A Robust Fingerprint Indexing Scheme Using Minutia Neighborhood Structure and Low-Order Delaunay Triangles

Xuefeng Liang, Arijit Bishnu, and Tetsuo Asano

***Abstract*—Fingerprint indexing is a key technique in automatic fingerprint identification systems (AFIS). However, handling fingerprint distortion is still a problem. This paper concentrates on a more accurate fingerprint indexing algorithm that efficiently retrieves the top $\mathcal{N}$ possible matching candidates from a huge database. To this end, we design a novel feature based on minutia neighborhood structure (we call this minutia detail and it contains richer minutia information) and a more stable triangulation algorithm (low-order Delaunay triangles, consisting of order 0 and 1 Delaunay triangles), which are both insensitive to fingerprint distortion. The indexing features include minutia detail and attributes of low-order Delaunay triangle (its handedness, angles, maximum edge, and related angles between orientation field and edges). Experiments on databases FVC2002 and FVC2004 show that the proposed algorithm considerably narrows down the search space in fingerprint databases and is stable for various fingerprints. We also compared it with other indexing approaches, and the results show our algorithm has better performance, especially on fingerprints with distortion.**

***Index Terms*—Fingerprint distortion, fingerprint indexing, low-order Delaunay triangle, minutia detail, order k-Delaunay triangulation, triplet.**

I. 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 [1].

- A verification system authenticates a person's identity by comparing the captured fingerprint characteristic with his or her own fingerprint template prestored in the system [2]. It conducts a 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 comparisons to establish the identity of the individual [3]. 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 the false acceptance rate (FAR) are not accepted by an identification system using this approach. Since all matches are performed, the required processing will have an unacceptably long response time, and the identification performance will be far too low.

To reduce these problems, fingerprint classification is employed [4]–[6]. All fingerprints in the database are classified into five classes (right loop, left loop, whorl, arch, and tented arch) and stored in a partial databases per class. The input fingerprint is also classified, and is only matched to the fingerprints of the corresponding class in the 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 would be 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) [1]. Furthermore, classification error and rejected fingerprints must be considered when classification is performed automatically. These lead to classification approaches which do not narrow down the search space adequately in the database for an efficient identification of a fingerprint.

Another approach, called indexing, is employed to overcome this problem. In indexing, the crucial idea is to measure the local similarity among input and template prints to report the top $\mathcal{N}$ most similar template prints. Since it is not necessary to consider every model as in final matching, fingerprint indexing signifi-

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

Manuscript received January 12, 2007; revised August 21, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Davide Maltoni.

X. Liang is with Ubiquitous Functions Research Group (UFRG), Intelligent Systems Research Institute (ISRI), National Institute of Advanced Industrial Science and Technology, Lbaraki 305-8568, Japan (e-mail: liangxuef@gmail.com; xf-ryan@aist.go.jp).

A. Bishnu is with the Computer Science and Engineering Department, Indian Institute of Technology, Kharagpur 721302, India (e-mail: arijit.bishnu@iitkgp.ac.in).

T. Asano is with the Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan (e-mail: t-asano@jaist.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2007.910242

cantly reduces the number of candidates to be considered by the verification step. Thus, fingerprint identification can be divided into the following sequential steps: 1) fingerprint indexing and 2) fingerprint verification.

There have been several attempts at fingerprint indexing. Cappelli *et al.* proposed an indexing approach with reasonable performance and identification time [7]. Germain *et al.* used the triplets of minutiae measuring local similarities in their indexing procedure [8]. Boer *et al.* improved on these studies by combining multiple features (orientation field, FingerCode, and minutiae triplets) [9]. Bhanu and Tan improved the work on minutiae triplets [10]. In their work, they replaced the features by some novel ones (e.g., triangle angles, handedness, type, and direction). Bebis *et al.* applied Delaunay triangulation instead of exhausting all triplets of the minutiae set [11]. This improvement saves computation cost and reduces the possibility of mismatch. However, with massive experiments, we find Delaunay triangulation may not be stable if even a tiny distortion is applied on prints.

To make the indexing algorithm more robust, fingerprint distortion must be considered. In this paper, we follow the aforementioned works, but design two novel features that are insensitive to distortion.

1) Minutia detail. Minutia detail is an approximate structure of minutia. Thus, it represents not only the type but also the shape. Minutia detail decreases the possibility of finding inaccurate correspondences during indexing.
2) Low-order Delaunay triangle (LoD triangle). It is known that Delaunay triangulation of a minutiae set may change under small shifts of minutiae caused by distortion. The LoD triangle is designed to prevent indexing performance degeneration due to such changes in Delaunay triangulation. Thus, it has a higher ability to tolerate distortion. Meanwhile, LoD triangles also inherit some of the properties of Delaunay triangles we need for fingerprint indexing. They are:
    1) uniqueness;
    2) creating only $O(n)$ triangles (This property reduces the time and space complexities, and indirectly narrows down the search space in database.);
    3) describing the topological structure of the three closest minutiae in a fingerprint.

    This algorithm is more reliable against distortion than other triangulation algorithms.

In the next section, we first describe the definition and algorithm of minutia detail; second, we sketch the motivation and definition of the LoD triangle; third, we design the indexing elements and matching conditions. In Section III, the whole indexing algorithm is given. Experimental results and analysis are presented in Section IV. Section V is the conclusion.

II. Features for Indexing

Due to cross-correlating changes of minutiae under elastic distortion, it is better to choose indexing features that are more invariant to distortion. The features chosen greatly affect the accuracy and response time of an identification system. Therefore, a good feature, which significantly narrows down the search space in a database, should be invariant for rigid and nonrigid transformation, offer lower computing cost, be available for most fingerprints, and so on. The following observation about the property of finger tips will help us design our method. Even in the presence of elastic distortion, every minutia keeps its own shape and similar neighboring structure (it may not be isomorphic). With the requirements just shown and observation, minutia detail and the LoD triangle are designed to perform the aforementioned tasks.

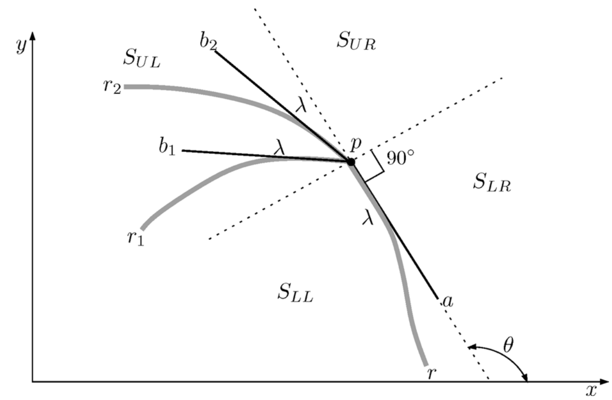


Fig. 1. Minutia detail of bifurcation at p.

A. Minutia Detail

1) Definition: Let p be a bifurcation minutia with three ridges incident upon it, where r is the ridge before bifurcation, r_1, and r_2 are the two ridges after bifurcation. For each 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 $\overline{ap}$ be measured in the counterclockwise direction with regard to the x axis. We call the group of the three line segments $\overline{b_2p}$, $\overline{b_1p}$, and $\overline{ap}$, the bifurcation detail $\mathcal{B}(p, a, b_1, b_2)$ for minutia p.

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 four quadrants, which are called S_{LR} (lower right region), S_{UR} (upper right region), S_{UL} (upperleft region), and S_{LL} (lowerleft region), as shown in Fig. 2. Each b_1 and b_2 can lie in any one of these four regions, thereby making $4 \times 4 = 16$ possibilities. Out of these 16 possibilities, however, there will be ten cases having distinct positions of b_1 and b_2 in relation to each other, considering the interchangeability 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 two different cases in 16 possibilities, are the same in the later ten cases. These ten cases are enumerated in Table I. 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 obtain $7 + 6 = 13$ different bifurcation details, which are shown in Figs. 2 and 3.

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 from r_1 and r_2. Keeping these points in consideration, out of the aforementioned ten cases, cases 3, 4, and 7

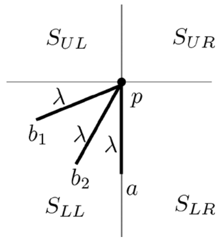


Fig. 2. Bifurcation detail (case 1).

TABLE I
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}	

cannot be possibly used as valid bifurcation minutiae. Hence, we have only $13 - 3 = 10$ possible differently shaped bifurcation details.

As there is only one ridge r incident upon ridge ending p, the ridge ending detail $\mathcal{T}(p, a)$ has just one line segment $\overline{ap}$ corresponding to r, which is defined as the same as the bifurcation detail. So, only the line segment $\overline{ap}$ contributes to ridge ending detail and its shape is unique. Thus, we do not take it into account in the indexing procedure.

2) Algorithm: The theoretical definition of line segments $\overline{b_1p}$, $\overline{b_2p}$, and $\overline{ap}$ states that they are tangent to the corresponding ridges at minutia point p_i, where $i = 1, \ldots, n$, n is the number of minutiae in a fingerprint. But in the discrete domain, it is a bit complicated to construct the line segment tangent to the ridge in practice. As an alternative approach, we simply connect the fifth skeleton point with the minutia point as the line segment on each ridge. The algorithm of line segment generation is given as follows:

Algorithm 1: Minutia detail generation.

Input: Skeletons of fingerprint ridges.

Output: Minutia detail of the minutia p_i.

1) **for** each minutia point p **do**.

2) **for** each branch of p **do**.

3) **while** the current point is not the fifth one on the ridge skeleton **do**.

4) go to the next point.

5) **end while**.

6) connect p with the fifth skeleton point to generate the corresponding line segment.

7) **end for**.

8) compute the mutual positions of b_1 and b_2 and report minutia detail.

9) **end for**.

As minutiae details represent the microstructure of minutiae in a fingerprint, they are unlikely to change under distortion. Moreover, they can be obtained in the minutiae extraction step and our indexing step does not require additional computing time for them [12]. Hence, minutiae details match the requirements of a good index.

B. Order k-Delaunay Triangle

To improve index selectivity, we need more invariants. The approaches in [8] and [10] consider triangles of minutiae to describe mutual relations of minutiae, and add attributes of triangles to the index. These approaches exhaust all possible triangles of minutia set in an image to ensure the maximum possible correspondences. As all possible triangles are considered, redundant triangles also come into play. These strategies lead to two negative side effects: 1) Some larger triangles may cover almost the entire fingerprint region. Obviously, such triangles are greatly changed under distortion and become useless for indexing and 2) more redundant triangles lead to a higher possibility of mismatch. On the contrary, Delaunay triangulation partitions a whole fingerprint region into many smaller pieces and exactly describes the closest neighbor structures of minutiae. It only creates $O(n)$ triangles, which are much less than the $O(n^3)$ triangles used in [8] and [10] (n is the number of minutiae).

A Delaunay triangulation is a geometric structure defined for a set of n points on the plane with the property that the interior of the circumcircle of each triangle contains no points of the set [13]. The empty circumcircles make Delaunay triangulation well shaped and unique. With these particular qualities, Delaunay triangulation has found use in several areas of fingerprint recognition [11], [14], [15].

Delaunay triangulation is defined by infinitely precise point coordinates. However, finger tips are soft tissues. When we press finger tips on the hard surface of a sensor, captured prints always contain more or less distortion, because of various sorts of pressures. Distortion changes the coordinates of neighboring minutiae. Thus, a natural question arises: Is Delaunay triangulation stable under distortion? After many experiments, our answer is possibly not.

In Fig. 4, we select an instance to show this issue. The diagram shows a subset of minutiae in a print and its Delaunay triangulation. We assume distortion is applied to it, causing the point p to move from the left side of the dashed line segment $\overline{ab}$ to the right side. This tiny shift of p changes the subgraph of Delaunay triangulation (represented by solid lines) completely. Regarding this issue, Abellanas *et al.* [16] and Khanban and Edalat [17] discussed the tolerance of Delauany triangulation from different viewpoints, exploring the supremum of shifts of points so that Delauany triangulation does not change. Here, we focus on

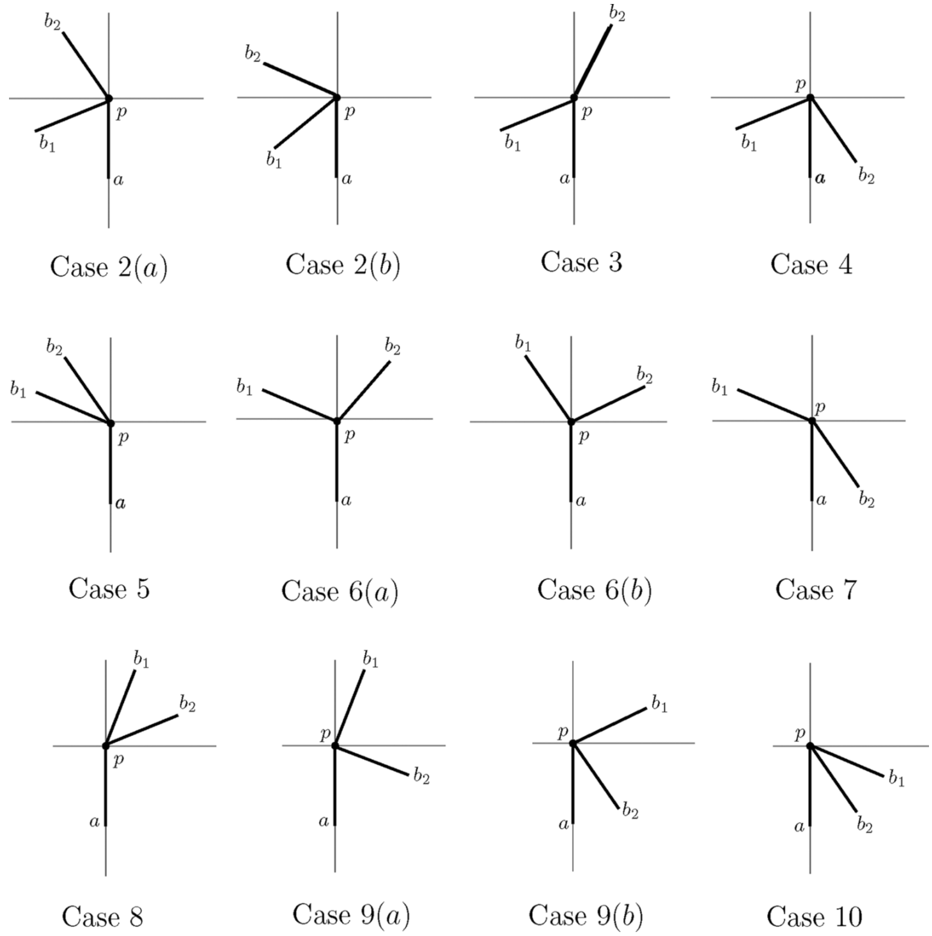


Fig. 3. Bifurcation details (cases 2–10).

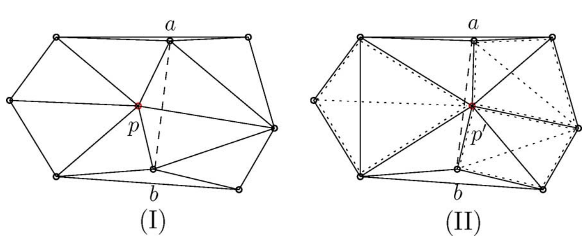


Fig. 4. Comparison of different Delauany triangulations on two pretty similar minutiae sets.

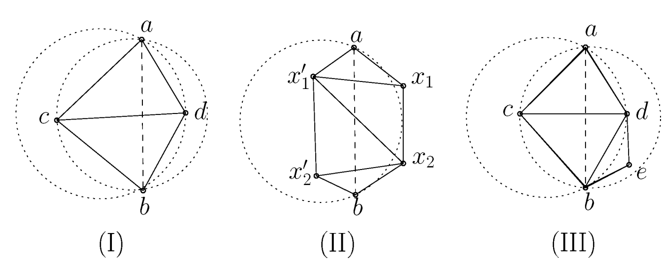


Fig. 5. Property of circles through order k-Delaunay edges.

a more robust triangulation algorithm that may restore the isomorphic subgraph [represented by the dotted lines in Fig. 4(II)] even while the point shifts out of tolerance. This leads us to use order k-Delaunay triangulation [18]–[21].

1) Definition and Properties [18], [19]: Refer to Fig. 5(I) for the following discussion. We first let $\overline{ab}$ denote the edge between vertices a and b; $C(a,b,c)$ denotes the circle through three vertices a, b, and c; and Δabc denotes the triangle defined by a, b, and c.

Definition 1: Given a finite set of points $S \in \Re^2$ and an integer $k \geq 0$. For any vertices $a, b, c \in S$:

- an edge $\overline{ab}$ is an order k-Delaunay edge, if a circle exists through a and b that contains, at most, k points of S;
- a triangle Δabc is an order k-Delaunay triangle if $C(a,b,c)$ contains, at most, k points of S;
- a triangulation of S is an order k-Delaunay triangulation if every triangle in the triangulation is an order k-Delaunay triangle.

Obviously, an order 0-Delaunay edge (respectively, triangle and triangulation) is a Delaunay edge (respectively, triangle and triangulation).

Lemma 1: Any order k-Delaunay triangle is also an order k'-Delaunay triangle, if $k' > k$. This also holds for order k-Delaunay triangulation.

Proof: Proof is trivial. By Definition 1, the circle though three vertices of an order k-Delaunay triangle contains, at most,

k points of S. Equivalently, the circle contains no more than k' points of S while $k' > k$. Thus, Lemma 1 holds. ■

Lemma 2: Every edge of an order k-Delaunay triangle is an order k-Delaunay edge; analogously, every edge of an order k-Delaunay triangulation is an order k-Delaunay edge.

Proof: Consider a circle through three vertices of the order k-Delaunay triangle. If we slightly move the center of the circle toward two vertices by making these two vertices on the circle and the third vertex just outside, the new circle still contains, at most, k points. This shows that the edge of those two vertices is an order k-Delaunay edge. By Definition 1 and Lemma 1, the remaining part of Lemma 2 holds from the aforementioned proof. ■

But, the reverse of Lemma 2 may not hold. Fig. 5(III) shows that not every order 1-Delaunay triangle can be included in an order 1-Delaunay triangulation. $C(a, b, d)$ only contains c, then Δabd is an order 1-Delaunay triangle. However, with $C(a, c, b)$ containing d and e, it is an order 2-Delaunay triangle. Therefore, the order 1-Delaunay triangle Δabd is included in an order 2-Delaunay triangulation.

There are particular properties of order k-Delaunay edges. See Fig. 5(I) for an illustration.

Observation 1: For an order k-Delaunay edge $\overline{ab}$ and a Delaunay edge $\overline{cd}$ that intersects $\overline{ab}$, the circle $C(a, b, c)$ contains d, and $C(a, d, b)$ contains c.

From Observation 1, we have a direct lemma. Please refer to Fig. 5(II).

Lemma 3: Given an order k-Delaunay edge $\overline{ab}$ and a circle through a and b, if the circle contains no vertex on one side of $\overline{ab}$, it contains all vertices, which are incident upon Delaunay edges intersecting $\overline{ab}$, on the opposite side of $\overline{ab}$.

Proof: $\overline{x_i x'_j}$ denotes the Delaunay edge intersecting $\overline{ab}$ $(i = 1, \ldots, k, j = 1, \ldots, l)$; x_i and x'_j are located on different sides of $\overline{ab}$, respectively. Assume a circle C' exists through a and b, containing no vertex on the x_i side, and also not containing x'_j on the other side of $\overline{ab}$. As in Observation 1, $C(a, b, x_j)$ contains x'_j. This shows that $C(a, b, x_j)$ must cover more area than C' on the x'_i side. From geometry theory, circle C' through a and b that does not contains a vertex on the x_i side of $\overline{ab}$ must cover a larger area than circle $C(a, b, x_j)$ on the x'_i side, a contradiction. ■

2) Order 1-Delaunay Triangles in Fingerprint Image: When a fingerprint image is scanned from the hard surface of a sensor, various manners of scanning and types of pressures distort the mutual position of minutiae. For instance, suppose there are a template print I_T and input print I_R from the same finger tip. Assume they have the same minutiae set S. Consequently, I_T and I_R have their own Delaunay triangulation $\mathrm{DT}_T(S)$ and $\mathrm{DT}_R(S)$ of minutiae set S, respectively. Distortion may change the local structure of Delaunay triangulation of minutiae. More specifically, those minutiae with changed positions may produce new local Delaunay triangulation $\mathrm{LDT}_R(s)$ which replaces the corresponding local one $\mathrm{LDT}_T(s)$, leading to $\mathrm{DT}_T(S)$ and $\mathrm{DT}_R(S)$ not being isomorphic, where $s \subseteq S$, $\mathrm{LDT}_R(s) \subseteq \mathrm{DT}_R(S)$, and $\mathrm{LDT}_T(s) \subseteq \mathrm{DT}_T(S)$.

This issue significantly decreases the performance of Delaunay triangulation-based indexing algorithm in cases with severe distortion. Our effort is engaged in constructing an order k-Delaunay triangulation of I_R, and finding the order k-Delaunay triangles that do not exist in $\mathrm{DT}_R(S)$ but might exist in $\mathrm{DT}_T(S)$. With these additional order k-Delaunay triangles, the fingerprint indexing algorithm can be enhanced.

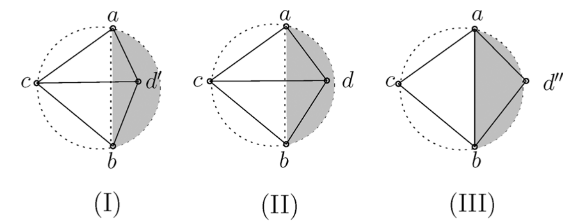


Fig. 6. Delaunay triangulations of the same points at different positions.

Fingerprint distortion cannot change the mutual positions of closely located minutiae to a great extent. We capture this closeness using Delaunay triangulation. Therefore, distortion may only change the local structure of the Delaunay triangulation by moving the minutiae close to the boundary of the tolerance region of the Delaunay triangles [16], [17]. Fig. 6 shows four minutiae with different mutual positions generating different Delaunay triangulations. Minutia d' in Fig. 6(I) lie in the middle of the tolerance region (the gray region). Distortions are unlikely to move it out of the tolerance region. Thus, Delaunay triangulation rarely changes under distortion in this case. Minutia d in Fig. 6(II) lie close to the boundary of the tolerance region. If the distortion moves d by a small distance to d'' and out of the tolerance region, Delaunay triangulation of minutiae changes [see Fig. 6(III)].

Δadc in Fig. 6(II) is a Delaunay triangle. Its corresponding order k-Delaunay triangle $\Delta ad''c$ in Fig. 6(III) can be obtained by flipping the Delaunay edge $\overline{ab}$ [13]. By observation 1 and Lemma 3, the circle $C(a, d'', c)$ must contain b because of the intersection of $\overline{cd''}$ and $\overline{ab}$. Moreover, distortion rarely moves d to d'' so much that $C(a, d'', c)$ contains more minutiae than b (same in the case of $C(c, d'', b)$ and a). In practice, we therefore only accept an order 1-Delaunay triangle $\Delta ad''c$ $(\Delta cd''b)$, whose edge $\overline{cd''}$ is an order 1-Delaunay edge in our algorithm. So we have the following definition of these additional triangles (called low-order Delaunay triangles) that can enhance a Delaunay triangulation-based fingerprint indexing algorithm.

Definition 2: Low-order Delaunay triangles (LoD triangles) is a union of Delaunay triangles of $\mathrm{DT}_R(S)$ and order 1-Delaunay triangles obtained by edge flipping in $\mathrm{DT}_R(S)$. These additional order 1-Delaunay triangles may be isomorphic to Delaunay triangles with the same vertices in $\mathrm{DT}_T(S)$.

For each convex quadrilateral in $\mathrm{DT}_R(S)$, it seems order 1-Delaunay triangles could be produced by simply flipping the shared Delaunay edge of two Delaunay triangles [e.g., flip the edge $\overline{cd}$ to $\overline{ab}$ in Fig. 5(III)]. Unfortunately, edge flipping may produce high-order Delaunay triangles. The computation of order 1-Delaunay triangles can be deduced by using the following Lemma.

Lemma 4: Finding valid order 1-Delaunay triangles is equivalent to finding any order 1-Delaunay edge that can be extended to order 1-Delaunay triangulation.

Proof: As per Definition 1, any order 1-Delaunay triangulation is composed of only Delaunay triangles and order

1-Delaunay triangles. Again, an order 1-Delaunay triangle must have at least one order 1-Delaunay edge. Thus, LoD triangles will be obtained when an order 1-Delaunay edge is found in a order 1-Delaunay triangulation. ■

We give the following algorithm to compute all order 1-Delaunay edges that are incident upon valid order 1-Delaunay triangles in a given Delaunay triangulation. Let $Q(adbc)$ denote convex quadrilaterals in the Delauany triangulation, and the shared Delaunay edge by two Delaunay triangles of $Q(adbc)$ is $\overline{ab}$. Refer to Fig. 6.

Algorithm 2: Order 1-Delaunay triangle generation.

Input: Delaunay triangulation of minutiae set $\mathrm{DT}(S)$.

Output: Valid order 1-Delaunay triangles.

1) **for** each $Q(abdc)$ in $\mathrm{DT}(S)$ **do**.

2) flip $\overline{ab}$ to produce order k-Delaunay triangles.

3) **if** both $C(adc)$ and $C(cdb)$ only have b and a inside, respectively **then**

4) report Δadc and Δcdb as valid order 1-Delaunay triangles.

5) **end if.**

6) **end for**.

Since the number of the convex quadrilaterals is linear in the number of Delaunay triangles (which is linear [13]), this algorithm takes linear time over and above the Delaunay triangulation construction which is $O(n \log n)$. Thus, the entire construction of the order 1-Delaunay triangles takes $O(n \log n)$ time.

We add obtained order 1-Delaunay triangles into the set of Delaunay triangles of $\mathrm{DT}(S)$, and employ all of them as elements of an index table. Since our specific order 1-Delaunay triangulation is an extension of Delaunay triangulation, it also inherits some other advantages: 1) Insertion of a new point in an order 1-Delaunay triangulation affects only the triangles whose circum circles contain that point. As a result, noise affects the order 1-Delaunay triangulation only locally. 2) Delaunay triangles are not skinny, because Delaunay triangulation maximizes the minimum angle [13] over all triangulations. 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 [22].

C. Indexing Elements

Minutia detail and the features derived from the LoD triangle of minutiae form the index, and are ordered by significance

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

- $\mathcal{M}_i^j$ is the minutia detail of each vertex of the triangle. If the vertex is a bifurcation, it is identified as the corresponding bifurcation detail case $\mathcal{B}_i^j$, where i is the index of the vertices of triangle and j is the index of minutia detail case as given in Table I. If it is a ridge ending, $\mathcal{T}_i$ is marked.

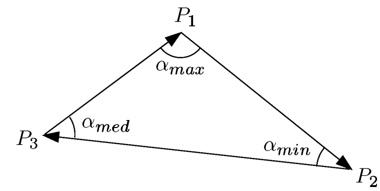


Fig. 7. Definition of triangle labels.

- $\alpha_{\min}$ and α_{med} are the minimum and median angles in the triangle, respectively. According to the magnitude of the angles, the vertices of angles $\alpha_{\max}$, $\alpha_{\min}$, and α_{med} are labeled as P_1, P_2, and P_3, respectively (see Fig. 7).
- 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 $H = \mathrm{sign}(Z_{21} \times Z_{32})$.
- $L_{\max}$ 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 be 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 the clockwise direction

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

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

Here, H, $\alpha_{\min}$, α_{med}, and $L_{\max}$ have the same definitions as in [10].

D. Conditions of Triangle Match

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

$$\begin{aligned} \mathcal{M}_i &= \mathcal{M}_i', \\ H &= H', \\ |\alpha_{\min} - \alpha_{\min}'| &< T_\alpha; \quad |\alpha_{\mathrm{med}} - \alpha_{\mathrm{med}}'| < T_\alpha, \\ \frac{|L_{\max} - L_{\max}'|}{L_{\max}} &< T_L, \\ |\phi_i - \phi_i'| &< T_\phi \end{aligned}$$

where T_α, T_L, and T_ϕ are the thresholds.

III. Indexing Score and Algorithm

Suppose I_R is the input fingerprint and I_{Ti} is the fingerprint in the database, which have minutiae set S and S_i, respectively, where $i = 1, 2, \ldots, N$ and N is the number of fingerprints in the database. Suppose there are n potential corresponding minutiae m_j in the pair of prints I_R and I_{Ti}, $m_j \in S \cap S_i$ and $j = 1, 2, \ldots, n$. r_j is the number of matched triangles, in

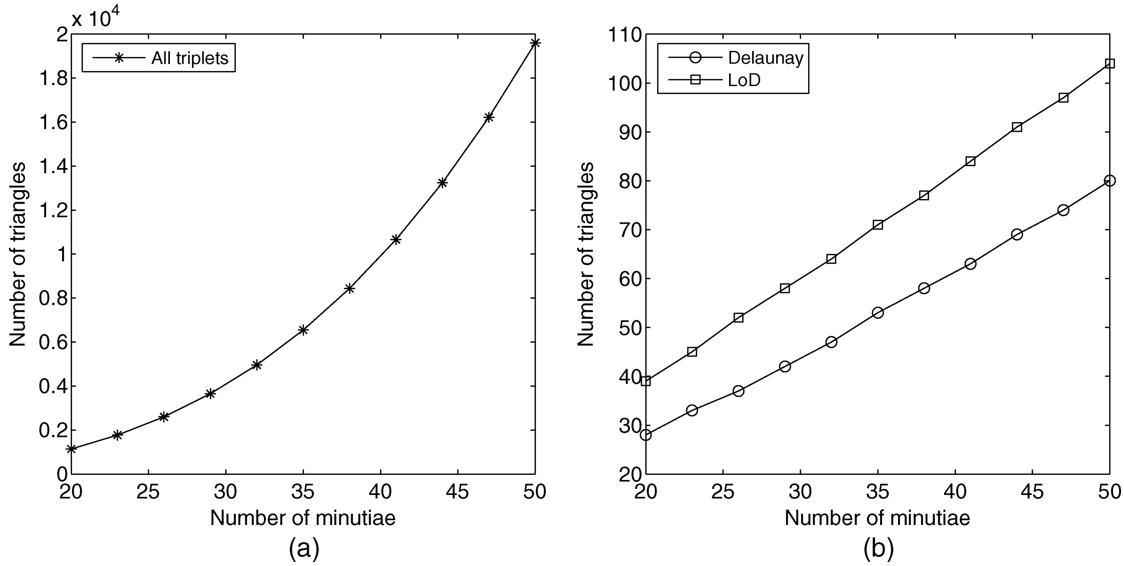


Fig. 8. Number of triangles constructed by three algorithms (all triplets, Delaunay, and LoD).

which m_j is incident, between I_R and I_{Ti}. We define the index score of the image I_{Ti} as

$$\mathcal{E}_i = c\sum_{j=1}^{n} r_j$$

where c is a constant for easier processing of $\mathcal{E}_i$. The indexing algorithm is now as follows.

For each triangle in I_{Ti}, compute ($\mathcal{M}_i^j$,H, $\alpha_{\min}$, α_{med}, $L_{\max}$, ϕ_i), and use these values to construct hash tables for fast indexing in a database.

Algorithm 3: Fingerprint indexing algorithm.

Input: Minutiae set S of input fingerprint I_R.

Output: Top $\mathcal{N}$ possible match print.

{*%The following pseudocode does the indexing.%*}

1) Apply Delaunay triangulation on S, then get $\mathrm{DT}_R(S)$.

2) Apply Algorithm 2 on $\mathrm{DT}_R(S)$, then add order 1-Delaunay triangles into the Delaunay triangle set.

3) **For** each triangle, **do**.

4) Compute ($\mathcal{M}_i^j$, H, $\alpha_{\min}$, α_{med}, $L_{\max}$, ϕ_i) for the triangle.

5) Search index space by ($\mathcal{M}_i^j$, H, $\alpha_{\min}$, α_{med}, $L_{\max}$, ϕ_i) under conditions given in Section II-D.

6) if the triangle satisfies the conditions, then

7) Take it as successful correspondence of the triangle in the database.

8) **end if**.

9) **end for**.

{*%The following pseudocode does the retrieval.%*}

10) **for** each I_{Ti} **do**.

11) count the number of corresponding triangles M_i.

12) **end for**.

13) $M_{\max} = \max\{M_i\}$.

14) **if** $M_{\max} < T_M$ **then**

15) reject the input fingerprint, where T_M is a threshold. And go to **End**.

16) **Else**.

17) Compute indexing score $\mathcal{E}_i$ based on M_i for I_{Ti}.

18) **end if**.

19) sort $\{\mathcal{E}_1, \ldots, \mathcal{E}_N\}$ in descending order, output top $\mathcal{N}$ possible match print.

IV. Experimental Results

A. Databases and Parameters

We evaluated our method by testing it on two databases: FVC2002(DB1) and FVC2004(DB1). Each of them consists of 880 fingerprints, eight prints each of 110 distinct fingers. All of these images are captured by two different optical sensors with the same resolution of 500 dpi, resulting in images of 388 × 374 pixels and 640 × 480 in 8-b gray scale, respectively. Fingerprints in the FVC2004 database are markedly more difficult to match than those in FVC2002, due to the perturbations deliberately introduced (such as low and high pressure against the sensor surface, exaggerated skin distortion and rotation, dried and moistened prints, etc.). We randomly choose three prints from these eight prints of each finger to construct the database. This database is now used for comparing three approaches, viz., all triplets [10], Delaunay [11], and proposed LoD. The remaining prints are used to test the indexing performance. Thresholds are different for the two data sets: for database FVC2002 $T_\alpha = 5°$, $T_L = 10\%$ pixels, $T_\phi = 5°$; for database FVC2004, $T_\alpha = 8°$, $T_L = 15\%$ pixels, $T_\phi = 10°$.

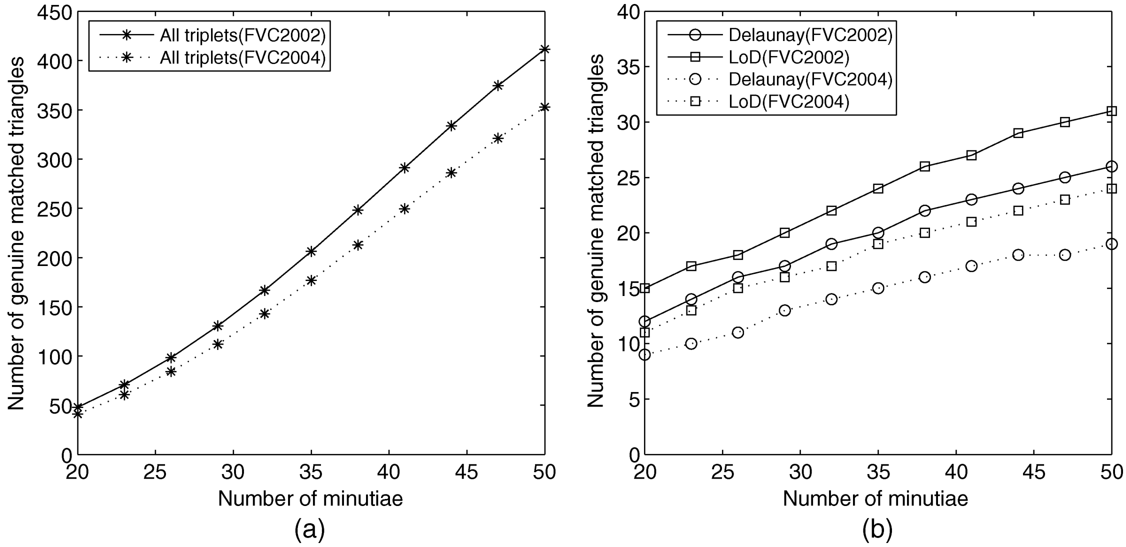


Fig. 9. Number of genuine matched triangles by three algorithms (all triplets, Delaunay, and LoD).

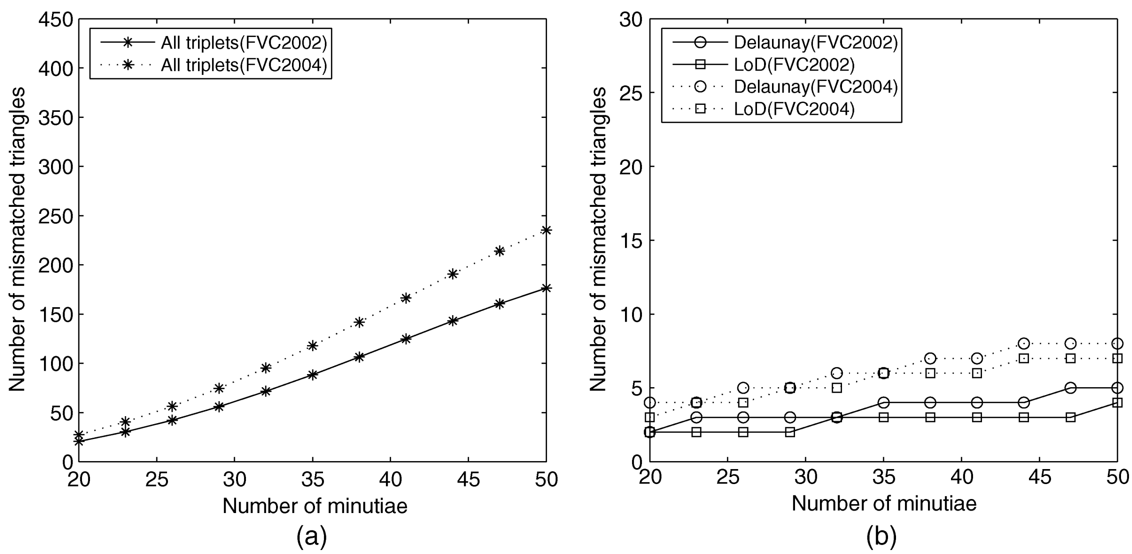


Fig. 10. Number of mismatched triangles by three algorithms (all triplets, Delaunay, and LoD).

B. Evaluation Measures for Triangle Matching

We evaluated these three triangulation algorithms (all triplets, Delaunay, and proposed LoD) on three measures:[2]

1) Number of constructed triangles (see Fig. 8).
2) Number of genuine matched triangles (Genuine matched triangle means the vertices of matched triangle are genuine minutiae, and they correspond to input and template prints, see Fig. 9).
3) Number of mismatched triangles (mismatched triangle means the vertices of the matched triangle may not be genuine minutiae, and some or all of them do not correspond to input and template prints, see Fig. 10).

In the following two subsections, we first discuss Section IV-B-1 our indexing strategy in Section IV-B-1, based on minutiae detail and LoD vis-a-vis the two other indexing methods based on triangles. Second, in Section IV-B-2, we justify the usage of LoD over the Delaunay-based method by placing them on the same pedestal by using the same minutiae detail index.

1) Different Indices: In this comparison, matching conditions of three algorithms are based on their own index elements.

Fig. 8 shows that all-triplets-based indexing algorithms will construct an unacceptably large number of triangles if a print contains more than 30 minutiae. So it consumes significant memory space. On the contrary, Delaunay-based and LoD-based algorithms construct considerably fewer triangles, both related linearly to the number of minutiae (see [13] and Algorithm 2). More important, an LoD-based algorithm constructs, at most, a constant number of triangles more than the number of Delaunay triangles. This property guarantees that the performance of the proposed indexing algorithm is improved without greatly increasing computation cost.

We select the genuine matched triangles resulting from the three algorithms and plot them against the number of minutiae

[2]In order to compare these three algorithms fairly, we obviously employ the same minutiae extraction algorithm [23] to obtain the same set of testing minutia for them.

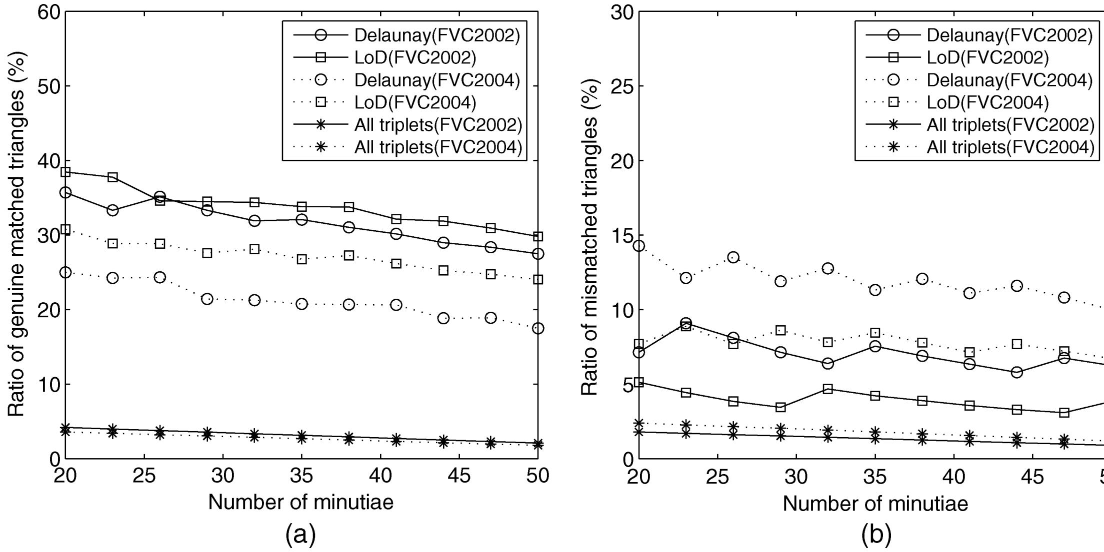


Fig. 11. Ratios of genuine matched and mismatched triangles by three algorithms (all triplets, Delaunay, and LoD).

in Fig. 9. An all-triplets-based algorithm acquires the most genuine matched triangles in both databases because of its large number of constructed triangles. On FVC2002, an LoD-based algorithm has more genuine matched triangles than Delaunay-based algorithms. It implies that even the fingerprints scanned under conditions of strict supervision still contain slight deformations that may change the Delaunay triangulation of the minutiae set. Matching on FVC2004 is more difficult due to deliberate distortions that change the Delaunay triangulation of minutiae. It directly decreases the number of matched triangles in the Delaunay-based algorithm. In comparison, remarkably more genuine matched triangles are achieved by the LoD-based algorithm because of matches of additional order 1-Delaunay triangles.

We also enumerate the number of mismatched triangles and plot them in Fig. 10. An all-triplets-based algorithm also obtains the most mismatches due to the same reason as to why it received the most genuinely matched triangles. The quantities of mismatched triangles generated by LoD-based and Delaunay-based algorithms are almost the same.

For a clearer comparison, we plot ratios of genuine matched and mismatched triangles versus the total number of triangles in Fig. 11. An all-triplets-based algorithm obtains both the lowest ratios because of the huge triangles it generates. A comparison of LoD-based and Delaunay-based algorithms is interesting. Although the LoD triangle set is larger than the Delaunay triangle set in both databases, additional order 1-Delauany triangles efficiently increase the ratio of genuine matched triangles of the proposed algorithm. Meanwhile, minutia details also prevent mismatches, and allow the ratio of mismatched triangles of an LoD-based algorithm to be always lower than the Delaunay-based algorithm. Therefore, more experiments have been conducted to evaluate the performance of minutia detail.

2) Same Index: Since an all-triplets-based algorithm is worse than the other two, we only compare LoD-based and Delaunay-based algorithms in this subsection. Also, genuine matched triangles are the same as before. So we just compare the mismatches' difference.

In this comparison, two algorithms first use the same index elements without minutia detail. Then, they use the same index elements with minutia detail. Please see Figs. 12 and 13 for illustration.

These two figures indicate that the minutia detail also can prevent a mismatch for the Delaunay-based algorithm. However, it cannot increase the number of genuine matches definitely. The proposed LoD-based algorithm obtains more mismatched triangles than the Delaunay-based algorithm when minutia detail is not employed, especially in database FVC2002. The explanation is that LoD generates or aggregates more triangles which increase the possibility of mismatch. Then, a discriminating index (e.g., minutia detail) is necessary to prevent the overmismatch, finally resulting in a better solution.

To sum up the aforementioned experimental results, we see that although the all-triplets-based algorithm obtains the largest quantity of genuine matched triangles, it also entails huge computation cost and a corresponding large number of mismatches. The Delaunay-based algorithm significantly saves computation cost, but may not find adequate genuine matched triangles, especially for distorted prints. The proposed LoD-based algorithm not only has the performance of the Delaunay-based algorithm, but desirably compensates for the Delaunay triangulation changes caused by distortion without greatly increasing computation cost.

C. Indexing Results

Two experiments have been carried out to evaluate indexing performances when the Delaunay- and LoD-based techniques use their own indices. Indexing performance curves plotting the correct index against the penetration rate at various thresholds are presented in Figs. 14 and 15 on the two databases. The correct index is defined as the percentage of correct fingerprints selected. Two figures indicate 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.

Since we randomly choose three prints from the eight prints per fingerprint as templates, here we report the average results.

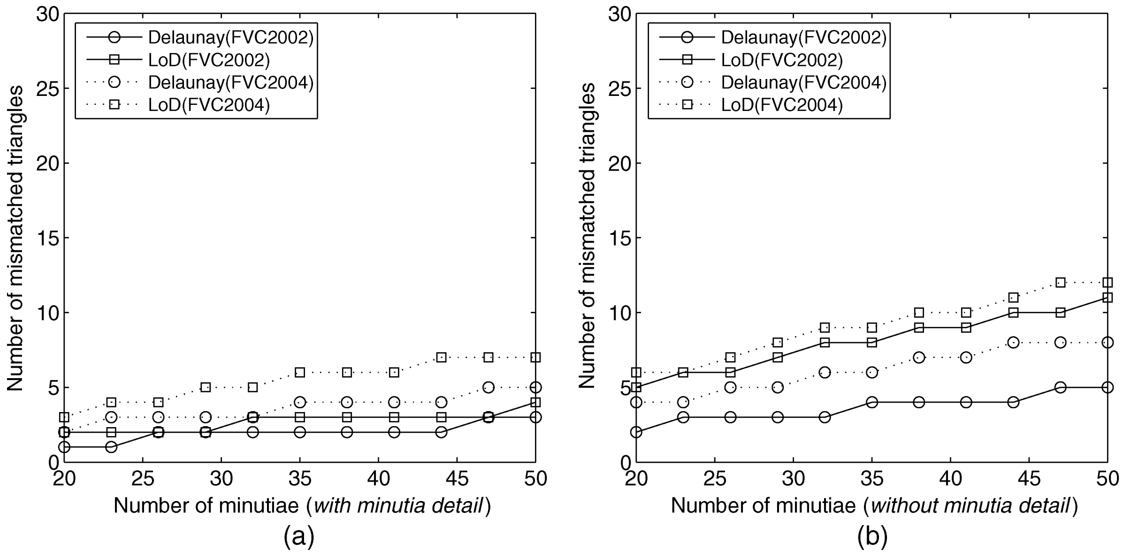


Fig. 12. Number of mismatched triangles from Delaunay- and LoD-based algorithms using and not using minutiae detail.

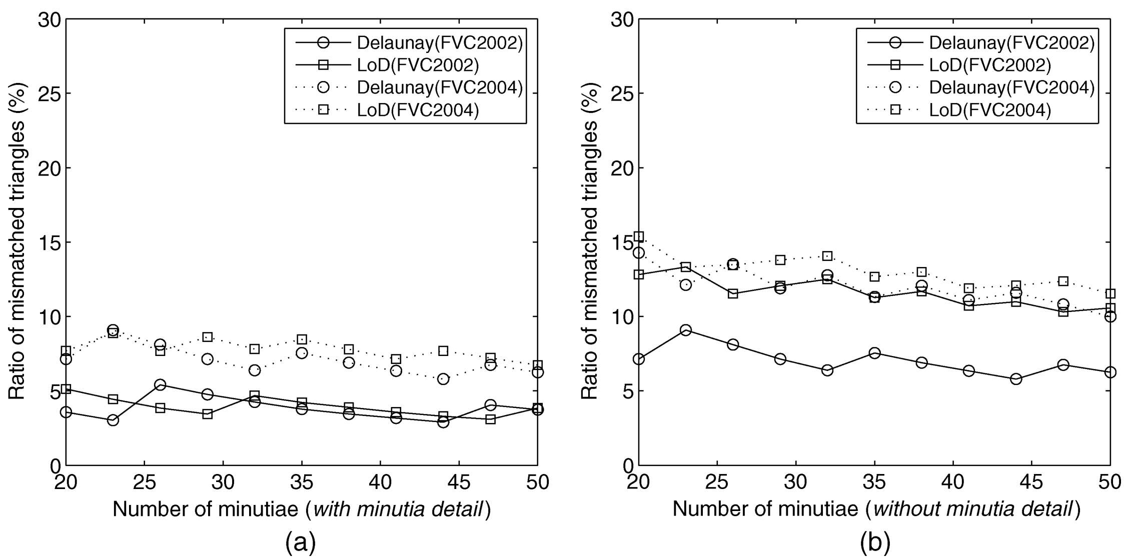


Fig. 13. Ratios of mismatched triangles from Delaunay- and LoD-based algorithms using and not using minutiae detail.

The results in Figs. 14 and 15 show that our proposed algorithm requires a smaller search space, and an all-triplets-based algorithm requires the largest search space when the three algorithms achieve the same correct index rate. This might be explained by the fact that an all-triplets-based algorithm creates $O(n^3)$ triangles compared to $O(n)$ triangles in our proposed and Delaunay-based algorithms. Although the all-triplets-based algorithm finds more matched triangles between two possible matched fingerprints, the side effect of $O(n^3)$ triangles also leads to more mismatches. Therefore, its search space in the database is larger than those of the other two when using the same T_M. Compared with the Delaunay-based algorithm, our proposed algorithm has better performance on both databases, thanks to the additional order 1-Delaunay triangles, which increase the possibility of matching genuine triangles. Meanwhile, minutia details prevent mismatches of order 1-Delaunay triangles.

More interesting results are shown in Table II. The performance of a Delaunay-based algorithm on FVC2004 becomes worse than its performance on FVC2002. The gap is around 10%, while the correct index achieves 100%. Performances of our proposed method and all-triplets-based algorithms, however, do not decrease so much. Their gaps are around 2.8% at a 100% correct index. This result shows that distortion changes the topology of Delaunay triangulation of the same minutiae set, and certainly makes the Delaunay-based algorithm degenerate and unstable. On the contrary, the LoD-based algorithm adds order 1-Delaunay triangles into the triangle set. These additional triangle-matches effectively compensate for distortion within a certain range, meanwhile minutia details prevent mismatches. Thus, the proposed algorithm has stable performance. Since an all-triplets-based algorithm exhausts all of the $O(n^3)$ triangles of minutiae, distortion changes the corresponding shape of the triangles only, but does not miss any of them. Therefore, it is also stable on different fingerprints but takes more time and space.

D. Tolerance of Spurious and Missing Minutiae

In practical AFIS systems, it is unlikely that noise removal and minutiae extraction can be performed with 100% accuracy.

TABLE II
INDEXING PERFORMANCE: PENETRATION RATE AT VARIOUS CORRECT INDEX RATES

Algorithms	95%		99%		100%	
	FVC2002	FVC2004	FVC2002	FVC2004	FVC2002	FVC2004
LoD triangle	2.7%	3.6%	8.1%	10%	18.1%	20.9%
Delaunay triangle	3.6%	7.2%	11.8%	18.1%	22.7%	32.7%
All triplets	7.2%	8.1%	23.6%	27.2%	38.1%	40.9%

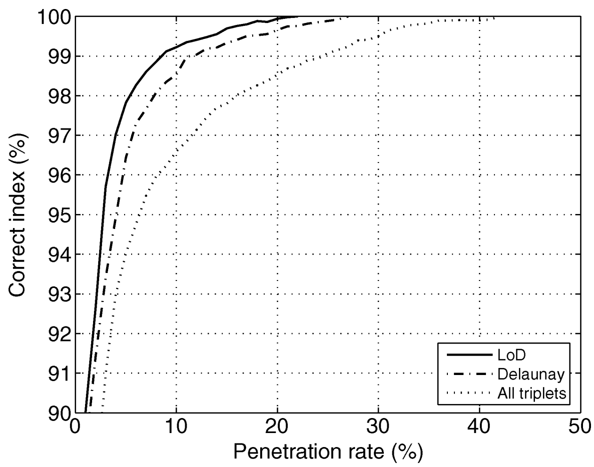


Fig. 14. Comparison of indexing performances on database FVC2002.

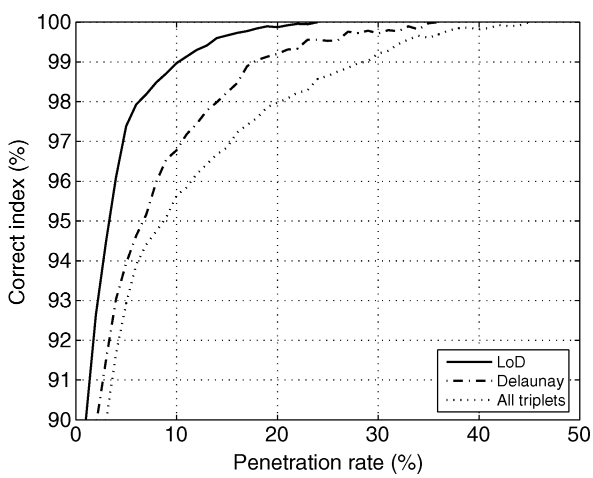


Fig. 15. Comparison of indexing performances on database FVC2004.

Therefore, AFIS systems require the ability of tolerating spurious and missing minutiae. We conducted two experiments on modified prints of database FVC2002 to evaluate the deterioration of three algorithms. Our minutiae extraction algorithm [23] was applied on all prints. We used the best one of the eight prints, which were from the same fingertip, as a template. The other seven prints were used for test.

1) Spurious Minutiae: We add $\lceil n\times 10\%\rceil$, $\lceil n\times 20\%\rceil$, $\lceil n\times 30\%\rceil$, and $\lceil n\times 40\%\rceil$ spurious minutiae uniformly into the fingerprint region of the print under test (n is the number of minutiae in one print).

We select penetration rates of three approaches when their correct indices achieve 100%, and plot them against the spurious minutiae percentage of the number of minutiae automatically detected by our system in Fig. 16(a).

This figure shows that the Delaunay-based algorithm can tolerate spurious minutiae within a small range (<20%) because the Delaunay triangulation can tolerate a lower amount of noise. When noise percentage increases, the performance decreases rapidly since it may lead to completely different triangulation.

Our proposed LoD-based algorithm can tolerate noise better than the Delaunay-based approach because of more genuine matched triangles. The geometric reasoning is as follows. Consider a subset of minutiae (a,b,c,d,e,f). We add one spurious minutia g into them [please see Fig. 17(I) and (II) for illustration]. Now, if we look into the Delaunay triangulations of these two subsets, we can see there is no possible matching triangle between them. After applying our proposed algorithm on subset (a,b,c,d,e,f,g), additional triangles Δbcd and Δdef in Fig. 17(III) can be matched with the corresponding triangles in Fig. 17(I). However, our experiments show that the proposed LoD-based approach will degrade rapidly when the spurious minutiae occupy more than 30% of the number of original minutiae. The reason is the same as in the case of the traditional Delaunay-based approach.

The all-triplets-based algorithm has the best ability to tolerate spurious minutiae, because spurious minutiae only increase the number of triangles, but do not decrease the number of genuine matched triangles.

2) Missing minutiae: $\lceil n\times 10\%\rceil$, $\lceil n\times 20\%\rceil$, $\lceil n\times 30\%\rceil$, $\lceil n\times 40\%\rceil$ minutiae of the fingerprint region of the print under test are uniformly deleted. The comparison is plotted in Fig. 16(b).[3]

This figure shows that all three algorithms perform a little worse in tolerating missing minutiae than spurious minutiae. The reason is that the number of triangles decreases with the increase of missing minutiae. It can be observed from the plotted curves in Fig. 16 that our LoD-based approach outperforms the Delaunay-based approach and can even compete with the all-triplets-based method up to a certain percentage (<40%) of missing and spurious minutiae, but beyond that, it is inferior to an all-triplets-based method. Fortunately, very high noise ranges are rare as almost all AFIS systems have an image enhancement algorithm.

To sum up the aforementioned experiments, our proposed algorithm has the efficiency of the Delaunay triangle-based algorithm, and the stability of the all-triplets-based algorithm on various fingerprints.

V. CONCLUSION

In this paper, an indexing algorithm using minutia detail and a low-order Delaunay triangle is proposed. It has been shown that this algorithm is able to search a fingerprint database more

[3]The analysis is similar to that for spurious minutia. But the input minutiae set is (a,b,c,d,e,f,g), and g plays a missing minutia in Fig. 17(I).

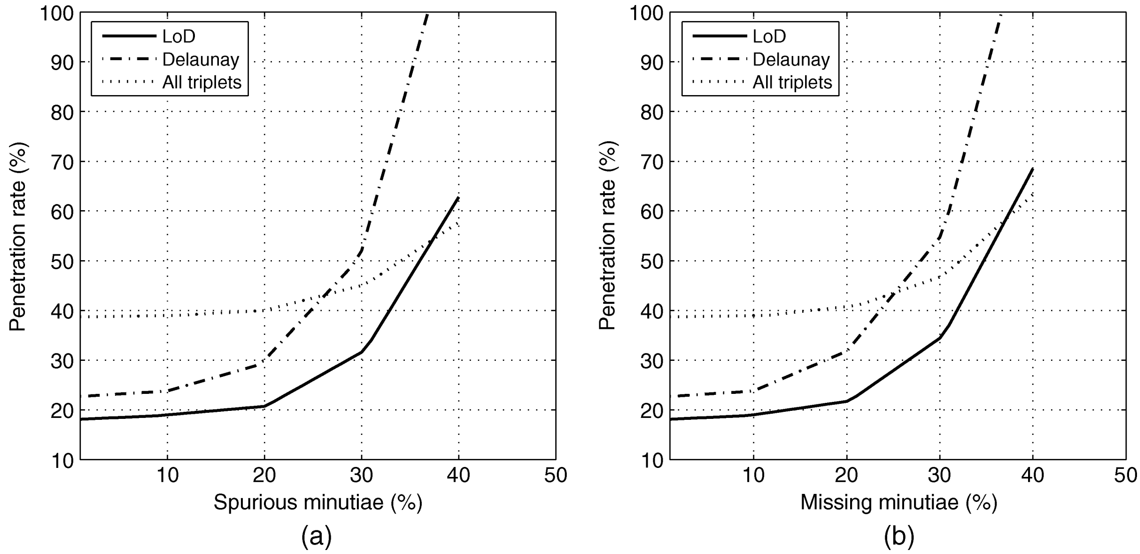


Fig. 16. Deterioration of indexing performance with spurious and missing minutiae increasing. Penetration rates are plotted for a 100% correct index.

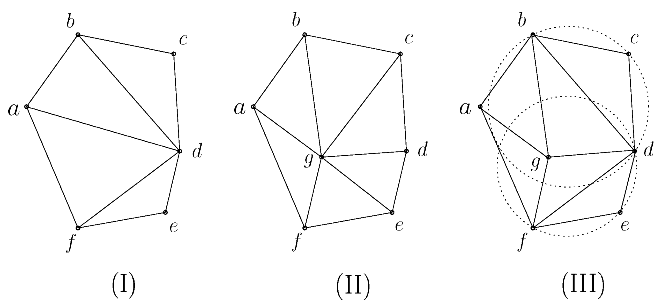


Fig. 17. Instance of tolerating spurious or missing minutia of the proposed LoD-based algorithm.

efficiently and stably than previous triangle-based algorithms. The reasons are as follows.

- Minutia detail provides more minutiae classes than minutia types, and further reduces search space. Since they can be obtained during the minutiae extraction step, the proposed indexing algorithm does not require additional computing time.
- LoD triangles include Delaunay and order 1-Delaunay triangles. These small number of additional order 1-Delaunay triangles increase the quantity of genuine matched triangles, especially on distorted fingerprints. Thus, it is more insensitive to elastic distortion. Since the proposed triangulation algorithm creates $O(n)$ triangles compared to $O(n^3)$ triangles for an all-triplets-based algorithm, many redundant or incorrectly matched triangles are avoided. Simultaneously, the search space in a database is reduced when using the same T_M.

The experiments on database FVC2002 and FVC2004 also prove that the proposed algorithm has better abilities to tolerate distortion and reduce the number of possible matching prints for the next step. A further improvement might be achieved by combining other features of fingerprints, including other conditional order k-Delaunay triangles and class rank method.

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

ACKNOWLEDGMENT

The authors would like to thank P. Bhowmick for useful discussions and the anonymous reviewers for their useful suggestions to improve this paper.

REFERENCES

[1] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. New York: Springer-Verlag, 2003.
[2] A. K. Jain, L. Hong, and R. M. Bolle, "On-line fingerprint verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 302–313, Apr. 1997.
[3] A. K. Jain and S. Pankanti, "Automated fingerprint identification and imaging systems," in *Advances in Fingerprint Technology*, 2nd ed. New York: Elsevier Science, 2001.
[4] K. Karu and A. K. Jain, "Fingerprint classification," *Pattern Recognit.*, vol. 33, no. 29, pp. 389–404, 1996.
[5] A. K. Jain, S. Prabhakar, and H. Lin, "A multichannel approach to fingerprint classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 348–359, Apr. 1999.
[6] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, "Fingerprint classification by directional image partitioning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 5, no. 21, pp. 402–421, May 1999.
[7] R. Cappelli, D. Maio, and D. Maltoni, "Indexing fingerperint databases for efficicent $1:n$ matching," presented at the 6th Int. Conf. Control, Automation, Robotics and Vision, Singapore, 2000.
[8] R. S. Germain, A. Califano, and S. Colville, "Fingerprint matching using transformation parameter clustering," *IEEE Comput. Sci. Eng.*, vol. 4, no. 4, pp. 42–49, Oct.-Dec. 1997.
[9] J. D. Boer, A. M. Bazen, and S. H. Cerez, "Indexing fingerprint database based on multiple features," presented at the ProRISC Workshop Circuits, Systems Signal Processing, Veldhoven, The Netherlands, Nov. 2001.
[10] B. Bhanu and X. J. Tan, "Fingerprint indexing based on novel features of minutiae triplets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 616–622, May 2003.
[11] G. Bebis, T. Deaconu, and M. Georgiopoulos, "Fingerprint identification using Delaunay triangulation," in *Proc. Int. Conf. Information Intelligence and Systems*, 1999, pp. 452–459.
[12] X. F. Liang, "Fingerprint image analysis using computational geometric techniques," Ph.D. dissertation, School Inf. Sci., Jpn. Adv. Inst. Sci. Technol., Nomi, Ishikawa, Japan, 2006.
[13] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry (Algorithms and Applications)*. Berlin, Germany: Springer-Verlag, 1997, ch. 9.

[14] N. Liu, Y. Yin, and H. Zhang, "A fingerprint matching algorithm based on Delaunay triangulation net," in *Proc. 5th Int. Conf. Computer and Information Technology*, 2005, pp. 591–595.

[15] G. Parziale and A. Niel, "A fingerprint matching using minutiae triangulation," in *Proc. 1st Int. Conf. Biometric Authentication*, Hong Kong, China, 2004, pp. 241–248.

[16] M. Abellanas, F. Hurtado, and P. A. Ramos, "Structural tolerance and Delaunay triangulation," *Inf. Process. Lett.*, vol. 71, no. 5, pp. 221–227, 1999.

[17] A. A. Khanban and A. Edalat, "Computing Delaunay triangulation with imprecise input data," in *Proc. 5th Canadian Conf. Computational Geometry*, 2003, pp. 94–97.

[18] M. Abellanas, P. Bose, J. García, F. Hurtado, M. Nicolás, and P. A. Ramos, "On properties of higher order Delaunay graphs with applications," in *Proc. 21th Eur. Workshop Comput. Geom.*, 2005, pp. 119–122.

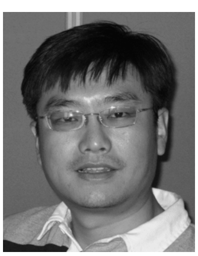
[19] J. Gudmundsson, M. H. Hammar, and M. van Kreveld, "Higher order Delaunay triangulations," *Comput. Geom. Theory Appl.*, vol. 23, no. 1, pp. 85–98, 2002.

[20] J. Gudmundsson, H. Haverkort, and M. van Kreveld, Constrained Higher Order Delaunay Triangulations Univ. Utrecht, Dept. Comput. Sci., Utrecht. The Netherlands, Tech. Rep. UU-CS-2004-007, 2004. [Online]. Available: http://www.archive.cs.uu.nl/pub/RUU/CS/techreps/CS-2004/2004-007.pdf.

[21] T. de Kok, M. van Kreveld, and M. Lffler, "Minimizing local minima in terrains with higher-order Delaunay triangulations," in *Proc. 21st Eur. Workshop Comput. Geom.*, 2005, pp. 115–118.

[22] Y. Lamdan, J. Schwartz, and H. Wolfson, "Affine invariant model-based object recognition," *IEEE Trans. Robot. Autom.*, vol. 6, no. 5, pp. 578–589, Oct. 1990.

[23] X. F. Liang, A. Bishnu, and T. Asano, "A combinatorial approach to fingerprint binarization and minutiae extraction using euclidean distance transform," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 21, no. 7, Nov. 2007.

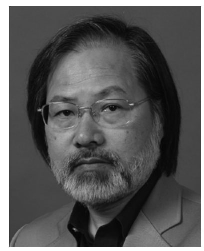


Xuefeng Liang received the M.Sc. degree in computer science from Xi'dian University, Xi'an, China, in 2003 and the Ph.D. degree in information science from the Japan Advanced Institute of Science and Technology, Nomi, Japan, in 2006.

He held a JST Research Fellowship at JAIST and worked on biometrics. Currently, he is conducting postdoctoral work with Ubiquitous Functions Research Group, Intelligent Systems Research Institute, the National Institute of Advanced Industrial Science and Technology, and on the vision system in robotics. His research interests include computational geometry, intelligent algorithm, biometrics, image processing, and computer vision.

Arijit Bishnu received the M.Sc. and Ph.D. degrees in computer science from the Indian Statistical Institute, Kolkata, India, in 1998 and 2003, respectively.

He was an Associate in the School of Information Sciences, Japan Advanced Institute of Science and Technology, Nomi, Japan. Currently, he is an Assistant Professor in the Computer Science and Engineering Department, Indian Institute of Technology, Kharagpur, India. His research interests are in topics related to geometry and its applications.



Tetsuo Asano was born in Kyoto, Japan, in 1949. He receieved the B.E., M.E., and Ph.D degrees from Osaka University, Japan, in 1972, 1974, and 1977, respectively.

In 1977, he joined Osaka Electro-Communication University as a Lecturer and moved to the Japan Advanced Institute of Science and Technology (JAIST) in 1997. Currently, he is a Professor in the School of Information Science. His research interest includes algorithms and data structures, especially in computational geometry, combinatorial optimization, computer graphics, computer vision using geometric information, and very large scale integrated (VLSI) layout design. He has been serving as an editor of several journals including *Discrete and Computational Geometry*, *Computational Geometry: Theory and Applications*, *International Journal of Computational Geometry and Applications*, and *Theory of Computing Systems*.

Dr. Asano was the Chair of a Special Interest Group on Algorithms of Information Processing Society of Japan from 1994 to 1996. He is a Fellow of the Association of Computing Machinery (2001) and Information Processing Society of Japan (2004).