
1_7	10.22	8.44	7.93	8.02
1_8	10.14	8.54	7.56	7.75
1_9	10.24	8.66	7.58	7.71
1_10	10.15	8.69	7.76	7.88
1_11	10.23	8.46	7.10	7.81
1_12	10.27	8.76	7.26	8.46

matched against the first sample of the remaining fingers to compute the False Match Rate(FMR). To sum up, a total of $103,680$ $(720 * 12 * 12)$ genuine matching and $258,840$ (C_{720}^2) imposter matching are conducted for each cross-matching. And as for regular matching, only $47,520$ $(720 * C_{12}^2)$ genuine matching is done.

Tab. 2 illustrates the different global ridge distances of fingerprints from the first finger in each sub-databases, which imply the scale ratios of fingerprints from each two sub-databases. Tab. 3 summarizes the comparison results of regular matching and cross-matching over all four sub-databases respectively. There is no need to scale in regular matching, so it's unnecessary to given the results of the MCC with scale for regular matching.

Based on the experimental results, the following conclusions can be drawn out:

1). EERs of regular matching (whether MCC-based method or VeriFinger 6.1 SDK) are far more satisfying than those of cross-matching. Furthermore, EERs in cross-matching between the same kind of sensors (URU vs. UPEK, AES vs.SW) obviously outperforms that between different kinds of sensors. The comparison confirms the matching performance is greatly deteriorated by non-linear distortion and different distortion patterns.

2). TPS model achieves high accuracy of cross-matching, even better than VeriFinger 6.1 SDK on those databases in different modes of acquisition(URU vs. AES, UPEK vs. AES, URU vs. SW), which proves its capability to describe the non-linear transformation.

3). EERs of MCC-based algorithm for regular matching are a little worse than those of VeriFinger 6.1 SDK. While EERs of MCC-based algorithm for cross matching are normally better than those of VeriFinger 6.1 SDK. As VeriFinger 6.1 SDK is a state-of-the-art commercial matcher, the results confirm that MCC is an effective feature for fingerprint matching.

4). MCC with scale achieves lower EERs than MCC, especially on the sub-databases with different resolutions(UPEK vs. AES, UPEK vs. SW), the modification of MCC according to scale is effective. Meanwhile, EERs of MCC with scale are mostly comparable to those of VeriFinger 6.1 SDK and those of TPS model, which proved that MCC with scale is effective for fingerprint cross matching.

5). Adding a scale process does improve the accuracy of cross-matching though it costs more time for ridge distance calculation, especially for the fingerprints for those databases in different modes of acquisition.

6). EER of the VeriFinger 6.1 SDK is lowest, but there are some other factors should be consider when compare with other algorithms. First, in the template generation stage, VeriFinger 6.1 SDK adopts quality control. If the quality of input fingerprint image is lower than a threshold, the image will be discarded and its template can not be generated. So VeriFinger 6.1 SDK discards some low quality fingerprint images and improves the matching performance on enrolled image databases. Second, in the matching stage, unlike our method uses only minutiae, VeriFinger 6.1 SDK uses more features, which can also improve the matching performance.

Based on the above experimental results and analysis, we can conclude that MCC is an effective and efficient feature for fingerprints cross matching, especially after modified with scale.

6. Conclusion

In this paper, we conduct a series of experiments with MCC on a cross-device matching database to evaluate its performance on fingerprints from multi-type capture sensors. Then a scale step is added to deal with the different resolutions and the creation of MCC is modified accordingly. The modified MCC with scale and two other matchers are tested on the same database. The experimental results proved that MCC is an effective and efficient feature for fingerprints cross matching, especially after modified with scale.

In future, we will do further research of MCC on fingerprints from multi-type sensors and try to combine multi-scale to MCC to improve its performance on cross matching.

References

[1] R. Cappelli, M. Ferrara, and D. Maio. A fast and accurate palmprint recognition system based on minutiae. *IEEE Transactions on systems, man, and cybernetics - part B: cybernetics*, 42(3):956–962, June 2012.

[2] R. Cappelli, M. Ferrara, and D. Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2128–2141, December 2010.

Table 3. The performance of regular matching and cross-matching of MCC-based algorithms and comparison algorithms on four FingerPass sub-databases.

	Method	EER(%)	FMR100(%)	FMR1000(%)	ZeroFMR(%)	AvgTime(s)
URU	MCC	0.023	0.007	0.019	0.858	0.051
	VeriFinger	0.018	-	-	0.421	0.001
UPEK	MCC	0.056	0.015	0.047	1.229	0.047
	VeriFinger	0.045	-	-	0.137	0.001
AES	MCC	0.053	0.017	0.044	1.191	0.049
	VeriFinger	0.014	-	-	0.042	0.001
SW	MCC	0.073	0.028	0.061	2.517	0.052
	VeriFinger	0.028	-	-	0.109	0.001
URU vs.UPEK	MCC	27.41	89.23	98.65	99.99	0.067
	MCC+Scale	0.283	0.128	0.457	2.936	0.117
	TPS	0.298	0.128	0.512	1.552	0.145
	VeriFinger	0.272	-	-	0.714	0.001
AES vs. SW	MCC	2.482	3.670	7.681	24.93	0.065
	MCC+Scale	2.242	3.309	5.827	11.42	0.121
	TPS	2.017	3.546	7.328	15.63	0.183
	VeriFinger	1.083	-	-	3.511	0.001
URU vs. AES	MCC	27.97	90.01	98.44	99.97	0.069
	MCC+Scale	2.432	4.186	10.89	31.06	0.115
	TPS	2.288	3.747	10.42	27.69	0.187
	VeriFinger	2.675	-	-	6.631	0.001
UPEK vs. AES	MCC	3.305	5.358	10.50	33.10	0.078
	MCC+Scale	2.632	3.581	6.137	15.26	0.103
	TPS	1.948	2.444	5.758	8.683	0.160
	VeriFinger	2.907	-	-	8.159	0.001
URU vs. SW	MCC	26.41	87.26	97.39	99.86	0.080
	MCC+Scale	3.326	7.854	15.47	28.37	0.126
	TPS	3.158	7.329	13.56	25.73	0.194
	VeriFinger	3.487	-	-	10.92	0.001
UPEK vs. SW	MCC	5.210	9.701	18.59	47.19	0.069
	MCC+Scale	4.437	8.430	13.86	25.41	0.112
	TPS	4.382	8.592	13.94	25.83	0.187
	VeriFinger	4.263	-	-	19.58	0.001

[3] X. Chen, J. Tian, J. Cheng, and X. Yang. Segmentation of fingerprint images using linear classifier. *EURASIP*, 2004(4):480–494, 2004.

[4] Y. He, J. Tian, Q. Ren, and X. Yang. Maximum-likelihood deformation analysis of different-sized fingerprints. In *AVBPA 2003*, volume 2688, pages 421–428, 2003.

[5] L. Hong, Y. Wan, and A. K. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8):777–789, 1998.

[6] J. Y. Jang, S. J. Elliott, and H. K. On improving interoperability of fingerprint recognition using resolution compensation based on sensor evaluation. In *ICB 2007*, volume 4642, pages 455–463, 2007.

[7] X. Jia, X. Yang, Y. Zang, N. Zhang, and J. Tian. A cross-device matching fingerprint database from multi-type sensors. In *Proceedings of 21st International Conference on Pattern Recognition*, 2012.

[8] Z. M. Kovács-Vajna, R. Rovatti, and R. Frazzoni. Fingerprint ridge distance computation methodologies. *Pattern Recognition*, 33:69–80, 2000.

[9] X. Luo and J. Tian. Knowledge based fingerprint image enhancement. In *In Proceedings of 15th ICPR*, volume 4, pages 783–786.

[10] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of fingerprint recognition*. Springer, New York, second edition, 2009.

[11] A. Ross and A. K. Dass, S.and Jain. A deformable model for fingerprint matching. *Pattern Recognition*, 38(1):95–103, 2005.

[12] A. Ross and R. Nadgir. A thin-plate spline calibration model for fingerprint sensor interoperability. *IEEE Trans. on Knowledge and Data Engineering*, 20(8):1097–1110, 2008.

[13] X. J. Tan and B. Bhanu. Fingerprint matching by genetic algorithms. *Pattern Recognition*, 39:465–477, 2006.

[14] VeriFinger SDK. http://www.neurotechnology.com/verifinger.html.

[15] D. R. Wan and J. Zhou. Fingerprint recognition using model-based density map. *IEEE Trans. on Image Processing*, 15(6):1690–1696, 2006.

[16] Y. Wang, J. Hu, and P. Damien. A fingerprint orientation model based on 2d fourier expansion (fomfe) and its application to singular-point detection and fingerprint indexing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(4):573–585, 2007.