Distorted Fingerprint Indexing Using Minutia Detail and Delaunay Triangle

Xuefeng Liang, Tetsuo Asano
Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi, Ishikawa 923-1292, Japan
{xliang, t-asano}@jaist.ac.jp

Arijit Bishnu
Indian Institute of Technology
Kharagpur 721302, India
Arijit.Bishnu@iitkgp.ac.in

Abstract

This paper is concentrated on an accurate and efficient fingerprint indexing algorithm, which efficiently retrieves the top $\mathcal{N}$ possibly matched candidates from a huge database. In order to have ability of coping with distorted fingerprints, the proposed algorithm uses novel features, which are insensitive to distortion, formed by the Delaunay triangulation of minutiae set as the representation unit. These features include minutia detail and Delaunay triangle (its handedness, angles, maximum edge, and related angle between orientation field and edges). Experiments on database FVC 2000 and scanned fingerprints with heavy distortion show our algorithm considerably narrows down the search space in fingerprint databases and is also available for distorted fingerprints. We also compared with other indexing approaches, and results show our algorithm has a better performance, especially on fingerprints with heavy distortion. This algorithm has another significant advantage that is it provides the control points for fingerprint distortion compensation.

Key words: *fingerprint indexing, minutia detail, Delaunay triangulation, triplet, fingerprint distortion*

1 Introduction

A fingerprint recognition system is essentially a pattern recognition system that recognizes a person by determining the authenticity of a fingerprint characteristic possessed by that person. An important issue in designing a practical fingerprint recognition system is to determine how an individual is recognized. Depending on the application context, a fingerprint recognition system may be called either a *verification* system or an *identification* system.

- A **verification system** authenticates a person's identity by comparing the captured fingerprint characteristic with his/her own fingerprint template pre-stored in the system [6]. It conducts one-to-one comparison to determine whether the identity claimed by the individual is true. A verification system either rejects or accepts the submitted claim of identity, in other words, it answers the question: *Am I whom I claim I am* ?
- An **identification system** recognizes an individual by searching the entire template database for a match. It conducts one-to-many comparison to establish the identity of the individual [7]. In an identification system, the system establishes a subject's identity (or fails if the subject is not enrolled in the system database) without the subject having to claim an identity. It answers the question: *Who am I* ?

A naive identification system would just compare the given fingerprint with all entries in the database. However, for modern databases containing more than several million prints, *penetration rate P*[1] and *false acceptance rate FAR* are not accepted by any identification system when using this approach. Since all matches are performed, the required processing will have unacceptably long response time and the identification performance will be far too low.

To reduce these problems, *fingerprint classification* is employed. All fingerprints in the database are classified into five classes (Right Loop, Left Loop, Whorl, Arch, and Tented Arch) and stored in partial databases per class. The input fingerprint is also classified, and only matched to the fingerprints with corresponding class in partial database. If fingerprints were equally distributed into these five classes, the penetration rate would be reduced to $P = 0.2$. Therefore, the processing time and *FAR* are greatly reduced. However, the number of classes is small and real fingerprints are unequally distributed among them: more than 90% of the fingerprints belong to only three classes (Loops and Whorl). Furthermore, classification error and rejected fingerprint must be considered when classification is performed automatically. These lead to classification ap-

[1] Penetration rate measures the expected number of comparisons to be made. $P = \frac{E[\textit{number of comparisons}]}{n}$

proaches do not narrow down the search space enough in the database for an efficient identification of a fingerprint.

Another approach, called indexing, is employed to overcome this problem. Fingerprint indexing can significantly reduce the number of possibly matched candidates to be considered by verification step. Thus, fingerprint identification can be divided into following sequential steps: 1. fingerprint indexing; 2. fingerprint verification.

There have been several attempts to account for fingerprint indexing. R. Cappelli et al. proposed an indexing approach which gets a reasonable performance and identification time [4]. R.S. Germain et al. used the triplets of minutiae in their indexing procedure [5]. J.D. Boer et al. improved them by combining of multiple features (orientation field, FingerCode, and minutiae triplets) [3]. B. Bhanu and X. Tan did a improvement work on minutiae triplets [2]. In their work, they replaced the features by some novel ones. And experiments showed the improved approach had a better performance.

To make indexing algorithm more robust, fingerprint distortion must be considered. In this paper, we follow above works but use two more invariant features: *minutia detail* and *Delaunay triangle* of minutiae. As minutia detail represents not only the type but also the shape, it decreases the possibility of finding a wrong correspondence during indexing; Delaunay triangulation of a set of minutiae only creates $O(n)$ triangles. They both further reduce the search space in database. Moreover, Delaunay triangulation has an ability of describing local similarity of fingerprint. It is more reliable to the distortion than other triangulation algorithms. Experiments on database FVC 2000 and scanned fingerprints with heavy distortion show our indexing approach has a better performance.

The rest of this paper is organized as follows: Section 2 describes the definition of minutia detail and features of Delaunay triangle for indexing. Section 3 gives out the indexing score and algorithm. Experimental results and analysis are represented in Section 4. Finally, we conclude in Section 5.

2 Features for indexing

Due to cross correlation changing of minutiae under elastic distortion, it is better to choose features that are more invariant to distortion as index. How to choose the best features greatly affects the accuracy and response time of an identification system. Therefore, a good feature should be invariant for rigid and nonrigid transformation, lower computing cost, available for most fingerprints, and so on. As property of finger tips, there is an observation: even if an elastic distortion is applied to a fingerprint image, every minutiae always keeps its own shape and the same neighbor structure. With above requirement and observation, minutia detail and Delaunay triangle are employed to describe the local structure.

2.1 Minutia detail

2.1.1 Definition

Let p be a bifurcation minutia with three ridges incident upon it, namely, where r is the ridge before bifurcation, r_1 and r_2 are the two ridges after bifurcation. For each of r, r_1 and r_2, consider a line segment, which has length λ and is tangent to the corresponding ridge at p. Let these three line segments be $\overline{b_1p}, \overline{b_2p}$ and $\overline{ap}$ corresponding to r_1, r_2 and r, respectively (see Fig. 1). Let θ be the angle made by $\overline{ap}$, measured in counterclockwise direction with regard to x-axis. We call the group of the three line segments $\overline{b_2p}, \overline{b_1p}$ and $\overline{ap}$ as the *bifurcation detail* $\mathcal{B}(p, a, b_1, b_2)$ for minutia p.

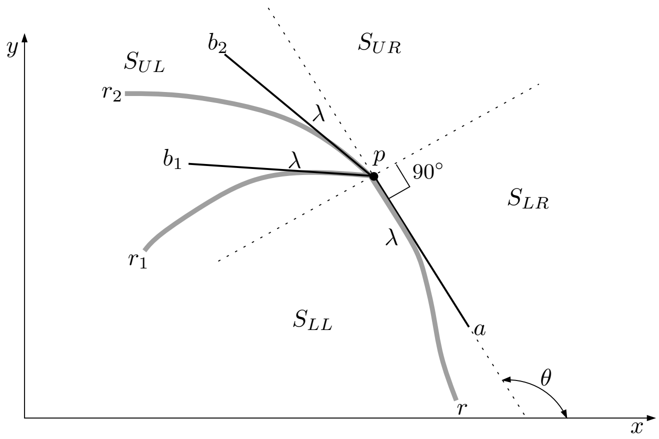

Figure 1. Minutia detail of bifurcation minutia at p.

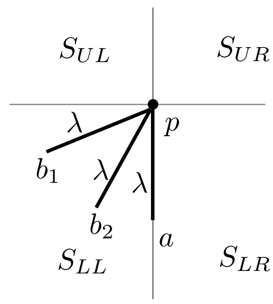

Figure 2. Bifurcation detail (case 1).

The Bifurcation detail $\mathcal{B}$ can have different shapes, depending on the mutual orientations of $\overline{b_1p}, \overline{b_2p}$ and $\overline{ap}$. The region around p can be divided into 4 quadrants, which are named as S_{LL} (lower left region), S_{UL} (upper left region), S_{UR} (upper right region), and S_{LR} (lower right region), as shown in Fig. 2. Each of b_1 and b_2 can lie in any one of these 4 regions, thereby making $4 \times 4 = 16$ possibilities. Out of these 16 possibilities, however, there will be 10 cases having distinct mutual positions of b_1 and b_2, considering the inter-changeability of b_1 and b_2. That is, for example, the case of $b_1 \in S_{LR}$ and $b_2 \in S_{UR}$, and the case of $b_2 \in S_{LR}$ and $b_1 \in R_{UR}$, which are 2 different cases in 16 possibilities, are the same in the later 10 cases. These 10 cases are enumerated in Table 1. Cases 2, 6 and 9 can have two subcases each, depending on the relative x-axis (or, y-axis) coordinates of b_1 and b_2, which will have differently shaped $\mathcal{B}$. Therefore, we get $7 + 6 = 13$ different bifurcation details, which are shown in Fig. 2 and Fig. 3.

Table 1. Different cases for bifurcation detail

Case	$b_1 \in$	$b_2 \in$	Condition
1	S_{LL}	S_{LL}	
2 (a)	S_{LL}	S_{UL}	$x[b_1] \leq x[b_2]$
2 (b)	S_{LL}	S_{UL}	$x[b_1] > x[b_2]$
3	S_{LL}	S_{UR}	
4	S_{LL}	S_{LR}	
5	S_{UL}	S_{UL}	
6 (a)	S_{UL}	S_{UR}	$y[b_1] \leq y[b_2]$
6 (b)	S_{LL}	S_{UR}	$y[b_1] > y[b_2]$
7	S_{UL}	S_{LR}	
8	S_{UR}	S_{UR}	
9 (a)	S_{UR}	S_{LR}	$x[b_1] \leq x[b_2]$
9 (b)	S_{UR}	S_{LR}	$x[b_1] > x[b_2]$
10	S_{LR}	S_{LR}	

For a valid bifurcation minutia p, the angle $\angle(b_1, p, b_2)$ should be less than both $\angle(b_1, p, a)$ and $\angle(b_2, p, a)$, which helps us to distinguish r form r_1 and r_2. Keeping this point in consideration, out of the above 10 cases, cases 3, 4 and 7 are not possible for a valid bifurcation minutia. Hence we have only $13 - 3 = 10$ possible differently shaped bifurcation details.

As there is only one ridge r incident upon ridge ending p. A ridge ending detail $\mathcal{T}(p, a)$ has just one line segment $\overline{ap}$ corresponding to r, which is defined as the same as bifurcation detail. So, only the line segment $\overline{ap}$ contributes to ridge ending detail, and its shape is unique. Thus, we take no account of it in indexing procedure.

2.1.2 Algorithm

The theoretical definition of line segments $\overline{b_1p}$, $\overline{b_2p}$ and $\overline{ap}$ defines that they are tangent to the corresponding ridges at

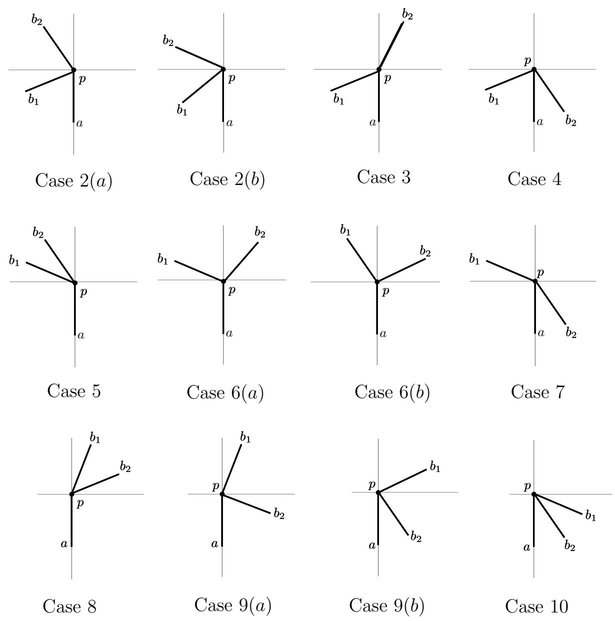

Figure 3. Bifurcation details (case 2 – 10).

minutia point p. But in discrete domain, it is a bit complicate in practice. An alternative approach simply connects the fifth skeleton point with minutia point as a line segment on each ridge. The algorithm of line segment generation is given out as follows:

First, three basic definitions are illustrated below.

1. *Start point*: The start point is one on the boundary of minutiae region (please refer Chapter 6 of [9]). The start points are considered as sources or elementary cells of the skeleton which grows up emitting from it. The boldfaced Euclidean distance transform (EDT) values in Fig. 4 are start points. Their non-minutia neighbors, which have a *9 EDT circle* covering two background components, are italic.

2. *Branch generation*: The set of points $\{p_i\}$ is called the *directional-neighborhood* of p, denoted by D_p, if they are in the 8-neighborhood of p and located within $\pm 45°$ slope changes from the current medial axis orientation of p. For example, using the 8-neighbors labeled $p_1, p_2, \ldots, p_8$ counterclockwise from the positive x axis of p, if p_7 and p are the skeleton points, the points p_2, p_3, p_4 are the directional neighbors of p, i.e. $D_p = \{p_2, p_3, p_4\}$. Several cases are illustrated below:

$$\begin{matrix} p_4 & p_3 & p_2 \\ . & p & . \\ . & p_7 & . \end{matrix} \qquad \begin{matrix} . & . & p_2 \\ p_5 & p & p_1 \\ . & . & p_8 \end{matrix}$$

$$\begin{array}{ccc} p_4 & p_3 & \cdot \\ p_5 & p & \cdot \\ \cdot & \cdot & p_8 \end{array} \qquad \begin{array}{ccc} \cdot & p_3 & p_2 \\ \cdot & p & p_1 \\ p_6 & \cdot & \cdot \end{array}$$

Note that the set of directional-neighborhood contains always three elements. The branch generation is to add the point which is the maximum of p's directional-neighborhood. That is to say:

$$p_{next} = \max_{p_i \in D_p} \{p_i\}$$

3. *Minutia point*: The minutia point is the one being the maximum (i.e. the largest EDT value locally) in extracted minutiae region. Occasionally, there would exist more than one maximum points in one minutiae region. In this case, the midpoint among them is chosen as the minutia point.

Input: Fingerprint image I_{EDT} represented by EDT values and minutiae regions (please refer Chapters 4 and 6 in [9]).

Output: Line segments incident upon the minutia p.

1. *Pick up a start point from minutiae region as initial skeleton point.*
2. *Choose its non-minutia neighbor that has a 9 EDT circle covering two background components as next skeleton point. If number of eligible neighbors is more than one, choose the midpoint of them.*
3. *Starting from start points and its chosen neighbor, the branch generation is used to add more skeleton points. For each branch, the branch generation halts at the fifth skeleton point.*
4. *Connect minutia point with the fifth skeleton point on each ridge to generate the corresponding line segment.*

Fig. 4 shows a instance of the skeleton of bifurcation minutia. Where the EDT values with underline construct the skeleton.

As minutia details can be obtained in minutiae extraction step, our indexing step does not cost additional computing time for them [9].

2.2 Delaunay triangle

To make identification system robust and efficient, we address ourselves to following points:

1. *Distortion*: There is the observation that, even if an elastic distortion is applied to a fingerprint image, every minutia always keeps the same neighbor structure. In other words, distortion does not change local similarity much. Some triangles that are used in [5, 2] may almost cover entire fingerprint region in an image. Obversely, they are greatly changed under heavy distortion. On the contrary, Delaunay triangulation partitions a whole fingerprint region into many smaller pieces and exactly describes the closest neighbor structures of minutiae. So, Delaunay triangle is more insensitive to distortion.

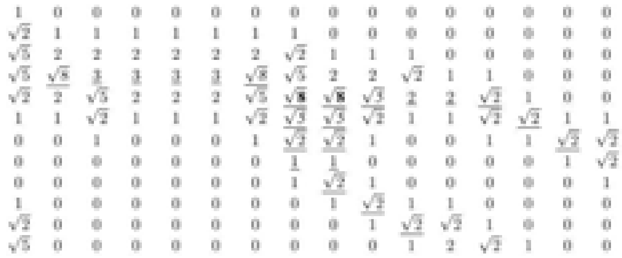

Figure 4. Skelton of bifurcation.

2. *Computing cost*: Fingerprint identification is a real-time application. Thus, a fast response is necessary. The approaches in [5, 2] exhaust all of possible triangles of minutia set in an image to ensure the maximal possible correspondences. Therefore, $O(n^3)$ triangles have to be compared during indexing (n is the number of minutiae). It is well-known, Delaunay triangulation creates $O(n)$ triangles. Therefore, The computing cost greatly decreases using Delaunay triangle when $n > 9$.

With the properties of Delaunay triangulation [1], there are some other advantages: 1. Insertion of a new point in a Delaunay triangulation affects only the triangles whose circum circles contain that point. As a result, noise affects the Delaunay triangulation only locally. 2. Delaunay triangles are not *skinny*. This is also very desirable in our application since the computation of the geometric transformations between fingerprints is based on corresponding minutiae triangles. Using skinny triangles can lead to instabilities and errors [8].

Minutia detail and the features derived from Delaunay triangle of minutiae form the index, and are ordered by fundamentality.

$$(\mathcal{M}_i^j, H, \alpha_{min}, \alpha_{med}, L_{max}, \phi_i)^2{}_{i=1,2,3;j=1,2,\ldots,10}$$

- $\mathcal{M}_i^j$ is the minutia detail of each vertex of triangle. If the vertex is a bifurcation, it is identified as the corresponding bifurcation detail case $\mathcal{B}_i^j$. If it is a ridge ending, $\mathcal{T}_i$ is marked.

[2] $H, \alpha_{min}, \alpha_{med}, L_{max}$ have same definitions in [2].

- α_{min} and α_{med} are the minimal and medial angles in the triangle respectively. According to angles, the vertexes of angle $\alpha_{max}, \alpha_{min}, \alpha_{med}$ are labeled as P_1, P_2, P_3 (See Fig. 5 for illustration).

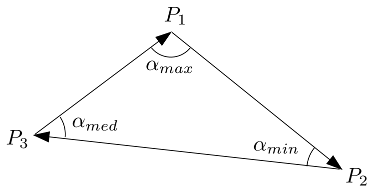

Figure 5. Definition of triangle labels.

- H is the triangle handedness. Let $Z_i = x_i + jy_i$ be the complex number corresponding to the location (x_i, y_i) of point P_i. Define $Z_{21} = Z_2 - Z_1, Z_{32} = Z_3 - Z_2$ and $Z_{13} = Z_1 - Z_3$. Let triangle handedness $H = sign(Z_{21} \times Z_{32})$.

- L_{max} is the length of the longest edge in the triangle.

- ϕ_i is the difference between angles of two edges of P_i and orientation field at P_i. For instance, let θ_1^r is the angle between edge $\overline{P_1P_2}$ and orientation field at P_1; similarly, θ_1^l is the angle between edge $\overline{P_3P_1}$ and orientation field at P_1. ϕ_1 is the difference between θ_1^r and θ_1^l in clockwise direction.

$$\phi_i = \theta_i^r - \theta_i^l$$

where $-180° < \phi_i < 180°$, $0° < \theta_i^r < 180°$, and $0° < \theta_i^l < 180°$.

2.3 Conditions of triangles match

To find the correct corresponding Delaunay triangles among the input fingerprint and fingerprints in the database, following conditions are given out:

$$\mathcal{M}_i = \mathcal{M}_i'$$

$$H = H'$$

$$|\alpha_{min} - \alpha'_{min}| < T_\alpha; \quad |\alpha_{med} - \alpha'_{med}| < T_\alpha$$

$$|L_{max} - L'_{max}| < T_L$$

$$|\phi_i - \phi'_i| < T_\phi$$

where T_α, T_L, T_ϕ are the thresholds.

3 Indexing score and algorithm

I and I_i are input fingerprint and fingerprint in database respectively, $i = 1, 2, \ldots, N$, where N is the number of fingerprints in the database. Suppose that there are n potential corresponding minutiae m_j $(j = 1, 2, \ldots, n)$ in each fingerprint. r_j is the number of matched triangles, which include m_j, in I and I_i. We define the index score of image I_i as:

$$S_i = c \sum_{j=1}^{n} r_j$$

where c is a constant.

Indexing algorithm is given out as follows:

For each triangle in I_i, compute $(\mathcal{M}_i^j, H, \alpha_{min}, \alpha_{med}, L_{max}, \phi_i)$, and use them to construct hash tables for fast indexing in a database.

1. *Apply Delaunay triangulation on minutiae set in input fingerprint I.*
2. *For each triangle in I, do 3 and 4.*
3. *Compute $(\mathcal{M}_i^j, H, \alpha_{min}, \alpha_{med}, L_{max}, \phi_i)$ for the triangle.*
4. *Search the index space using $(\mathcal{M}_i^j, H, \alpha_{min}, \alpha_{med}, L_{max}, \phi_i)$ under conditions given out in section 2.3. If the triangle satisfies the conditions, then take it as successful correspondence of the triangle in database.*
5. *Suppose that M_i corresponding triangles, which belong to the fingerprint I_i, are found. Let $M_{max} = \max\{M_i\}$. If $M_{max} < T_M$, then reject the input fingerprint, where T_M is a threshold. Otherwise, do 6 and 7.*
6. *Compute indexing score S_i based on M_i for I_i.*
7. *Sort S_i in a descending order, output top $\mathcal{N}$ possible matched prints.*

4 Experimental results

We evaluated our method by testing it on Database FVC2000 which consists of 880 fingerprints, 8 prints each of 110 distinct fingers. These images are captured by a capacitive sensor with a resolution of 500dpi, resulting in images of 364×256 pixels in 8 bit gray scale. The first of these prints is used to construct the database, while the other seven prints are used to test the indexing performance. In addition, we also scanned 180 fingerprints with heavy distortion from a FUJITSU Fingerprint Sensor (model: FS-210u), 3 prints each of 60 distinct fingers. Size and resolution are 300×300 and 500dpi respectively. Similarly, the first of these prints is used to construct the database, the others are used to test. Thresholds are different for the two data sets: for database FVC 2000, $T_\alpha = 5°, T_L = 10$ pixels, $T_\phi = 5°$; for scanned fingerprints with heavy distortion, $T_\alpha = 8°, T_L = 20$ pixels, $T_\phi = 10°$.

Indexing performance curves plotting the correct index against the penetration rate at various thresholds are presented in Fig. 6, 7, 8. Where correct index is defined as the percentage of correct fingerprints are selected.

First, a experiment has been performed to investigate the effect of using minutia detail against minutia type for indexing the database.

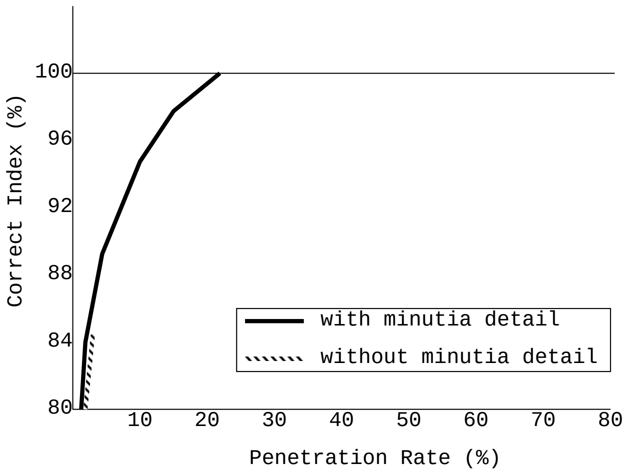

Figure 6. Comparison of indexing algorithms with minutia detail and without minutia detail.

Fig. 6 indicates the size of the part of the database that has to be searched in order to achieve a fixed probability that the corresponding fingerprint is found.

Although, real data prove that more than 70% bifurcation minutiae belong to bifurcation detail cases 5, 6, and 8. Minutia detail provides more classes than simple minutia types, and narrows down the search space in a database. Fig. 6 shows that the search space of indexing algorithm using minutia detail is around 10% less than the space of algorithm using minutia type when they achieve 100% correct index rate.

Second, two experiments have been carried out to evaluate the performances of our algorithm and the algorithm based on triplets on database FVC 2000 and scanned fingerprints.

On database FVC 2000, Fig. 7 shows our algorithm gets a smaller search space when two algorithms achieve the same correct index rate. This might be explained by the fact that the algorithm based triplets creates $O(n^3)$ triangles. This number is much greater than $O(n)$, which is the number of triangles in our algorithm, when $n > 9$. Obviously, triplets based algorithm could find more matched triangles between two possible matched fingerprints and has more computing cost. Therefore, its search space in database is larger than ours when using a same T_M.

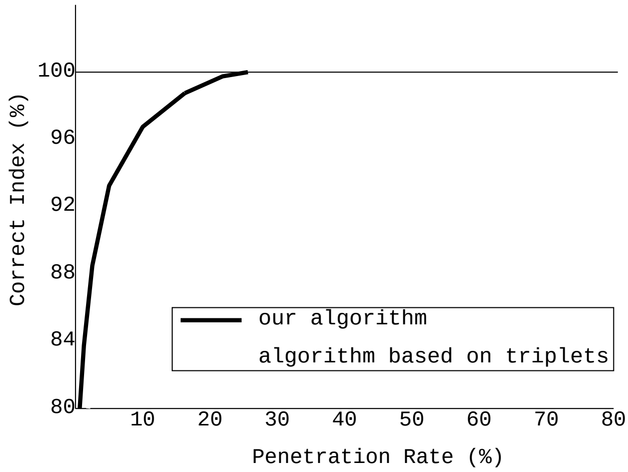

Figure 7. Comparison of our algorithm and the algorithm based on triplets on database FVC 2000.

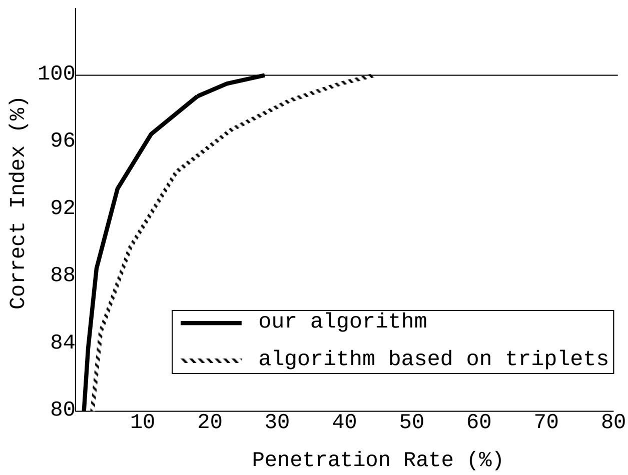

Figure 8. Comparison of our algorithm and the algorithm based on triplets on scanned fingerprints with heavy distortion.

On the scanned fingerprints with heavy distortion, Fig. 8 shows a similar conclusion as above. Since elastic distortion changes the geometric relationship among minutiae, we decrease T_M in algorithm. As the same reason in previous paragraph, triplets based algorithm finds more matched triangles even they are not real corresponding between two fingerprints.

5 Conclusions

In this paper, an indexing algorithm using minutia detail and Delaunay triangle is proposed. It has been shown that this algorithm is able to search a fingerprint database more efficiently. The reasons are follows:

- Minutia detail provides more minutiae classes than minutia types, and further reduce search space. As they can be obtained in minutiae extraction step, our indexing algorithm does not cost additional computing time.

- Delaunay triangle has an ability of description of local similarity. Thus, our algorithm is more insensitive to elastic distortion. Since Delaunay triangulation creates $O(n)$ triangles, it is much less than the number of triangles created by algorithm based on triplets. More redundant or wrong matched triangles are avoided. Simultaneously, the search space in a database is reduced when using a same T_M.

The experiments on database FVC 2000 and scanned fingerprints with heavy distortion also prove our algorithm has a better ability of reducing the number of possible matched prints for next step. A further improvement might be achieved by combining other features of fingerprint, suitable combination rule and class rank method.

This indexing algorithm has another significant advantage. All of correspondences found in indexing can work as the control points for estimating the nonlinear mapping function between two fingerprints during distortion compensation.

6 Acknowledgment

Research for X.F. Liang was conducted as a program for the "Fostering Talent in Emergent Research Fields" in Special Coordination Funds for Promoting Science and Technology by Ministry of Education, Culture, Sports, Science and Technology. Research for T. Asano was partially supported by the same Ministry, Grant-in-Aid for Scientific Research (B) and Exploratory Research. FUJITSU LABORATORIES LTD. was extremely helpful in providing devices for our research.

References

[1] M. Berg, M. Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry (Algorithms and Application)*, chapter 9. Springer-Verlag, Berlin Heidelbarg, 1997.

[2] B. Bhanu and X. Tan. Fingerprint indexing based on novel features of minutiae triplets. *IEEE Trans. on PAMI*, 25(5):616–622, 2003.

[3] J. Boer, A. Bazen, and S. Cerez. Indexing fingerprint database based on multiple feature. In *Proc. of the ProRISC 2001 Workshop on Circuits, Systems and Singal Processing*, 2001.

[4] R. Cappelli, D. Maio, and D. Maltoni. Indexing fingerperint databases for efficicent $1 : n$ matching. In *Proc. Sixth Int. Conf. on Control, Automation, Robotics and Vision*, Singapore, 2000.

[5] R. Germain, A. Califano, and S. Colville. Fingerprint matching using transformation parameter clustering. *IEEE Computational Science and Eng.*, 4(4):42–49, 1997.

[6] A. Jain, L. Hong, and R. Bolle. On-line fingerprint verification. *IEEE Trans. on PAMI*, 19(4):302–313, 1997.

[7] A. Jain and S. Pankanti. Automated fingerprint identification and imaging systems. In *Advances in Fingerprint Technology*. Elsevier Science, 2 edition, 2001.

[8] Y. Lamdan, J. Schwartz, and H. Wolfson. Affine invariant model-based object recognition. *IEEE Trans. on Robotics and Automation*, 6(5):578–589, 1990.

[9] X. Liang. *Fingerprint image analysis using computational geometric techniques*. PhD thesis, Japan Advanced Institute of Science and Technology, 2006.