

Fingerprint Matching Incorporating Ridge Features With Minutiae

Heeseung Choi, Kyoungtaek Choi, and Jaihie Kim

Abstract—This paper introduces a novel fingerprint matching algorithm using both ridge features and the conventional minutiae feature to increase the recognition performance against nonlinear deformation in fingerprints. The proposed ridge features are composed of four elements: ridge count, ridge length, ridge curvature direction, and ridge type. These ridge features have some advantages in that they can represent the topology information in entire ridge patterns existing between two minutiae and are not changed by nonlinear deformation of the finger. For extracting ridge features, we also define the ridge-based coordinate system in a skeletonized image. With the proposed ridge features and conventional minutiae features (minutiae type, orientation, and position), we propose a novel matching scheme using a breadth-first search to detect the matched minutiae pairs incrementally. Following that, the maximum score is computed and used as the final matching score of two fingerprints. Experiments were conducted for the FVC2002 and FVC2004 databases to compare the proposed method with the conventional minutiae-based method. The proposed method achieved higher matching scores. Thus, we conclude that the proposed ridge feature gives additional information for fingerprint matching with little increment in template size and can be used in conjunction with existing minutiae features to increase the accuracy and robustness of fingerprint recognition systems.

Index Terms—Breadth first search, ridge count, ridge features, ridge-based coordinate system.

I. INTRODUCTION

FINGERPRINT recognition has been widely adopted for user identification due to its reliable performance, usability, and low cost compared with other biometrics such as signature, iris, face, and gait recognition [1]. It is used in a wide range of forensic and commercial applications, e.g., criminal investigation, e-commerce, and electronic personal ID cards. Although significant improvement in fingerprint recognition has been achieved, many challenging tasks still remain. Among them, nonlinear distortions, presented in touch-based fingerprint sensing, make fingerprint matching more difficult. As shown in Fig. 1, even though these two fingerprint images are from the same individual, the relative positions of the minutiae

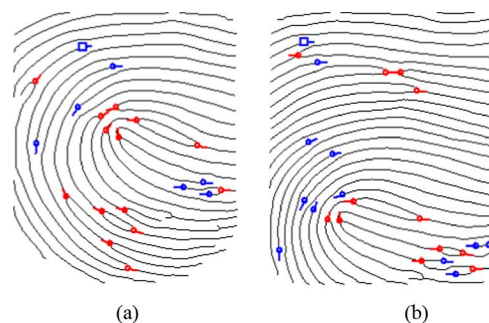


Fig. 1. Example of skin distortions.

are very different due to skin distortions. This distortion is an inevitable problem since it is usually associated with several parameters [2], [3], including skin elasticity, nonuniform pressure applied by the subject, different finger placement with the sensor, etc.

To deal with the distortions in fingerprint images and improve the matching performance, various methods have been proposed by many researchers. These can be roughly classified into several groups: modeling the distortion of fingerprints [2], [4], [5]; detecting the distortions using special hardware or video sequences [6], [7]; allowing some amount of distortion in the minutiae matching stages [8], [9]; and using local similarity measures [10], [11]. Cappelli *et al.* [5] proposed a plastic distortion model of a fingerprint to calculate the nonlinear deformation of fingerprints. They first defined three distinct regions in fingerprint images and described the distortion mechanism using some distortion parameters. This method has successfully been applied to the generation of synthetic fingerprints of the same finger [12]. However, it is hard to estimate the parameters accurately due to insufficient information. Bazen *et al.* [4] and Ross *et al.* [2] used the thin plate spline (TPS) model to compensate for deformations, but this sort of alignment process typically requires too much computational power to be used in practical fingerprint recognition systems. Ratha *et al.* [6] proposed a method to directly measure the forces and torques on the scanner using special hardware and Dorai *et al.* [7] estimated the distortions by observing fingerprint video sequences. These two methods prevent the capture of severely distorted images, but they cannot detect the distortion in data sets that were collected in the past and the need of additional hardware restricts their use in practical situations. Luo *et al.* [8] used changeable tolerance boxes in the minutiae matching process. The size of the tolerance boxes is incrementally increased, moving from the center towards the borders of the fingerprint area, to deal with the effect of distortion. Lee *et al.* [9] applied distance normalization and local alignment during minutiae matching. In local minutiae matching, these approaches can be considered

Manuscript received July 15, 2010; revised December 10, 2010; accepted December 18, 2010. Date of publication January 06, 2011; date of current version May 18, 2011. This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Biometrics Engineering Research Center at Yonsei University R112002105070010(2010). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Arun Ross.

H. Choi and J. Kim are with the Biometric Engineering Research Center, Yonsei University, Seoul 120-749, Korea (e-mail: mcnas@yonsei.ac.kr; jhkim@yonsei.ac.kr).

K. Choi is with the LIG Nex1 Company, Yong In, Gyeonggi, 148-1, Korea (e-mail: kyoungtaekchoi@lignex1.com).

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

Digital Object Identifier 10.1109/TIFS.2010.2103940

as an effective tool. However, as the size of the tolerance boxes had to be increased, the probability of falsely matching fingerprints from different fingers also increases. Some methods used local similarity measures to improve the robustness of the distortions since fingerprint images are less affected by distortions in the local area. Jiang *et al.* [10] proposed a method using the similarity measure defined between local structural features to align fingerprint images. Kovacs–Vajna [11] proposed a method using triangular matching by checking the correspondence of gray-scale profiles between every pair of minutiae from the template and the corresponding positions in the input images. And the dynamic time warping method was used to tolerate small location errors of features. However, for widespread use of these approaches, a consolidation step may be implemented to check whether the local similarity holds at the global level [1].

Moreover, although many fingerprint matching methods have been developed to cope with distortions, most of them are minutiae-based. Thus, they cannot use more topological information (such as ridge shape) covering the entire fingerprint image and the limitation of information still exists. In addition, these methods use complex data structures and many parameters for fingerprint matching. Accordingly, it is hard to understand and implement these methods accurately. Considering the facts mentioned above, instead of developing complex distortion models or elaborate minutiae alignment algorithms, we propose a new and simple matching scheme by incorporating conventional minutiae features and additional ridge features associated with corresponding minutiae sets. To extract the ridge features, a ridge-based coordinate system is also defined. The ridge features consist of four elements: ridge count (rc), ridge length (rl), ridge curvature direction (rcd), and ridge type (rt). These features are invariant to any geometric transformations (rotation, translation) of the fingerprints and concisely represent the relationships between the minutiae since the maintenance of ridge structures is robust to distortions. Moreover, since the correlation between the proposed ridge features and conventional minutiae features is low, combining these features leads to an improvement in the overall recognition performance with a small increment in template size. Our ridge features require only 5 bytes (ridge count—1 byte; ridge length—2 bytes; ridge curvature direction—1 byte; and ridge type—1 byte) for each minutiae pair.

This paper is organized as follows. In Section II, we introduce and analyze the proposed ridge features extracted from the ridge-based coordinate system. In Section III, we introduce a fingerprint matching algorithm incorporating conventional minutiae and the proposed ridge features. Experimental results are shown in Section IV and Section V offers conclusions and suggestions for further research.

II. FINGERPRINT PREPROCESSING AND RIDGE FEATURE EXTRACTION

A. Fingerprint Preprocessing

Before extracting the proposed ridge features, we need to perform some preprocessing steps (see Fig. 2). These steps include typical feature extraction procedures as well as additional procedures for quality estimation and circular variance estimation. We first divide the image into 8×8 pixel blocks. Then, the mean and variance values of each block are calculated to segment the

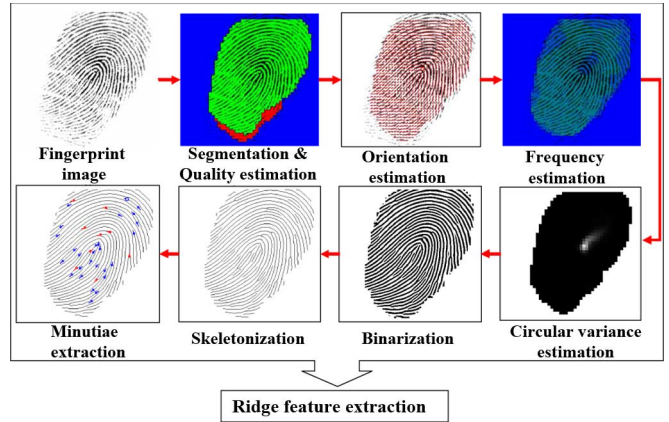


Fig. 2. Overall preprocessing steps.

fingerprint regions in the image. We then apply the method described in [13] to estimate the ridge orientation and the ridge frequency is calculated using the method presented in [14]. The Gabor filter [15] is applied to enhance the image and obtain a skeletonized ridge image. Then, the minutiae (end points and bifurcations) are detected in the skeletonized image. The quality estimation procedure [16] is performed in order to avoid extracting false minutiae from poor quality regions and to enhance the confidence level of the extracted minutiae set. Furthermore, in regions where ridge flows change rapidly, such as the area around a singular point, it is hard to estimate the ridge orientations accurately or to extract the thinned ridge patterns consistently. Therefore, to detect regions which have large curvature, we apply circular variance estimation [17]. The circular variance of the ridge flows in a given block is calculated as follows:

$$\text{Var}(\theta) = 1 - \frac{1}{n} \left[\left(\sum_{i=1}^n \cos \theta_i \right)^2 + \left(\sum_{i=1}^n \sin \theta_i \right)^2 \right] \quad (1)$$

where θ_i and n represent the estimated orientation of the i th block and the number of neighboring blocks around the i th block, respectively. In our experiments, we use eight neighboring blocks. Quality estimation and circular variance values are used to avoid generating feature vectors in poor quality regions or in regions around singular points. Moreover, we adopt some postprocessing steps [18] to remove falsely extracted ridges, such as short ridges and bridges. We can then extract the ridge structures consistently against various noise sources.

B. Ridge Feature Extraction

1) *Proposed Ridge-Based Coordinate System*: After performing the preprocessing steps, we obtain the skeletonized ridges and minutiae information from the fingerprint image. We can then define ridge coordinates and extract ridge features between two minutiae. As shown in Fig. 3, each ridge-based coordinate system is defined by a minutia O (called origin) and vertical and horizontal axes starting from the origin minutia. First, the vertical axis is defined by drawing a line passing through the origin and orthogonal to the orientation of the origin. The axis also traverses the ridge flows orthogonally. In addition, to define the sign of the vertical axis according to the origin, the cross product between the orientation of the

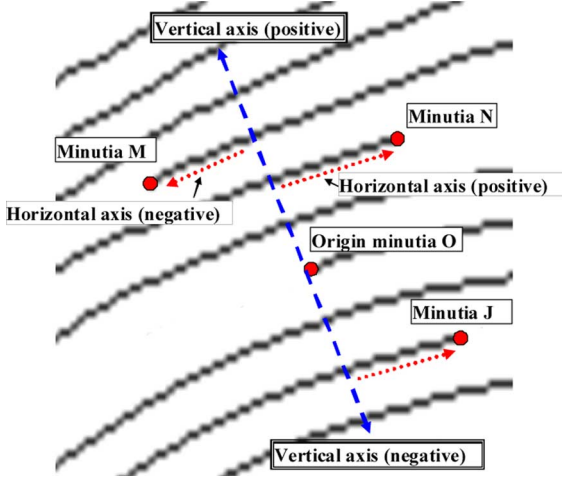


Fig. 3. Ridge-based coordinate system.

origin and the vector pointing from the origin to the side of the vertical axis is calculated as follows:

$$V_s = \text{sign} \left(\vec{O} \times \vec{V}_n \right) \quad (2)$$

where V_s , \vec{O} , and \vec{V}_n represent the sign of the vertical axis, the minutia orientation vector, and the unit vector of the vertical axis, respectively. Thus, we determine the positive and the negative side of the vertical axis by checking the sign value of V_s .

To represent the relative position of the minutiae (minutiae M , N , and J in Fig. 3) according to the origin, horizontal axes should be defined. The horizontal axes are defined as ridges intersecting the vertical axis. To define the sign of each horizontal axis, the cross product between the vectors pointing from the intersection to the vertical and horizontal axes is calculated as follows:

$$H_s = \text{sign} \left(\vec{H}_n \times \vec{V}_n \right) \quad (3)$$

where H_s , \vec{H}_n , and \vec{V}_n represent the sign of the horizontal axis, the vector pointing from the intersection to the horizontal and the vertical axis, respectively. In the ridge-based coordinate system, the ridge features that describe the relationship between the origin (minutia O in Fig. 3) and an arbitrary minutia (minutiae M , N , and J in Fig. 3), are described as follows:

$$\vec{V} = (rc, rl, rcd, rt) \quad (4)$$

where rc , rl , rcd , and rt represent the ridge count, ridge length, ridge curvature direction, and ridge type, respectively. These four components form a ridge-based feature vector between two minutiae and this feature vector is used in the matching process. In the following sections, we will explain in detail these ridge features were selected and the methods for extracting these features.

2) *Ridge Feature Extraction*: In the general ridge count methods [10], [11], the number of ridges that intersect the straight line between two minutiae in the spatial domain is counted. However, when the ridge-counting line is parallel to the ridge structures, the line may meet the same ridge at

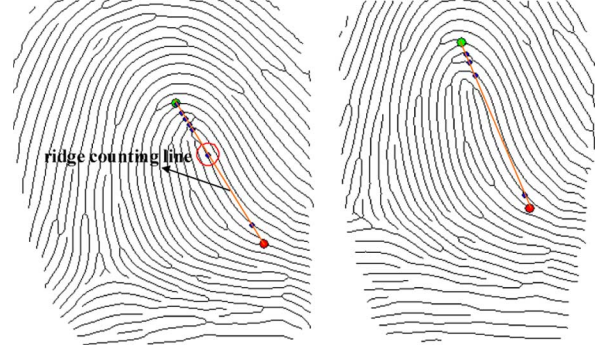


Fig. 4. Example of ridge-counting errors using the general ridge counting methods. Even though the two images are from the same fingerprints, the ridge count numbers between the two corresponding minutiae are different due to skin deformation.

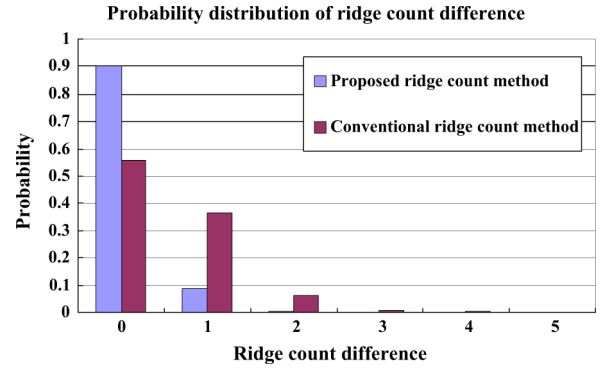


Fig. 5. Comparison of the ridge counting methods.

one point, at more than two points, or at no point, due to skin deformation (see Fig. 4).

Therefore, unlike existing ridge-counting methods, here, the ridge count (rc) is calculated by counting the number of ridges along the vertical axis until the axis meets the ridge attached to the neighboring minutia. The vertical axis is perpendicular to the ridge structures. Thus, the counted numbers are less affected by skin deformation than in the results of the general ridge counting methods. In order to prove the effectiveness of the proposed ridge counting method, we used 50 fingers from FVC 2002 DB1-A [22] and manually paired the corresponding minutiae among the five images from each finger. After pairing two corresponding minutiae, we estimated the probability distributions of the absolute difference of the ridge counting numbers in each method. Fig. 5 shows that the absolute difference of the ridge counting numbers using our method is smaller than that using the conventional ridge count method. Therefore, we can conclude that our ridge count feature is more robust to skin deformation. Furthermore, to increase the discriminating power of the ridge count (rc) feature, we also consider the direction of the ridge count line. The ridge count (rc) is not always a positive number and the sign of the ridge count follows the sign of the vertical axis. If two minutiae are directly connected by the same ridge, the ridge count would be zero.

The ridge length (rl) is the distance on the horizontal axis from the intersection of the vertical and horizontal axis to a minutia. To prove the usefulness of the ridge length feature, we conducted an experiment similar to the analysis of the ridge

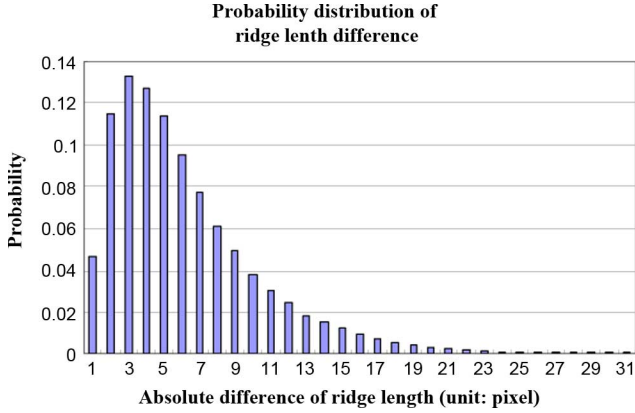


Fig. 6. Probability distribution of the absolute difference of ridge length.

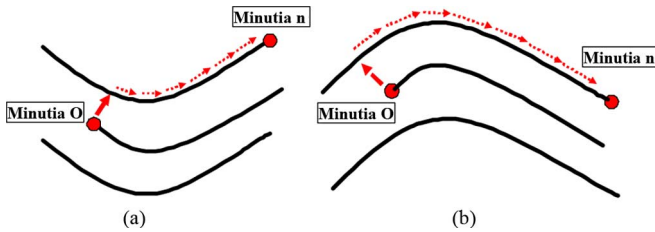


Fig. 7. Ridge curvature direction. (a) Concave shape. (b) Convex shape.

count feature. Fig. 6 shows the probability distribution of the absolute difference of the ridge length feature. As shown in the figure, the absolute differences of ridge length elements are mostly less than 16 pixels. Therefore, we can set the threshold of the ridge length feature to determine the same fingerprint as 16 pixels. The ridge length value also has a sign and follows the sign of the related horizontal axis to improve the discriminating power.

To use more topology information in ridge patterns for matching, the ridge curvature direction is also considered. As shown in Fig. 7, even though the ridge count and ridge length values are very similar, the shapes of the ridge patterns may be different [concave shape—Fig. 7(a); convex shape—Fig. 7(b)].

The ridge curvature direction is defined as follows:

$$\text{rcd} = \text{sign} \left(\sum_{i=1}^N \vec{v}_i \times \vec{v}_{i-1} \right) \quad (5)$$

where v_i represents the i th vector between the sampling points along the horizontal axis from the intersection of the vertical and horizontal axes to the minutia n (see Fig. 7) and N represents the number of sampling points. In our experiment, we set the sampling point every 8 pixels on the ridges. Then, by checking the sign of this value, we can determine the ridge curvature direction. The ridge curvature direction feature is robust to skin deformation but some errors may still occur. First, ridges may have more than two inflection points, which makes it hard to define this feature. Second, some ridges are too straight to define a curved direction. Therefore, to avoid the error caused by more than two inflection points, we empirically limit the maximum length of ridges to 80 pixels (10 sampling points \times 8 pixels =

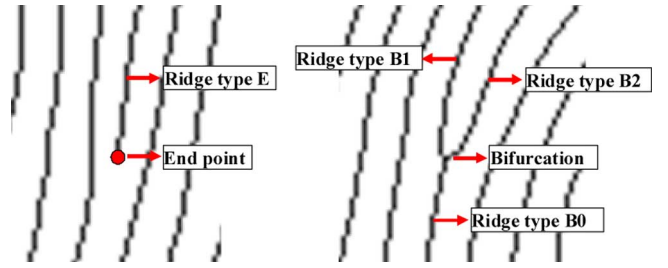


Fig. 8. Examples of ridge types.

80 pixels). Additionally, to avoid the error caused by a straight ridge, we defined the ridge curvature direction as 0.

Due to the feature extraction error, skin condition changes, and different finger pressures, end points may appear as bifurcations and vice versa. Therefore, considering these facts and to further improve the discriminating power of ridge features, the ridge type (rt) is used as one of the ridge features instead of a minutia type. To determine the ridge type (rt), each minutia is first classified as an end point or a bifurcation. If a minutia is an end point, there is only one ridge belonging to the minutia. If a minutia is a bifurcation, there are three ridges connected to the minutiae. Next, the type of ridge associated with the minutia is determined as one of four types according to the type of the minutia and the relative position of the ridges. As shown in Fig. 8, if a minutia is an end point, the ridge type is defined as E . In a bifurcation case, the three ridges are labeled by checking the angle between each ridge and the minutia orientation. A triangle is created by three points on the ridges (equidistant from the bifurcation). If the vertex of the triangle is not on the shortest side of the triangle, then the ridge belongs to the vertex and is defined as type $B0$. The other two ridges are classified as type $B1$ and $B2$, moving in a clockwise direction from $B0$. Generally speaking, ridge type E can change only into ridge type $B1$ or $B2$. However, type E cannot be converted into type $B0$. Therefore, we use this information in the fingerprint matching.

The overall procedure for extracting ridge features is as follows:

- 1) Perform preprocessing steps and extract a ridge image from a fingerprint.
- 2) Traverse the ridge-valley structures along the vertical axis from each minutia origin.
 - a) If the vertical axis intersects with the ridges attached to a minutia, extract ridge features (ridge count, ridge length, ridge curvature direction, and ridge type) from the origin to the minutia and form a ridge feature vector between the origin and the minutiae.
 - b) Keep traversing all the ridges until one of three terminating conditions is satisfied (see below).
- 3) If all minutiae are used as the origin minutiae, terminate the procedure. Otherwise, return to step 2).

The termination conditions include the following three cases:

- 1) The vertical axis reaches a background region in the fingerprint image.
- 2) The vertical axis reaches a poor quality region in the fingerprint image.
- 3) The vertical axis reaches a high circular variance region in the fingerprint image.

III. FINGERPRINT MATCHING

The ridge feature vectors between the minutiae in the ridge coordinate system can be expressed as a directional graph whose nodes are minutiae and whose edges are ridge feature vectors. Thus, we can adopt graph matching methods to utilize the ridge feature vectors in fingerprint matching. Chikkerur *et al.* [19] proposed a graph-based fingerprint minutiae matching method in a Euclidean space. They first defined the local neighborhood of each minutia, called K -plet, which consists of the K -nearest minutiae from a center minutia. The comparison of two K -plets is performed by computing the distance between the two strings obtained by concatenating the K neighboring minutiae, sorted by their radial distance with respect to the center minutia. Neighborhoods are matched by dynamic programming and a match of local neighborhoods is propagated with a breadth-first fashion. Thus, we apply this matching scheme to our ridge-based coordinate system, since the ridge-based coordinate system can be represented as a graph and each coordinate system makes a local neighborhood. Moreover, the data structure of the ridge-based coordinate system is very similar to the K -plet structure proposed in [19].

The overall flow of the proposed fingerprint matching algorithm is as follows:

- 1) Initially match any pair of ridge-based coordinate systems extracted from the enrolled fingerprint image and the input fingerprint image using dynamic programming.
- 2) Select the top N degree of matched ridge-based coordinate pairs.
- 3) For every initially matched pair, a breadth-first search (BFS) is performed to detect the matched ridge-based coordinate pairs incrementally.
- 4) Check the validity of the matched coordinate pairs using the relative position and orientation of the minutiae and count the number of matched minutiae.
- 5) Iterate steps 3) and 4) N times and then return the maximum number of matched minutiae.
- 6) Compute the matching score.

Dynamic programming is applied to find the optimal solution in matching two string sequences in the enrolled and input ridge-based coordinates. The ridge feature vectors in a ridge-based coordinate system are arranged in the order of their ridge count feature component (rc), then the order is invariant intrinsically. Therefore, the feature vectors in a ridge-based coordinate system can be stored as the elements of an ordered sequence. Thus, all the enrolled and input ridge-based coordinates are compared one by one and a similarity score is computed for the dynamic programming. The similarity score is based on the *Bayesian* decision rule [20] and is calculated as follows:

$$\begin{cases} \text{score} = P\left(\frac{w_1}{X}\right), & \text{when } P\left(\frac{w_1}{X}\right) > P\left(\frac{w_2}{X}\right) \\ \text{score} = 0, & \text{otherwise} \end{cases} \quad (6)$$

where X is the absolute difference between two feature vectors, w_1 is the correctly matched class, and w_2 is the incorrectly matched class. In order to calculate the posterior probability, we assumed that the prior probabilities of w_1 and w_2 are equal. We estimated the conditional probability density functions $p(x/w_1)$ and $p(x/w_2)$ by using a Parzen window of a uniform kernel in the training set of FVC 2002 DB1. For the ridge feature vector,

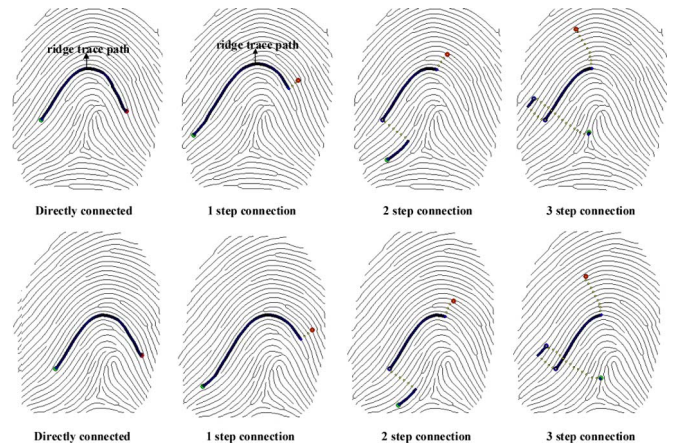


Fig. 9. Examples of corresponding ridge feature vectors according to the number of connection steps (upper and lower row images are from the same finger).

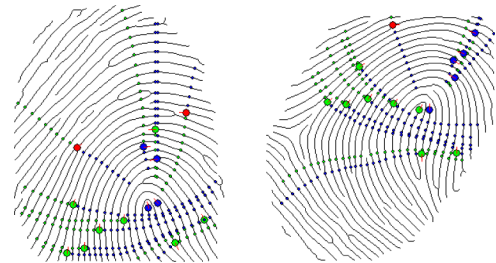


Fig. 10. Example of matched minutiae using the proposed ridge feature vectors (solid circles represent matched minutiae and dotted lines represent the vertical axis of each minutia).

TABLE I
EER COMPARISONS OF TWO MATCHING METHODS ON FVC DATABASES

| Database | EER(%) | |
|--------------|-----------------|------------------------------------|
| | Proposed method | Conventional minutiae-based method |
| FVC 2002 DB1 | 1.8 | 4 |
| FVC 2002 DB2 | 0.8 | 2.9 |
| FVC 2002 DB3 | 3.5 | 7.3 |
| FVC 2004 DB1 | 4.3 | 8.9 |

the three feature elements (ridge count, ridge length, and ridge curvature direction) are used to calculate the scores and the ridge type feature is used to check the validity of the candidate pairs. After that, we select the top N degree of matched ridge-based coordinate pairs. In this paper, we set the value N as 10. For every initially matched pair, we perform a BFS to increment the match for other neighboring ridge-coordinate systems. However, there is not always a path for every minutiae pair because we do not extract ridge features in the fingerprint regions which have low quality or a high curvature. Therefore, we find a detour path to perform the BFS [21]. For example, even if it is not possible to directly extract the ridge feature vector between minutia A and B due to the absence of a path, it is still possible to obtain the ridge feature vector by including minutia C (as $A - C - B$). Fig. 9 shows some examples of the corresponding ridge feature vectors using the detour, as the number of connection steps increases.

We check the validity of the matched coordinate pairs using the relative position and orientation of the minutiae used in conventional minutiae-based matching. If the relative position

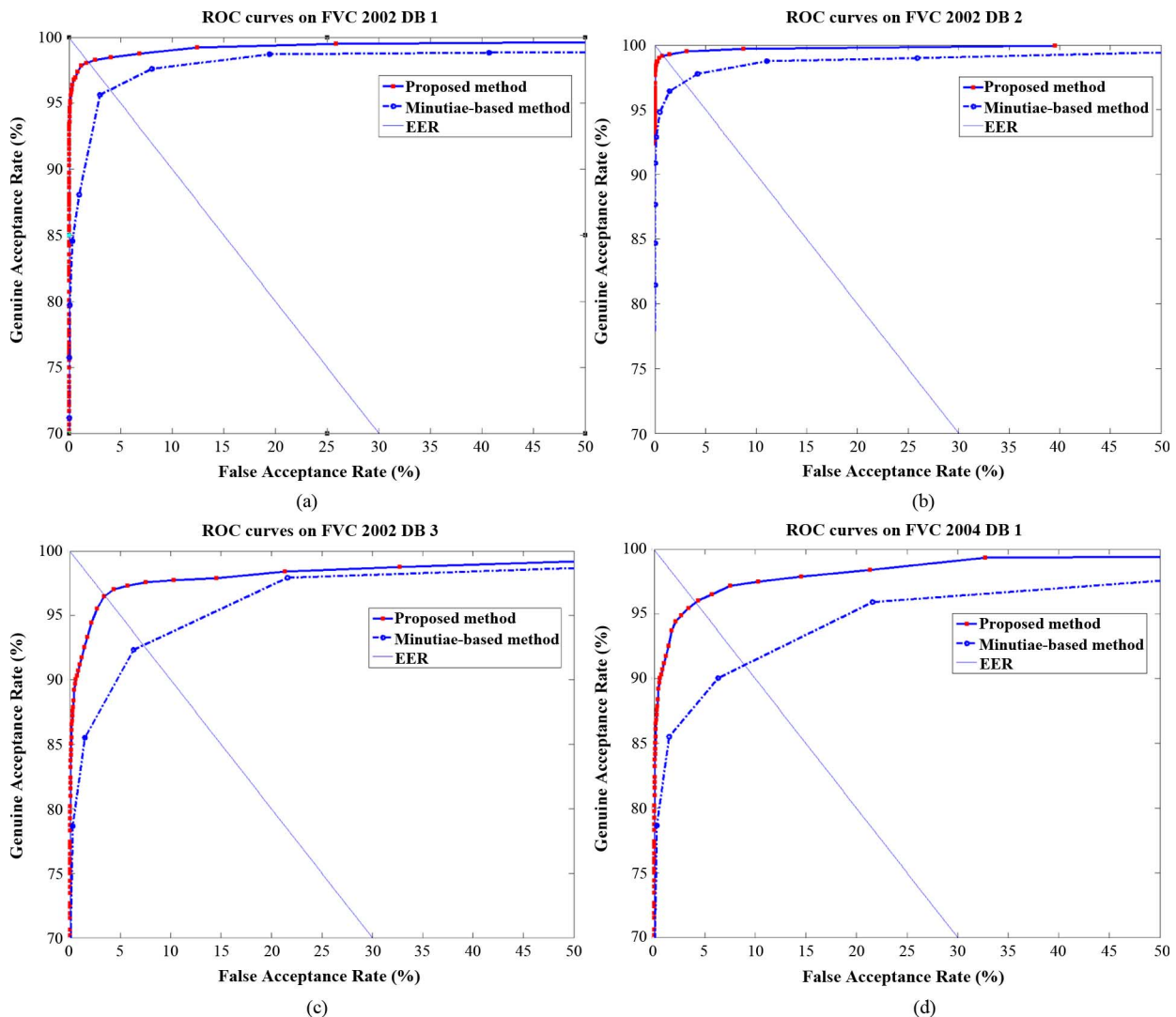


Fig. 11. ROC curves on each database.

and orientation of the minutiae in the coordinate pair are also matched, we can be sure that these minutiae are correctly matched. We then count the number of matched minutiae and store them. Finally, after the execution of the BFS procedure for every initial matched pair, we find the maximum number of matched minutiae between two fingerprints. Fig. 10 shows an example of matched minutiae using the proposed method. As shown in the figure, even if two impressions of the same finger are different due to skin distortion, many minutiae are matched correctly.

To compute the matching score, we must consider both the degree of overlap between two impressions and the degree of similarity of the overlapped region. Thus, the matching score can be computed as follows:

$$s_m = \frac{L \times L}{m_o \times n_o} \times \frac{L \times L}{N_1 \times N_2} \quad (7)$$

where L , N_1 , and N_2 are the number of matched minutiae, the number of minutiae in an input image, and the number of minutiae in a template image, respectively. m_o and n_o are the number of minutiae in the overlapping regions of the query and template images, respectively. The overlapped regions are where

two fingerprints intersect after the linear transformation (translation and rotation) using the matched minutiae.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

We compared the recognition performances of two algorithms (the conventional minutiae-based matching method [9] and the proposed method). To demonstrate the effect of the proposed ridge features more generally, we chose the conventional minutiae-based method, which is based on popular minutiae features such as minutiae position, minutiae orientation, and minutiae type [9] instead of the state-of-the-art minutiae-based algorithms which use additional specific matching techniques. The conventional method utilizes several reference points for local alignment and an adaptive tolerance box is used to calculate the number of matched minutiae.

For the experiments, we used the databases FVC 2002 DB1, DB2, DB3, and FVC 2004 DB1, released on the Web [22], [23]. Regarding fingerprint quality, FVC 2002 DB3 and FVC 2004 DB1 have lower quality fingerprints than other databases because the users were explicitly requested to exaggerate distortions [1]. Therefore, it is reasonable to analyze the robustness of the proposed method against skin distortions by using these

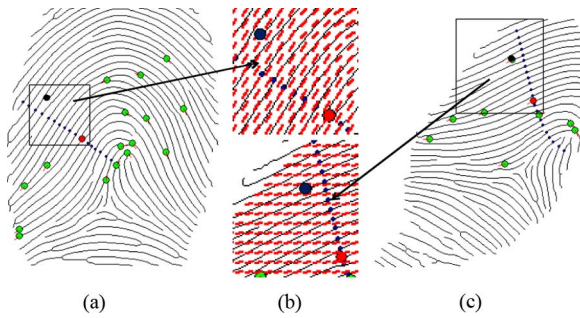


Fig. 12. Example of ridge length error caused by the ridge orientation estimation error.

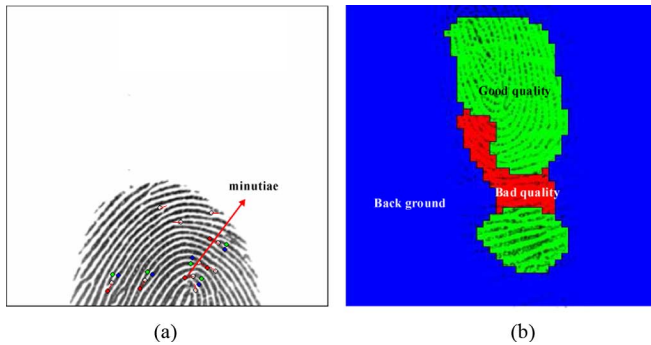


Fig. 13. Examples of fingerprint images with few ridge features.

databases. Each database is composed of 800 fingerprint images from 100 different fingers (eight impressions per finger). For genuine matches, each impression of each finger is compared with other impressions of the same finger. Therefore, 2800 genuine matches were executed in each database. For imposter matches, each impression of each finger is compared with all impressions of the different fingers. Therefore, 316 800 imposter matches were conducted in each database. Table I shows the equal error rate (EER) comparisons of two matching methods on the FVC databases and Fig. 11 shows the ROC curves on each database. From the experimental results, we can see that the proposed method is superior to the conventional minutiae-based one for all the databases. Even though the performances for FVC 2002 DB3 and FVC 2004 DB1 are lower than those for FVC 2002 DB1 and DB2, we can maintain that our ridge features can support the minutiae features when they are used together in the matching stage. Some recognition errors occurred and can be analyzed in the following way.

The first cause is a wrongly estimated orientation error. As shown in Fig. 12(b), the vertical axis starting from a minutia (red-circle) is slightly tilted towards the vertical axis determined from the corresponding minutia in Fig. 12(a) since the orientation fields in Fig. 12(c) are poorly estimated. As a result, there is a large variation between the ridge length (r_l) features in the corresponding sets. Therefore, by enhancing the orientation estimation process, the performance can be improved. Second, there are some minutiae pairs offering no ridge feature vectors because some images had small foreground regions or their levels of quality were too low, as can be seen in Fig. 13. In Fig. 13(a), the foreground region was very small and there were few minutiae. Moreover, only a few minutiae were located in the region (high circular variance region) around a singular point. In Fig. 13(b), the foreground region was split in two by a poor

quality region, so there was no connection between the minutiae in the upper good quality region and those in the lower good quality region. This disturbed the generation of ridge feature vectors in the whole fingerprint region, reducing the discriminating power. And the experiments were conducted on a PC with Core2 Duo 2.4 GHz. The average matching time of the proposed method was 83 ms.

V. CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

In this paper, we proposed a novel fingerprint matching algorithm using both ridge features and the minutiae. The ridge features consist of four elements (ridge count, ridge length, ridge curvature direction, and ridge type) that describe the relationship between the minutiae. With the proposed ridge features and conventional minutiae features (minutiae type, orientation, and position), we proposed a novel matching scheme using a BFS to detect the matched minutiae pairs. The experimental results show that the proposed method gives higher matching scores compared to the conventional minutiae-based one. Hence we can conclude that the proposed ridge features give additional information for fingerprint matching with little increment of template size. And, for future work, we will try to incorporate these features into the state-of-the-art minutiae-based matchers for further improvement of the matching performance. Also, our matching method needs to be improved for images with a small foreground area and those of low quality. Therefore, in future work, we will develop the use of global knowledge of fingerprints, such as singular point position, to enhance the matching accuracy. We will also develop a robust preprocessing method to reduce enhancement errors. Moreover, our ridge features can be used in other applications. In the area of fingerprint identification, it is important to be able to extract alignment-free features since it needs no time to align a query feature set with the N enrolled feature sets one by one [24]. In cancellable fingerprints, without a fiducial corresponding pair such as a core point, it is difficult to align a transformed feature set with an enrolled one [25]. The proposed ridge features are invariant to any transform, thus they can be used in addition to conventional alignment-free features in the fingerprint identification or cancellable fingerprint area.

REFERENCES

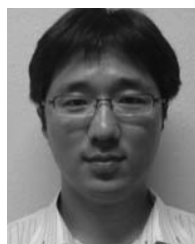
- [1] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. New York: Springer-Verlag, 2003.
- [2] A. Ross, S. Dass, and A. K. Jain, "A deformable model for fingerprint matching," *Pattern Recognit.*, vol. 38, no. 1, pp. 95–103, 2005.
- [3] X. Chen, J. Tian, X. Yang, and Y. Zhang, "An algorithm for distorted fingerprint matching based on local triangle feature set," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 169–177, Jun. 2006.
- [4] A. M. Bazen and S. H. Gerez, "Fingerprint matching by thin-plate spline modelling of elastic deformations," *Pattern Recognit.*, vol. 36, no. 8, pp. 1859–1867, Aug. 2003.
- [5] R. Cappelli, D. Maio, and D. Maltoni, "Modeling plastic distortion in fingerprint images," in *Proc. ICAPR*, 2001, pp. 369–376.
- [6] N. K. Ratha and R. M. Bolle, "Effect of controlled acquisition on fingerprint matching," in *Proc. 14th ICPR*, 1998, vol. 2, pp. 1659–1661.
- [7] C. Dorai, N. Ratha, and R. Bolle, "Detecting dynamic behavior in compressed fingerprint videos: Distortion," in *Proc. CVPR*, Hilton Head, SC, Jun. 2000, pp. 2320–2326.
- [8] X. P. Luo, J. Tian, and Y. Wu, "Aminutia matching algorithm in fingerprint verification," in *Proc. 15th ICPR*, Sep. 2000, vol. 4, pp. 833–836.
- [9] D. Lee, K. Choi, and J. Kim, "A robust fingerprint matching algorithm using local alignment," in *Proc. 16th Int. Conf. Pattern Recognition*, Quebec City, Que., Canada, Aug. 2002, vol. 3, pp. 803–806.

- [10] X. Jiang and W. Y. Yau, "Fingerprint minutiae matching based on the local and global structures," in *Proc. 15th Int. Conf. Pattern Recognition*, Barcelona, Spain, Sep. 2000, vol. 2, pp. 1038–1041.
- [11] Z. M. Kovacs-Vajna, "A fingerprint verification system based on triangular matching and dynamic time warping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1266–1276, Nov. 2000.
- [12] R. Cappelli, A. Erol, D. Maio, and D. Maltoni, "Synthetic fingerprint-image generation," in *Proc. 15th Int. Conf. Pattern Recognition*, Barcelona, Spain, Sep. 2000, pp. 3475–3478.
- [13] C. Lee, S. Lee, J. Kim, and S. Kim, "Preprocessing of a fingerprint image captured with a mobile camera," in *Proc. IAPR Int. Conf. Biometrics (ICB)*, Hong Kong, Jan. 2006, pp. 348–355, Springer LNCS-3832.
- [14] D. Maio and D. Maltoni, "Ridge-line density estimation in digital images," in *Proc. 14th ICPR*, 1998, vol. 1, pp. 534–538.
- [15] L. Hong, Y. Wan, and A. K. Jain, "Fingerprint image enhancement: Algorithm and performance evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 777–789, Aug. 1998.
- [16] S. Lee, H. Choi, and J. Kim, "Fingerprint quality index using gradient components," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 4, pp. 792–800, Dec. 2008.
- [17] K. V. Mardia and P. E. Jupp, *Directional Statistics*. New York: Wiley, 2000.
- [18] K. Choi, H. Choi, S. Lee, and J. Kim, "Fingerprint image mosaicking by recursive ridge mapping," *Special Issue Recent Adv. Biometrics Syst., IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 5, pp. 1191–1203, Oct. 2007.
- [19] S. Chikkerur, A. N. Cartwright, and V. Govindaraju, "K-plet and coupled BFS: A graph based fingerprint representation and matching algorithm," *Lecture Notes Comput. Sci., Adv. Biometrics*, vol. 3832, pp. 309–315, 2005.
- [20] S. Theodoridis and K. Koutroumbas, *Pattern Recognit.*, 3rd ed. San Diego, CA: Academic, 2006.
- [21] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. New York: McGraw-Hill Book Company, 2001.
- [22] Mar. 25, 2010 [Online]. Available: <http://bias.csr.unibo.it/fvc2002>
- [23] Mar. 25, 2010 [Online]. Available: <http://bias.csr.unibo.it/fvc2004>
- [24] B. Bhanu and X. Tan, "Fingerprint indexing based on novel features of minutiae triplets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 616–622, May 2003.
- [25] F. Farooq, R. Bolle, T. Jea, and N. Ratha, "Registration-free anonymous fingerprint recognition," in *Proc. IEEE Comput. Soc. Workshop Biometrics CVPR*, Minneapolis, MN, Jun. 2007.



Heeseung Choi received the B.S. and the M.S. degrees in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2004 and 2006, respectively. He is currently working toward the Ph.D. degree in electrical and electronic engineering from Yonsei University, Seoul, Korea.

He has been a research member of the Biometrics Engineering Research Center (BERC). His research interests include computer vision, biometrics, image processing, and pattern recognition.



Kyoungtaek Choi received the B.S. degree in electrical and electronics engineering from Chung-ang University, Seoul, Korea, in 2001, and the M.S. and Ph.D. degrees in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2003 and 2008, respectively.

Currently, he is a research engineer in LIG Nex1, Yong In, Korea. His research interests include computer vision, biometrics, image processing, and pattern recognition.



Jaihie Kim received the B.S. degree in electronic engineering from Yonsei University, Seoul, Korea, in 1979, and the M.S. degree in data structures and the Ph.D. degree in artificial intelligence from Case Western Reserve University, Cleveland, OH, in 1982 and 1984, respectively.

Since 1984, he has been a professor in the School of Electrical and Electronic Engineering, Yonsei University. He is currently the Director of the Biometric Engineering Research Center in Korea. His research areas include biometrics, computer vision, and pattern recognition.

Prof. Kim is currently the Chairman of Korean Biometric Association.