PERGAMON

Pattern Recognition 32 (1999) 877–889

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

Fingerprint minutiae extraction from skeletonized binary images

Alessandro Farina, Zsolt M. Kovács-Vajna*, Alberto Leone

D.E.I.S., University of Bologna, Viale Risorgimento 2, I-40136 Bologna, Italy

Received 26 June 1997, in revised form 25 June 1998

Abstract

Fingerprint comparison is usually based on minutiae matching. The minutiae considered in automatic identification systems are normally ridge bifurcations and terminations. In this paper we present a set of algorithms for the extraction of fingerprint minutiae from skeletonized binary images. The goal of the present work is the extraction of the real 40–60 minutiae of a fingerprint image from the 2000–3000 contained in typical skeletonized and binarized images. Besides classical methodologies for minutiae filtering, a new approach is proposed for bridge cleaning based on ridge positions instead of classical methods based on directional maps. Finally, two novel criteria and related algorithms are introduced for validating the endpoints and bifurcations. Statistical analysis of the results obtained by the proposed approach shows efficient reduction of spurious minutiae. The use of the fingerprint minutiae extraction algorithms has also been considered in a fingerprint identification system in terms of timing and false reject or acceptance rates. The presented minutiae extraction algorithm performs correctly in dirty areas and on the background as well, making computationally expensive segmentation algorithms unnecessary. The results are confirmed by visual inspections of validated minutiae of the NIST sdb 4 reference fingerprint image database. © 1999 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Image processing; Binary image; Post-processing fingerprint; Minutiae; NIST sdb 4

1. Introduction

Research in the design of automatic fingerprint identification systems (AFIS) is currently focusing on fingerprint classification into five classes [1] and on fingerprint matching, whose goal is the identification of a person by means of the fingerprints. If the stored image database is huge, the identification process needs both the classification and the matching stages. In this paper we focus our attention on a part of the matching stage, which is usually based on preprocessing, feature extraction and matching algorithms.

There are several useful features in fingerprints, related to ridge topology, called minutiae. The American National Standards Institute has proposed a classification of minutiae into four main groups: terminations, bifurcations, crossovers and undetermined [2]. The most important minutiae types are terminations and bifurcations.

In a fingerprint identification system the fingerprint image is obtained by a sensor or camera and the matching process starts with image filtering. The minutiae are extracted and they are compared with those of the

*Corresponding author. Fax: (+39) 51 6443073; e-mail: zkovacs@deis.unibo.it

PII: S0031-3203(98)00107-1

already stored images in order to establish the correspondence. The minutiae extraction can be based on gray-scale [3] or binary images.

In this paper we focus our attention on a binary image-based technique, where we assume that the fingerprint image has been either acquired directly in binary or binarized from a gray-scale image. We assume also that the image has been skeletonized. The problem in this methodology comes from the fact that a minutia in the skeleton image does not always correspond to a meaningful point in the real image. In fact, there are more than a thousand apparent ending or bifurcation points, while the real minutiae are less than 100. Such a behavior arises as a consequence of under-inking, over-inking, wrinkle, scars and excessively worn prints, and, therefore, spurious minutiae can appear in the skeleton image after pre-processing.

In order to simplify or to preserve the reliability of the AFIS and to lower computational costs, post-processing of the skeleton image is necessary to reduce the number of spurious minutiae.

Several approaches have been already proposed in the last few years to enhance skeletonized images.

The AFIS proposed by Stosz and Alyea [4] uses pore positions coupled to other minutiae extracted from live scanned images: the quality of the valley skeleton is first improved by analyzing and extracting segments that represent pores (*cleaning*), then syntactic processing is used to remove undesirable artifacts: two disconnected ridges are connected if their distance is less than a given threshold and endpoint directions are almost the same; wrinkles are detected by analyzing information on neighboring branch points.

Chen and Kuo [5] adopt a three-step false minutiae identification process: (i) ridge breaks are repaired somehow using the ridge directions close to the minutiae; (ii) minutiae associated with short ridges are dropped; (iii) crowded minutiae in a noisy region are dropped. Malleswara's methods [6] are used to eliminate those false minutiae which are caused by noise or imperfect image processing.

A post-processing algorithm applied to binarized and not thinned images is proposed by Fitz and Green [7]. These techniques are employed to remove small holes, breaks and lines in and along the ridges and they are implemented by convolving morphological operators with the image.

Spikes caused by the skeletonizing process can be removed by the adaptive morphological filter proposed by Ratha et al. [8]. Minutiae are detected by counting the BLACK neighbors in a 3×3 window. False endpoints and bifurcations are removed by means of three heuristic criteria: (i) two endpoints with the same orientation and whose distance is below a threshold are deleted (ridge break elimination); (ii) an endpoint connected to a bifurcation and below a certain threshold distance is removed (spike elimination); (iii) minutiae below a certain distance from the boundary of the foreground region are cancelled (boundary effect treatment).

The duality between valley image and ridge image has been exploited by Hung [9] considering that bifurcations and bridges in one of the two images correspond to endings and breaks, respectively, in the dual image. Therefore, the same algorithm can be applied both to valley and ridge images to remove ridge breaks and bridges. A graph is defined for both images by assigning a vertex to each endpoint and bifurcation and by assigning an edge to each ridge. Each edge is characterized by the length of the corresponding ridge, while the degree of a vertex is given by the number of converging edges. Spurs, holes, bridges and breaks are then removed by considering some property of the ridge graph. The same algorithm can be mostly applied to the valley graph to remove ridge breaks, crossovers and other particular configurations.

Xiao and Raafat [10] assume that fingerprints have already been pre-processed and skeletonized. They propose a post-processing based on a combined statistical and structural approach. Fingerprint minutiae are characterized by a set of attributes like the ridge length, the direction for endpoints and bifurcations, the angle between two minutiae. Moreover, each endpoint or bifurcation is also characterized by the number of "facing" minutiae in its neighborhood. The neighborhood area depends on the local distance between ridges. Finally, the number of "connected" minutiae is evaluated. The post-processing algorithm connects facing endpoints, removes bifurcations facing with endpoints or other bifurcations and then connects the newly generated endpoints, and finally removes spurs, bridges, triangle structures and ladder structures. Unfortunately, it is not possible to assign a "global" ridge distance to the whole image, because it varies up to 300% of its local value in a typical fingerprint. Several methodologies of the local ridge distance map computation may be found in the literature [11].

The algorithm proposed in this paper works on the skeleton image obtained from a binarized version of the fingerprint image. Standard algorithms are used to remove close minutiae in noisy areas. Ridge repair is performed on the base of the directions of the two ridge pieces to be connected and the direction of the segment which should rebuild the broken ridge, while previous works suggested the use of structural and statistical considerations [10] or analysis of the distance between the two broken ridges [8]. The novel approach proposed to remove bridges is based on the ridge positions instead of evaluating CPU-consuming direction maps [9]. Short ridges are removed on the basis of the relationship between ridge length and average distance between ridges, instead of considering the average ridge length. Island filtering (already proposed in the literature) is performed in a more efficient way, since it is not based on a graph construction to detect closed paths. Finally, two new topological validation algorithms are presented to classify reliable endpoints and bifurcations: they are removed if topological requirements are not satisfied, they are classified as *less reliable* minutiae if requirements are not fully satisfied, else they are considered as *highly reliable* minutiae. Less reliable minutiae can be taken into account if the number of highly reliable minutiae cannot guarantee the required confidence level of the identification. Simulation results demonstrate the effectiveness of the proposed algorithm and display a sort of implicit segmentation that removes all minutiae related to regions outside the fingerprint.

2. The Proposed Algorithm

The minutiae extraction methodology proposed in this work is suitable for an AFIS like the one sketched in Fig. 1. We assume that the skeleton image has been defined and a local ridge distance map has been derived. The local ridge distance map defines the average ridge distance in each region of the image [11]. This is necessary because the ridge distance varies over the image and it is one of the most important reference parameters in the image both for filter design and minutiae extraction.

The algorithm performs the following steps, according to the sequence detailed in Fig. 1.

2.1. Pixel codification

The codification stage performs minutiae classification and removal of unclassified configurations. The skeleton image is processed in order to obtain a codified image where the value of each pixel represents the number of outgoing branches in the corresponding pixel of the skeleton image [3].

2.2. Pre-filtering

Low-quality areas are often present in the original scanned fingerprint image. Binarizing and thinning algorithms usually produce a large number of spurious high-density minutiae in these sections of the fingerprint with obvious difficulties for the identification algorithms. Prefiltering is necessary to remove as many as possible spurious minutiae without reducing, if possible, the number of details useful for the identification.

A first pre-filtering has already been performed in the codification phase, removing isolated points and blobs. The pre-filtering algorithm scans the codified image in the usual way (row by row from top-left to bottom-right) and acts according to the type of the minutia configuration. At this phase an endpoint next to another minutia is deleted. Adjacent bifurcations are removed and those where one or more crosses are detected in their neighborhood. Also, crosses are removed if another cross is detected in their neighborhood.

2.3. Skeleton enhancement

Skeleton enhancement allows for: ridge repair by connecting endpoints identified as ridge breaks, elimination of bridges, spurs and short ridges. Ridge breaks are caused by insufficient ink, wrinkle and scars. Two facing endpoints can be connected, repairing the ridge, if they are assumed to belong to the same ridge. The ridge repair algorithm works in three steps: search of facing endpoints, best endpoint selection and ridge reconstruction.

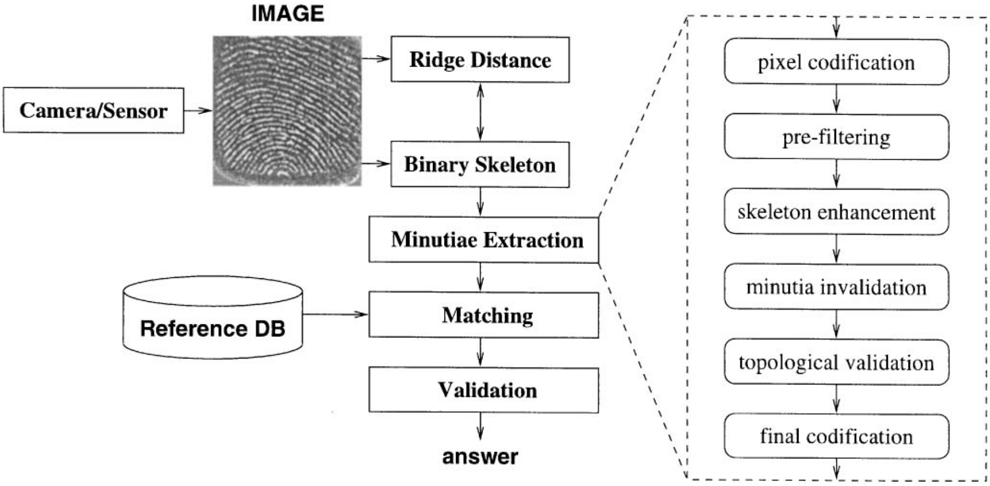

Fig. 1. Generic structure of an automatic fingerprint identification system and detail of the "Minutiae Extraction" block.

2.4. Minutiae invalidation

Minutiae originated by bridges and spurs are invalidated, once these configurations have been recognized. Close minutiae are also removed.

2.4.1. Elimination of bridges and spurs

Bridges and spurs give rise to false minutiae and they must be removed as in the examples shown in Figs 2a and 3a. It is important to apply this algorithm before the close minutiae are searched for, in order to keep as many as possible valid minutiae. A novel method is proposed to detect and clear bridges and spurs. In previous works, bridges and spurs are detected using a local dominant directional map [9]. The novel algorithm, proposed in this work, uses a visual consideration: bifurcations found when these two spurious patterns are detected, are different from real bifurcations. In these "spurious" bifurcations only two branches are aligned while the direction of the third branch is generally different. In bridges, the third branch is almost orthogonal to the other two.

The absolute value of the scalar product between branch unit vectors is used to estimate the alignment or orthogonality of the two branches. In this implementation one orthogonality threshold θ_{bs1} and two alignment thresholds θ_{bs2} and θ_{bs3} have been used. Two directions are considered to be orthogonal if the absolute value of the scalar product is lower than $\theta_{bs1} = 0.55$, i.e. when the angle is greater than 56°. Two directions are considered to be aligned if the absolute value of the scalar product is greater than θ_{bs2} (in step 1e of the algorithm) or θ_{bs3} (in step 2a). $\theta_{bs2} = 0.85$ and $\theta_{bs3} = 0.80$ (corresponding to an angle less than 32° and 37°, respectively).

Algorithm. Elimination of bridges and spurs
For each bifurcation:

(1) if there are at least $N_{lr} = \lambda$ points in each branch, where λ is the local ridge distance in pixels:

(a) estimate the direction of each branch by linear regression using at least N_{lr} points;

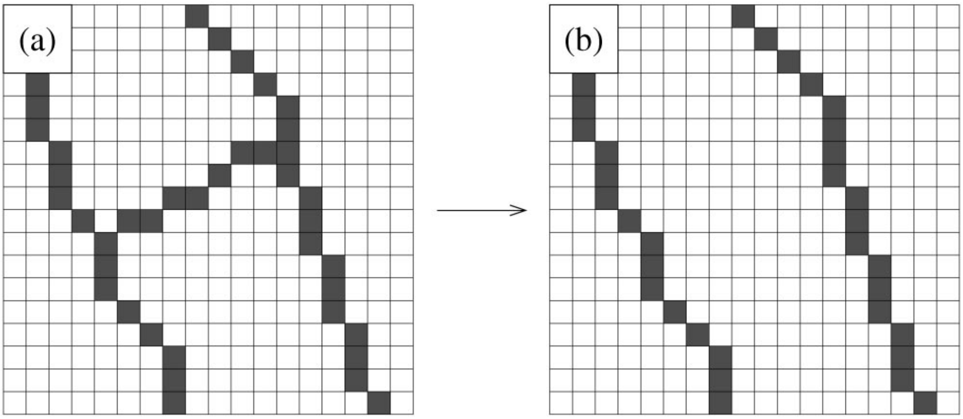

Fig. 2. Bridge and processed slide.

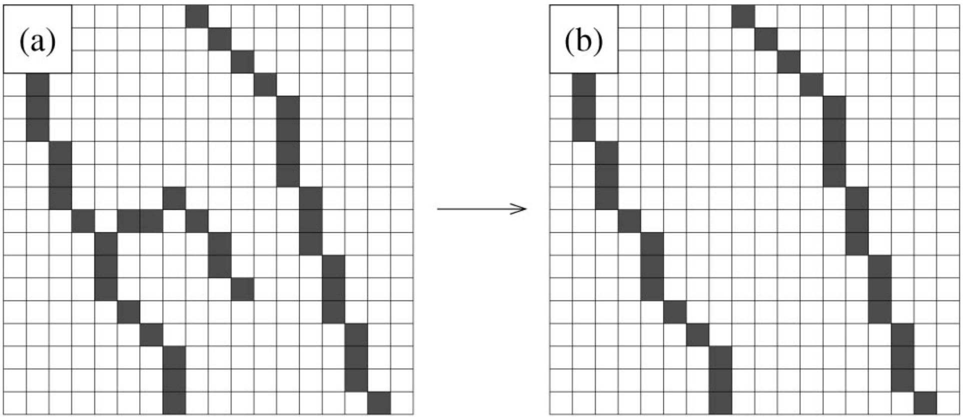

Fig. 3. Spur and processed slide.

(b) if two branches are aligned and the third is almost orthogonal: search for another minutia along the orthogonal branch in the first $N_{bs1} = 5\lambda/6$ points;

(c) if another minutia is found: remove the bridge or spur;

(d) if two branches are aligned, but the third is not orthogonal: search for an endpoint in the first $N_{bs2} = 3\lambda/2$ points;

(e) if the endpoint is found: remove the spur;

(2) if only two branches contain at least N_{lr} points:

(a) if these two branches are aligned: search along the third one and remove this short bridge or spur.

2.4.2. Elimination of close minutiae

The algorithm first removes the close endpoints and then the other close minutiae. Any endpoint found in a small square area is removed. The minimum acceptable distance between two endpoints is given by $N_{ce} = \lambda/2$ pixels, where λ is the local ridge distance. The search region is a square of side $2N_{ce} + 1$ centered on the current endpoint.

There are less than 100 minutiae in a good quality rolled fingerprint image and their relative distance is rarely less than the local ridge distance, therefore too close minutiae on the same branch are removed.

2.5. Topological validation

The topological validation algorithms are based on neighboring ridge topology:

Island removal: Bifurcations belonging to a short closed path are invalidated.

Bifurcation and endpoint validation: Neighboring ridge layout is considered to verify the existence of the correct pattern. These two novel algorithms will introduce reliable and less reliable minutiae codes.

2.5.1. Island elimination

Closed paths are commonly referred to as islands. Even if this particular minutia can occur in a common fingerprint, islands are often generated in the skeleton image by noisy areas and where large irregular ridges are thinned as shown in Fig. 4, giving rise to false minutiae.

The term island is usually employed for two facing bifurcations which form a closed path. It may be noted that this is the most frequent situation only in high-quality images. We also experienced closed paths generated by the interaction of three or more bifurcations and crosses, as the example of Fig. 5. An island can be generated also by a single bifurcation where two branches flow into one another.

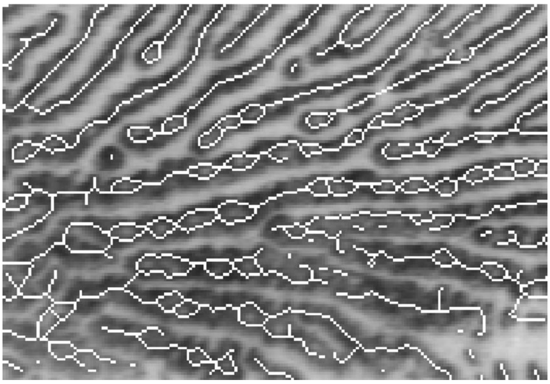

Fig. 4. Islands caused by large ridges.

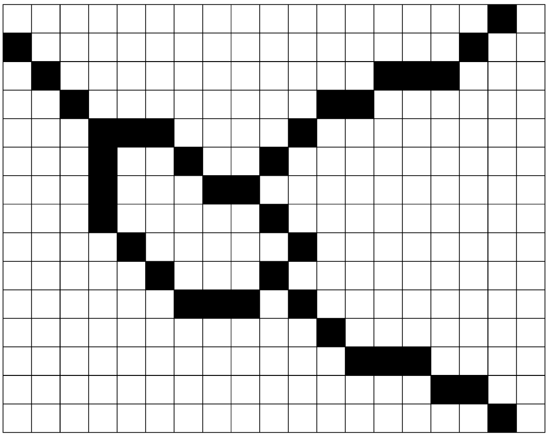

Fig. 5. Three bifurcations belonging to an island.

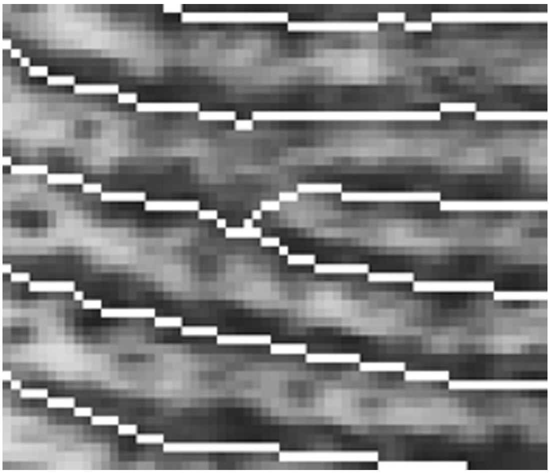

Fig. 6. Common bifurcation topology.

2.5.2. Bifurcation validation

Fig. 6 shows the topology of a typical valid bifurcation. In fact, with the exception of bifurcations located near the *core* or *delta* of the fingerprint, ridges near the bifurcation run parallel to the outgoing branches. Different bifurcation structures can be evidenced in the fingerprint margins or in textured areas. Therefore, the above property is suitable for removing isolated bifurcations in meaningless areas.

The proposed validation algorithm can be divided into two phases: the first is necessary to distinguish between a valid and an invalid bifurcation, the second is required to find a reliable bifurcation. The first phase checks the existence of lateral ridges for each branch departing from the bifurcation (Fig. 7). The second phase checks the geometrical properties around the bifurcation (Fig. 8).

Algorithm. Bifurcation validation
For each valid bifurcation:

(1) compute the three branch directions on $N_{bv1} = 3\lambda/2$ points;
(2) if a minutia is found within $N_{bv2} = \lambda$ points: invalidate this bifurcation and go further;
(3) move along each branch for $N_{bv3} = \lambda/2$ points and verify that a lateral edge is present within $N_{bv4} = 3\lambda/2$ points (see Fig. 7);

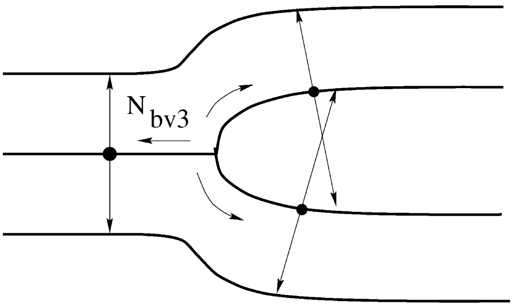

Fig. 7. Bifurcation validation.

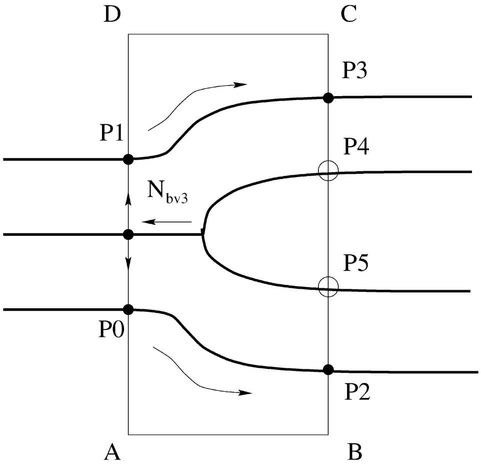

Fig. 8. Reliability determination.

(4) if a lateral ridge is not found: invalidate this bifurcation and go further;
(5) if lateral ridges are found: mark the bifurcation as a *less reliable* bifurcation;
(6) define a rectangular area ABCD (see Fig. 8) with:

$$\overline{AB} = 3\lambda/2,$$

$$\overline{AD} = 4\lambda;$$

(7) move along the lateral ridges (see Fig. 8) from the left intersections with the rectangle P0 and P1 to the right intersections P2 and P3;
(8) if an endpoint is found: invalidate this bifurcation and go further;
(9) if P2 and P3 are reached: mark the bifurcation as *reliable*.

2.5.3. Endpoint validation

The distance between two adjacent ridges remains almost constant where no other minutiae exist in the valley between the two ridges. The same consideration applies to the dual representation of the fingerprint analyzing the valley behavior. This regularity is perturbed by the presence of an endpoint when a ridge is interrupted or by its dual structure, a bifurcation, when a valley is ended.

Fig. 9 shows the typical ridge behavior around an endpoint: neighbor ridges close each other up when the ridge between them is interrupted. It can be noted how the distance between adjacent ridges remains almost constant.

The proposed endpoint validation algorithm is based on this visual consideration and removes most of the

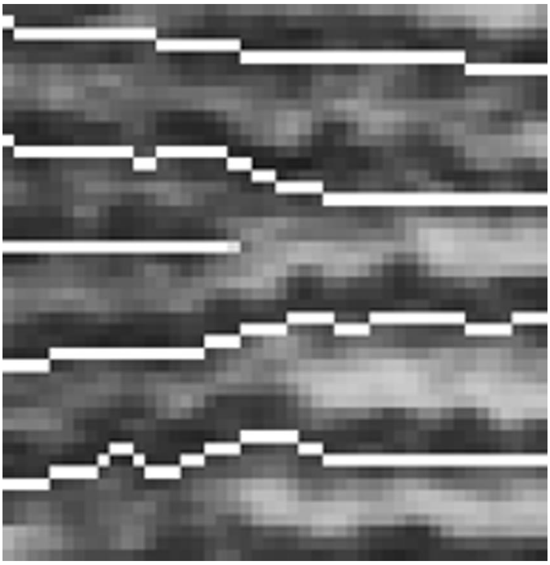

Fig. 9. Common endpoint topology.

spurious endpoints and those originated by fingerprint borders.

Algorithm. Endpoint validation
For each valid endpoint:

(1) evaluate the endpoint direction defined as the direction of the broken ridge over $N_r = \lambda$ points;
(2) if the direction cannot be evaluated for the existence of a minutia within N_r points: invalidate the endpoint and go further;
(3) move along the endpoint ridge for $N_{ev1} = \lambda/2$ points;
(4) search orthogonally to the endpoint ridge direction for the neighboring ridges;
(5) if within $N_{ev2} = 3\lambda/2$ points the ridges are not intercepted: invalidate the endpoint and go further;
(6) define a rectangular area ABCD (see Fig. 10) with:

$$\overline{\mathrm{AB}} = 3\lambda/2,$$

$$\overline{\mathrm{AD}} = 3\lambda,$$

(7) lateral ridges are scanned from P0 and P1 to the exit from the rectangle (points P2 and P3) or a minutia occurrence;
(8) if a lateral ridge crosses the rectangle in DC or AB, while the other crosses the rectangle in BC: mark the endpoint as a *less reliable* endpoint;
(9) if a minutia is found: invalidate the endpoint and go further;
(10) if both lateral ridges cross the rectangle in BC:
(a) if the lateral ridges are not *convergent* (as defined in Section 2.5.4): mark the endpoint as a *less reliable* endpoint else mark the endpoint as a *highly reliable* endpoint;
(b) if a ridge is detected between P2 and P3: invalidate the endpoint.

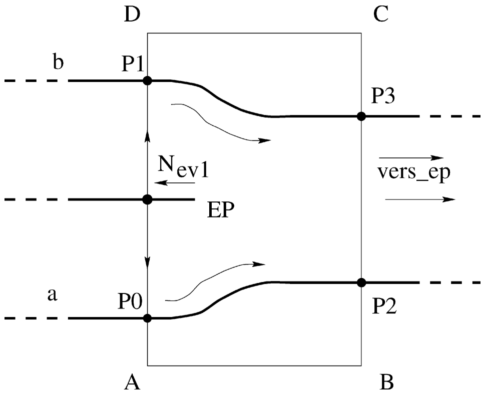

Fig. 10. Rectangular area.

2.5.4. Ridge convergence

The points of lateral ridges contained in the rectangle ABCD are used to estimate their directions. If $\mathbf{u}_1$ and $\mathbf{u}_2$ are the unit vectors of the estimate directions, the degree of convergence is estimated by the vertical component of the vector product:

$$C = (\mathbf{u}_1 \wedge \mathbf{u}_2) \cdot \mathbf{k} \qquad (1)$$

where $\mathbf{k}$ is the unit vector of the axis normal to the image plane. Lateral ridges are considered *convergent* if the coefficient $C \geqslant \theta_{ep} = 0.085$ corresponding to an angle of 4°. If the ridges are almost parallel, we have $C \cong 0$, while $C < 0$ for divergent ridges.

If an endpoint does not verify the convergence condition it is marked as *less reliable* but it is not invalidated since lateral ridges of valid endpoints can be almost parallel and characterized by very slow convergence.

3. Results

The proposed minutiae extraction procedure has been tested on 500 fingerprint images from the *NIST Special Database* 4 [12] (figs_0 data). Fingerprint images are stored as 512 × 512 pi xel 8-bit gray-scale images. The database is organized to guarantee the existence of several different fingerprint topologies and different quality images.

3.1. Statistical results

The performance evaluation of this kind of algorithm is not a well-defined task. However, qualitative considerations can be based on a contextual visual check. The goal of post-processing is to minimize the number of spurious minutiae and to maximize the amount of reliable minutiae to allow reliable fingerprint matching. Statistical analysis has been carried out on three different initial image conditions: (i) raw image; (ii) segmented image: boundary areas without information have been removed, then images have been binarized and

Table 1
Reduction of the number of minutiae (average and standard deviation) after application of the proposed algorithm

Preliminary processing	Initial		Final		Reduction factor
	$\mu_{iniz.}$	$\sigma_{iniz.}$	$\mu_{fin.}$	$\sigma_{fin.}$	
No processing	2933	1114	49(29)	17	98.33%
Segmentation	1845	703	47(29)	17	97.45%
Segmentation and filtering	722	363	70(54)	15	90.30%

Table 2
Minutiae average number and relative reduction

Algorithm	End.	Bif.	Endpoint reduction	Bifurcation reduction	Total reduction
Pre-filtering	379	266			
Ridge breaks	311	266	17.94	0.00	10.54
Bridge and spur	264	124	15.11	53.38	32.76
Short ridge	217	124	17.80	0.00	12.11
Close endpoint	170	124	21.66	0.00	13.78
Close minutiae	141	67	17.06	45.97	29.25
Island	141	55	0.00	17.91	5.77
Validate bif.	141	34(10)	0.00	38.18(56.36)	10.71(15.82)
Validate ep.	36(6)	34(10)	74.47(78.72)	0.00	60.00(67.27)

thinned; (iii) segmented and filtered image: segmented image has also been improved by non-linear enhancement [13].

The minutiae extraction algorithms applied to the three different sets of test images produce the results shown in each row of Table 1, where the average and the standard deviation of the number of minutiae in the original images are shown in columns 2 and 3, respectively, while average and standard deviation estimated after the application of the proposed algorithm are shown in columns 4 and 5. Column 5 also reports the average number of *less reliable minutiae* (between round brackets). The last column reports the percentage of minutiae removed by the algorithm.

Obviously, segmented and filtered images contain a lower initial number of spurious minutiae, because the image quality is higher. Even for this class of images, the number of minutiae is reduced by an order of magnitude. It must also be noted how the standard deviation is reduced. Therefore, the number of valid minutiae is "more uniform" in the post-processed image set, which displays the real minutiae distribution in fingerprints. Better results are obtained for raw and segmented-only images. It should be noted how a lower number of minutiae have been validated in these lower-quality images. This, together with the visual check, highlights an implicit segmentation property of the proposed algorithms. Furthermore, the algorithms eliminate minutiae located in noisy inner fingerprint image regions.

Table 2 gives the statistics related to segmented and filtered images, reporting outcome of application of each step of the proposed algorithm in each row: columns 2 and 3 show the average number of endpoints and bifurcations after application of the algorithm specified in column 1. The number of *less reliable* minutiae is reported in brackets. The last three columns show the endpoint, bifurcation and total reduction factors after application of each algorithm (numbers in brackets show the reduction factors, including *less reliable* minutiae).

Table 3
Total execution time (s)

Preventive action	Execution time Average	Standard deviation
No action	1.706	0.137
Segmentation	1.661	0.121
Segmentation and filtering	1.497	0.087

Table 4
Algorithm execution time (segmented and filtered images)

Algorithm	Execution time	Standard deviation	Normalized exec. time
Codification	0.051	0.009	0.000087
Pre-filtering	0.040	0.009	0.000068
Ridge breaks	0.089	0.029	0.000247
Bridge and spurs	0.342	0.033	0.001702
Short ridges	0.158	0.011	0.000712
Close endpoint	0.147	0.010	0.000800
Close minutiae	0.153	0.010	0.000613
Island	0.160	0.014	0.002692
Bif. valid.	0.178	0.013	0.003624
Endp. valid.	0.179	0.014	0.001403

The algorithm which most reduces the endpoint number is the topological endpoint validation, since it takes into account image boundaries. For the bifurcation number, bridge and spur elimination and close minutia elimination are the most effective. This is due to low-quality image areas where the minutiae density is very high.

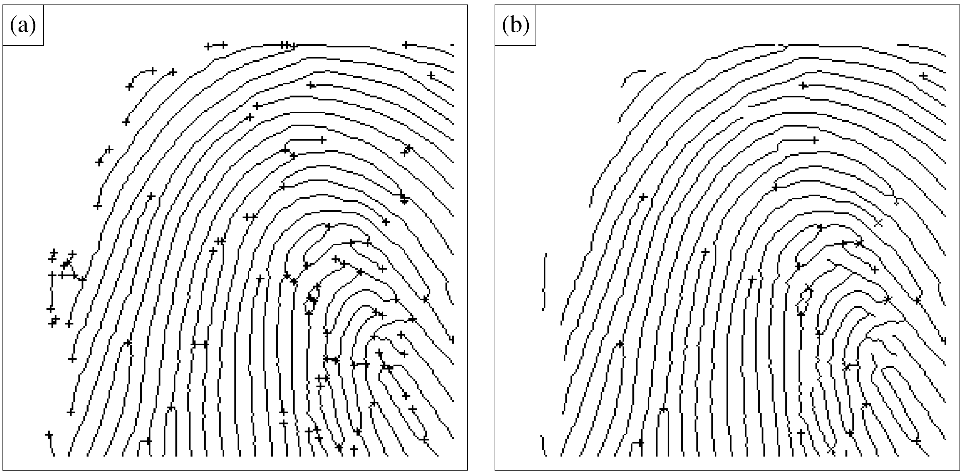

Fig. 11. Skeleton with minutiae points marked: (a) before post-processing; (b) after post-processing.

The statistics of raw images or segmented-only images display a major impact of the algorithms on close minutiae and topological validation. This is due to the higher number of spurious minutiae that are present in the raw or segmented-only images. This result also shows that the proposed algorithm performs an implicit image segmentation, since the final number of minutiae obtained from raw images turns out to be comparable to that obtained from high-quality pre-processed images.

3.2. Algorithm performance

The algorithms have been executed on a *SUN SPARCstation* 20 with total execution time depending on the initial image treatment as shown in Table 3.

For each algorithm the following statistics have been evaluated on the 500 test images: average execution time; standard deviation of the execution time. The execution times of each step of the extraction procedure (relative to segmented and filtered images) are reported in Table 4. The normalized average execution time is also reported in column 4. Normalization is obtained by dividing the execution time by the number of *minutiae* processed by the corresponding algorithm.

The bridge and spur elimination algorithms are the most CPU-expensive but this is due to the order of application of the different algorithms. If normalized times are considered the island elimination and the bifurcation topological validation turn out to be computationally more intensive. The sequence of the different sub-algorithms has been adjusted to optimize the final result. However, since the most CPU-expensive steps are performed when the number of minutiae has already been significantly reduced, execution times appear to be almost equally distributed among the different procedures.

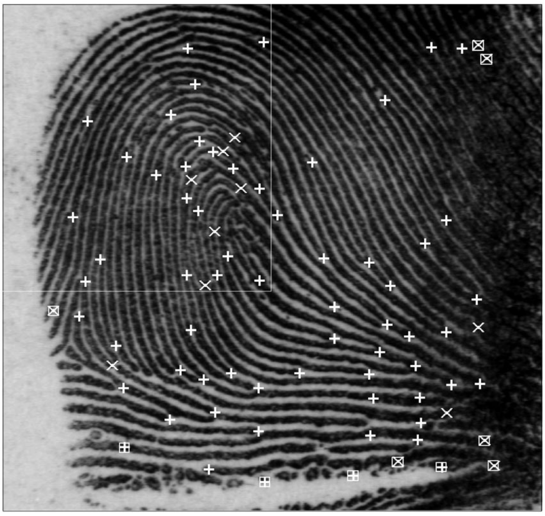

Fig. 12. Extracted minutiae points superimposed on the gray-level image. False minutiae are marked by a square.

3.3. Minutiae extraction algorithms in an AFIS

The effect of the minutiae extraction algorithm set has been investigated in terms of false acceptance and false reject rates. The AFIS where the algorithms are applied has the structure reported in Fig. 1. The last operation, before the answer is generated, is a validation stage where the fingerprint regions are checked between matching minutiae. This final inter-minutiae check is very strict and it is able to discard the matchings where a number of false minutiae generate a valid matching configuration. Thanks to this stage, the false acceptance rate does not increase if the minutiae are not completely filtered. The false reject rate does not increase by adding false points to the correct minutiae. These considerations have been experimentally checked on the 500 image pairs, turning off the single minutiae filtering algorithms. The obvious difference between turning on or off the minutiae extraction algorithms is in the execution time, which increases with the number of additional minutiae considered in the matching phase. For example, without the bifurcation and endpoint validation routines, the average matching time is twice as long as with the application of these routines. Since the execution time is a very important parameter in a real-world application, the relevance of the minutiae extraction algorithms is confirmed.

3.4. Visual considerations

Since there are no reference databases or standard validation techniques for the minutiae extraction stages of AFIS, visual considerations are the only possible means for reporting on the behavior of the presented algorithms. For this reason a visual check has been carried out on a set of 10 images extracted from the figs_0 *NIST Special Database* 4. These images are characterized by an average initial minutia number of 556 and a final one of 86. The goal of the algorithm is to maintain a sufficient number of minutiae to allow an AFIS matching algorithm to operate correctly.

In the following, the results on a specific image (f0084_03.pct in figs_0 *NIST Special Database* 4 directory) are presented.

Fig. 11 shows a region extracted from a skeleton before and after the application of the proposed algorithm. The number of feature points in Figs. 11a and b are 96 and 27, respectively. The graphical symbols " + " and " × " are

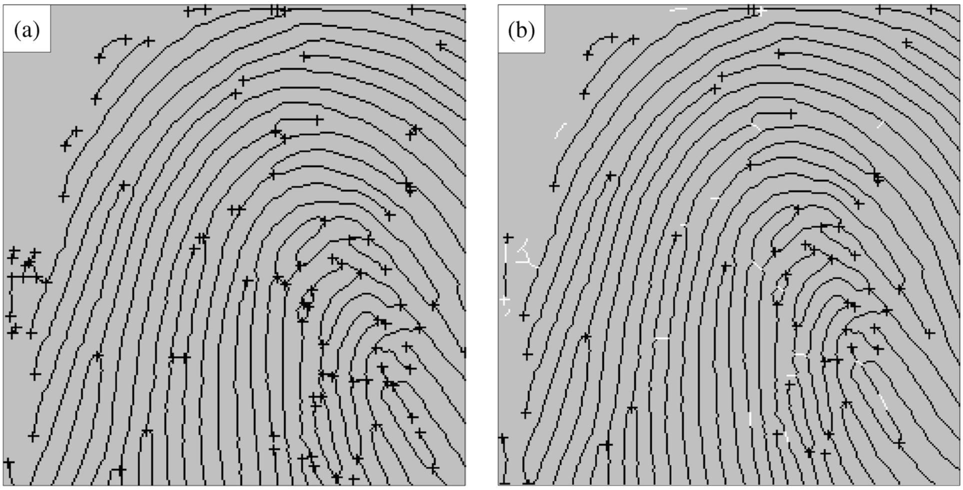

Fig. 13. Skeleton with minutiae points marked: (a) after pre-filtering; (b) after skeleton enhancement.

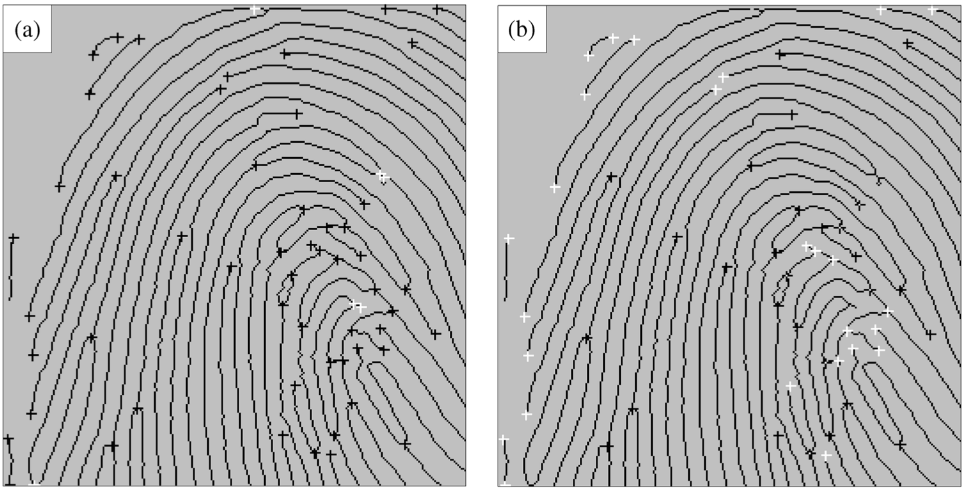

Fig. 14. Skeleton with minutiae points marked: (c) after relative distance elimination; (d) after topological validation.

used to mark *reliable* and *less reliable* minutiae, respectively. Examples of bridge elimination, spur elimination and ridge repair can be found. Boundary minutiae have been completely deleted. In Fig. 12 the minutiae on the input gray-scale image are shown and the extracted region of Fig. 11 is highlighted. Minutiae associated with low-quality regions have been almost completely cancelled. False minutiae are related to low-quality regions and to the skeletoning technique used. False minutiae are marked in Fig. 12 with a square.

Figs. 13 and 14 show the processing step by step. In Fig. 13a valid minutiae after the pre-filtering phase are marked with a black " + ". In Fig. 13b the slide after the skeleton enhancement phase is shown. Skeleton enhancements are shown in white: reconstructed segments, deleted bridges and spurs, erased short ridges. Fig. 14c reports the results of the distance-based elimination: invalidated minutiae are marked in white. The portion of the image under analysis corresponds to a quite regular region of the skeleton. If a "bad" region is considered, this algorithm is very effective, as shown in Fig. 15. Finally, in Fig. 14d the topological validation results are shown. As in previous images invalidated minutiae are marked in white.

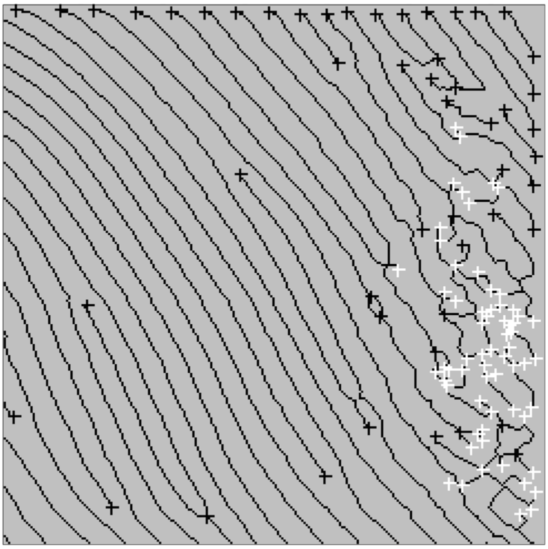

Fig. 15. Low-quality region.

In Figs. 16–18 extracted minutiae points are shown on a gray-level image representing a portion of the original fingerprint. The region is the same marked in Fig. 12a. In each figure the valid features after the phases of pre-filtering (Fig 16a), skeleton enhancement (Fig. 16b), relative distance elimination (Fig. 17c) and topological validation (Fig. 17d) are shown. In Figs. 17d and 18e *highly* and *less reliable* minutiae are marked with the symbols " + " and " × ", respectively. The fingerprint image is the same used in Figs. 13 and 14 (f0084_03.pct). In Table 5 the number of still valid minutiae after the application of each algorithm is reported (the number of *less reliable* minutiae is shown in brackets):

4. Conclusions

Since fingerprint comparison is usually based on minutiae matching, in this paper we presented a set of algorithms suitable for the extraction of minutiae from skeletonized binary images. The problem of minutiae extraction from skeletonized images has been derived from the fact that there are thousands of apparent minutiae, while those of the real fingerprint are less than 100.

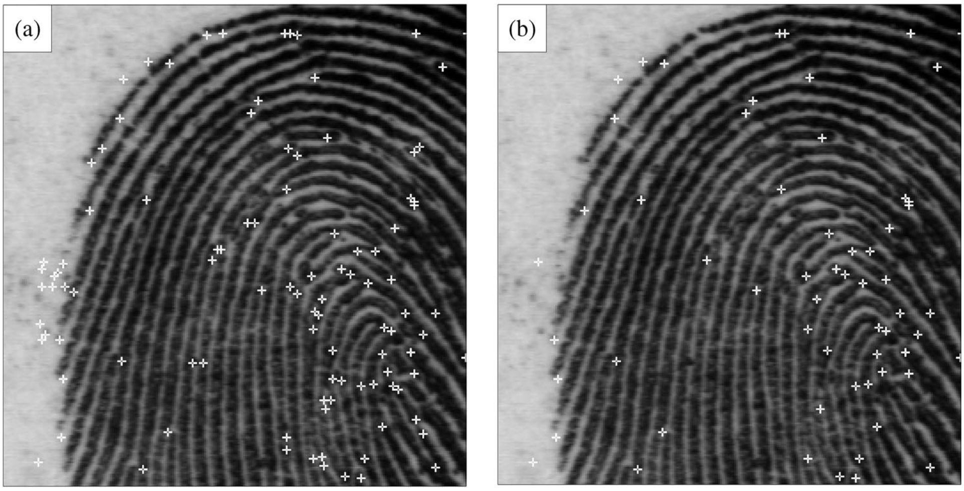

Fig. 16. Extracted minutiae points marked on the gray-level image: (a) after pre-filtering, (b) after skeleton enhancement.

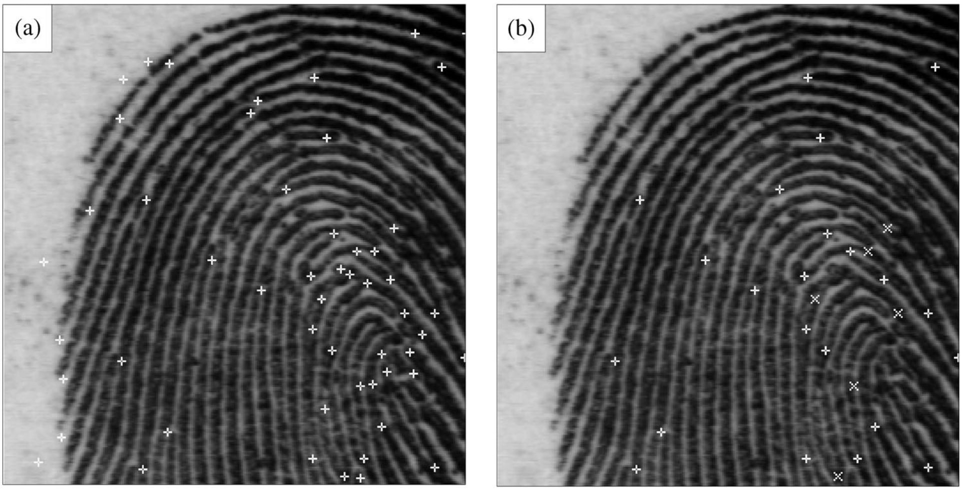

Fig. 17. Extracted minutiae points marked on the gray-level image: (c) after relative distance elimination, (d) after topological validation.

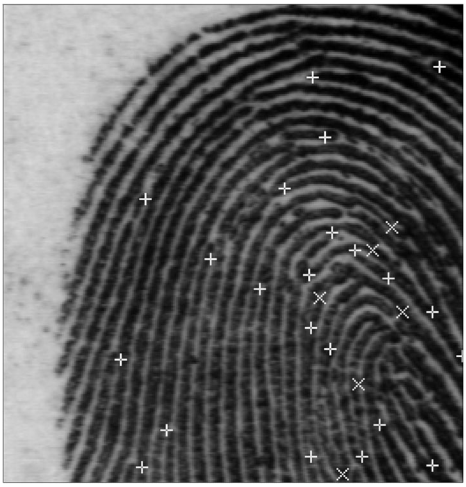

Fig. 18. Extracted minutiae points superimposed on the gray-level image at the end.

Table 5
Number of minutiae after application of each algorithm

Algorithm	Endpoints	Bifurcations
Pre-filtering	291	227
Ridge breaks	265	227
Bridge and spurs	232	119
Short ridges	204	119
Close endpoint	152	119
Close minutiae	129	66
Island	129	55
Bif. valid.	129	39(13)
Endp. valid.	40(2)	39(13)

Besides classical methodologies which remove false minutiae, new approaches were also proposed for bridge filtering based on ridge positions instead of classical methods based on directional maps, which are computationally expensive. Finally, two novel criteria and related algorithms were introduced to validate endpoints and bifurcations. These validation techniques define reliable and less reliable minutiae, which can be used in different security level applications. The statistical analysis of the results obtained by the proposed approach shows the effective reduction of spurious minutiae. A very useful side effect of the proposed minutiae extraction algorithm is that since it performs correctly in dirty areas and on the background too, the application of computationally expensive segmentation algorithms is not necessary. The results are confirmed by visual inspection of images extracted from the NIST sdb 4 reference fingerprint image database.

References

[1] The Science of Fingerprints: Classification and Uses, United States Department of Justice, Federal Bureau of Investigation, Washington, DC, rev. (1988) 12–84.

[2] Fingerprint Identification — Data Format for Information Interchange, American National Standards Institute, New York, 1986.

[3] D. Maio, D. Maltoni, Direct gray-scale minutiae detection in fingerprints, IEEE Trans. PAMI 19 (1997) 27.

[4] J.D. Stosz, L.A. Alyea, Automated system for fingerprint authentication using pores and ridge structure, Proc. SPIE 2277 (1995) 210.

[5] Z. Chen, C.H. Kuo, A topology-based matching algorithm for fingerprint authentication, IEEE 2 (1992) 84.

[6] T.C. Malleswara, Feature extraction for fingerprint classification, Pattern Recognition 8 (1976) 181.

[7] A.P. Fitz, R.J. Green, Fingerprint classification using a hexagonal fast Fourier transform, Pattern Recognition 29 (1996) 1587.

[8] N.K. Ratha, S. Chen, A.K. Jain, Adaptive flow orientation-based feature extraction in fingerprint images, Pattern Recognition 28 (1995) 1657.

[9] D.C. Douglas Hung, Enhancement and feature purification of fingerprint images, Pattern Recognition 26 (1993) 1661.

[10] Q. Xiao, H. Raafat, Fingerprint Image Postprocessing: a combined statistical and structural approach, Pattern Recognition 24 (1991) 985.

[11] Zs.M. Kovács-Vajna, M. Frazzoni, R. Rovatti, Fingerprint ridge distance computation methodologies, Pattern Recognition (1997) submitted.

[12] C.I. Watson, C.L. Wilson, NIST Special Database 4, Fingerprint Database, National Institue of Standard and Technology, 1992.

[13] G.T. Candela, P.J. Grother, C.I. Watson, R.A. Wilkinson, C.L. Wilson, PCASYS — A pattern-level classification automation system for fingerprints, 1995.

About the Author—ALESSANDRO FARINA received the Dr Eng. degree in electrical engineering in 1996 at the University of Bologna, Italy. He is presently serving the Italian Army.

About the Author—ZSOLT M. KOVÁCS-VAJNA received the Dr Eng. degree from the University of Bologna, Italy, in 1988. From 1989 he has been with the Department of Electrical Engineering of the same university where he received the Ph.D. degree in electrical engineering and computer sciences in 1994 for his research on optical character recognition and circuit simulation techniques. He is currently Assistant Professor in Electronics. His research interests include pattern recognition (OCR, ICR, finger-print identification), neural networks and circuit simulation techniques. He is a member of the Institute of Electrical and Electronics Engineers (IEEE), of the International Association for Pattern Recognition (IAPR-IC) and of the International Neural Network Society (INNS).

About the Author—ALBERTO LEONE graduated in Electrical Engineering in 1989 at the University of Bologna, Italy. In the same year he received a Masters degree in Information Technology at CEFRIEL, Milano, Italy. In 1990 he joined the Department of Electrical Engineering of the University of Bologna, working on the "hydrodynamic" model for the transport simulation in submicron devices. There he received the Ph.D. degree in electrical engineering and computer sciences in 1995 for his research on image compression. He is pre sently studying problems related to the modeling of multiterminal devices in smart–power technologies.