PERGAMON

Pattern Recognition 36 (2003) 1847–1857

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Learning fingerprint minutiae location and type[☆]

Salil Prabhakar[a,*], Anil K. Jain[b], Sharath Pankanti[c]

[a] *Digital Persona Inc., 805 Veterans Blvd., Suite 301, Redwood City, CA 94063, USA*
[b] *Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA*
[c] *IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA*

Received 8 March 2002; received in revised form 2 October 2002; accepted 2 October 2002

Abstract

For simplicity of pattern recognition system design, a sequential approach consisting of sensing, feature extraction and classification/matching is conventionally adopted, where each stage transforms its input relatively independently. In practice, the interaction between these modules is limited. Some of the errors in this end-to-end sequential processing can be eliminated, especially for the feature extraction stage, by revisiting the input pattern. We propose a feedforward of the original grayscale image data to a feature (minutiae) verification stage in the context of a minutiae-based fingerprint verification system. This minutiae verification stage is based on reexamining the grayscale profile in a detected minutia's spatial neighborhood in the sensed image. We also show that a feature refinement (minutiae classification) stage that assigns one of two class labels to each detected minutia (ridge ending and ridge bifurcation) can improve the matching accuracy by ~1% and when combined with the proposed minutiae verification stage, the matching accuracy can be improved by ~3.2% on our fingerprint database.

Keywords: Fingerprint matching; Feature extraction; Feedforward; Minutia verification; Minutia classification; Gabor filters; Learning vector quantization

1. Introduction

The human visual system relies on the entire input image data for decision making because of the richness of the image context. Ideally, we would like to design pattern recognition systems that make decisions based on *all* the information available in the input image. However, traditionally, for simplicity of design, a sequential approach to feature extraction and matching is often adopted, where each stage transforms its input information relatively independently and the interaction between these inputs is limited. Often, this rather simplistic model used in each component (stage) is not sufficient to utilize the entire sensed input data. One of the problems with the sequential approach is that the limited use of information in each stage results in feature extraction and matching artifacts. Even though the sequential approach is efficient from design and processing point of view, it may introduce errors in the feature extraction and recognition stages. We believe that by reexamining the original image data, some of the errors in the end-to-end sequential processing can be eliminated, resulting in an improvement in system accuracy. Additionally, by attaching additional discriminative attributes to the features (feature refinement) and designing an appropriate similarity metric that exploits these attributes, the matching accuracy can be further improved. Fig. 1 shows our proposed modifications to a sequential pattern recognition system. We illustrate the above approach in the fingerprint matching domain.

Most of the existing automatic fingerprint verification systems are based on minutiae features (ridge bifurcation and ending; see Fig. 2). Such systems first detect the minutiae

[☆] An earlier version of this paper was presented at the 15th International Conference on Pattern Recognition (ICPR), Barcelona, September 3–8, 2000.

[*] Corresponding author. Tel.: +1-650-261-6070; fax: +1-650-261-6079.

E-mail addresses: salilp@digitalpersona.com (S. Prabhakar), jain@cse.msu.edu (A.K. Jain), sharat@watson.ibm.com (S. Pankanti).

0031-3203/03/$30.00 © 2003 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.
PII: S0031-3203(02)00322-9

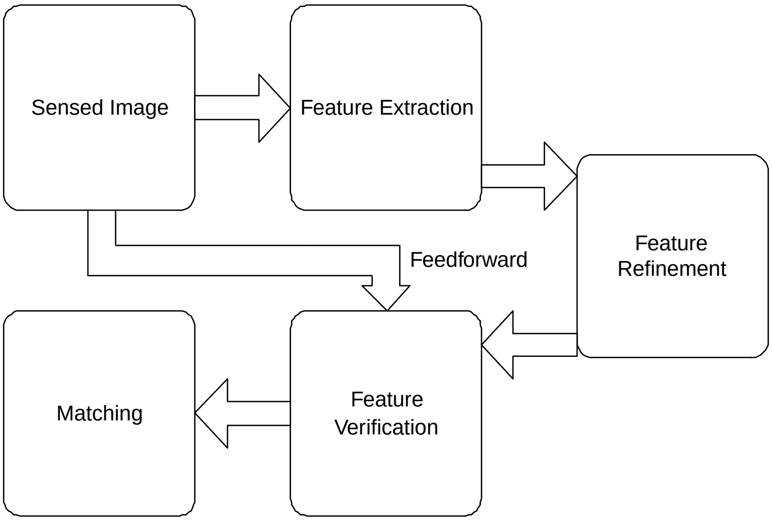

Fig. 1. A general pattern recognition system with proposed feature refinement stage and a feedforward of original image data for feature verification.

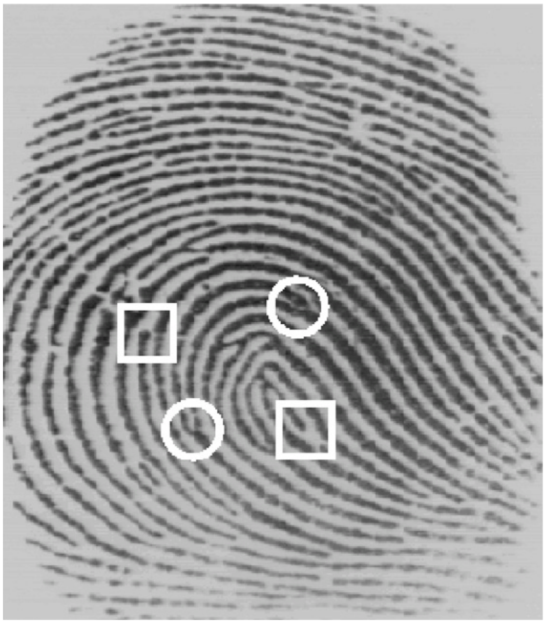

Fig. 2. Examples of fingerprint minutiae: ridge endings (□) and bifurcations (○).

in a fingerprint image and then match the input minutiae set with the stored template [1,2]. Several methods have been proposed for minutiae extraction that involve binarizing and thinning steps [3–5]. An algorithm described in Ref. [1] is a typical example of a sequential approach to feature extraction (see Fig. 3). The feature extraction module first binarizes the ridges in a fingerprint image using masks that are capable of adaptively accentuating the local maximum grayscale values along a direction normal to the local ridge direction. Minutiae are determined as points that have either one neighbor or more than two neighbors in the skeletonized image (see Fig. 4). However, the orientation estimation in a poor quality fingerprint image is extremely unreliable. This leads to errors in precisely locating the fingerprint ridges and results in the detection of many false minutiae (see Fig. 5). A fingerprint enhancement algorithm [6] is often employed prior to minutiae extraction to obtain a more reliable estimate of the ridge locations. Several researchers have also proposed minutia-pruning in the post-processing stage to delete spurious minutiae [1,7–11], but the pruning is based on rather ad hoc techniques. Bhanu et al. [12] proposed a minutiae verification algorithm that used hand-crafted fixed binary templates. A later version of this algorithm by Bhanu and Tan [13] learned the templates from example fingerprint images. However, in both these approaches, the templates were applied to the binarized ridge images for minutiae verification. This is an example of a sequential approach to system design and suffers from the disadvantage that the information lost during earlier stages cannot be recovered. Maio and Maltoni [14] developed a neural network-based minutiae verification algorithm to verify the minutiae detected by their direct grayscale minutiae detection algorithm [2]. This algorithm achieved only a marginal improvement in overall minutiae detection rate (false minutiae rate + missed minutiae rate). The results were based on a very small database (they used 31 fingerprint for training and 31 fingerprints for testing). Moreover, they did not perform a goal-directed test [15] and it is not known how the minutiae verification algorithm affected the overall fingerprint verification system accuracy.

In this paper, we propose a minutiae verification stage that is based on an analysis of the grayscale profile of the

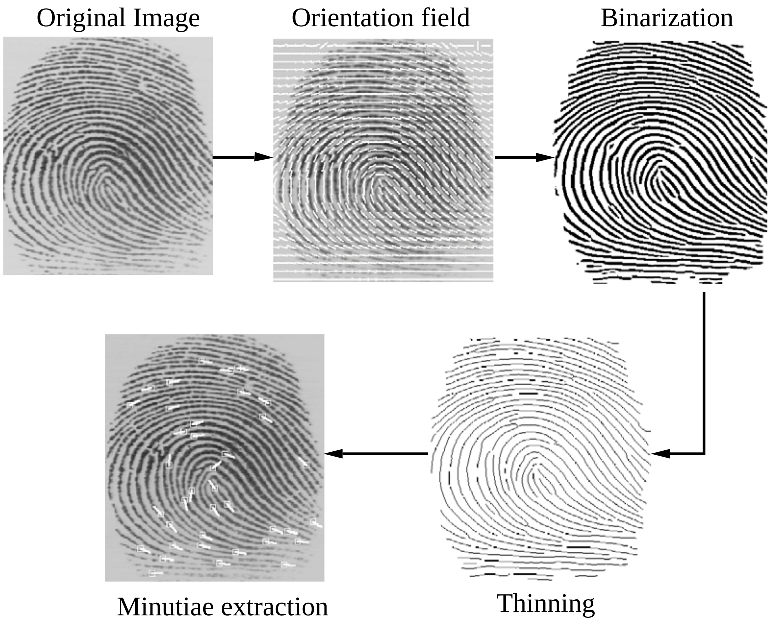

Fig. 3. Various stages in a typical minutiae extraction algorithm [1].

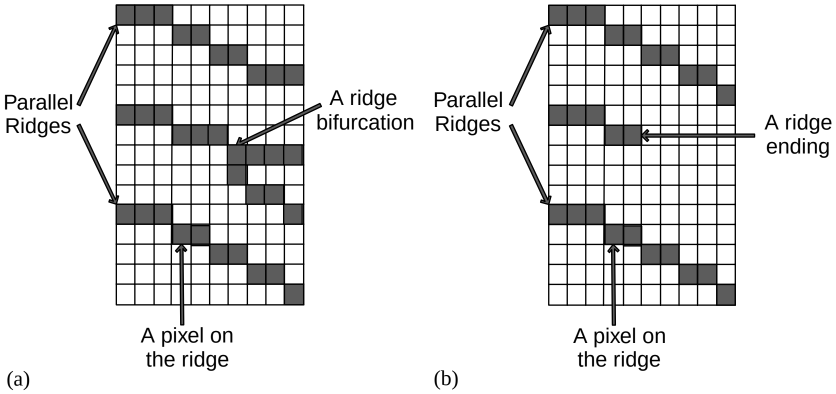

Fig. 4. Examples of a ridge bifurcation and a ridge ending in a thinned fingerprint image. In (a) and (b), all the pixels that reside on the ridge have two 8-connected neighbors. In (a), the pixel with three neighbors is a ridge bifurcation and in (b), the pixel with only one neighbor is a ridge ending.

original input image in the neighborhood of potential minutiae. The minutiae verification stage first learns the characteristics of minutiae in grayscale images, which is then used to verify each detected minutia. This stage will replace the rather ad hoc minutia-pruning stage used in Ref. [1]. The minutiae are extracted using the algorithm described in Ref. [1] (see, Fig. 3). Each detected minutia goes through this verification stage and is either accepted or rejected based on the learned grayscale characteristics in the neighborhood of minutiae. Our minutiae verification stage is based on supervised learning using learning vector quantization (LVQ) [16]. We chose to use LVQ for our verification problem due to its fast learning speed and good accuracy. We also propose a minutiae classification stage where the minutiae are classified into two major classes: ridge bifurcation and ending. We show that both the minutiae verification and minutiae minutia classification improve the fingerprint matching accuracy.

In Section 2 we describes the details of the proposed minutiae verification scheme. Section 3 describes the minutiae classification. Section 4 presents the experimental results and section 5 presents discussion and future directions.

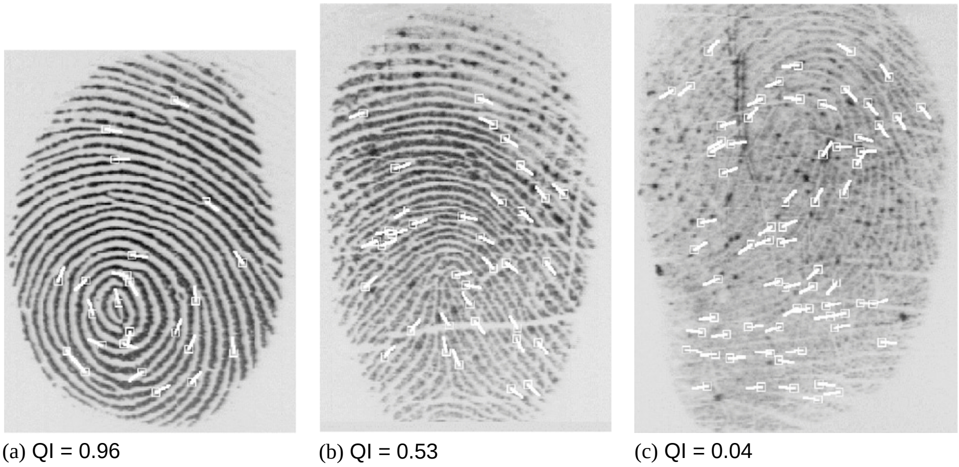

Fig. 5. Sample images from our database with varying quality index (QI). No false minutiae were detected in (a), 7 in (b), and 27 in (c) by the automatic minutiae detection algorithm [1].

2. Minutiae verification

In this section, we first explain the feature extraction process for minutiae and nonminutiae training examples, and then describe the design of the LVQ minutiae verifier (training and validation) using these features.

2.1. Feature extraction

A potential minutia has the following three attributes: the $\{x, y\}$ position and the direction of the ridge on which it resides (θ). Our goal is to build a verifier that takes this $\{x, y, \theta\}$ information and makes a YES/NO decision, as to whether a minutia is present at this location and orientation by analyzing the grayscale image neighborhood. Given $\{x, y, \theta\}$ and a 500 dpi fingerprint image, we first extract a 64×64 region centered at the x and y position oriented in the direction of θ. The grayscale intensities in this 64×64 region are normalized to a constant mean and variance to remove the effects of sensor noise and grayscale variations due to finger pressure differences. Let $I(x, y)$ denote the grayscale value at pixel (x, y), M and V, the estimated mean and variance of grayscale values in this 64×64 window, respectively, and $N(x, y)$, the normalized grayscale value at pixel (x, y). For all the pixels in the window, the normalized image is defined as

$$N(x,y) = \begin{cases} M_0 + \sqrt{\frac{V_0 \times (I(x,y)-M)^2)}{V}} & \text{if } I(x,y) > M, \\ M_0 - \sqrt{\frac{V_0 \times (I(x,y)-M)^2)}{V}} & \text{otherwise,} \end{cases} \quad (1)$$

where M_0 and V_0 are the desired mean and variance values, respectively. Normalization is a pixel-wise operation and does not change the clarity of the ridge and valley structures. For our experiments, we set the values of both M_0 and V_0 to 100. The values of M_0 and V_0 should be the same across all the training and test sets.

After the normalization, we enhance the contrast of the ridges by filtering this 64×64 normalized window with an appropriately tuned Gabor filter [18]. An even symmetric Gabor filter has the following general form in the spatial domain:

$$G(x, y; f, \theta) = \exp\left\{\frac{-1}{2}\left[\frac{x'^2}{\delta_x^2} + \frac{y'^2}{\delta_y^2}\right]\right\}\cos(2\pi f x'), \quad (2)$$

$$x' = x\sin\theta + y\cos\theta, \quad (3)$$

$$y' = x\cos\theta - y\sin\theta, \quad (4)$$

where f is the frequency of the sinusoidal plane wave along the direction θ from the x-axis, and δ_x and δ_y are the space constants of the Gaussian envelope along x and y axes, respectively. We set the frequency f of the Gabor filter to the average ridge frequency $(1/K)$, where K is the average inter-ridge distance. The average inter-ridge distance is approximately 10 pixels in a 500 dpi fingerprint image. The values of parameters δ_x and δ_y for Gabor filters were empirically determined and each is set to 4.0 (about half the average inter-ridge distance). Since the extracted region is in the direction of the potential minutia, the filter is tuned to 0° direction. See Fig. 6 for the 0°-oriented Gabor filter used here. We perform the filtering in the spatial domain with a mask size of 33×33. The filter values smaller than 0.05 are ignored and the symmetry of the filter is exploited to speed up the convolution. We extract a 32×32 region from the center of the 64×64 filtered region to avoid boundary problems of convolution. Each pixel in this 32×32 region is scaled to eight grayscales and the rows are concatenated to form a 1024-dimensional feature vector. See Fig. 7 for an illustration of the intermediate stages in the feature extraction process.

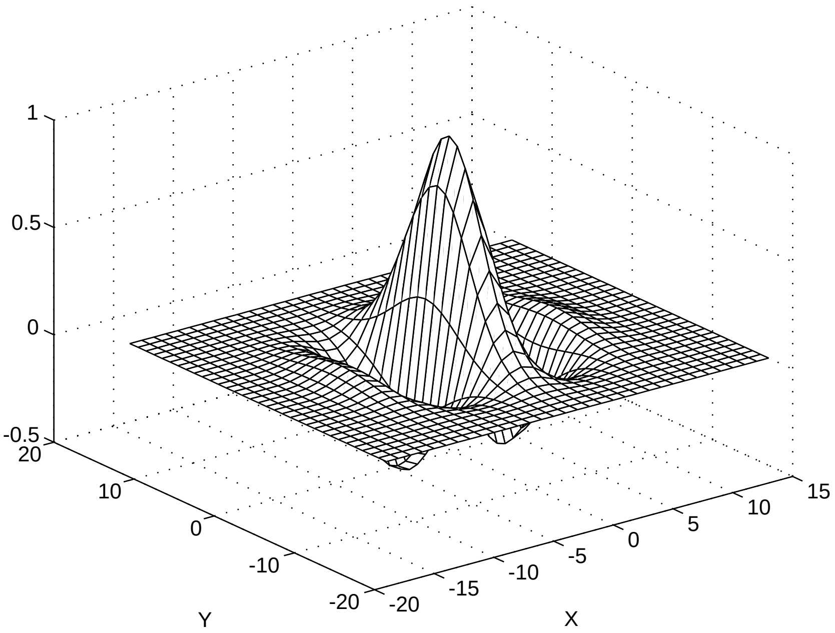

Fig. 6. Gabor filter (orientation = 0°, mask size = 33 × 33, $f = 0.1$, $\delta_x = 4.0$, $\delta_y = 4.0$).

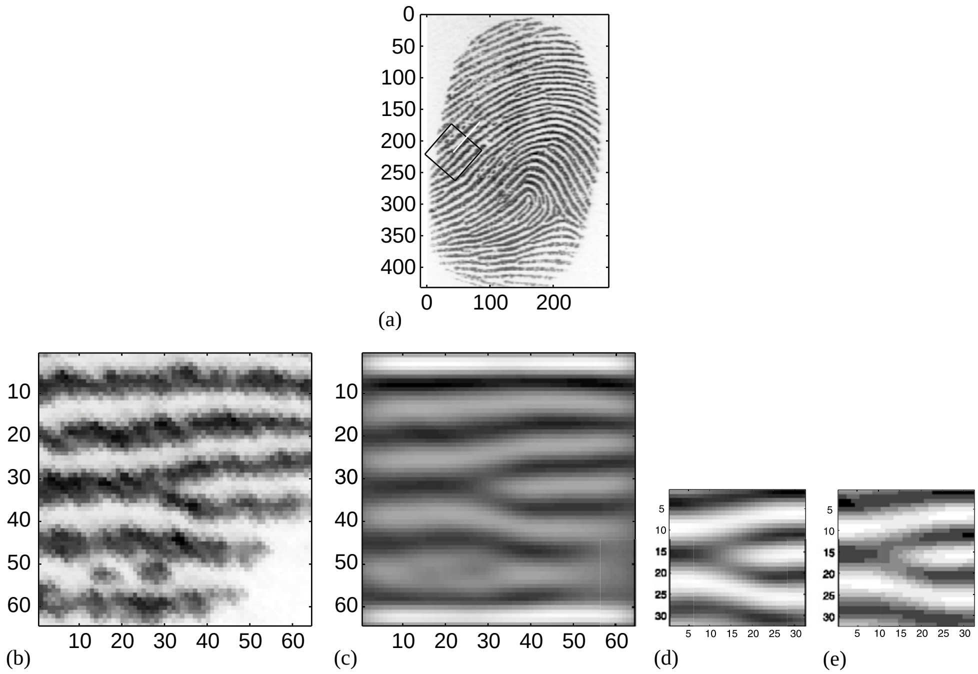

Fig. 7. Stages in feature extraction for minutiae verification. A true minutia location and the associated direction is marked in (a); the 64 × 64 area centered at the minutia location and oriented along the minutia direction is also shown. In (b), the grayscale values in the 64 × 64 neighborhood are shown. The output of 0°-oriented Gabor filter applied to (b) is shown in (c); note the problems at the boundary due to convolution. The central 32 × 32 region is extracted and shown in (d). (e) shows the same 32 × 32 region as in (d) but the grayscale range has been scaled to integers between 0 and 7.

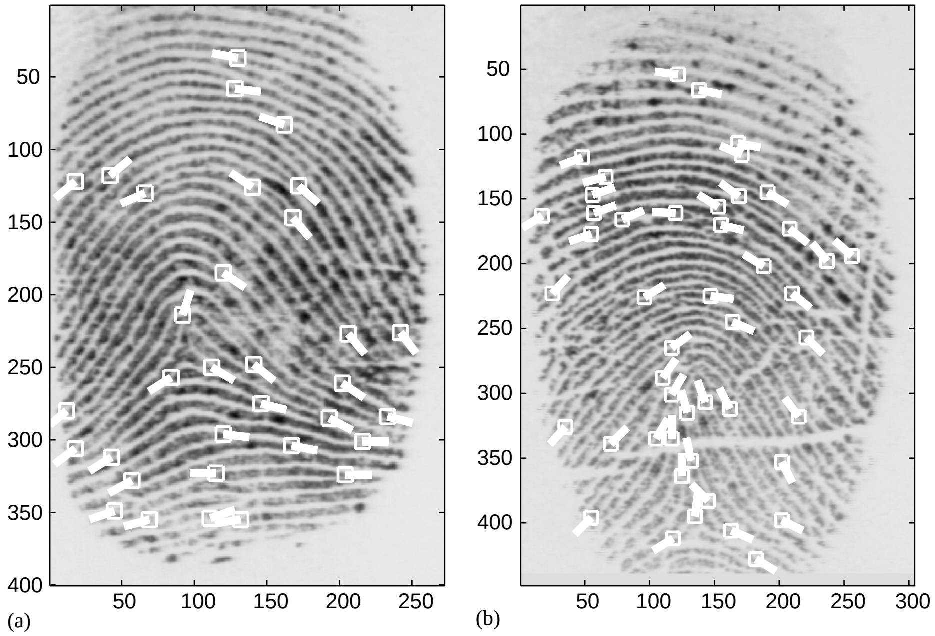

Fig. 8. Two examples of images in the GT database. The ground truth minutiae provided by an expert are marked on the image.

2.2. *Verifier design*

In the training phase, feature vectors are extracted from the ground-truth minutiae and nonminutiae regions and fed to a LVQ to learn the characteristics of minutiae and nonminutiae regions. We use a database (called GT) that contains 900 fingerprint images from 269 different fingers. A fingerprint expert has marked the "true" fingerprint minutiae locations as well as orientations (but not the minutiae types) in the fingerprint images in this database (see Fig. 8).This database contains multiple impressions for each finger that were taken at different times. All the images have been scanned at 500 dpi resolution with 256 grayscales. We chose this database because other fingerprint databases available to us did not have the associated minutiae ground truth marked in them. We use the first 450 fingerprint images in the database (GT-training set) for training the LVQ classifier and the remaining 450 fingerprint images from different fingers (GT-validation set) for validation. The distribution of the quality of fingerprints in this database is shown in Fig. 9. The quality index was determined using an automatic fingerprint quality checker algorithm. This algorithm checks for the grayscale variance and consistency of orientation in a fingerprint image to determine good quality areas. The quality score is the ratio of the good quality area to the total foreground area in the fingerprint image [19]. It will be shown later that the proposed minutiae verification is more effective for medium to poor quality fingerprint images.

We extract approximately 15,000 feature vectors (each feature vector has 1024 components) corresponding to all the true minutiae from the images in the GT-training set. We also extract an equal number of negative samples (nonminutiae) by randomly sampling the images in the training set and making sure that there is no minutiae in its immediate 32 × 32 neighborhood. For the true minutiae, we use the direction of the minutiae provided by the expert. For the negative examples, we compute the direction of the 32 × 32 block using the hierarchical orientation-field algorithm [1]. See Fig. 10 for examples of extracted minutiae and nonminutiae features.

We validate our LVQ-based minutiae verifier on the independent GT-validation set. In the LVQ method, each class (e.g., minutiae and nonminutiae) is described by a relatively small number of codebook vectors, usually more than one per class. These codebook vectors are placed in the feature space such that the decision boundaries are approximated by the nearest neighbor rule [16]. The advantage of using the LVQ method over nearest neighbor method is that the number of codebook vectors in typically much smaller than the number of training examples; thus LVQ is much faster and requires less storage space. The best verification accuracy of ∼95% on the GT-training set and ∼87% on the GT-validation set were achieved with one hundred codebook vectors for each of the minutiae and nonminutiae classes. Note that the results on the GT-validation are much worse than the GT-test set. This is expected because the resubstitution errors (errors on

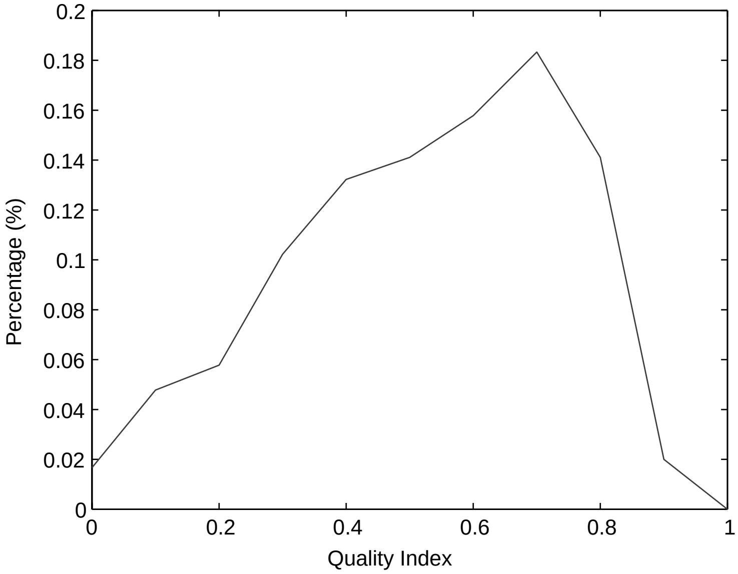

Fig. 9. Distribution of quality of fingerprints in the GT database.

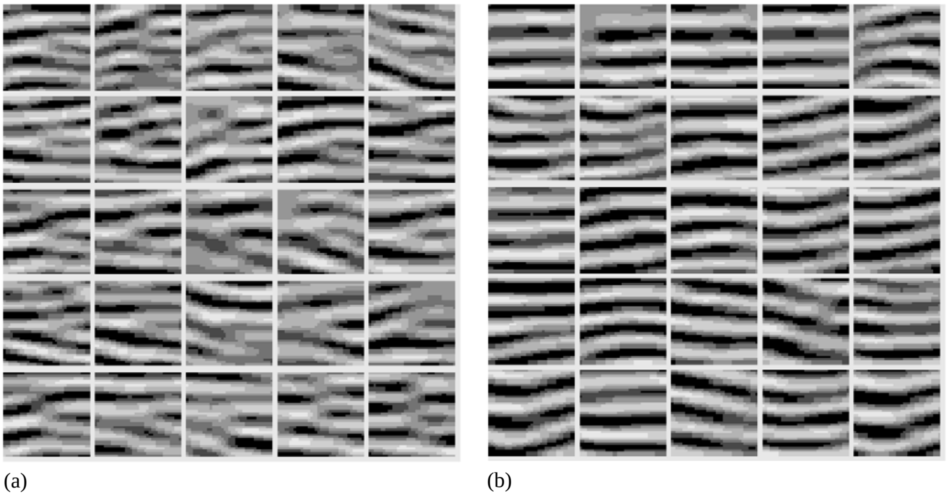

Fig. 10. Examples of grayscale profiles in the neighborhood of (a) minutiae and (b) nonminutiae. These 32 × 32 subimages that are scaled to 8 grayscales, are used for training a LVQ classifier.

the training set) are positively biased due to "overfitting" [20].

3. Minutiae classification

The American National Standards Institute (ANSI) proposes four classes of minutiae: ending, bifurcation, trifurcation, and undetermined. The most discriminable categories are ridge ending and bifurcation (see Fig. 2). Several of the fingerprint matching algorithms reported in the literature do not use minutia type information because of the difficulty in designing a robust classifier to identify minutiae type. We use a rule-based minutia classification scheme and show that the resulting classification of minutiae can indeed improve the overall matching accuracy. In minutiae extraction algorithm, if a pixel in the thinned image has more than two neighbors, then the minutia is classified as a bifurcation, and if a pixel has only one neighbor, then the minutia is classified as an ending (see Fig. 4). The matching algorithm in Ref. [1] is modified to match minutiae endings only with minutiae endings and minutiae bifurcations only with

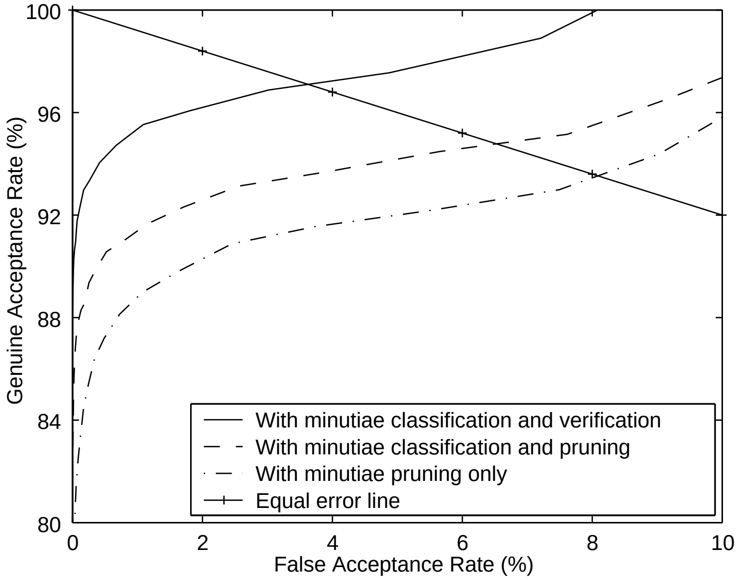

Fig. 11. ROC for fingerprint matching when both minutiae classification and verification are used.

minutiae bifurcations. Note that this classification of minutiae does not affect any other part of the feature extraction stage. In our experience, there are significantly more number of minutiae endings present in a typical fingerprint than bifurcations (according to a study conducted by Stoney and Thronton [17], the probability of occurrence of a ridge ending is greater than the probability of occurrence of a ridge bifurcation).

4. Experimental results

We test the effectiveness of the methods proposed in this paper on the accuracy of a fingerprint verification system on a database. We chose the GT-validation set described in the previous section as our test set but do not use the ground-truth minutiae information available with the fingerprint images in the GT database. We will call this database GT-test set; it contains the same fingerprint images as the GT-validation sets without the ground truth. The accuracies of the fingerprint verification system on the GT-test set corresponding to the methods proposed here are reported by plotting receiver operating characteristics (ROC) curves. In Fig. 11, the ROC curve shown in dash-dotted line shows the accuracy of the state-of-the-art minutiae-based fingerprint verification system described in [1] without any modifications.

A goal-directed test [15] of the minutia verification stage would measure the benefits of replacing the minutia-pruning stage in Ref. [1] with the proposed minutia verification stage by computing the change in matching accuracy of the fingerprint verification system. So, we first remove the minutiae-pruning stage from Ref. [1]. Then, the minutiae are extracted from the fingerprint images in the GT-test set for verification. Since an automatically detected minutia may be slightly perturbed from its true location because of the noise introduced during the binarizing and thinning processes, we extract twenty five 32 × 32 windows (overlapping in steps of 4-pixels in x and y) in the neighborhood of each detected minutia. A 1024-dimensional feature vector is extracted from each window by the feature extraction algorithm described in Section 2.1 and verified using the minutiae verifier described in Section 2.2. The decisions from the verification of these 25 windows are combined in a simple manner. If the verifier determines that a minutia is present in any of the 25 windows, the minutia is accepted. Figs. 12(a), (c), and (d) illustrate minutiae detection by the extraction algorithm in Ref. [1] without pruning, results of the proposed minutia verification, and results of minutiae pruning [1], respectively, for a medium quality fingerprint image. Note that minutia verification is more effective for medium to poor quality fingerprint images.

In Fig. 11, the solid line shows the improvement in accuracy when the minutiae-pruning stage in Ref. [1] is replaced with the proposed minutiae verification scheme and the proposed minutiae classifier (into ridge and ending) is used. Thus, the techniques proposed in this paper improve the overall fingerprint verification system accuracy by ∼3.2% at the equal error rate (point on the ROC where the false accept rate is equal to the false reject rate).

To illustrate the contributions of the minutiae verification (Section 2) and the minutiae classification (Section 3) separately, we additionally plot the ROC curve when the minutiae classification is used alone. This ROC is shown

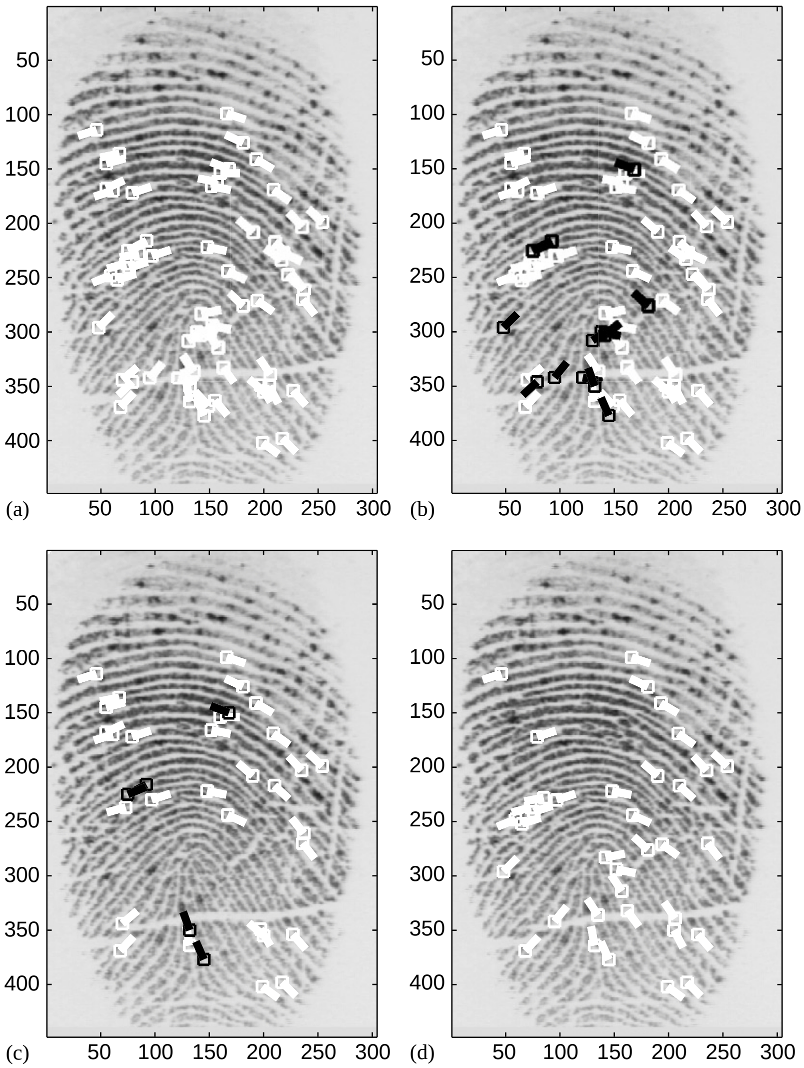

Fig. 12. Minutiae detection and classification: (a) minutiae detection using the algorithm in Ref. [1] without pruning; (b) result of classifying minutiae, minutia bifurcations are marked with black and endings are marked with white; (c) result of minutiae verification; (d) the results of minutiae pruning and no minutiae classification are shown for a comparison. Note that visually, the results of minutiae verification proposed in this paper are better than the rather ad-hoc minutiae pruning used in Ref. [1].

in dashed line in Fig. 11 and demonstrates that the improvement in fingerprint verification accuracy when using the minutia classification alone is ∼1% at the equal error rate. Additionally, see Fig. 12(b) for an illustration of the minutiae classification results on an example of a fingerprint image of medium quality.

5. Discussions and future work

We have shown that the fingerprint verification system accuracy can be improved by a feedforward of the original image grayscale data to a feature verification stage that verifies each minutia detected by the feature extraction algorithm by an analysis of the grayscale profile of its spatial neighborhood in the original image. We have also shown that the accuracy of a minutiae-based fingerprint verification system can be improved if the features (minutiae) are refined and augmented with more discriminable attributes (minutia type information) before matching and the matching is modified to take advantage of this additional information.

The minutiae verification approach suffers from the problem of missed minutiae, i.e., the true minutiae in the fingerprint image that are missed by the feature extraction algorithm cannot be recovered by the minutiae verification algorithm. Minutiae verification algorithm can only reject the falsely detected minutiae. Therefore, the minutiae detection algorithm should be operated at a very low false rejection rate so that it misses very few potential minutiae. We accomplished this by removing the post-processing stage from the feature extraction algorithm in Ref. [1]. However, there are still many missed minutiae in the fingerprint images that cannot be recovered.

The current minutiae verification algorithm is applied only on the minutiae already extracted by the algorithm in Ref. [1] from the thinned binarized fingerprint ridges. The use of minutiae verification algorithm presented here can be extended to detect the minutiae directly in the grayscale fingerprint image [14]. However, the current implementation of the minutiae verification algorithm cannot be used for the minutiae detection problem due to its poor accuracy. For example, consider a 320×320 fingerprint image scanned at 500 dpi resolution. Our minutiae verification algorithm places a 32×32 region around each minutiae and cannot tolerate more than 8-pixel displacement in the minutiae location. Therefore, at least 1600 $\left(\frac{32\times32}{8\times8} \times \frac{320\times320}{32\times32}\right)$ candidate minutiae locations in the fingerprint image will need to be sampled. First of all, this will be computationally expensive. Secondly, with the current 87% accuracy of our minutiae verification algorithm, there will be 208 errors made by the minutiae detection algorithm in the image. In a 320×320 fingerprint image scanned at 500 dpi resolution, there are typically 30–40 minutiae and 208 errors cannot be gracefully handled by the matching algorithm. Therefore, techniques to improve the accuracy of the minutiae verification algorithm should be explored. At the same time, an intelligent scheme to apply this algorithm for minutiae detection at only selected locations instead of the whole image should also be explored.

In our training of the minutiae verification algorithm, the minutiae examples are representative of the total pattern variation in the minutiae types. However, the nonminutiae examples selected from random locations in the fingerprint images are not representative of all the nonminutiae patterns. A more representative nonminutiae training set or a better method of using the training patterns for a more effective training should be explored to improve the accuracy of the minutiae verification algorithm.

In a fingerprint image, core(s)/delta(s) are points of global singularity in parallel fingerprint ridges and can be used to align two fingerprints for matching. An alignment algorithm that is based on these singularities suffers from errors in reliably locating these points [18]. The design of a core/delta point learning and verification algorithm similar to the minutiae learning and verification algorithm described in this paper will help such an alignment scheme. The current limitation in developing such an algorithm is the unavailability of large ground truth database of cores and deltas.

We believe that a continuous classification of the minutiae into several categories (one of the categories being nonminutiae) can also be achieved. Such a classifier would perform the minutiae verification and classification in a single step as opposed to the two-step process used in this paper. A classification label and a confidence value assigned to each minutia and a modified matching algorithm that takes these confidence values into account can improve the fingerprint verification system accuracy.

Acknowledgements

We would like to thank Dr. Aditya Vailaya of Agilent Technologies for his help with the Learning Vector Quantization software.

References

[1] A.K. Jain, L. Hong, S. Pankanti, R. Bolle, An identity authentication system using fingerprints, Proc. IEEE 85 (9) (1997) 1365–1388.

[2] D. Maio, D. Maltoni, Direct gray-scale minutiae detection in fingerprints, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1) (1997) 27–40.

[3] M.R. Verma, A.K. Majumdar, B. Chatterjee, Edge detection in fingerprints, Pattern Recognition 20 (5) (1987) 513–523.

[4] B. Moayer, K. Fu, A tree system approach for fingerprint pattern recognition, IEEE Trans. Pattern Anal. Mach. Intell. 8 (3) (1986) 376–388.

[5] L. O'Gorman, J.V. Nickerson, An approach to fingerprint filter design, Pattern Recognition 22 (1) (1989) 29–38.

[6] L. Hong, Y. Wan, A.K. Jain, Fingerprint image enhancement: algorithm and performance evaluation, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 777–789.

[7] N. Ratha, S. Chen, A.K. Jain, Adaptive flow orientation-based feature extraction in fingerprint images, Pattern Recognition 28 (11) (1995) 1657–1672.

[8] D.C.D. Hung, Enhancement and feature purification of fingerprint images, Pattern Recognition 26 (11) (1993) 1661–1671.

[9] Q. Xiao, H. Raafat, Fingerprint image postprocessing: a combined statistical and structural approach, Pattern Recognition 24 (10) (1991) 985–992.

[10] A. Farina, Z.M. Kovacs-Vajna, A. Leone, Fingerprint minutiae extraction from skeletonized binary images, Pattern Recognition 32 (5) (1999) 877–889.

[11] X.P. Luo, J. Tian, Knowledge based fingerprint image enhancement, Proceedings of the International Conference on Pattern Recognition (ICPR), Vol. 4, Barcelona, Spain, September 2000, pp. 783–786.

[12] B. Bhanu, M. Boshra, X. Tan, Logical templates for feature extraction in fingerprint images, Proceedings of the International Conference on Pattern Recognition (ICPR), Vol. III, Barcelona, Spain, September 2000, pp. 850–854.

[13] B. Bhanu, X. Tan, Learned template for feature extraction in fingerprint images, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. II, Hawaii, USA, December 2001, pp. 591–596.

[14] D. Maio, D. Maltoni, Neural network based minutiae filtering in fingerprints, Proceedings of the 14th International

Conference on Pattern Recognition, Brisbane, Australia, August 1998, pp. 1654–1658.

[15] O. Trier, A.K. Jain, Goal-directed evaluation of binarization methods, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 1191–1201.

[16] T. Kohonen, J. Kangas, J. Laaksonen, K. Torkkola, LVQ_PAK: a program package for the correct application of learning vector quantization algorithms, Proceedings of the International Joint Conference on Neural Networks, Baltimore, USA, June 1992, pp. 1725–1730.

[17] D.A. Stoney, Distribution of epidermal ridge minutiae, Am. J. Phys. Anthropol. 77 (1988) 367–376.

[18] A.K. Jain, S. Prabhakar, L. Hong, S. Pankanti, Filterbank-based fingerprint matching, IEEE Trans. Image Process. 9 (5) (2000) 846–859.

[19] R. Bolle, S. Pankanti, Y.-S. Yao, System and method for determining quality of fingerprint images, US Patent No. US5963656, 1999.

[20] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley, New York, USA, 2001.

About the Author—SALIL PRABHAKAR was born in Pilani, Rajasthan, India, in 1974. He received his BTech degree in Computer Science and Engineering from Institute of Technology, Banaras Hindu University, Varanasi, India, in 1996. During 1996–1997 he worked with IBM, Bangalore, India, as a software engineer. He received his Ph.D. degree in Computer Science and Engineering from Michigan State University, East Lansing, MI 48824, in 2001. He currently leads the Algorithms Research Group at DigitalPersona Inc., Redwood City, CA 94063 where he works on 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 20 technical publications and has two patents pending. He has also coauthored the book *Handbook of Fingerprint Recognition* (Springer 2003).

About the Author—ANIL K. JAIN is a University Distinguished Professor in the Department of Computer Science and Engineering at Michigan State University. He was the Department Chair between 1995–1999. He has made significant contributions and published a large number of papers on the following topics: statistical pattern recognition, exploratory pattern analysis, neural networks, Markov random fields, texture analysis, interpretation of range images, 3D object recognition, document image analysis, and biometric authentication. Among others, he has coedited the book *Biometrics*: *Personal Identification in Networked Society* (Kluwer 1999) and coauthored the book *Handbook of Fingerprint Recognition* (Springer 2003). Several of his papers have been reprinted in edited volumes on image processing and pattern recognition. He received the best paper awards in 1987 and 1991, and received certificates for outstanding contributions in 1976, 1979, 1992, 1997 and 1998 from the Pattern Recognition Society. He also received the 1996 IEEE Transactions on Neural Networks Outstanding Paper Award. He is a fellow of the IEEE and International Association of Pattern Recognition (IAPR). He received a Fulbright Research Award in 1998 and a Guggenheim fellowship in 2001.

About the Author—SHARATH PANKANTI is with the Exploratory Computer Vision and Intelligent Robotics Group, IBM T.J. Watson Research Center, Yorktown Heights, NY. From 1995 to 1999, he worked on the Advanced Identification Solutions Project dealing with reliable and scalable fingerprint identification systems for civilian applications. For the past few years he has been working on analysis and interpretation of video depicting human activities. His research interests include biometrics, pattern recognition, computer vision, and human perception.