Computer Vision and Image Understanding 113 (2009) 979–992

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Minutiae-based template synthesis and matching for fingerprint authentication

Tamer Uz [a], George Bebis [a,*], Ali Erol [a], Salil Prabhakar [b]

[a] Computer Vision Laboratory, University of Nevada, Reno, NV, USA
[b] Digital Persona, Inc., Redwood City, CA, USA

ARTICLE INFO

Article history:
Received 28 March 2008
Accepted 14 April 2009
Available online 18 May 2009

Keywords:
Fingerprint authentication
Fingerprint matching
Template synthesis
Delaunay triangulation

ABSTRACT

Fingerprint matching is often affected by the presence of intrinsically low quality fingerprints and various distortions introduced during the acquisition process. An effective approach to account for within-class variations is by capturing multiple enrollment impressions of a finger. The focus of this work is on effectively combining minutiae information from multiple impressions of the same finger in order to increase coverage area, restore missing minutiae, and eliminate spurious ones. We propose a new, minutiae-based, template synthesis algorithm which merges several enrollment feature sets into a "super-template". We have performed extensive experiments and comparisons to demonstrate the effectiveness of the proposed approach using a challenging public database (i.e., FVC2000 Db1) which contains small area, low quality fingerprints.

1. Introduction

Fingerprint matching is among the most widely used biometric technologies with a broad range of both government and civilian applications such as driver's license, social security, passport control, ATM/credit card, and medical records management, laptop and cell phone access control [1]. The key challenge in fingerprint matching is getting a match decision between a pair of fingerprints from the same finger under various within-class variations. These variations can be caused by several factors such as non-linear geometric distortions due to skin elasticity, inconsistent finger placement and contact pressure, small sensing area, environment conditions, and sensor noise. As a result, impressions of the same finger may be quite different from each other, making matching very difficult. For example, noise results in missing/spurious minutiae, while small sensing area results in small overlap.

An effective approach to account for within-class variations is by capturing multiple enrollment impressions. Information from multiple enrollment impressions can be integrated in two different ways. The first approach involves matching a given impression (i.e., query) against each of the enrollment impressions. The final matching is obtained by fusing the individual matching results either at the score level (e.g., maximum score) or at the decision level (e.g., majority voting). The second approach involves merging the enrollment impressions into a "super-fingerprint" by registering the enrollment impressions together and matching the query against the super-fingerprint. The first approach has shown to increase accuracy to desired levels, however, its main drawback is that it increases both storage and time requirements. On the other hand, the second approach is less space and time consuming but registering the enrollment impressions accurately is a challenging issue.

Merging a number of enrollment impressions into a super-fingerprint increases fingerprint area and accounts for low quality impressions. Fig. 1 shows an example where the overlapping area between two fingerprints from the same finger is less than 40% of the overall fingerprint area. Merging can be performed either at the image level or at the minutiae level. Image level merging, also known as "mosaicking", creates a super-fingerprint image by registering the enrollment impressions. Minutiae-level merging creates a super-fingerprint template (i.e., "super-template") by registering the corresponding enrollment minutiae templates. This approach is simpler and can tolerate non-linear distortions better than image mosaicking. However, the resulting super-template could only be used with minutiae-based, fingerprint matching algorithms.

In this paper, we propose a new, minutiae-based, template synthesis approach which employs a novel hierarchical matching strategy to combine a number of enrolment minutiae feature sets into a super-template. Assuming that each finger is represented by multiple enrollment impressions, the key objective of our approach is merging the feature sets of each finger into a super-template to improve the quality of features used, reduce space requirements, and improve speed. To build a super-template from a number of enrollment feature sets, one of the enrollment templates is selected first to initialize the super-template. Then, the remaining enrollment feature sets are aligned and merged with the super-template on an incremental basis. Minutiae in the

[*] Corresponding author. Fax: +1 775 784 1877.
E-mail addresses: uz@cse.unr.edu (T. Uz), bebis@cse.unr.edu (G. Bebis), aerol@cse.unr.edu (A. Erol), SalilP@digitalpersona.com (S. Prabhakar).

1077-3142/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2009.04.002

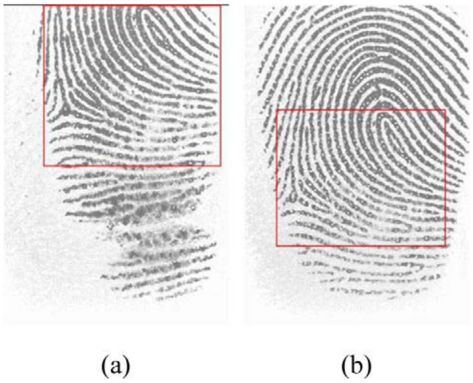


Fig. 1. Small overlapping area problem; the two images shown come from the same finger, however, the overlapping area between them is less than 40% of the overall fingerprint area.

super-template are assigned a weight based on the frequency of their occurrence in the enrollment feature sets. These weights serve as a quality measure of the minutiae. To merge an enrollment feature sets with the super-template, we search for minutiae correspondences between the enrollment template and the super-template using a hierarchical matching algorithm.

Our hierarchical minutiae matching algorithm builds upon our earlier work on Delaunay triangulation-based fingerprint indexing and matching [2]. We exploit the invariance of features extracted from Delaunay triangles and accumulate evidence about matching triangles in a transformation space using voting. To account for missing and spurious minutiae, the proposed approach explores the super-template minutiae hierarchically. This is performed by organizing the minutiae into a hierarchy of Delaunay triangulations, where higher levels of the hierarchy correspond to higher quality minutiae and lower levels of the hierarchy correspond to medium and low quality minutiae. Matching is performed hierarchically, starting at the top level of the hierarchy and moving down to lower levels until matching has been accomplished or the last level of the hierarchy has been reached.

Our hierarchical matching and template synthesis approaches are closely interrelated. The template synthesis relies on hierarchical matching to merge the enrollment feature sets into a super-template while the hierarchical matching relies on template synthesis to build the super-templates. Therefore, once the super-templates have been built, a similar hierarchical matching algorithm is employed for recognition. The proposed template synthesis and hierarchical matching approach is less sensitive to missing and/or spurious minutiae. Also, it addresses the small overlapping area problem since the coverage of the super-template increases with the number of enrollment feature sets merged together. Combining the enrollment feature sets into a super-template also eliminates the problem of template selection [3]. Finally, this approach is better suited for updating the template over time. We have evaluated and compared our approach with related approaches on a challenging public database (i.e., FVC2000 Db1) which contains small coverage area, low quality fingerprint images.

The rest of the paper is organized as follows: In Section 2, we review the problem of merging a number of fingerprint images or minutiae feature sets into a super-fingerprint image or a super-template. Section 3 presents the details of the proposed hierarchical matching strategy. Our minutiae template synthesis algorithms is presented in Section 4. Section 5 discusses fingerprint authentication; results and comparisons are given in Section 6. Finally, Section 8 presents conclusions and discusses directions for future research.

2. Background

Algorithms that have been proposed in the literature to combine multiple impressions of a finger are typically based on two main approaches: (i) mosaicking [4,5], which combines the enrollment impressions at the image level, and (ii) template synthesis [6,7], which combines the enrollment impressions at the feature level.

In [4], Ratha et al. used several blending algorithms to tile the image sequence of a rolling fingerprint grabbed by a large area scanner. Since the images were obtained by rolling the fingerprint on the sensor, successive images were assumed to be spatially registered. In [8], Jain and Ross combined multiple enrollment impressions by aligning the ridges of two images using the iterative closest point (ICP) algorithm. However, they did not account for non-linear deformations. In [9], Zhang et al. mosaicked the stream of swipe fingerprint frames using a minimum mean absolute error criterion. This technique cannot handle rotation, scale, or shear in individual frames of the swipe fingerprint images. Moreover, it does explicitly take non-linear deformations in consideration.

In [10], Choi et al. tried mosaicking different parts of a fingerprint which were collected by having the subject roll and slide his/her finger on the surface of a small area fingerprint scanner. Rolling and sliding over a small area sensor can cause severe plastic distortion and smudging, which degraded the performance of their system. Although they addressed the small overlapping area problem, handling missing/spurious minutiae was not explicitly considered. Shah et al. [5] performed mosaicking by employing a thin plane spline as a transformation model to account for non-linear distortions in fingerprints. However, they could not get a correct alignment in 16% of the cases; they handled these failures using score-level fusion. Although they showed better results using mosaicking than the individual enrollment impressions, their system is semi-automatic and requires intervention when mosaicking does not work.

Among the template synthesis methods, Yau et al. [11] compared three transformations for aligning two sets of minutiae for template synthesis: affine, projective, and topological. Their results indicated that affine transformation performed better. They demonstrated that template synthesis can lower the number of false rejections. A modified ICP algorithm was used by Moon et al. [12] to search for the optimal registration before merging two minutiae sets. In Ryu et al. [13], a Bayesian estimation approach was used to merge several enrollment minutiae sets. Toh et al. [6] created a synthesized template using multiple enrollment feature sets. Even though they showed significant improvements using the combined template as compared to a single feature set that included the center of the finger, the reported accuracy rates were much lower than current benchmarks.

In [7], Jiang and Ser created and updated a super-template as the user provided new samples during verification. Similar to our study, they assigned a weight to each minutia according to its frequency of occurrence. Depending on the weight values, spurious minutiae can fade away over time while missing minutiae can appear at some point in time. In contrast to our approach that utilizes minutiae of various quality for matching, they simply discarded low quality minutiae by thresholding the weights. They reported an improvement in accuracy relative to using individual feature sets; however, no comparisons with other fusion approaches were shown.

In a recent study comparing image mosaicking versus template synthesis using thin-plate splines [14], it was found that both methods improve matching performance, however, template synthesis outperformed image mosaicking. In a related study [12], it was found that image mosaicking worked better than template synthesis when the size of the component images decreased. The authors justified this result by the fact that the number of spurious minutiae gets smaller as the image size decreases.

3. Fingerprint matching using Delaunay triangulation hierarchies

Combining a number of enrollment feature sets into a super-template requires finding an alignment transformation to register each of the enrollment feature sets with the super-template. Given a pair of minutiae sets, one from enrollment and the other from the super-template, the alignment transformation can be computed by finding corresponding minutiae in the two sets. Our matching strategy for template synthesis builds upon our previous work on fingerprint matching using Delaunay triangulation [2] with some important extensions to account for missing and spurious minutiae. Readers who are not familiar with this approach are encouraged to review Appendices A and B.

A key issue when employing Delaunay triangulation for matching is that missing or spurious minutiae could change the triangulation locally by introducing spurious triangles or eliminating important triangles. As a result, matching quality could be degraded, especially when the overlap between minutiae sets to be matched is poor. When multiple impressions of the same fingerprint are available, one can make more reliable decisions about the presence or quality of fingerprint features by combining information from each impression. In this section, we present our novel minutiae matching algorithm which builds upon the algorithm in Appendix B.

To address the issue of missing and spurious minutiae, we propose performing the matching hierarchically based on minutiae quality. The key idea is representing the super-template minutiae in terms of a hierarchy of Delaunay triangulations where every level of the hierarchy corresponds to a subset of minutiae, having certain quality only. Specifically, the Delaunay triangulation at the lowest level of the hierarchy contains all minutiae, independently of quality, while the Delaunay triangulations at higher levels of the hierarchy contain minutiae of higher quality only. Matching is performed hierarchically, starting at the top of the Delaunay hierarchy, which contains high quality minutiae only, and moving down to lower levels, effectively considering lower quality minutiae on an incremental basis. At each level, a number of alignment hypotheses are computed, as described in Appendix B, and refined in an iterative fashion using affine transformations [2].

To discuss the proposed hierarchical matching algorithm, the only thing to bear in mind is that the minutiae in the super-template have quality weights associated with them based on their number of occurrences. Fig. 2 shows an example of a super-template which was built by merging three enrollment feature sets. In this case, minutiae weights can have three possible values (i.e., 1, 2, or 3) with higher weight values indicating higher quality minutiae.

3.1. Hierarchical Delaunay triangulation

Missing and/or spurious minutiae as well as non-linear distortions can severely affect the Delaunay triangulation as illustrated in Fig. 3. In this example, the yellow triangle shown in Fig. 3(a) could not be detected in Fig. 3(b) due to a spurious minutia (e.g., shown with a white rectangle in Fig. 3(b)), which is the result of smudging. Fig. 3(c) and (d) contains no missing minutiae; however, non-linear dislocations due to the elasticity of the skin have altered the topology and shape of corresponding triangles.

Assuming that minutiae weights range from 1 to k, we form k minutiae groups and organize their Delaunay triangulations in a hierarchy. The group at the bottom of the hierarchy contains minutiae having weights greater or equal to one (i.e., all possible super-template minutiae, independently of their quality). A group at some level i, where $1 \leqslant i \leqslant k$, contains minutiae having weights greater or equal to i (i.e., lower quality minutiae are excluded as we move to higher levels). Finally, the group at the top level of the hierarchy contains minutiae having weights equal to k (i.e., highest quality super-template minutiae).

Fig. 2(d) shows the super-template built by merging the templates shown in Fig. 2(a)–(c). Fig. 4 illustrates the corresponding Delaunay triangulation hierarchy which contains three levels in this case. In particular, Fig. 4(a) shows the Delaunay triangulation corresponding to minutiae whose weight is equal to 3. The triangulation shown in Fig. 4(b) corresponds to minutiae whose weight is equal to 2 or 3. Finally, Fig. 4(c) shows the triangulation corresponding to minutiae whose weight is equal to 1, 2, or 3 (i.e., all possible minutiae).

3.2. Hierarchical matching

When comparing a feature set (whether enrollment or query) to a given super-template, matching is performed hierarchically, starting at the top level of the hierarchy (i.e., considering hight quality minutiae), and ending at the bottom level of the hierarchy (i.e., considering all possible minutiae). Traversing the hierarchy in a top-down way is equivalent to taking lower quality minutiae into consideration in an incremental fashion. At each level, matching is performed as described in Appendix B, that is, matching triangles from the triangulation of the input feature set to triangles corresponding to the triangulation of the super-template at that level. Referring to Fig. 4, when matching a new feature set to a super-template, its Delaunay triangulation might have to be compared with all three triangulations in order to calculate an optimum alignment between the feature set and the super-template.

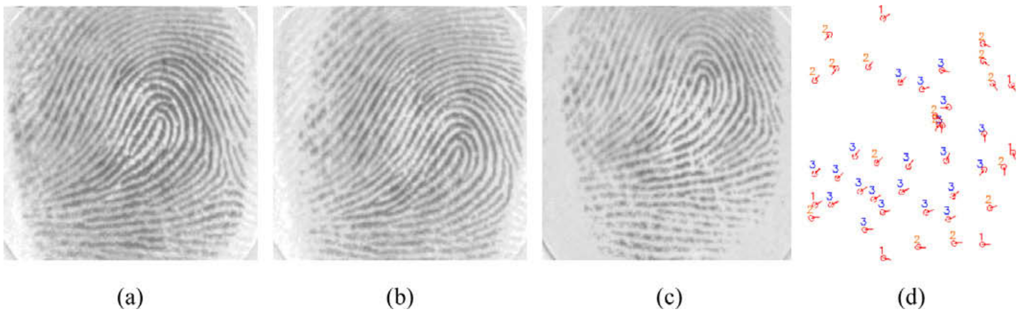


Fig. 2. Illustration of the merging step using three templates. Minutiae in a super-template is associated with a weight which corresponds to their frequency of occurrence in the enrollment templates and characterizes their quality.

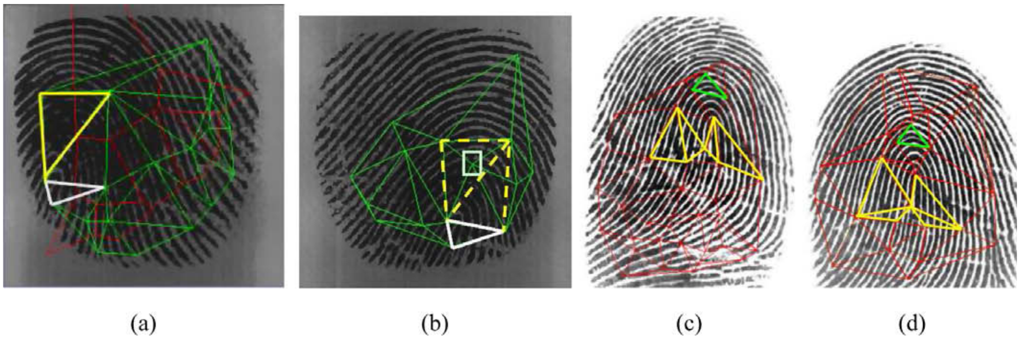


Fig. 3. Problems associated with the Delaunay triangulation of a set of minutiae: (a) and (b) show the effect of missing/spurious minutiae, (c) and (d) show the effect of non-linear distortions. In the first case, spurious triangles might be introduced while original triangles might not be formed. In the second case, the topology and shape of certain triangles might change.

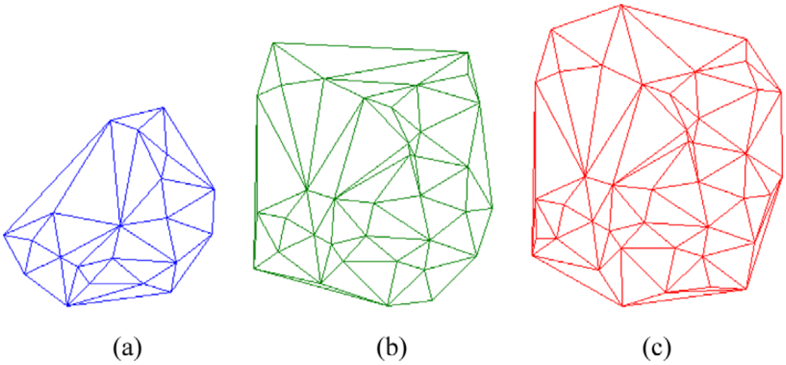


Fig. 4. The Delaunay triangulation hierarchy corresponding to the super-template shown in Fig. 2: (a) triangulation using minutiae whose weight is equal to 3; (b) triangulation using minutiae whose weight is equal to 3 or 2; and (c) triangulation using minutiae whose weight is equal to 3, or 2, or 1.

Using minutiae weights ensures that low quality minutiae will be considered only at the last stages of matching. Fig. 5 shows an example to illustrate the benefits of the hierarchical matching approach. In this example, minutiae could have a weight equal to 1 or 2; therefore, the Delaunay triangulation hierarchy has two levels. Using the bottom level of the hierarchy, we could match the white triangles shown in Fig. 5(a) and (b) but not the yellow triangles. This is because we cannot form the yellow triangles in Fig. 5(b) due to the presence of a spurious minutia. Using the top level of the hierarchy, however, allows the formation and matching of the yellow triangles as shown in Fig. 5(c).

Note that the Delaunay triangles at the $i+1$ level of the hierarchy are not necessarily a subset of the Delaunay triangles at the i level of the hierarchy. To improve matching results by increasing support in the transformation space, when considering the i level, our matching algorithm considers not only Delanuay triangles at this level but also Delaunay triangles at higher levels assuming that (i) they are different from those at level i and, (ii) have been matched successfully with the input template in previous iterations. To illustrate this idea, Fig. 6(a) shows the combined triangulations of Fig. 4(a) and (b), while Fig. 6(b) shows the combined triangulations of Fig. 4(a)–(c). It can be observed that the triangles shown in Fig. 4(b) are not all the same to the triangles shown in

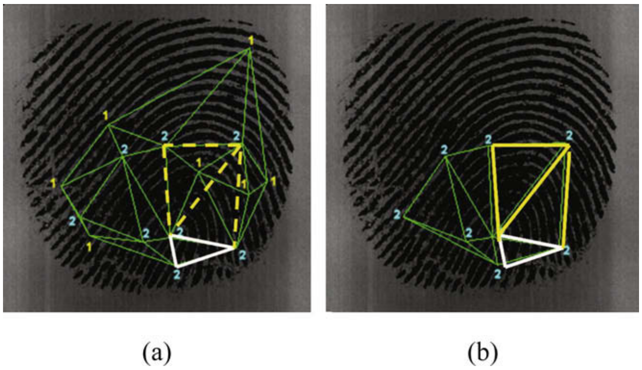


Fig. 5. An example illustrating the benefits of hierarchical matching. The Delaunay triangulation corresponding to the input template is shown in Fig. 3(a). (a) Delaunay triangulation of super-template minutiae having weights equal to 1 or 2; (b) Delaunay triangulation of super-template minutiae having weights equal to 2. Using the triangulation in (a), the yellow triangles in Fig. 3(a) and (b) can be matched. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

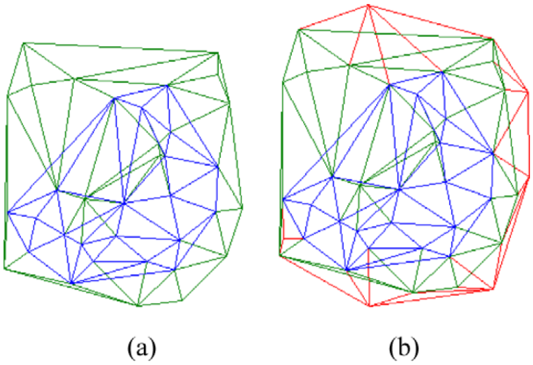


Fig. 6. (a) The triangulations shown in Fig. 4(a) and (b) combined together; (b) the triangulations in Fig. 4(a)–(c) combined together.

Fig. 6(a). The same is true for the triangles shown in Figs. 4(c) and 6(b).

3.3. Refinement using affine transformations

Due to non-linear distortions present in fingerprints, using rigid transformations to register a pair of minutiae sets at any level of the Delaunay hierarchy might not yield good results. Since it is very important to align the enrollment templates with the super-template as accurately as possible to avoid registration errors, we improve the estimated alignment (i.e., based on rigid transformations) by applying affine refinements in an iterative fashion as in [2]. The basic idea is finding additional minutiae correspondences iteratively and re-estimating the parameters of the alignment. First, a set of matching minutiae pairs is found by aligning the minutiae sets using the estimated rigid transformation. An initial affine transformation is computed using the matching minutiae pairs. Then, the set of matching minutiae pairs is updated by aligning the minutiae using the estimated affine transformation. Using the updated set of matching minutiae pairs, a new affine transformation is estimated to further improve the alignment transformation. This process is repeated until no more alignment improvements are possible. In each iteration, a more conservative criterion is used to establish matching minutiae pairs (i.e., the allowed distance between matching minutiae pairs decreases in each iteration).

Although affine transformations cannot model complex, non-linear fingerprint distortions, they can still account for certain kinds of deformations such as shearing. Fig. 7 shows an example where the minutiae in Fig. 7(b) need to be registered with the minutiae in Fig. 7(a). Fig. 7(c) shows the results of registering the two sets using a rigid transformation while Fig. 7(d) shows the results after a single affine refinement only.

4. Fingerprint super-template synthesis

In this section, we describe our template synthesis algorithm. First, the super-template is initialized by choosing one of the enrollment feature sets based on quality criteria. Then, the remaining enrollment templates are aligned and merged with the current super-template sequentially. During this process, new minutiae might be added to the super-template. Super-template minutiae are associated with several attributes including their spatial coordinates and the number of times they appear in the enrollment templates (i.e., frequency of occurrence). This information serves as a quality measure and is the key to the hierarchical matching algorithm.

In the rest of the paper, we refer to the enrollment feature set that is chosen to initialize the super-template as the ‘‘prime” feature set. The order of merging the rest of the enrollment feature sets with the current super-template is determined based on their similarity with the current super-template. Given a number of enrollment templates, the super-template is built as follows:

(1) Set $t = 1$, choose the highest quality enrollment feature set as the prime template T_p and initialize the super-template $S_1 = T_p$. Set the minutiae frequencies in the initial super-template to one.
(2) Compute the similarity of each of the remaining enrollment feature sets with the current super-template using hierarchical matching and select the template T_n being most similar to the current super-template.
(3) If T_n is ‘‘similar” enough with the current super-template, then merge it with the current super-template: $S_{t+1} = S_t \cup T_n$.
(4) If there are more enrollment feature sets left for merging, then set $t = t + 1$ and go to 2; otherwise exit.

4.1. Prime selection and order of merging

We found that the selection of the prime feature set has a significant impact on the performance of the synthesis algorithm. For example, selecting a low quality feature set might cause subsequent registration operations to fail. We associate a quality index with the minutiae and select the template having the highest quality overall. To associate a quality index with minutiae, we used NIST’s fingerprint processing software [15] to compute a quality map for a fingerprint image. The software computes a block-wise (i.e., 8×8 blocks) quality index map where the quality index is an integer in the range [0–4] (i.e., 0 represents the lowest quality index and 4 represents the highest quality index). To compute a global quality measure for a given fingerprint image, we extract its minutiae and assign a quality index to each of them by overlaying them on the quality map (i.e., see Fig. 8). The average of quality indices over all minutiae provides a global quality measure of the underlying fingerprint image.

Once the prime has been selected, the order of merging is determined by computing the similarity of the remaining enrollment feature sets with the current super-template and choosing the one being most similar to the current super-template. The presence of very low quality enrollment templates could actually hamper the construction of the super-templates because low quality images would not be aligned accurately with the current super-template, increasing spurious minutiae. To deal with this issue, we merge it only if it is similar enough with the current super-template. We used a hysteresis-based decision making approach. If an enrollment template has a similarity score higher than a threshold s_1, the merging takes place. If its similarity score is less than some other threshold s_2, where $s_2 < s_1$, then no merging takes place.

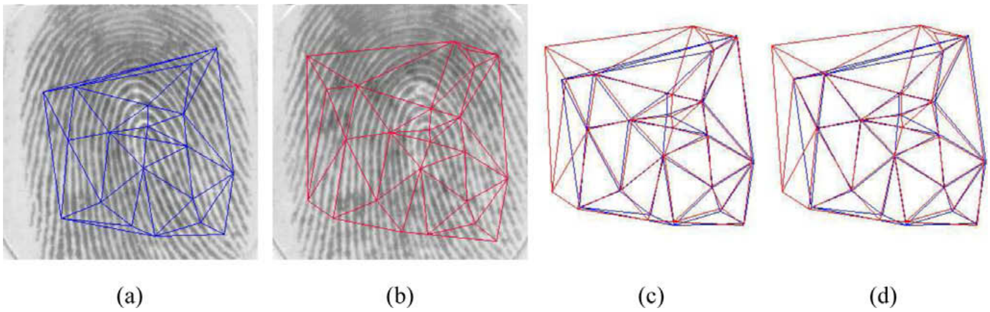


Fig. 7. Effect of affine refinements: (a and b) minutiae sets to be registered; (c) alignment results using a rigid transformation; and (d) registration results after a single affine refinement.

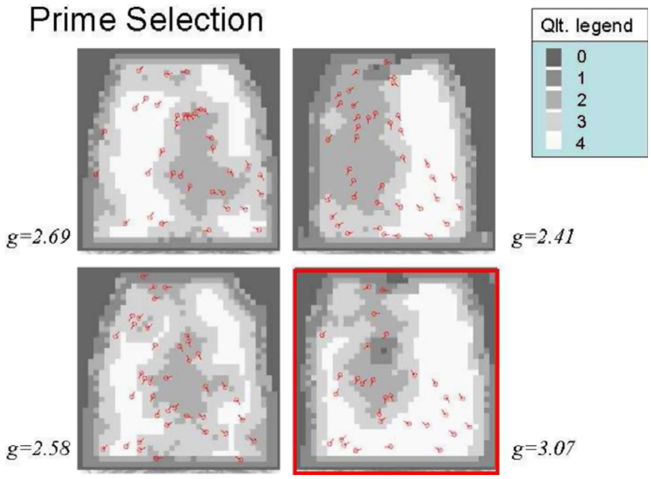


Fig. 8. Quality index map with minutiae overlayed on it. Lighter areas correspond to higher quality regions. The overall quality measure corresponds to average all minutiae quality indices. Using this criterion, the lower-right fingerprint is selected as prime.

However, if its similarity score is between s_2 and s_1, we look at the similarity scores computed for this enrollment template in previous iterations (i.e., assuming an earlier version of the current super-template). If one of these scores was more than s_1, then we merge it with the current super-template; otherwise, no merging takes place. In our experiments, we set s_1 to 20 and s_2 to 10.

4.2. Super-template updating

When an enrollment feature set is merged with the current super-template, then all the minutiae in the super-template that have corresponding minutiae in the enrollment feature set have their weights increased by one. Moreover, minutiae in the enrollment feature set that do not have correspondences in the super-template are added to the super-template and their weights are set to one. Fig. 9 illustrates how the super-template minutiae are updated using the enrollment feature sets shown in Fig. 10. In particular, Fig. 9(a) shows the prime template used to initialize the super-template (i.e., same as Fig. 10(a)), while Fig. 9(b)–(d) shows how the current super-template is updated. As new feature sets are merged with the current super-template, the area of the super-template grows, addressing the small area problem. Moreover, the weights of stable minutiae increase, reflecting their frequency of occurrence in the enrollment templates, addressing the spurious minutiae problem.

5. Fingerprint authentication

During fingerprint authentication, the identity of a new feature set is verified by matching it with the super-template of the subject's finger whose identity is claimed. Although any well known, minutiae-based, matching algorithm could be used for verification, here we used the same hierarchical matching strategy with minor modifications and stricter parameter settings to better account for imposters. This is essential since during merging, the enrollment feature sets and current super-template all come from the same finger. However, this might not be the case during authentication where new templates might belong to different people than those whose identity is claimed.

Specifically, when matching an new feature set to a given super-template for verification purposes, we do not update the super-template although one can envision incremental learning schemes where super-template is updated over time. Second, although affine refinements have shown to improve merging results, they have a tendency to increase imposter scores during authentication. This is because the query template and the super-template might not come from the same finger. Therefore, we do not apply any affine refinements during authentication. Finally, stricter matching thresholds are required for verification in order keep imposters low.

6. Experimental results and comparisons

In this section, we report experimental results to illustrate the performance of the proposed template synthesis and matching approach as well as to compare it with other approaches. For minutiae extraction, we used Verifinger [16] which outputs both the coordinates and orientation of the detected minutiae.

6.1. Database

In choosing a database for experimentation, we looked at several public databases from the Fingerprint Verification Competition (i.e., FVC2000, FVC2002, and FVC2004). Our objective was to choose a database containing multiple impressions from each finger (e.g., more than six), small coverage area, and low quality fingerprint images. Synthetically generated databases were excluded from our consideration. FVC2000 [17] contains four databases; among them, Db1 seemed to satisfy our criteria best (i.e., see details below). It should be noted that according to the FVC2000 competition results, Db3 was the hardest database, however, fingerprint coverage area was much larger compared to Db1. FVC2002 contains four databases too; among them, Db3 satisfied our criteria best, however, competition results on this database were better overall than on FVC2000 Db1. Considering the FVC2004 databases, Db3 satisfied our criteria, however, this database was obtained using a thermal sensor. Based on these observations, we decided that FVC2000 Db1 would be the most appropriate database to be used in our experiments. To obtain statistically significant results, we used cross-validation as explained in Section 6.2.

FVS2000 Db1 was created using a low cost small area optical scanner called *Secure Desktop Scanner* by KeyTronic. It contains a total of 800 300 × 300 images composed of 8 different impressions from 100 different fingers. It corresponds to a typical, small area, low quality database. Fingerprint images were captured from untrained people in two different sessions and no efforts were made to assure a minimum acquisition quality. Fig. 11 shows several representative samples from this database.

6.2. Experiments

In our experiments, we tested three different approaches: (i) Score-Level Fusion (SLF), (ii) Template Selection (T_SEL), and (iii) Template Synthesis (T_SYN). In the case of SLF, multiple enrollment feature sets were stored in the enrollment database for each finger. To verify a new (i.e., query) feature set, we computed its similarity with each of the enrollment feature sets of the user's finger whose identity was claimed and took the maximum matching score. We also considered taking the average matching score, which is known to work well in practice, however, it did not work as good as the maximum score in our case. In the case of T_SEL, only one of the enrollment feature sets was chosen to represent each finger. We used the prime selection methodology described in Section 4.1 to select a representative enrollment feature set for each enrolled finger user. In both approaches, we used the matching algorithm described in Appendix B.

For experimentation, the database was randomly divided into two partitions, keeping N impressions in the first partition and

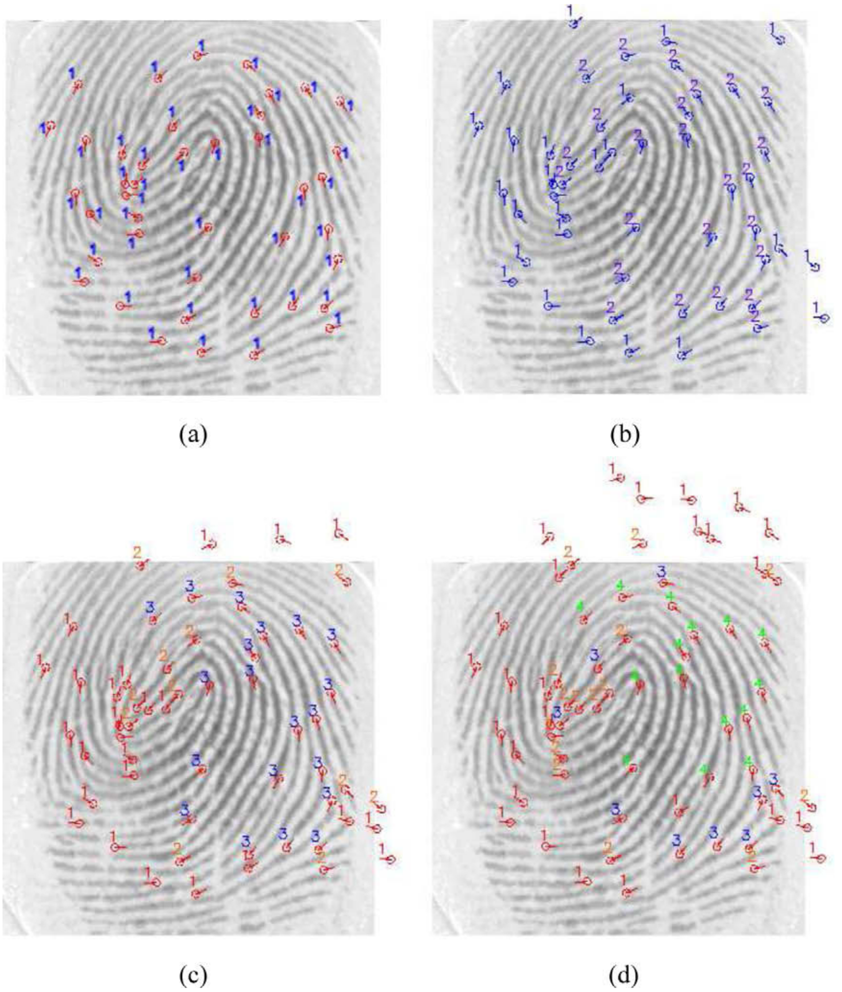


Fig. 9. Super-template updating: (a) prime feature set used to initialize the super-template; (b) current super-template after merging with the enrollment feature set shown in Fig. 10(b); (c) current super-template after merging with the enrollment feature set shown in Fig. 10(c); and (d) current super-template after merging with the enrollment feature set shown in Fig. 10(d).

the rest in the second partition. We used the first partition to create the enrollment database and the second partition for testing. To show the effect of choosing a different number of enrollment impressions, we conducted experiments using $N = 2$, 3, 4, and 5. Since randomness is involved in the selection of the enrollment and test impressions, we repeated each experiment 30 times and reported average performance.

For error rate estimation, we randomly picked one sample from each finger in the test set and compared it against the enrollment data of all other fingers yielding a total of 9900 imposter scores. The number of genuine scores varied, depending on the number of enrollment impressions. Using 2, 3, 4, and 5 impressions for enrollment, the number of genuine scores were 600, 500, 400, and 300, respectively. Although the number of genuine scores is not very high, the use of cross-validation yields statistically significant results. For evaluation, we computed the Receiver Operating Characteristics (ROC) curve, which shows the variation of False Acceptance Rate (FAR) versus False Rejection Rate (FRR). Specific FRR readings at certain FAR rates (i.e., FAR = 0.0%, 0.1%, and 1.0%) are also reported for a more detailed comparison.

7. Results

Fig. 12 compares each method using different number of templates. The graphs have been logarithmically scaled for clarity. Clearly, using more feature sets in enrollment improves the performance of T_SLF and T_SYN, however, it does not have any effect on the performance of T_SEL. Since FVC2000 Db1 is a low quality database, it can be clearly observed that using multiple feature sets in enrollment does make a difference in terms of accuracy and consistency. Using one template, however, as in the case of T_SEL, yields lower performance due to noisy features and small coverage area. Overall, T_SEL performed worst compared to SLF and T_SYN, especially when increasing the number of feature sets in enrollment.

Comparing SLF with T_SYN, we found that SLF performed slightly better than T_SYN but performance differences got smaller with increasing the number of feature sets in enrollment or FAR rate as shown in Fig. 13. Table 1 shows some specific FRR readings at certain FAR values for a more detailed comparison. These results were somewhat surprising to us as we expected T_SYN to outperform SLF. An analysis of our results, however, revealed that the main reason that T_SYN did not outperform SLF was because there were inaccuracies in registering the enrollment feature sets with the current super-template during merging.

Although SLF performed slightly better than T_SYN in terms of verification accuracy, SLF had higher storage and time requirements compared to T_SYN. When $N = 4$, the average storage requirements of SLF, T_SEL, and T_SYN were 1540, 385, and 729 bytes, respectively, per finger. In terms of time, T_SEL was the fastest, requiring 0.56 s on the average. T_SYN was faster on the average than SLF, taking 1.84 s versus 2.24 s.

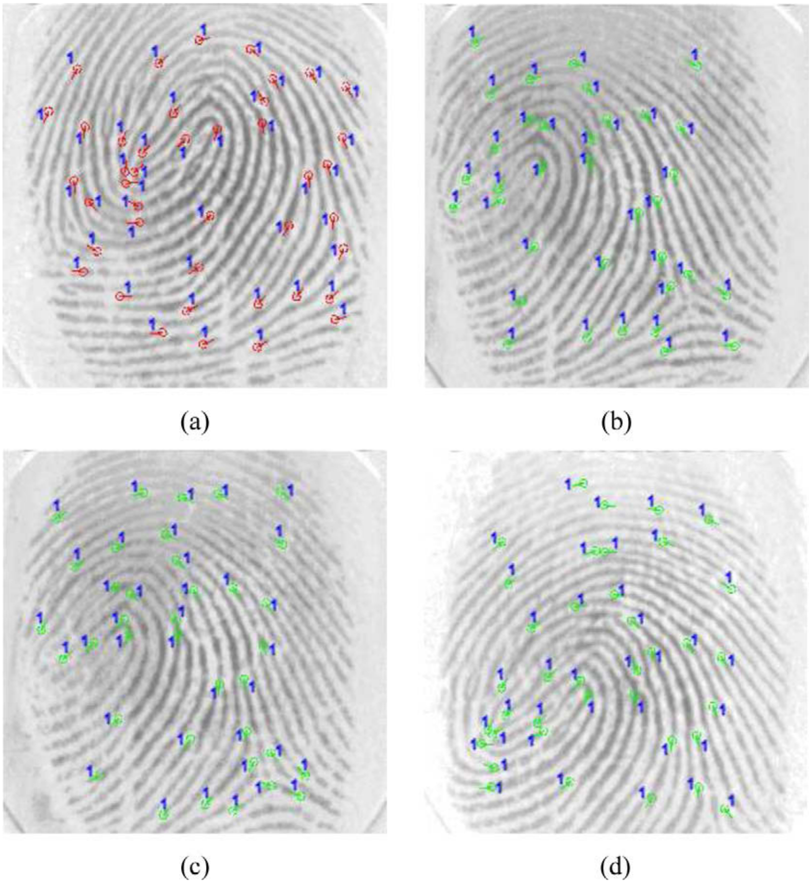
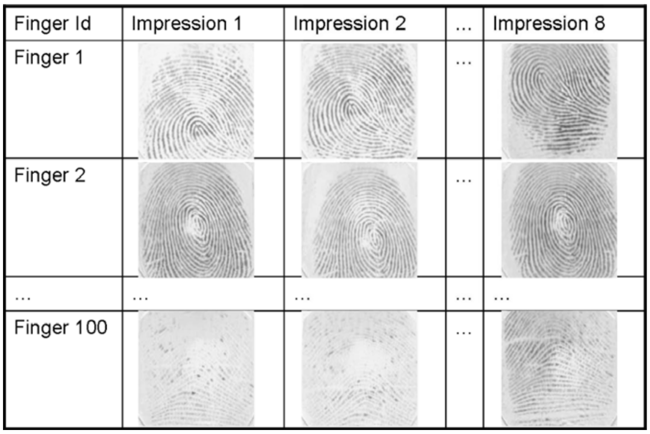

(a) (b) (c) (d)

Fig. 10. The enrollment feature set used to build the super-template shown in Fig. 9.



Finger Id	Impression 1	Impression 2	...	Impression 8
Finger 1			...	
Finger 2			...	
...	...	...	...	...
Finger 100			...	

Fig. 11. Typical samples from Db_1 FVC 2000 database.

Table 2 provides some statistics on super-template minutiae counts, as well as distribution of weights. Compared to individual feature sets, the number of super-template minutiae increased by 25.95%, 47.46%, 65.53%, and 81.46% when merging 2, 3, 4, and 5 enrollment impressions, respectively. When merging two impressions, the percentage of minutiae having weights more than 1 was 48.46% of the total number of minutiae in the super-template. The corresponding percentages for 3, 4, and 5 impressions were 55.97%, 59.95%, and 60.83%.

8. Conclusion

We have considered the problem of integrating information from multiple enrollment impressions in order to account for small coverage area and missing or spurious minutiae. In this context, we proposed a new, minutiae-based, template synthesis algorithm which employs a novel matching strategy based on Delaunay triangulation hierarchies. The key idea was performing the matching hierarchically by considering higher quality minutiae first. Lower quality minutiae were considered for matching on an incremental basis. The same hierarchical matching strategy, with minor modifications, was used for authenticating new fingerprints. The proposed template synthesis approach and matching approach is less sensitive to missing or spurious minutiae and addresses the small overlapping area problem. Our main contributions can be summarized as follows:

- Proposed a novel hierarchical fingerprint matching strategy which accounts for missing and spurious minutiae by using Delaunay triangulation hierarchies.
- Proposed a new methodology for minutiae-template synthesis (i.e., building a "super-template" from a set of minutiae templates) which employs the hierarchical fingerprint matching strategy.
- Performed extensive experiments and comparisons with traditional methods to evaluate the performance of the proposed minutiae-template synthesis and matching methodology.

T_SYN is better suited for updating the template over time (e.g., during verification) compared to SLF. Moreover, although SLF per-

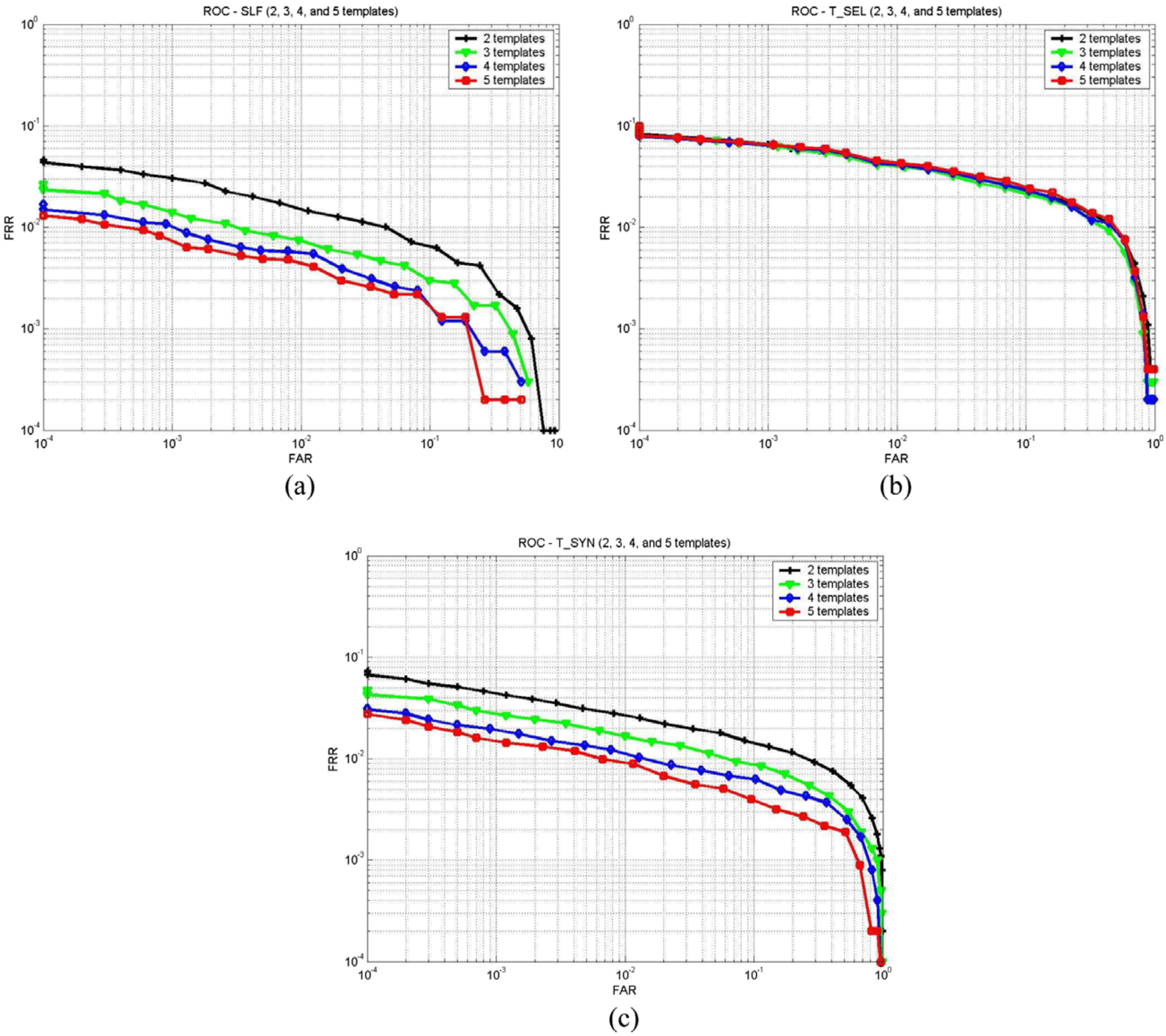


Fig. 12. ROC curves for each method assuming different number of templates: (a) SLF; (b) T_SEL; and (c) T_SYN.

formed slightly better than T_SYN in terms of verification accuracy, T_SYN has lower time and space requirements. If accuracy is more important, multiple super-templates can be generated from the available fingerprint images by choosing different prime fingerprint templates of even by changing the order of merging. Then, SLF could be applied using multiple super-templates.

For future research, we plan to improve the proposed approach in several ways. First, we plan to employ more powerful registration algorithms in order to build the super-templates more accurately (e.g., using thin-plate spines [18,19]). Second, we plan to consider additional minutiae features (e.g., ridge count information) in order to reduce the probability of false alignments during merging. Third, we plan to enhance our hierarchical matching strategy. Finally, we plan to perform additional experiments using more databases.

Appendix A. Delaunay triangulation

Given a set S of points $p_1, p_2, \ldots, p_N$, we can compute the Delaunay triangulation of S by first computing its Voronoi diagram. The Voronoi diagram decomposes the 2D space into polygonal regions around each point p_i such that all the points in the region of p_i are closer to p_i than to any other point in S. Given the Voronoi diagram, the Delaunay triangulation can be formed by connecting the centers of every pair of neighboring Voronoi polygons. Fig. A1(a) shows a set of 2D points; their Voronoi diagram is shown in Fig. A1(b) while their Delaunay triangulation is shown in Fig. A1(c).

The Delaunay triangulation has some nice properties, including: (1) it is unique assuming a non-degenerate set of points, (2) a circle through the three points of a Delaunay triangle contains no other points, and (3) the minimum angle across all the angles in all the triangles in a Delaunay triangulation is greater than the minimum angle in any other triangulation of the same points. Property 1 is very important as it allows us to use the Delaunay triangles for matching. Property 2 implies that inserting a new point affects only the Delaunay triangles whose circumcircles contain that point. This implies that missing/spurious minutiae would affect the Delaunay triangulation only locally. In a comparative study involving several well known topological structures [20], the Delaunay triangulation was found to have the best structural stability under random positional perturbations. The last property implies that the Delaunay triangles would not be *skinny*. This is very important in the context of our application since using skinny triangles to compute alignment transformations between fingerprints could lead to serious instabilities and errors.

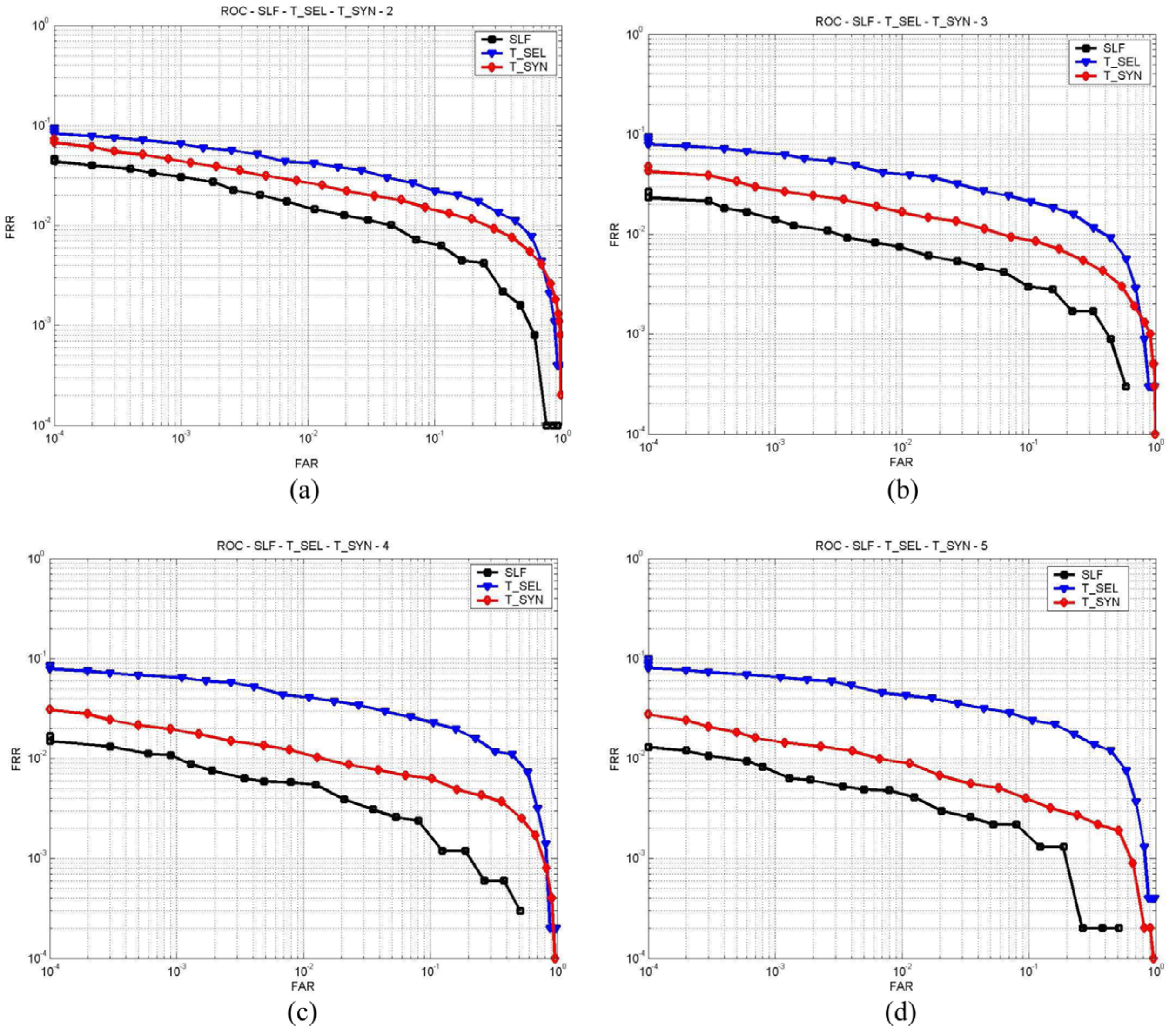


Fig. 13. ROC curves corresponding to 2, 3, 4, and 5 templates.

The Delaunay triangulation and the Voronoi diagram can be computed very efficiently since the number of edges in both cases is of the order of $O(N)$. Since each edge belongs to at most two triangles or polygons, the number of triangles generated by the Delaunay triangulation is also linear to the number of points. In this work, we used Fortune's implementation which is available from http://netlib.bell-labs.com/netlib/voronoi. The complexity of the triangulation algorithm is $O(NlogN)$.

Appendix B. Fingerprint matching using Delaunay triangulation

Employing Delaunay triangulation for fingerprint matching has received a lot of attention recently, for example see [21–26]. Our matching strategy for template synthesis builds upon our previous work on fingerprint matching using Delaunay triangulation [2]. Next, we review fingerprint matching using Delaunay triangulation. To improve matching, we have incorporated orientation information into the matching process.

B.1. Fingerprint matching using minutiae triangles

A common approach to matching a pair of minutiae sets is by comparing minutiae triangles. The main idea is forming triangles from minutiae triplets and matching them using invariant features [27,2]. Fig. B1 shows an example. In general, a pair of minutiae triangles provides enough information to compute a transformation that potentially aligns the minutiae sets. To compute good alignments, voting is applied in the transformation space to find transformation that are supported by many minutiae triangles. A number of hypothetical transformations is then found by considering transformations that have received high number of votes. Each hypothetical transformation is explicitly verified by using it to align the minutiae sets and counting the number of overlapping minutiae. The best alignment is the one maximizing the number of overlapping minutiae.

An important issue in matching is which minutiae triplets to choose in order to form the minutiae triangles. Considering all possible minutiae triplets is computationally prohibitive since there are $O(n^3)$ minutiae triplets. To keep complexity low, Germain et al. [27] suggested a number of heuristics based on the distance between minutiae. In a later study [2], we proposed associating a unique topological structure with the minutiae using Delaunay triangulation and using the Delaunay triangles for matching. This reduces the number of minutiae triangles to $O(n)$, speeding up matching considerably without affecting accuracy significantly. Fig. B3 shows the Delaunay triangulation of a set of minutiae, overlaid on the corresponding fingerprint image.

Table 1
Specific FRR readings at certain FAR values for specific comparisons.

		Number of templates			
		2	3	4	5
SLF	FRR (@FAR = 0.000)	5.00% (std:0.83)	2.92% (std:0.69)	1.83% (std:0.58)	1.53% (std:0.61)
	FRR (@FAR = 0.001)	3.25% (std:0.72)	1.57% (std:0.48)	1.03% (std:0.40)	0.91% (std:0.48)
	FRR (@FAR = 0.010)	1.70% (std:0.50)	0.76% (std:0.26)	0.58% (std:0.25)	0.48% (std:0.39)
T_SEL	FRR (@FAR = 0.000)	9.88% (std:1.38)	9.77% (std:1.55)	9.30% (std:2.20)	9.20% (std:2.31)
	FRR (@FAR = 0.001)	6.77% (std:0.95)	6.75% (std:0.73)	6.73% (std:1.16)	6.85% (std:1.21)
	FRR (@FAR = 0.010)	4.40% (std:0.74)	4.14% (std:0.62)	4.31% (std:0.92)	4.48% (std:0.94)
T_SYN	FRR (@FAR = 0.000)	8.71% (std:2.62)	5.23% (std:1.55)	3.90% (std:1.13)	3.62% (std:1.19)
	FRR (@FAR = 0.001)	4.68% (std:0.74)	3.10% (std:1.03)	2.03% (std:0.50)	1.60% (std:0.54)
	FRR (@FAR = 0.010)	2.81% (std:0.62)	1.75% (std:0.47)	1.23% (std:0.47)	0.99% (std:0.48)

Table 2
Super-template minutiae and weight statistics.

	Number of templates				
	1	2	3	4	5
Avg. # of minutiae	32.64	41.11	48.13	54.03	59.23
Weight 5	N/A	N/A	N/A	N/A	10.40
Weight 4	N/A	N/A	N/A	11.79	7.37
Weight 3	N/A	N/A	13.54	9.65	7.86
Weight 2	N/A	19.92	12.30	10.95	10.40
Weight 1	32.64	22.29	21.19	21.64	23.20

B.2. Invariant features

Once the Delaunay triangulation of a set of minutiae has been computed, we consider two groups of features from each triangle based on the sides and angles of the triangle. Specifically, given a minutiae triangle (e.g., see Fig. B2(a)), the first group of features, denoted as V_t, includes three attributes which are invariant to rigid transformations:

$$V_t = \left[\frac{l_1}{l_3}, \frac{l_2}{l_3}, \cos(A)\right] \tag{B1}$$

where $l_1 \leqslant l_2 \leqslant l_3$ and A is the largest angle of the triangle (i.e., the angle across from the largest side). We used the cosine of A instead of A itself because the cosine is less sensitive to noise. The second group of features, denoted as V_m, involves the angles of the minutiae. Specifically, we order the minutiae forming the triangle with respect to the length of the side across them and take their angles as the second set of features:

$$V_m = [\angle m_1, \angle m_2, \angle m_3] \tag{B2}$$

It should be mentioned that very large angles yield triangles whose points are almost collinear (i.e., skinny). Such triangles are not desirable since small errors in minutiae locations can lead to large errors in the computation of the parameters of the alignment transformation. Although the Delaunay triangulation tends to avoid skinny triangles, it is not always guaranteed unless inserting extra points [28]. To deal with this issue, we reject triangles whose largest angle is greater than a threshold (e.g., 168 deg). Fig. B3 shows an example of minutiae features extracted by our algorithm.

$$V_m = [\angle m_1, \angle m_2, \angle m_3] \tag{B3}$$

B.3. Hypothesis generation

The goal of this step is generating a number of hypothetical alignments between minutiae sets. This is performed by finding corresponding minutiae triangles using the invariant attributes. To improve accuracy, we have modified our original algorithm by incorporating minutiae orientation information in matching. Minutiae orientation is defined by the orientation of the ridge containing the feature. Specifically, let **T** and **Q** correspond to two different minutiae templates. Then, triangles in **T** are compared to triangles in **Q** using the following three constraints:

(1) *Similarity consistency:* This constraint tests the similarity between two minutiae triangles using the invariant features. As shown in Fig. B3, each minutiae triangle is represented by six invariants. If the differences between corresponding pairs of invariants are all below a threshold, then we consider that the triangles match. In our experiments, the thresholds used for the spatial and angular features were 0.3 and 0.5, respectively. These thresholds were chosen relatively high in order to account for non-linear minutiae dislocations. Many false matches are subsequently eliminated by the third criterion.

(2) *Planarity consistency:* This constraint tests whether matching minutiae triangles can be brought into alignment using in-plane transformations only. Fig. B4 illustrates this criterion with an example. If we order the minutiae in Fig. B4 starting from the first one in each triangle and going clockwise, the ordering in the left triangle would be $m_{11}m_{12}m_{13}$ while the ordering in the right triangle would be $m_{21}m_{23}m_{22}$. Obviously, an out-of-plane transformation is required in order to align the triangles in this case. Such inconsistencies can

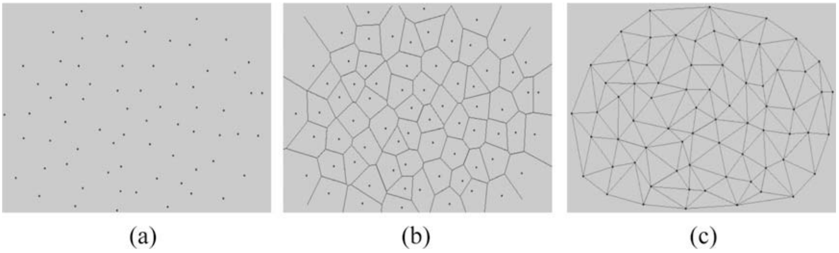


Fig. A1. Delaunay triangulation of a 2D point set: (a) a set of points; (b) the Voronoi diagram; and (c) the Delaunay triangulation.

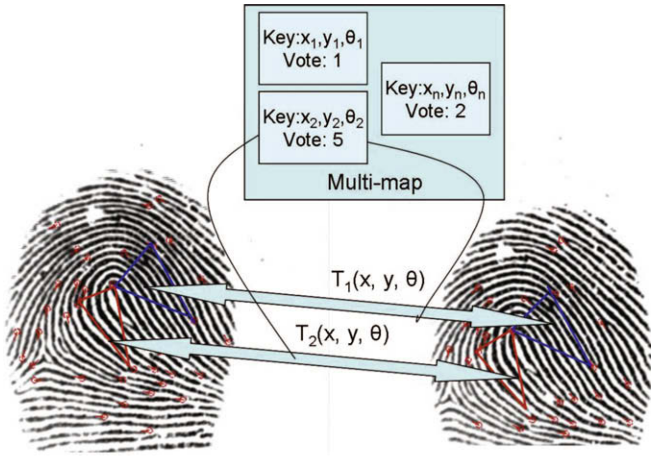


Fig. B1. Matching by comparing minutiae triangles. Good alignment are computed by using voting in the transformation space to find transformations that are supported by many matching triangles.

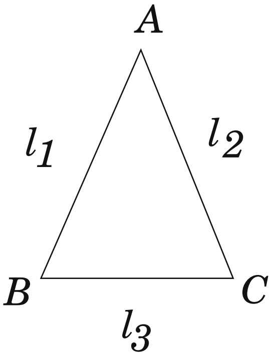


Fig. B2. A triangle defined by a triplet of minutiae.

be fixed by changing the order of the minutiae in the left triangle (e.g., starting from m_{12}). Then, the similarity consistency between the two triangles can be tested using the new ordering. We consider all three possible orders.

(3) *Minutiae orientation consistency*: The purpose of this constraint is to test whether corresponding minutiae have similar orientations. This is performed by estimating the rigid transformation that aligns corresponding triangles and computing the orientation differences of the corresponding minutiae. If the average orientation difference is below a threshold (e.g., 30 deg), then we consider that the corresponding minutiae have similar orientations.

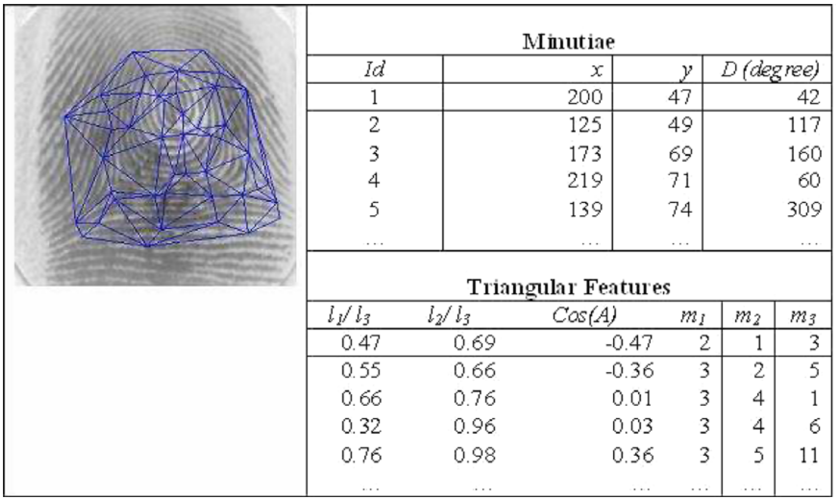
If all three constraints are satisfied for a given pair of minutiae triangles, then we concluded that the minutiae triangles match. Given a pair of matching triangles, a rigid transformation can be computed which aligns them. These, "locally optimum", transformations are used to substantiate a number of hypothetical alignment transformations between the minutiae sets which are supported by many corresponding triangles. To find these, "globally consistent", transformations we employ a voting scheme where each matching pair of triangles casts a weighted vote in the transformation space. The weight of a vote is inversely proportional to the average minutiae orientation differences. To compensate for quantization errors in the transformation space, we also cast votes to the immediate neighbors of the estimated transformation using lower weights (i.e., 2/3 of the vote cast to the estimated transformation). Alignment transformations that receive high votes are considered for further verification.

Fig. B5 shows the structure of the entries in the transformation space. Each entry holds (i) the transformation parameters, (ii) the number of votes, and (iii) a list of corresponding minutiae that have voted for this transformation. It should be mentioned that the transformation space needs to be quantized coarsely enough to let the entries receive enough votes and build reasonable histograms. Since we align the minutiae triangles using rigid transformations, the transformation space is three-dimensional (i.e., *x*, *y*, and θ). However, since we use $\cos(\theta)$ and $\sin(\theta)$ in our calculations frequently, we decided to store this information instead of the actual angle for efficiency.

After all the votes have been accumulated, local maxima in the transformation space are detected and considered as possible candidates for aligning the minutiae sets. The resulting set of transformations yields a set of hypotheses which are verified in the next stage.

B.4. Hypothesis verification

In this stage, the minutiae templates are aligned using the hypothetical transformations in order to determine the highest number of overlapping minutiae. First, each candidate transforma-



Minutiae

Id	x	y	D (degree)
1	200	47	42
2	125	49	117
3	173	69	160
4	219	71	60
5	139	74	309
…	…	…	…

Triangular Features

l_1/l_3	l_2/l_3	Cos(A)	m_1	m_2	m_3
0.47	0.69	-0.47	2	1	3
0.55	0.66	-0.36	3	2	5
0.66	0.76	0.01	3	4	1
0.32	0.96	0.03	3	4	6
0.76	0.98	0.36	3	5	11
…	…	…	…	…	…

Fig. B3. An example of minutiae features extracted by our algorithm.

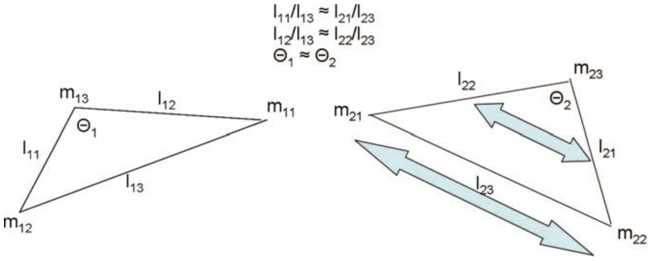


Fig. B4. An example illustrating an inconsistency in the ordering of the minutiae between triangles.

X	Y	cos θ	sin θ	Vote	p_{11}	p_{12}	…	p_{1n}
					p_{21}	p_{22}		p_{2n}

Fig. B5. Structure of entries in the transformation space.

tion is refined (i.e., re-estimated) using all matching triangles that have voted for that transformation. Unlike local transformation computations, which are based only on a pair of minutiae triangles, the refinement process computes globally consistent transformations by considering minutiae correspondences scattered over a diverse region of the fingerprint. Then, the quality of each hypothesis is evaluated by aligning the minutiae sets and computing the number of overlapping minutiae.

The overlap between the minutiae sets is determined by considering the differences between corresponding minutiae locations and orientation angles. The number of overlapping minutiae is normalized to calculate a similarity score between the minutiae sets. Specifically, let t and q be the number of minutiae in the two sets, respectively, if m is the number of matching minutiae, then the similarity score s is calculated as follows:

$$s = \frac{2m}{t+q} \times 100 \tag{B4}$$

The hypothesis yielding the highest number of overlapping minutiae is taken as the best hypothesis. It should be mentioned that we tried several other ways to compute similarity scores, however, the formula above gave the best results.

References

[1] D. Maltoni, D. Maio, A. Jain, S. Prabhakar, Handbook of Fingerprint Recognition, Springer-Verlag, 2003.

[2] G. Bebis, T. Deaconu, M. Georgiopoulos, Fingerprint identification using Delaunay triangulation, in: IEEE Conference on Information, Intelligence and Systems, 1999, pp. 452–459.

[3] A. Jain, U. Uludag, A. Ross, Biometric template selection: a case study in fingerprints, in: Audio- and Video-based Biometric Person Authentication Conference, 2003, pp. 335–342.

[4] N. Ratha, J. Connell, R. Bolle, Image mosaicking for rolled fingerprint construction, in: International Conference on Pattern Recognition, 1998, pp. 1651–1653.

[5] S. Shah, A. Ross, J. Shah, S. Crihalmeanu, Fingerprint mosaicking using thin plate splines, in: The Biometric Consortium Conference, September 2005.

[6] K. Toh, W. Yau, X. Jiang, T. Chen, J. Lu, E. Lim, Minutiae data synthesis for fingerprint identification applications, in: IEEE International Conference on Image Processing, 2001, pp. 262–265.

[7] X. Jiang, W. Ser, Fingerprint template improvement, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (8) (2002) 1121–1126.

[8] A. Jain, A. Ross, Fingerprint mosaicking, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002.

[9] Y. Zhang, J. Yang, H. Wu, A hybrid swipe fingerprint mosaicking scheme, in: Audio- and Video-based Biometric Person Authentication Conference, Rye Brook, NY, July 2005, pp. 131–140.

[10] K. Choi, H. Choi, J. Kim, Fingerprint mosaicking by rolling and sliding, in: Audio- and Video-based Biometric Person Authentication Conference, Rye Brook, NY, July 2005, pp. 260–269.

[11] W. Yau, K. Toh, D. Jiang, T. Chen, J. Lu, On fingerprint template synthesis, in: International Conference on Control, Automation, Robotics and Vision, 2001.

[12] Y. Moon, H. Yeung, K. Chan, S. Chan, Template synthesis and image mosaicking for fingerprint registration: an experimental study, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004, pp. 409–412.

[13] W.C. Ryu, Y. Han, H. Kim, Super-template generation using successive Bayesian estimation for fingerprint enrollment, in: Audio- and Video-based Biometric Person Authentication, 2005, pp. 710–719.

[14] A. Ross, S. Shah, J. Shah, Image versus feature mosaicking: a case study in fingerprints, in: SPIE Conference on Biometric Technology for Human Identification III, 2006, pp. 620208-1–620208-12.

[15] Nist fingerprint image software, NFIS, National Institute of Standards and Technology.

[16] Verifinger fingerprint identifications system. Available from: <http://www.neurotechnologija.com/verifinger.html>.

[17] D. Maio, D. Maltoni, R. Capelli, J. Wayman, A. Jain, Fvc2000: Fingerprint verification competition, in: International Conference on Pattern Recognition, 2000.

[18] D. Kwon, I. Yun, S. Lee, A robust warping method for fingerprint matching, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–6.

[19] A. Ross, S. Dass, A. Jain, A deformable model for fingerprint matching, Pattern Recognition 38 (2005) 95–103.

[20] M. Tuceryan, T. Chorzempa, Relative sensitivity of a family of closest-point graphs in computer vision applications, Pattern Recognition 24 (5) (1991) 361–373.

[21] H. Deng, Q. Huo, Minutiae matching based fingerprint verification using Delaunay triangulation and aligned-edge-guided triangle matching, in: Audio- and Video-based Biometric Person Authentication Conference, 2005, pp. 270–278.

[22] A. Ross, R. Mukherjee, Augmenting ridge curves with minutiae triplets for fingerprint indexing, in: SPIE Biometric Technology for Human Identification IV, vol. 6539, 2006.

[23] N. Liu, Y. Yin, H. Zhang, A fingerprint matching algorithm based on Delaunay triangulation net, in: IEEE International Conference on Computer and Information Technology, 2005, pp. 591–595.

[24] G. Parziale, A. Niel, A fingerprint matching using minutiae triangulation, in: International Conference on Biometric Authentication, 2004, pp. 241–248.

[25] C. Wang, M. Gavrilova, Delaunay triangulation algorithm for fingerprint matching, in: IEEE International Symposium on Voronoi Diagrams in Science and Engineering, 2006, pp. 208–216.

[26] X. Liang, T. Asano, A. Bishnu, Fingerprint matching using minutia polygons, in: International Conference on Pattern Recognition, 2006, pp. 1046–1049.

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

[28] S. Skiena, The Algorithm Design Manual, Springer-Verlag, New York, 1998.

Tamer Uz. Tamer Uz received the B.S. degree in Electrical and Electronics Engineering from Middle East Technical University, Turkey, in 1997 and M.S. degree in Computer Science from the University of Nevada, Reno, in 2006. Currently, he is a software engineer at the International Game Technology (IGT), in Reno, NV. He works on slot machine communication protocols development.

George Bebis. George Bebis received the B.S. degree in mathematics and M.S. degree in computer science from the University of Crete, Greece in 1987 and 1991, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Central Florida, Orlando, in 1996. Currently, he is a Professor with the Department of Computer Science and Engineering at the University of Nevada, Reno (UNR) and Director of the UNR Computer Vision Laboratory (CVL). His research interests include computer vision, image processing, pattern recognition, machine learning, and evolutionary computing. His research is currently funded by NSF, NASA, ONR, and Ford Motor Company. Dr. Bebis is an associate editor of the Machine Vision and Applications Journal, and serves on the Editorial Board of the Pattern Recognition Journal and the International Journal on

Artificial Intelligence Tools. He has served on the program committees of various national and international conferences, and has organized and chaired several conference sessions. In 2002, he received the Lemelson Award for Innovation and Entrepreneurship. He is a member of the IEEE and the IAPR Educational Committee.

Ali Erol. Ali Erol received the B.S. degree in electrical and electronics Engineering from Bilkent University, Turkey in 1991 and the M.Sc. and Ph.D. degrees in electrical and electronics engineering from Middle East Technical University, Turkey in 1995 and 2001, respectively. He worked as a post-doctoral fellow in the Computer Vision Laboratory (CVL) of the Department of Computer Science and Engineering at the University of Nevada, Reno (UNR). Currently, he is working as a research scientist in Ocali Software, Turkey for the development of an image-based 3D reconstruction software. His research interests include computer vision, image processing and pattern recognition.

Salil Prabhakar. Salil Prabhakar received the B.Tech. degree in computer science and engineering from the Institute of Technology, Banaras Hindu University, Varanasi, India, in 1996. After working for IBM India for a year, he joined the Department of Computer Science and Engineering at Michigan State University, East Lansing, where he completed the Ph.D. degree in 2001. Since 2001, he has been the chief scientist at Digital-Persona, Inc., a leading provider of fingerprint-based biometric solutions. Dr. Prabhakar's research interests include pattern recognition, image processing, computer vision, machine learning, biometrics, data mining, and multimedia applications. He is coauthor of more than 30 technical publications and holds two patents. He coauthored the Handbook of Fingerprint Recognition (Springer, 2003), which received the PSP award from the Association of American Publishers. He is an associate editor of the Pattern Recognition Journal and IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI). He was one of the guest editors of the April 2007 special issue of IEEE TPAMI on "biometrics: progress and directions". He is a senior member of the IEEE.