A New Blind Fingerprint Alignment Algorithm used in Biometric Encryption

Xinglong Zhang, Quan Feng, Kang He
Engineering College, Gansu Agricultural University Lanzhou, Gansu Province, China
fquan@sina.com

Abstract - Fuzzy Vault is an effective method to ensure the security of template information. However, this is done after the alignment of the template and the sample. In this paper, an improved blind fingerprint alignment using maximum curvature points of fingerprint is proposed, which are extracted on the dense and reliable orientation field flow curves followed by a post-processing to purify the obtained maximum curvature points. Experimental results show the good alignment accuracy especially in those cases that the template and the query fingerprint images are partially overlapped.

Index Terms - fingerprint, alignment, biometric, encryption

1. Introduction

Nowadays, with more and more private and sensitive information exchanging on the Internet, the cryptographic technique has been widely used to guarantee the security. Conventional cryptography uses cryptographic keys, which are just bit strings long enough [1]. The biometric authentication techniques employ biometric characteristics as testimony to verify identity because it is difficult to guess or forge biometrics in general. The way to release a key based on biometric authentication is a promising technique, which would offer more reliability than traditional cryptographic methods [2].

Some biometric cryptosystems have been proposed in the literature [3-6], which integrate biometrics with cryptographic system, provides a reliable, secure method of protecting the cryptographic key. Fuzzy vault scheme is a remarkable cryptographic construction proposed by Juels and Sudan [7], and it can work with unordered and partial biometric data. Several fingerprint-based implementations have been presented [8-10]. Unlike the traditional fingerprint system, where the template is directly stored in the database, these methods stores only a transformed version of the template. Therefore, the main challenge here is the alignment between the query fingerprint and the original template. In the fuzzy vault scheme in [8], an automatically alignment algorithm that aligns the template and the query automatically without leaking the template's minutiae is proposed. They extract maximum curvature points from fingerprint orientation field and use them as helper data to align the template and query minutiae. In this paper, more dense and reliable orientation field flow curves are created and used in the stage to extract the maximum curvature points, followed by some post-processing algorithm to purify those extracted points. The experimental results demonstrate the effectiveness of the proposed improved method.

The rest of the paper is organized as follows. Section 2 introduces how to get maximum curvature points. The automatic fingerprint alignment algorithm based on maximum curvature points is described in section 3. The experimental results are presented in section 4, and section 5 gives the summarization and future work.

2. Helper Data Extraction

Maximum curvature points are extracted from the fingerprint orientation flow image and used as helper data to align the template and query fingerprint images. The helper data is stored as public information, and will not reveal any information about the template minutiae [8]. The steps of the helper data extraction are shown in Fig.1.

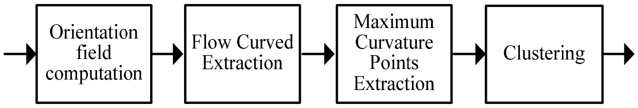
Orientation field computation
Flow Curved Extraction
Maximum Curvature Points Extraction
Clustering

Fig.1 The Flowchart of Maximum Curvature Points Extraction

Let I be a fingerprint image with R rows and C columns.

A. Orientation Field Estimation

The orientation field of I gives the direction of the ridge flow in a local neighborhood around s for all $s \in I$. We use a robust algorithm described in [11] to estimate of the orientation field of I. Thus we get the orientation o_s of s in the form of vector $(\cos x_s, \sin x_s)^T$ where x_s is the angle of the flow with respect to the horizontal axis.

B. Flow Curves Extraction

An OFFC(Orientation Field Flow Curve) [12] is a set of short linear segments whose tangent direction at each point is parallel to the orientation field direction at that point. Although flow curves are similar to fingerprint ridges, extraction of flow curves is not affected by breaks and discontinuities which are commonly encountered in ridge extraction. So it is better to use flow curves to extract maximum curvature points. An OFFC with a starting point s_0 can be defined as

$$s_j = s_{j-1} + d_j \cdot l \cdot o_{j-1} \tag{1}$$

where $j = 1, 2, \cdots, n$; d_j , with values among $\{-1, +1\}$, is the flow direction from s_{j-1} to s_j , l is the distance from s_{j-1} to s_j , and $o_{s_{j-1}}$ is the orientation vector of site s_{j-1} . The point s_n denotes the end point of the OFFC curve, which is achieved when the boundary of the image is reached or n exceeds the predefined number. The length l specifies the sampling interval of the OFFC. In our experiments, l is set as 6 for all the OFFCs. Each point s_0 generates two segments of an OFFC which are obtained by initializing d_1 firstly at +1, and after the first segment is obtained initializing d_1 at -1, so that the points s_j in (2) will trace opposite directions. The starting points $\{s_0\}$ are selected by the following method:

Let $s_q = (x_q, y_q)$ be the singularity which can be obtained by using the algorithm in [11], $s_t = (x_t, y_t)$ be the top boundaries of the fingerprint pattern area, and d be the sampling width. The points $\{s_0\}$ are determined as follows:

$$s_0 = s_t + k \cdot d \cdot V \qquad (2)$$

where $k = 1, 2, 3, \cdots$, and V is the direction vector from s_t to s_q .To get plenty of OFFCs, we select $d = 3$ in our experiments instead of 5 listed in [12]. Furthermore, it did not offer an efficient method to get d_j which is very important to get reliable an OFFC in [12]. To improve it, we use the following method.

Define $v_j = s_j - s_{j-1}$, $\Delta x_j = < v_{j-1}, v_j >$ (the angle from v_{j-1} to v_j), then we predict the value of v_{j+1}, pv_{j+1}, to be $R(x_j) \cdot v_j$ where

$$R(x_j) = \begin{bmatrix} \cos x_j & \sin x_j \\ -\sin x_j & \cos x_j \end{bmatrix}.$$

The pseudo code is :

```
{
    If $pv_{j+1} \cdot o_j \geq 0$, set $d_{j+1} = 1$;
    Else set $d_{j+1} = -1$.
}
```

Fig.2 gives the extracted results. Blue curves are flow curves, green circles are s_q and s_t , and yellow stars are s_0 .

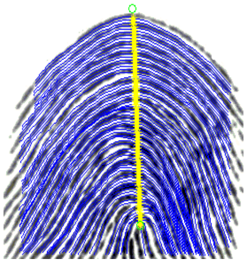

Fig. 2. Extracted Flow Curves

C. Helper Data Extraction

Coordinates of maximum curvature points in the flow curves along with their curvature values constitute the helper data used in our auto alignment. We use methods in [8] to extract the maximum curvature points.

The curvature of the point l_i in a flow curve is defined as $w_{l_i} = 1 - \cos a_i$ (a_i is the angle between the tangents to the flow curve at the points l_{i-T} and l_{i+T} , see Fig.3) when $T < i \leq N - T$, $\omega_{l_i} = 0$ when $i \leq T$ or $\text{i} > N - T$. The parameter T is related to the sampling interval of the flow curve. If the value of the local maximum is greater than some pre-defined threshold, the point is marked as a local maximum curvature point and described using the triple vector $h = (x, y, w)$, where (x, y) is the location and w is curvature value. H is used to represent the extracted helper data list. After maximum curvature points of all the flow curves are obtained we get the final helper data list $H_I = \{h_i\}_{i=1}^{R_I}$, where R_I is the number of local maximum curvature points in image I .

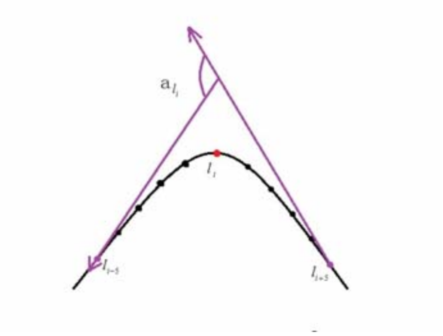

Fig. 3. a at l_i .

D. Post-processing

Local maximum curvature points tend to be distributed close to a straight line, so we can use the post-processing method to purify the extracted helper data in H_I . Firstly, a straight line is used to fit all these points in H_I , then the distance d_i from the point h_i to the fitting line is calculated, if d_i is larger than some predefine threshold , the corresponding e h_i is deleted from H_I . Fig. 4 shows the results of the helper data extraction. The red stars give the positions of the helper data.

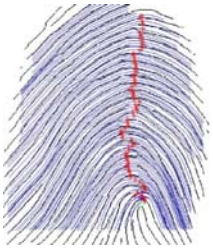

Fig. 4. An Example of the Extracted Helper Data.

3. Alignment Using Improved ICP

The Iterative Closest Point (ICP) algorithm in [13] is widely used to align 3-D point set, but it strictly assigns every query helper point to a template helper point, which will lead to some alignment errors when the overlap between the two sets is partial. To solve this problem, we use the improved ICP algorithm in [14] which ignores a proportion of points in the query set whose distance to the corresponding points in the template set is greater than a predefined value. Thus we can get the rigid transformation $F=(R,t)$ where R is the rotation matrix and t is the translation vector.

4. Experimental Results

To verify the performance, FVC2002 DB2 is used in our experiments. It consists of 800 fingerprint images from 100 fingers (8 impressions/finger). The image size is 296*560 with the image resolutions of 569 dpi. The parameter T is set to 6 and the threshold of the maximum curvature if set to 0.28.

FMR (false match rate), FNMR (false non-match rate) and EER (equal error rate) are given in our results. Here we use the system in [15] to evaluate the performance of the extracted helper data in an automatic alignment. The result of correct curve is shown in Fig.5. Fig.6 gives some results of correct blind alignment, especially in some case of small overlapped areas. Fig.7 shows some failure results.

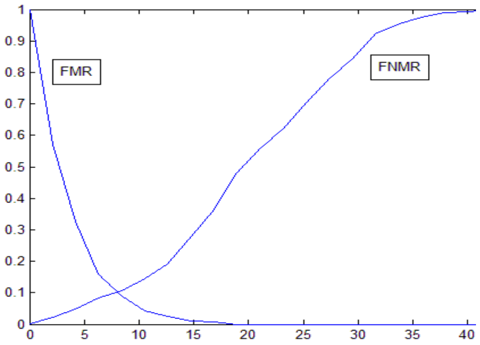

Fig. 5. FMR and FNMR after the blind alignment.

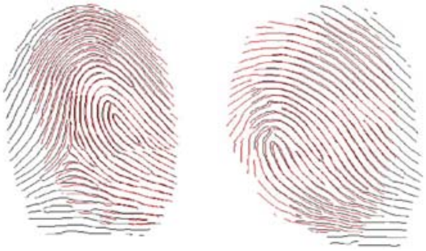

Fig.6. Some correct alignment.

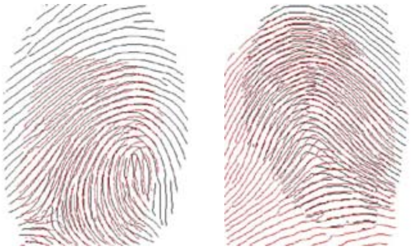

Fig.7. Some failure results.

For the more dense orientation flow curves give more information, so it is more robust to some extent in the improved algorithms in case of the small overlapped areas of the two fingerprint images.

5. Summary and Future Work

The security of template information is becoming more and more important. In this paper, an improved fingerprint blind alignment method is proposed. More dense and reliable orientation field flow curves are created and used in the stage to extract the maximum curvature points, followed by the post-processing algorithm to purify those extracted points. The experimental results demonstrate the effectiveness of the proposed improved method, even though the template and the query are partially overlapped.

Our future work would concentrate on improving our algorithm to get higher accuracy.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (61062012), the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, and the Fundamental Research Funds for the Central Universities (2011RC0121).

References

[1] A. Menezes, P. van Oorschot and S. Vanstone. *Handbook of applied cryptography*. Cincinnati, USA: CRC Press, 1996.

[2] Quan Feng, Fei Su, Anni Cai. The encrypted fuzzy vault and fingerprint-based implementation. *Fifth International Conference on Information Assurance and Security*, pp. 137-140, 2009.

[3] G. I. Davida, Y. Frankel, and B. J. Matt. On enabling secure applications through off-line biometric identification. *IEEE Symposium on Security and Privacy*, pp. 148–157, May 1998.

[4] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. K. V. Kumar. Biometric encryption. *ICSA Guide to Cryptography*, R. K. Nichols, Ed. McGraw Hill, 1999.

[5] F. Monrose, M. Reiter, Q. Li, and S. Wetzel. Cryptographic key generation from voice. *IEEE Symposium on Security and Privacy*, pp. 202–213, May 2001.

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

[7] A. Juels and M. Sudan. A Fuzzy Vault Scheme. *IEEE International Symposium on Information Theory*, p. 408, 2002.

[8] K. Nandakumar, A. K. Jain and S. Pankanti. Fingerprint-based fuzzy vault: Implementation and Performance. *IEEE Trans. Information Forensics and Security*, Vol.2, No.4, pp.744–757, 2007

[9] T. Clancy, D. Lin, and N. Kiyavash. Secure smartcard-based fingerprint authentication. *ACM SIGMM Workshop on Biometric Methods and Applications*, pp. 45–52, Nov. 2003.

[10] Quan Feng, Fei Su and Anni Cai. Fingerprint-based key binding/recovering scheme based on fuzzy vault. *Journal of Electronics (China)*, Vol. 25, No. 3, pp.415-421, Jun. 2008.

[11] S. C. Dass. Markov random field models for directional field and singularity extraction in fingerprint images. *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1358–1367, October 2004.

[12] S. C. Dass and A. K. Jain. Fingerprint classification using orientation field flow curves. *Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 650–655, December 2004.

[13] P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, February 1992.

[14] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. *International Conference on Pattern Recognition*, pp. 545–548, August 2002.

[15] Sun Peng. Singular point extraction in fingerprint recognition and minutiae matching suitable for resource-constraint. *Master thesis, of BUPT*.