Partial Fingerprint Recognition Based on Localized Features and Matching

Tsai-Yang Jea and Venu Govindaraju
Center for Unified Biometrics and Sensors, University at Buffalo, the State University of New York, Amherst, New York 14228
{tjea,govind}@cedar.buffalo.edu

1. Introduction

Fingerprint matching based on minutia features is a well researched problem. Most of these algorithms assume that the two templates are approximately of the same size. This hypothesis is no longer valid. Miniaturization of fingerprint sensors has led to small sensing areas and can only capture partial fingerprints.

Matching partial fingerprints to full (relatively larger or rolled) pre-enrolled images in the database presents several challenges: (i) the number of minutia points available in partial prints is few, thus discriminating power reduces; (ii) absence of singular points (core and delta) in partial prints is likely, therefore, a robust algorithm independent of these singularities is required; and (iii) uncontrolled impression environments result in unspecified orientations of partial fingerprints, and distortions like elasticity and humidity are introduced due to characteristics of the human skin.

In this paper, we propose a matching algorithm that uses the secondary features we presented in [1]. However, in this approach, the number of secondary features associated to a minutia varies according to the fingerprint's size. Moreover, secondary feature indexing and a local matching technique without global alignment are also presented.

2. Secondary Features

Minutiae information is extracted by using the algorithms described in [2] with some modifications to remove the false minutiae on the edges of the fingerprint foreground. A secondary feature consists of a central (M_c) and two of its neighboring minutiae (M_0 and M_1) (Fig. 1). Only relative information between the central and neighboring minutiae, such as distance (r), orientation (φ), and angle (δ) with respect to the central minutia, is used. Thus, secondary features are invariant to translation and rotation. To increase the usability and robustness of our algorithm, we do not include the ridge count and minutia type information that are used by other local fingerprint representations [3][4].

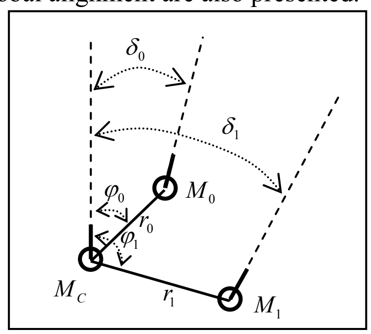

Fig. 1 Secondary feature of central minutia M_C.

In [1], each minutia associates with only one secondary feature. However, fewer minutiae exist in a small fingerprint, and the small amount of secondary features makes secondary feature matching difficult. We propose to associate more secondary features to a minutia, and the number of associating secondary features is determined by the size (number of minutiae) of the fingerprint. Instead of using the two nearest neighboring minutiae, k nearest minutiae around the central minutia are selected. A secondary feature is formed by the central minutia and *any* two of its neighboring minutiae. Therefore, we have C_2^k secondary features for each minutia. The value k is selected according to each fingerprint's size. For example, in our experiment, the k is 6, 7 and 10 when the number of minutiae on the fingerprint is larger than 30, between 20 and 30, and less than 20, respectively. A "neighborhood list" that consists of k neighboring minutiae of the central minutia is also maintained.

3. Feature Pruning and Indexing

Large amount (about 600) of secondary features is generated for every fingerprint by associating more than one secondary features with each minutia which affects the speed of finding the matched secondary features between the input and reference fingerprints. To address this problem, we remove the "poor" secondary features that would introduce ambiguities when performing secondary feature matching. A secondary feature, S_C, is "poor" if $\angle M_0M_CM_1$, is extremely large, close to 180°, or extremely small, close to 0°. Due to distortions of fingerprint images, these "poor" features may result in false match or false non-match of secondary features. By removing "poor" features, we can not only increase the matching accuracy but also speed up the matching process.

Furthermore, an indexing technique is used to further improve the efficiency of matching. Instead of indexing across the entire database [5], we propose to index the secondary features "within" a fingerprint. We put secondary features into different bins according to their neighbors' properties and positions that are relative to the central minutiae. We divide the space around a central minutia into several, say 8, quadrants, and each of the two neighboring minutiae will belong to one of the quadrants. Assume that the partition is aligned with the orientation of the central minutia and quadrants are labeled from **a** to **h** clockwise (Fig. 2). Thus, each secondary feature can be put into the bin that represents a specific neighboring minutiae configuration, for example **bd**, **dh**, or **hb**. In order to compensate the affects of distortion, we also label the neighboring minutia with the closest quadrant. Use Fig. 2 as an example, neighboring minutiae M_0 can be labeled as **a** and **b**, and M_1 can be labeled as **b** and **c**. Hence the secondary feature, *Sc*, can possibly reside in bins **ab**, **ac**, **bc**, and **bb**.

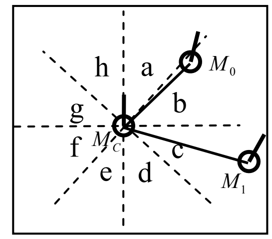

Fig. 2 The eight quadrants and the secondary feature, M_c, can be labeled as **ac**, **bc**, **ab**, and **bb**.

When matching is performed, only the feature distances of the secondary features within the same bin are examined. This improves the matching speed significantly. The average number of times of matching reduced from 600 times to 47 times per feature.

4. Matching

The matching is performed in two stages: (i) secondary feature matching and (ii) extending matching. The first stage attempts to obtain the possible secondary feature correspondence between the input and reference fingerprints. Using dynamic tolerance areas and a validation process that described in [1], a list of matched secondary features, called "seeds", is obtained. Then, we add those matched "seeds" into each other's "neighborhood lists".

The second stage of the matching is extending the pairing results of those "seeds". Beginning with any pair of seeds, we put the pair into a "matching list", which contains the pairing relation between minutiae on both input and reference fingerprints. Starting from the head of the "matched list", we match the minutiae in their neighborhood lists. If there are matched minutiae, we append the newly matched minutiae pair into the "matched list". This process continues until all the neighboring minutiae of those matched pairs in the "matched list" are examined. At the end of this process, the number of matched minutiae and pairing information are returned. This stage can be viewed as a special breadth first search that starts with a pair of seeds and then propagates the matching along their neighborhood lists. Finally, we choose the highest matched number as our matching result.

Table 1 Some results for partial fingerprint matching.

Fingerprint Size	Avg. Width (pixels)	Avg. Height (pixels)	Bozorth3		Proposed Matching	
			EER	Min. TER	EER	Min. TER
20%	92.42	133.30	43.90%	68.40%	20.97%	30.92%
30%	113.31	163.36	25.63%	33.18%	9.79%	16.20%
40%	130.90	188.70	12.35%	19.69%	6.16%	10.17%
50%	146.41	211.03	8.28%	11.77%	3.16%	5.16%

5. Experimental Results

Our algorithm is tested on the FVC 2002 Db1 and Db2 databases, and the performances are compared to a publicly available fingerprint matcher (Bozorth3 [2]) which uses the same feature extraction algorithm as our experiment. For the Db1 database, Bozorth3 matcher has equal error (EER) at 4.67% and the proposed algorithm has EER at 1.09%. For the Db2 database, Bozorth3 matcher has EER at 3.37% while proposed method achieves EER at 1.22%. To evaluate the system performance on partial fingerprints, we generated a series of partial fingerprint templates of various sizes (20%~90% of original image size) from the Db1 database and match the partial template against the original sized template of different impression. Some results are listed in Table 1. Our approach outperforms the Bozorth3 matcher in both full and partial fingerprint matching experiments.

6. Conclusions

The proposed algorithm has the following advantages: (i) secondary features are localized and their matching does not rely on any global alignment; (ii) number of features is determined by the fingerprint size and every fingerprint has roughly similar number of features; (iii) secondary features are indexed, thus the matching speed is fast; and (iv) the entire matching process is performed within the neighborhood of each minutia, thus the proposed algorithm can handle distortion better. Superior results are observed in the experiment.

7. References

[1]. "Security and matching of partial fingerprint recognition systems," in Proceeding of SPIE, T. Jea, V. K. Chavan, V. Govindaraju and J. K. Schneider, Vol. 5404 of SPIE Biometric Technology for Human Identification (Orlando, Florida 2004), pp39-50.

[2]. Craig I. Watson, Michael D. Garris, Elham Tabassi, Charles L. Wilson, R. Michael McCabe, and Stanley Janet, "User's Guide to NIST Fingerprint Image Software 2 (NFIS2)," National Institute of Standards and Technology, 2002

[3]. "Fingerprint Minutiae Matching Based on the Local and Global Structures," in Proc. of ICPR (15th), X. Jiang and W.Y. Yau, Vol. 2 of ICPR (Barcelona, Spain, Sept 2000), pp 1042-1045.

[4]. "A Topology-Based Matching Algorithm for Fingerprint Authentication," in Proc. of Int. Carnahan Conf. On Security Technology (25th), Z. Chen and C.H. Kuo, 1991, pp 84-87.

[5]. R.S. Germain, A. Califano, and S. Colville, "Fingerprint Matching Using Transformation Parameter Clustering," IEEE Computational Science & Engineering, Vol. 4, No. 4, 1997, pp 42-49.