Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 26 (2005) 1337–1345
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Fingerprint minutiae matching using the adjacent feature vector

Xifeng Tong [a,b,*], Jianhua Huang [a], Xianglong Tang [a], Daming Shi [c]

[a] *School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, P.R. China*
[b] *Daqing Petroleum Institute, Daqing, P.R. China*
[c] *School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore*

Received 29 October 2004
Available online 15 December 2004

Abstract

Minutiae matching is the most popular approach to fingerprint verification. In this paper, we propose a novel fingerprint feature named the adjacent feature vector (AFV) for fingerprint matching. An AFV consists of four adjacent relative orientations and six ridge counts of a minutia. Given a fingerprint image, the optimal matching score is computed in three stages: (1) minutiae candidate pairs searching based on AFVs; (2) coordinate transform for image rotation and translation; and (3) transformed minutiae matching to get matching score. The experimental results show that the proposed method provides a good trade-off between speed and accuracy.

Keywords: Fingerprint verification; Minutiae matching; Adjacent feature vector

1. Introduction

Fingerprints are texture on the top of human fingertips. Because of the uniqueness of fingerprints, fingerprint verification has been applied to many fields including criminal verification, financial security, access control, etc. (Leung et al., 1991). In the past three decades, automatic fingerprint verification has been applied more widely than other branches of biometrics such as face verification and signature verification. Fingerprint matching typically consists of two stages: feature extraction and feature matching. Feature matching is usually evaluated by the following factors:

- Matching error rate, including False Match Rate and False Non-Match Rate.
- Computational time cost.

* Corresponding author. Address: School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, P.R. China. Tel.: +86-45186417602; fax: +86-45186413631.
E-mail address: tongxifeng@eyou.com (X. Tong).

0167-8655/$ - see front matter
doi:10.1016/j.patrec.2004.11.012

In order to improve the above two factors, many techniques have been designed, for example, Jain et al.'s filterbank method (2000) and Isenor and Zaky's graph matching method (1986). But the most popular matching strategy for fingerprint verification is minutiae matching (He et al., 2003; Bazen and Gerez, 2003; Ratha et al., 2000). The simplest pattern of the minutiae-based representation consists of a set of minutiae, including ridge endings and bifurcations. Each minutia is described by its spatial location associated with the direction and the minutia type.

Many researchers have tried to make the minutiae-based method more robust. Ross (2001) described an algorithm using both minutiae and texture features. Kovács-Vajna (2000) proposed a triangular matching method to cope with the deformation and validated the matching by dynamic time warping. Ratha and Karu (1996) proposed a Hough transform based matching method. By specifying scale, rotation and shift parameters, a Hough transform is conducted on a minutiae set. A score can be obtained by specifying these three parameters. The maximum score is considered as the final score. He et al. (2003) introduced ridge information into the minutiae matching process to reduce the computational cost and made use of a variable sized bounding box to make the algorithm robust to deformation.

Jiang and Yau (2000) proposed a minutiae matching technique, which used both local and global structural information. The local structure described a rotation and translation invariant feature and they used the local structure to find the correspondence of two minutiae sets. They used the global structure to determine the uniqueness of a fingerprint. However, if fewer neighborhood minutiae are used, false reject may occur in the presence of spurious minutiae and the absence of genuine minutiae; if more neighborhood minutiae are used, the template size will increase tremendously.

Lee et al. (2002) proposed a local alignment method, in which ridge frequency was adopted to minimize distance error by normalizing the distance between minutiae. However, distance normalizing by frequency makes the algorithm more complex and time consuming.

Bazen and Gerez (2003) proposed a thin-plate spline model for minutiae matching, which includes a local and a global matching stage. After registration of a fingerprint according to the model, a very tight matching threshold can be used during the matching stage. The algorithm gives a higher matching score compared to the rigid matching algorithm for deformed fingerprints.

As we know, adjacent features of a minutia is very important for matching. In this paper, we aim to present a novel fingerprint adjacent feature of a minutia named the adjacent feature vector, or AFV for fingerprint matching. The rest of the paper is organized as follows. Section 2 describes our proposed AFV in detail. Our robust minutiae matching method is presented in Section 3. The experimental results are reported in Section 4. Section 5 concludes our paper.

2. Adjacent feature vector (AFV)

Given a fingerprint image, we initially calculate the orientation at each pixel by Bazen and Gerez's method (2002), and we assume all the orientation in this paper are radian variant or radian constant. Fig. 1 illustrates an Adjacent Feature Vector. If a is a minutia of a fingerprint, $\overrightarrow{ab}$ indicates the direction of minutia a, and t_1, t_2, t_3 and t_4 are the four adjacent points satisfying $|at_1| = |at_2| = |at_3| = |at_4| = ADis$, $\angle bat_1 = \pi/4$, $\angle bat_2 = 3\pi/4$, $\angle bat_3 = 5\pi/4$, and $\angle bat_4 = 7\pi/4$, where $ADis$ is a constant. We assume the orientation at t_1, t_2, t_3 and t_4 are θ_1, θ_2, θ_3 and θ_4, respectively. If t_i belongs to the

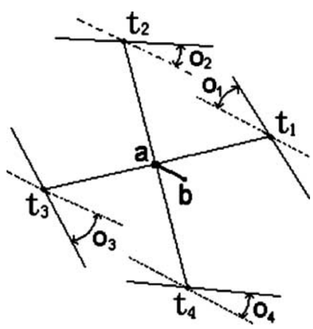

Fig. 1. Illustration of adjacent feature vector.