

Fingerprint Representation and Matching in Ridge Coordinate System

Jianjiang Feng and Anni Cai

Beijing University of Posts and Telecommunications

fjianjiang@263.net, annicai@bupt.edu.cn

Abstract

Fingerprints are generally represented and analyzed in Cartesian or polar coordinates. In this paper, however, fingerprints are represented and analyzed in a novel coordinate system, called ridge coordinate system (RCS), which is based on a ridge and an oriented point on the ridge. Using RCSs based on lots of ridges, we obtain a robust representation scheme of the ridge (skeleton) image of a fingerprint. Based on this representation, an alignment-based fingerprint matching algorithm is proposed. Experimental results on FVC2002 demonstrate the validity of the proposed algorithm.

1. Introduction

Fingerprints have been used in identification of individuals for many years because of the famous fact that each finger has a unique pattern. According to fingerprint representation, the existing fingerprint matching algorithm can be classified into three families: image correlation [1], minutiae-based [2, 3], ridge-based [4], and ridge feature based [5]. Among them the most widely used one is methods based on minutiae-based.

The approach in this paper is ridge-based. Comparing with raw images, ridge images are invariant to brightness, contrast, ridge thickness. Comparing with minutiae and FingerCode, ridge images preserve more information and more discriminating power.

In spite of these advantages of ridge-based representation, when designing a matching algorithm based on such representation, we still face some of the problems faced by the other families of approaches. The problems include unknown alignment parameters, noise and partial overlap.

To deal with these problems, instead of comparing two ridge images directly, we construct lots of representations from the ridge image. These representations are invariant to translation and rotation,

insensible to noise and partial overlap, and discriminative.

Each of these representations is constructed based a novel coordinate system, called ridge coordinate system (RCS), which is based on a ridge and an oriented point on the ridge. The coordinates of a point p in RCS contain two components: the ridge count between p and the projecting point p' of p on the spanning ridge, and the position of p' on the spanning ridge. (Details are given in section 2). Compared with Cartesian coordinates or polar coordinates, ridge coordinates are insensitive to nonlinear deformation and carry information about the intermediate image region.

Based on the RCS representations, a matching algorithm is proposed, which first find a pair of most similar RCS representations, and pairs the base ridges of this pair of RCSs, and then incrementally pairs other ridges. Our algorithm can be regarded as an alignment-based algorithm. Two alignment-based matching algorithms, minutiae orientation descriptors by Tico and Kuosmanen [2], and minutiae local structures by Jiang and Yau [3], have been implemented to compare with our algorithm. Experiments on FVC2002 databases [6] show the proposed algorithm is superior to the two others.

The rest of the paper is organized as follows. In section 2, RCS and fingerprint representation in RCS is presented. In section 3, fingerprint matching in RCS is described. In section 4, experimental results are given. Finally in section 5, conclusions and future work are presented.

2. Fingerprint representation in RCS

2.1. Ridge extraction

Given a gray-scale fingerprint image, a feature extraction algorithm developed in our laboratory is used to produce a thinned image and a set of minutiae. This algorithm follows the common steps adopted in most minutiae-based fingerprint matching systems,

including directional filtering, thresholding, thinning and minutiae detecting.

Then ridges are extracted from the ridge image and sampled at a fixed interval (6 pixels in our experiments). Thus, a set of ridges are represented as a list of sampled points. In the following, sampled point is called point for simplicity.

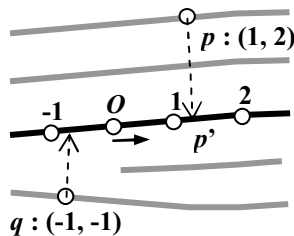


Figure 1. Ridge coordinates. O is the origin and its orientation is marked with arrow. The r coordinates of the points on the r -axis (black ridge) are labeled. The coordinates of two points, p and q , are shown.

2.2. Ridge coordinates

A ridge coordinate system (RCS) is defined based on a ridge (called r -axis) and an oriented point on it (called *origin*, denoted as O). See Figure 1 for an example. The orientation of the origin points to one of its two neighbors on the r -axis, and gives an order for the points on the r -axis.

The ridge coordinate of a point include two components: (r, c) , where r and c are analogous to discrete x and y in Cartesian coordinates, respectively. In the following, the r coordinates of the points on the r -axis are first defined, and then the r and c coordinates of any point are defined.

Let the r coordinate of O be 0. Define the side pointed by O as the positive side of the r -axis. Then the r coordinates of the points on the positive side of the r -axis are 1, 2, ..., and those on the negative side are -1, -2, -3, The orientation of a point p_i on the r -axis is defined as $\overrightarrow{p_i p_{i+1}}$, where i and $i+1$ are r coordinates of two points.

Given a point (e.x. p in Figure 1) in the fingerprint, draw a line passing p and normal to the local ridge direction at p . Assume that the line intersects at p' with the r -axis. This action is called projecting, the line the projecting line, and p' the projecting point of p . The r coordinate of p' is equal to that of the nearest sampled point on the r -axis, and the r coordinate of p is equal to that of p' .

The absolute value of the c coordinate of p is the ridge count (2 in this example) between p and p' (including the ridge of p , but excluding the ridge of p'). The c coordinate of the points on the r -axis are all 0.

The sign of the c coordinate of p is defined as follows. Move from p' towards the positive direction of the r -axis. If p is on the left side, the sign is +; otherwise, it is -.

2.3. Origin

This section discusses how to select origins. It is natural to use a minutia as origin and the ridge that the minutia belongs to as r -axis. As will be shown, it can also be generalized to other points and ridges. In the following subsections, we will treat the three cases where termination, bifurcation or other point acts as origin, respectively.

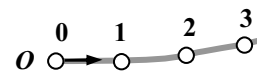


Figure 2. Termination is used as the origin (orientation is marked with arrow). The r coordinate of each point on the r -axis is labeled.

2.3.1 Termination as origin. The termination point is used as the origin, and its ridge is used as the r -axis. The r coordinate of the origin is 0, and that of the next sampled point is 1, and so on (see Figure 2).

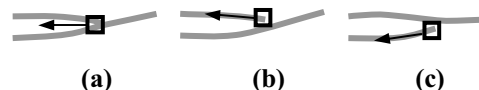


Figure 3. Regard one bifurcation (a) as two possible terminations (b or c).

2.3.2 Bifurcation as origin. In many cases, the difference between bifurcation and termination is subtle. Due to different pressure, skin states, and minutiae detection algorithms, bifurcations may appear as terminations and vice versa. To take this fact into account, we regard one bifurcation as two possible terminations (see Figure 3). Then the above definitions about terminations naturally apply to bifurcations. Therefore, there are two RCSs for a bifurcation. In this way, even if a pair of corresponding minutiae appears as different types in two fingerprints, it is guaranteed that one of the two RCSs of the bifurcation is consistent with the RCS of the termination each other.

2.3.3 Other point as origin. Based on a minutia, we can generalize the definition of origin to other points. Take a termination as an example (see Figure 4). Project O_0 on the surrounding ridges and obtain some projecting points, such as O_{-1} , O_1 , and O_2 in Figure 4. Define the positive direction of the projecting line as the orientation of O_0 plus 90 degree. The whole region

is segmented into left side and right side by the projecting line. The orientations of the projecting points are defined as pointing to the same side as O_0 , namely the right side. Then these projecting points can be used as origins and the ridges they belong as r -axes. O_0 is called parent origin of these origins, and these origins are called child origins of O_0 . Accordingly, RCS of O_0 are called parent RCS, and RCSs of child origins are called child RCSs. While projecting, the coordinates of O_0 in child RCSs are also obtained. Since the r coordinates are all 0, each child RCS can be uniquely determined by the c coordinate of O_0 , which is called the ID of the child RCS. In Figure 4, the subscript of each child origin is just its ID.

The main reason of using additional points as origins is to enlarge the region covered by a RCS, as the r -axis of a minutia may be too short to cover most region of a fingerprint. Considering the computational complexity, a parent origin has at most two child origins, namely O_{-1} and O_1 .

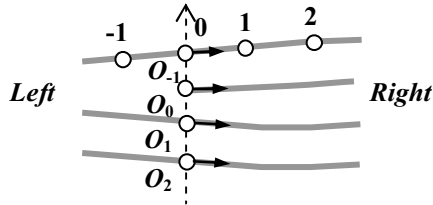


Figure 4. Based on O_0 , more origins, O_{-1} , O_1 , O_2 ,... (orientation is marked with arrow) are obtained. The r coordinates of points on one of the r -axes are labeled.

2.4. Fingerprint representation

To make the representation robust to noise and occlusion, we using each r -axis as the base r -axis, and record the properties of other r -axes (called associate r -axis) in the RCS of the base r -axis.

For a point p on an associate r -axis, in addition to the (r, c) coordinates, there are five additional properties of p , which can also be recorded to increase the discriminative power of the representation. They are the r coordinate of p in the associate r -axis (denoted as r'), the ID of the associate r -axis (denoted as c'), the ID of the parent minutia of the associate r -axis (denoted as m), the type of the parent minutia (denoted as t), and the relative orientation between p and the projecting point p' of p (denoted as s). If the parent minutia is a termination, $t=0$. If the parent minutia is a bifurcation, $t=1$ or 2 in the two cases (see Figure 3). The projecting line segments the whole region into two sides. If p and p' point to the same side, $s=1$; otherwise, $s=-1$.

Therefore, for each point in the RCS, a 7-dimension point vector $d_i = [r \ c \ r' \ c' \ m \ t \ s]$ is recorded. The set of all point vectors in a RCS is denoted as $D_j = \{d_i\}$. The set of all RCSs of a fingerprint is denoted as $D = \{D_j\}$.

3. Fingerprint matching in RCS

Based on the above representation, a simple greedy matching algorithm is designed. A table T is created to record the similarity score between each RCS in the first fingerprint and each RCS in the second fingerprint.

The similarity score between a pair of RCSs is computed by comparing each pair of point vectors in the two RCSs and counting the number of matched point vectors. A pair of point vectors, $d_1 = [r_1 \ c_1 \ r'_1 \ c'_1 \ m_1 \ t_1 \ s_1]$ and $d_2 = [r_2 \ c_2 \ r'_2 \ c'_2 \ m_2 \ t_2 \ s_2]$, is said to be matched, if $c_1 = c_2$ & $c'_1 = c'_2$ & $s_1 = s_2$ & $|r_1 - r_2| < 3$ & $|r'_1 - r'_2| < 3$. Organizing all point vectors according each dimension of vector can help to speed up the comparing process.

Based on the table, a greedy matching algorithm is designed. First, the pair of RCSs with the maximum score is assumed to be matched. Then all columns and rows in the table that violate one-by-one match constraint are cleared. The corresponding points on the two r -axes are matched. A rigid pose transformation is estimated based on the matched points. All minutiae pairs that are incompatible with the estimated transformation are identified and corresponding entries in the table are cleared. The 2nd, 3rd, ..., pairs of RCSs are found in turn by searching for the maximum score and updating the table. This process proceeds until all entries of the table are 0. Finally we get a list of matched r -axes, denoted as $\{a_i, b_i\}_{i=1}^m$, and all matched points on the matched r -axes are denoted as $\{p_i, q_i\}_{i=1}^n$.

The matching score between two fingerprints is computed as:

$$score = \frac{\sum_{i=1}^m T(a_i, b_i)}{1 + distortion}$$

$$distortion = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \frac{\|p_i p_j\| - \|q_i q_j\|}{\max(\|p_i p_j\|, \|q_i q_j\|)}$$

where $\|p_i p_j\|$ denote the length of the line segment $p_i p_j$.

4. Experiments

We have conducted experiments on db1_a and db3_a from FVC2002 databases [6]. The performance evaluation protocol used in FVC2002 has been adopted.

Two alignment-based fingerprint matching algorithms [2, 3] are implemented to compare with the

proposed algorithm. In the implementation of Jiang's algorithm, the same parameters as [3] have been used. In the implementation of Tico's algorithm, the sampling point configuration is (27, 10), (45, 16), (63, 22), (81, 28), and the orientation similarity function adopted is $\exp(-16x)$.

The ROC curves of the three algorithms on the two databases are given in Figure 5. Note that the range of scores has been adjusted to [0, 1]. It can be observed that our algorithm outperforms the others on both databases. This may be due to: i) RCS representation is robust, and ii) distortion is considered when computing matching scores.

Using a PC with P4 2.0 GHz, the average matching times of our algorithm on db1_a and db3_a are 409 ms and 310 ms, respectively. The construction of RCS representation is the most time consuming part.

5. Conclusions

In this paper, we have defined a novel coordinate system for fingerprint representation and matching. The fingerprint representation scheme in this coordinate system captures the relationships between ridges, which are ignored by many minutiae-based fingerprint matching algorithms. Experiments demonstrate the superiority of our algorithm over the two minutiae-based algorithms.

Although the preliminary result of our system is promising, there is still a lot of work that has to be done. On one hand, matching very poor quality fingerprints in RCS should be further explored. On the other hand, the ridge extraction algorithm should be improved and some ridge quality evaluation method should be integrated into the system.

Acknowledgement

This work is supported by National Natural Science Foundation of China under grant 60472069.

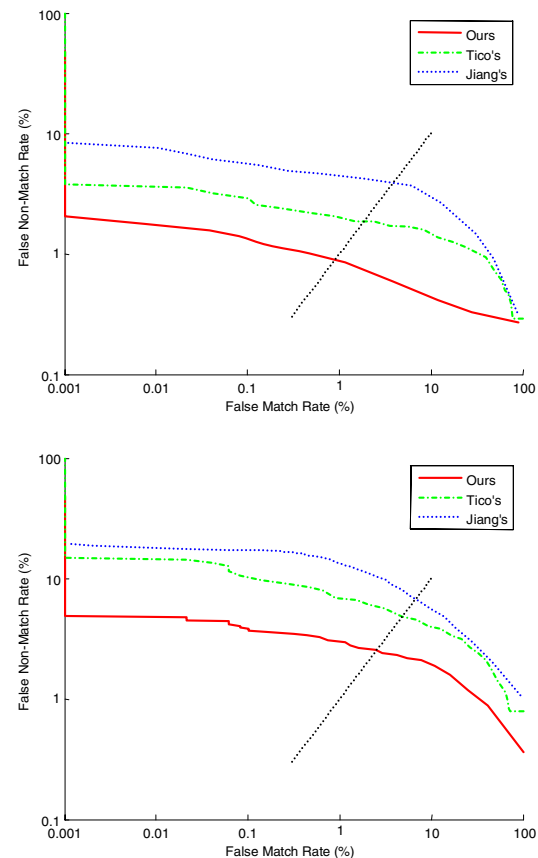


Figure 5. ROC curves on db1_a and db2_a.

References

- [1] C. Wilson, C. Watson, and E. Paek, "Effect of resolution and image quality on combined optical and neural network fingerprint matching", *Pattern Recognition*, 2000, 33 (2), pp. 317-331.
- [2] M. Tico and P. Kuosmanen, "Fingerprint matching using an orientation-based minutia descriptor", *IEEE Trans. PAMI*, 2003, 25 (8), pp. 1009-1014.
- [3] X. Jiang and W. Yau, "Fingerprint minutiae matching based on the local and global structures", in: *Proceedings of ICPR*, Barcelona, 2000, pp. 1038-1041.
- [4] D.K. Isenor and S.G. Zaky, "Fingerprint identification using graph matching", *Pattern Recognition*, 1986, 19 (2), pp. 113-122.
- [5] A.K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching", *IEEE Trans. Image Processing*, 2000, 9 (5), pp. 846-859.
- [6] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, A.K. Jain, "FVC2002: Second fingerprint verification competition", in: *Proceedings of ICPR*, Quebec City, Canada, 2002, pp. 744-747.