

Fingerprint Indexing Using Ridge Invariants

Jianjiang Feng and Anni Cai

Beijing University of Posts and Telecommunications

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

Abstract

Indexing large fingerprint databases is an important and challenging problem. In this paper, an invariant-based fingerprint indexing scheme is proposed. Minutia and surrounding ridges are combined to form a substructure. The invariants describe binary relations between substructures. Experimental results on FVC2002 database demonstrate the validity of the proposed algorithm.

1. Introduction

Human identification by fingerprints is widely used in forensic and civilian applications. Accuracy and speed is two key factors when choosing an automatic fingerprint recognition system. Most fingerprint matching algorithms can easily meet real-time requirement of verification systems. However, real-time fingerprint identification in large databases is a challenging problem. Assume that the time for matching a pair of fingerprints is t . It will take $M \times t$ to compare the input fingerprint against each model fingerprint in the database with M fingerprints. When the size of the database is huge (e.x., 10^6), the identification time is forbidden. Therefore some means should be adopted to reduce the number of candidate fingerprints that have to be dealt with by matching algorithms.

The existing approaches to reducing the number of candidates can be coarsely classified into three families.

- 1) *Exclusive classification*: A fingerprint is assigned to one of the several pre-specified types [5], such as five types. Then matching is performed only for fingerprints of the same class. This approach can not reducing the number of candidates greatly.
- 2) *Continuous classification*: In [2, 3], a fixed-length feature vector is computed from fingerprint directional field, and then fingerprint indexing is converted into a high dimension indexing problem.

- 3) *Invariants*: In preprocessing stage, invariants (or indices) are computed from all model fingerprints and stored in a hash table. In query stage, indices are computed from input fingerprints and used to search the hash table. The method by Germain et al. [4] uses invariants from minutiae triplet, including distances, angles, and ridge counts between each pair of minutiae. Bhanu and Tan [5] make some improvements on Germain's method.

The method proposed in this paper belongs to the third family, namely the invariant-based one. Different from the algorithms in [4] and [5], the invariants used in our algorithm are computed from ridges. In our approach, the role that a minutia plays is the reference of surrounding ridges. A minutia and surrounding ridges are combined to form a substructure. For two substructures in a fingerprint, using one of them as the base, we can construct lots of invariants to describe the relations between them.

Ridge invariants have the following advantages over the minutiae invariants used in [4, 5].

- 1) It carries more distinctive information. In some cases, the number of minutiae that can be paired between a query fingerprint and the corresponding model fingerprint is small. In an extreme case, such that the number is 3, only one minutiae triplet can be matched and therefore, the true model fingerprint obtains only one vote. However, our method might still produce many votes, depending on the distribution of ridges.
- 2) It is insensitive to nonlinear deformation and noise. Obviously, ridge counts are invariant with respect to nonlinear deformation. In addition, our ridge count estimation method is more robust than that in [4].
- 3) Evidence accumulating is simple and fast. In [4], evidence is accumulated by estimating the pose transformation and clustering. However, we just sum the votes of the same base.

We have tested the proposed algorithm and the minutiae triplet algorithm by Germain et al. on

FVC2002 databases [6]. Experimental results show the superiority of our algorithm over that of Germain et al.

The rest of the paper is organized as follows. In section 2, the overview of the indexing scheme and feature extraction algorithm are given. In section 3, invariants used are defined and index generation is described. In section 4, query is described. In section 5, experimental results are given. Finally in section 6, conclusions and future work are given.

2. Overview and feature extraction

The system consists of two stages: preprocessing stage and query stage. In the preprocessing stage, invariants are computed from model fingerprints, and are then inserted into the hash table together with source information (including model fingerprint ID and base substructure ID). In the query stage, invariants are computed from the input fingerprint and used to vote for (model, base) pairs with the same invariants. Finally, the first N candidate fingerprints sorted by the numbers of votes are outputted.

2.1. Feature 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 of minutiae-based fingerprint matching systems, including directional filtering, thresholding, thinning and minutiae detecting.

Then ridges are extracted from the thinned image and sampled at a fixed interval (6 pixels in our experiments). Thus, a set of ridges, denoted as SP , are represented as a 2-D list of sampled points, and the j th sampled point on the i th ridge is denoted as $SP(i, j)$, where i denotes the ridge number, and j the point number.

3. Index generation

This section is to describe how to generate indices from fingerprints. First, the definition of substructure is introduced. Then invariants (indices) are described.

3.1. Substructures

For each minutia, a substructure is constructed as follows. For a bifurcation, the substructure consists of 3 ridges that the bifurcation belongs to (see Figure 1 (a)). For a termination, in addition to the ridge that the termination belongs to, 2 more adjacent ridges are included (Figure 1 (b)). In the later case, each of the

two ridges is further segmented into two sub-ridges in the following way. Draw a line passing the termination, normal to the local ridge direction, which will segment each ridge into two sub-ridges. The two points (marked with circle in Figure 1 (b)) are called the reference points of the four sub-ridges. A minutia is the natural reference point of the ridges which it belongs to.

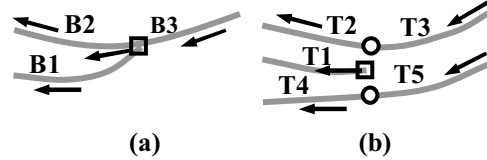


Figure 1. Substructure for bifurcation (a) and termination (b). Labels and flow directions of ridges are shown.

In a substructure, the flow directions (θ) of ridges are defined with respect to the direction of the minutia (see arrows in Figure 1). Note that the range of the flow direction is $[-180^\circ, 180^\circ]$.

Each ridge (or sub-ridge) in a substructure is assigned a label. For a bifurcation, the 3 ridges are labeled by checking the relative directions between the ridges. For a termination, the ridge that it belongs to is first labeled, and then the other 4 sub-ridges are labeled by checking the relative directions with respect to the first ridge. Figure 1 (a) and (b) show examples for labeling of substructures of bifurcation and termination respectively, where B and T mean bifurcation and termination, respectively. Note that the labeling is invariant with respect to rotation.

Given two substructures to be compared, we can determine the corresponding ridges in two substructures based on ridge labels. If the minutiae of two substructures are of the same type, the ridges with the same label are corresponding ridges; Otherwise, there are two possible correspondences: $\{(B1, T1), (B2, T2), (B3, T3)\}$ or $\{(B1, T4), (B2, T1), (B2, T5)\}$ (see Figure 1).

3.2. Invariants

The invariants (or indices) are generated by using binary relations between substructures. Since partial overlap is common in fingerprint matching, instead of using a single feature vector to characterize the relations between a pair of substructures, we use lots of local invariants, which describe the relations between sampled points in two substructures.

One of the two substructures is called base substructure (BS), and the other called associated substructure (AS). From each sampled point (called BP)

on each ridge (called BR) in BS, draw a line L (called scanning line) along $\alpha = \theta(BP) + \pi/2$, where $\theta(BP)$ denotes the flow direction of BP. If L intersects a ridge (AR) in AS at a point (the sampled point nearest to this point is called AP), an index is generated to describe this incident. The index consists of 5 components: $[l_b, p_b, l_a, p_a, c]$, where l_b denotes the label of BR, p_b the label of BP, l_a the label of AR, p_a the signed point count between AP and the reference point of AR, c the signed ridge count between BP and AP. Point count between two sampled points on the same ridge is the number of sampled points between them. Ridge count between BP and AP is the number of time that the scanning line between BP and AP intersects with ridges. The sign of p_b is positive, if $\alpha - \theta(AP) > 0$; otherwise it is negative. The sign of c is positive, if AR intersects with the positive side of the scanning line at AP; otherwise it is negative.

An example is given in Figure 2 to illustrate the generation of index. There are three minutiae m_1 , m_2 , and m_3 in the figure. Suppose that the substructure of m_1 is the base substructure. The substructures of the other two minutiae are associated substructures. One BP (marked with filled circle), its scanning line, and APs (marked with circles) are shown on each BR. Two of the 13 indices are given in the figure. This example also demonstrates that our method can produce lots of indices even if only three minutiae are available, whereas the minutiae triplet method can only produce one index in this case. In addition, for the only minutia triplet in this example, the ridge counts between minutiae are difficult to estimate.

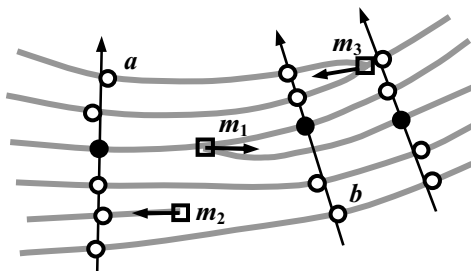


Figure 2. Index generation. The indices of a and b are $[B3\ 4\ B2\ -10\ 2]$ and $[B1\ 4\ T5\ -6\ -3]$, respectively.

Note that each substructure in a fingerprint will be used as the base. In the hash table, an index may have multiple (model, base) pairs.

4. Query

During the query stage, each index generated by the query fingerprint is used to lookup the same index in

the table and vote for all (model, base) pairs under this index. In practice, since the exchange of types of minutiae is allowed, not only the same index is looked up, all permitted indices are looked up. In addition, if the two base minutiae of query fingerprint and model fingerprint are of different types, for the base, two possible correspondences are voted separately.

After all indices from the query fingerprint have been used to vote, model fingerprints are sorted by vote numbers, and the top N model fingerprints are outputted.

5. Experiment

We have conducted experiments on DB1_A from FVC2002 databases. The image size is 388×374 , and the resolution is 500 dpi. The database consists of 100 fingers and each finger consists of 8 impressions. The first impression of each finger is used as the model fingerprint. Other 7 impressions are used as the query fingerprints. Totally 700 queries are performed.

We use the correct rate and penetration rate to evaluate a fingerprint indexing algorithm. A query is regarded as correct if the true fingerprint is contained in the candidate list outputted by the algorithm. The correct rate is the ratio of the correct queries to all queries. The penetration rate is the ratio of the size of the candidate list to the size of the whole database. For a given indexing algorithm, the correct rate increases when the penetration rate increases. The correct rate versus penetration rate graph, which shows the correct rates under different penetration rates, is an important tool to analyze the performance of an indexing algorithm.

The minutiae triplet algorithm by Germain et al. is implemented for comparison purpose. The correct rate versus penetration rate graphs of two algorithms are given in Figure 3. From the figure, it can be observed that our algorithm is slightly better than that of Germain et al. The query times for a single query fingerprint of both algorithms are less than 0.1 second.

An example is given to illustrate our algorithm. The query fingerprint and the corresponding model fingerprint are shown in Figure 4. The base substructures with the highest number of votes are highlighted. Using this query fingerprint, the numbers of votes of all model fingerprints in the database are plotted in Figure 5. The number of votes of the 1st model (the corresponding model) is 151. It is easy to distinguish the corresponding model from the others.

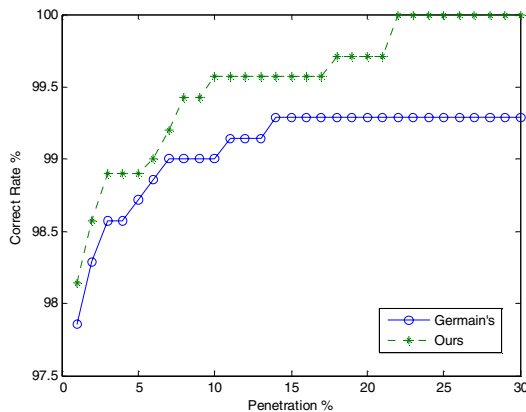


Figure 3. Correct rate versus penetration graphs of two algorithms on DB1 from FVC2002.

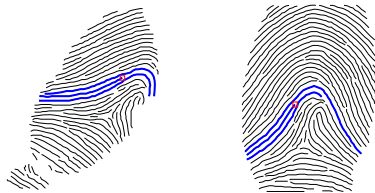


Figure 4. The left one is the query fingerprint, and the right one is the corresponding model fingerprint. Base substructures with the highest number of votes are highlighted.

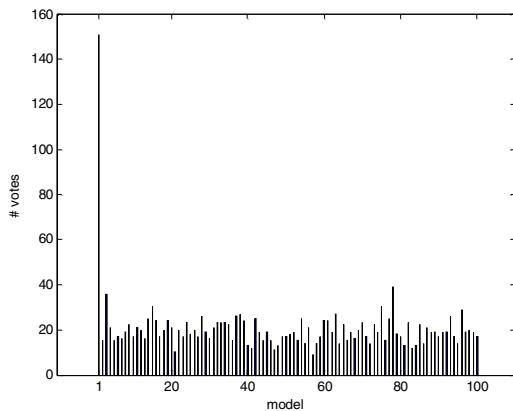


Figure 5. The number of votes is shown for each model in the database. The query fingerprint is the left one in Figure 4. The corresponding model (1st model) is easily recognized.

6. Conclusions

Indexing large fingerprint databases is an important and challenging problem. An efficient fingerprint indexing algorithm is proposed in this paper, which is based on ridge invariants. Compared to minutiae

triplets based indexing algorithm, our algorithm carries more information. The discrete nature of ridge invariants makes binning of invariants unnecessary. It is also easy to accumulate evidence. Promising results are obtained on public domain database.

Future work includes exploring performance on larger databases and fusion of multiple indexing schemes.

Acknowledgement

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

References

- [1] A.K. Jain, S. Prabhakar, and L. Hong, "A Multichannel Approach to Fingerprint Classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999, 21 (4), pp. 348-359.
- [2] A. Lumini, D. Maio, and D. Maltoni, "Continuous versus exclusive classification for fingerprint retrieval", *Pattern Recognition Letters*, 1997, 18 (10), pp. 1027-1034.
- [3] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, "Fingerprint Classification by Directional Image Partitioning", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999, 21 (5), pp. 402-421.
- [4] R.S. Germain, A. Califano, and S. Colville, "Fingerprint Matching Using Transformation Parameter Clustering", *IEEE Computational Science & Engineering*, 1997, 4 (4), pp. 42-49.
- [5] B. Bhanu and X. Tan, "Fingerprint indexing based on novel features of minutiae triplets", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, 25 (5), pp. 616-622.
- [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.