Incorporating Ridges with Minutiae for Improved Fingerprint verification

Ms.M.Indra[#1], Ms.S.Krishnaveni [#2], Ms.P.Suriyapriya[#3]

[#1,#2,#3]Asst.Prof, Department of Information Technology,
Sethu Institute of Technology, Virudhunagar District,TamilNadu, India.
indrahere8@gmail.com,krish.jan1st@gmail.com,spria.p16@gmail.com

Abstract

Next to DNA, fingerprint is the unique feature which identifies the individual. Distortions and skin deformations makes the fingerprint unreliable and it is difficult to match using minutiae alone. But when ridge features are incorporated with minutiae features (minutiae type, orientation and position) more topological information can be obtained. And also ridges are invariant to transformations such as rotation and translation[1]. Ridge based coordinate system is used to extract the ridge features such as ridge length, ridge count, ridge type and curvature direction in the skeletonized image. Breadth First Search is used to traverse the graph formed using the minutiae as the node and the ridge vector formed using the ridge features as the edge. The proposed ridge feature gives additional information for fingerprint matching with little increment in template size and can be used along with the existing minutiae features to increase the accuracy and robustness of fingerprint recognition systems.

Index Terms

Breadth first search, distortion, geometric transformation, ridge features, ridge-based coordinate system, ridge patterns, ridge vector.

1. INTRODUCTION

FINGERPRINT recognition is one of the most adopted technique for user identification. This is considered as a most reliable feature and the cost of implementing fingerprint recognition methods is very less than other biometric features. It is used in many forensic and commercial applications such as criminal investigation, electronic personal ID cards,etc. Although there is a significant improvement in fingerprint recognition, some challenging tasks may degrade the efficiency of fingerprint matching systems. Hence the fingerprint matching methods should be implemented in such a way that it should overcome all difficulties while matching. 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 are very different due to skin distortions. This distortion is an inevitable problem since it is usually associated with several parameters [6], [3], including elasticity of skin, nonuniform pressure applied by the subject, different finger placement with the sensor, etc.

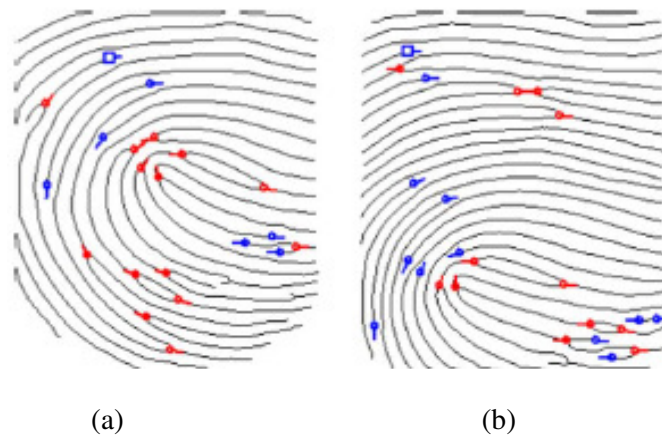(a)   (b)

Fig. 1. Example of skin distortions.

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 [6], [9]; detecting the distortions using special hardware or video sequences allowing some amount of distortion in the minutiae matching stages [8].

Luo et al. [8] used changeable tolerance boxes in the minutiae matching process. Starting from the center of the fingerprint image, the size of the tolerance boxes is increased till the border of the fingerprint area is reached. 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, a new and simple matching scheme by incorporating conventional minutiae features and additional ridge features associated with corresponding minutiae sets is proposed. 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.

2. FINGERPRINT PREPROCESSING AND RIDGE FEATURE EXTRACTION

A. Fingerprint Preprocessing

After preprocessing the image, the required fingerprint features for the proposed method were extracted (see Fig. 2). Some additional procedures for quality estimation and circular variance estimation is also introduced to improve the quality. Aftyer dividing the image into 8× 8 pixel

blocks, the mean and variance values of each block of the image are calculated to segment the fingerprint regions in the image.

Adaptive binarization is applied which selects varying threshold based on the intensity of pixels in each block. The orientation of ridges is estimated using the formula given below. Apply least square approximation for all the pixels in each block.
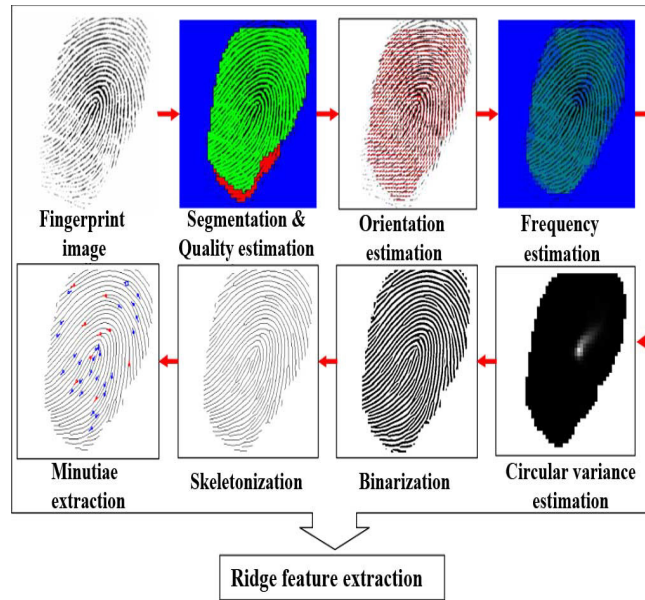

Fig. 2. Overall Preprocessing Steps

$$\text{Orientation} = 2 \sum \sum (g_x {}^* g_v) / \sum \sum (g_x{}^2 - g_v{}^2) \tag{1}$$

where g_x and g_y gradients taken along x and y axis respectively. After finished with the estimation of each block direction, those blocks without significant information on ridges and furrows are discarded based on the following formula,

$$E= \{ 2\sum\sum(g_x{}^*g_y) \pm \sum\sum(g_x{}^2 - g_y{}^2)\} / W{}^*W{}^*\sum\sum(g_x{}^2 + g_y{}^2) \tag{2}$$

For each block in the image, the certainty level E is compared with the fixed threshold value. Based on the certainty level some blocks are regarded as a background blocks which are considered to have no significant information. These blocks have no information needed for matching.

Histogram equalization is applied to enhance the image and to improve its contrast. When the input is taken through the sensor, some region in the image becomes invisible or broken. Histogram equalization method connects those false broken points. histeq() expands the pixel values of the image. Fingerprint enhancement is done by dividing the image into small blocks (32 by 32 pixels) and perform the Fourier transform for the pixels of the image taken throughout all the rows and columns of the input image Finally a skeletonized ridge image is obtained by the

process of thinning and other morphological operations. The end and bifurcation minutiae points are marked in the skeletonized image.

In order to avoid extracting false minutiae and to improve the confidence level and accuracy of the extracted minutiae set, the quality estimation procedure is performed. The direction of each block image of 16 by 16 pixels size is estimated by calculating the gradients along x and y axis. Two dimensional filter is created using fspecial('sobel') where sobel is used to emphasize the edges in the image.

The preprocessing stage alone is not enough to enhance the fingerprint image taken from the sensor. For example, false ridge breaks and ridge cross-connections due to over inking are not totally eliminated. These features result in the formation of spurious minutiae which will reduce the efficiency and accuracy of the matching process. So in order to develop a accurate fingerprint matching system, the spurious minutiae which are considered to be false minutiae have to be removed before extracting the ridge features.

B. Ridge Feature Extraction

1) Proposed Ridge-Based Coordinate System

After performing the preprocessing steps, the skeletonized ridges and minutiae information is obtained from the fingerprint image. Fingerprint Image is binarized to transform the 8-bit Gray image into a binary image which has distinct values and colours for ridges and furrows. The ridge coordinates are defined to extract ridge features between two minutiae. The ridge-based coordinate system is defined by considering a minutia (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. In addition, to define the sign of the vertical axis according to the origin, the cross product between the orientation of the origin and the vector pointing from the origin to the side of the vertical axis is calculated.
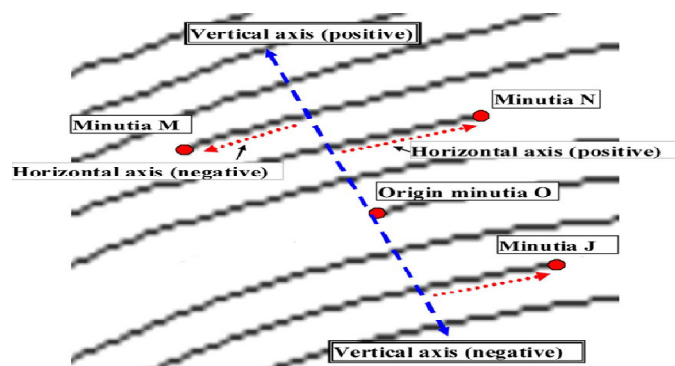

Fig. 3. Ridge Based Coordinate System

In the ridge-based coordinate system, the ridge features that describe the relationship between the origin and an arbitrary minutiae is given as follows.

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

(3)

where rc is the ridge count, rl is the ridge length, rcd is the ridge curvature direction and rt is the ridge type. These four components form a ridge-based feature vector between two minutiae and this feature vector is used in the matching process.

2) Ridge Feature Extraction

In the general ridge count methods [7], [10], 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 one point, at more than two points, or at no point, due to skin deformation 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. The ridge count feature is more robust to skin deformation. Furthermore, to increase the discriminating power of the ridge count (rc) feature, the direction of the ridge count line is also considered. 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 between the horizontal axis of the image and the axis of the corresponding minutiae. The ridge length value also has a sign and follows the sign of the related horizontal axis to improve the discriminating power. The probability distribution of the absolute difference of the ridge length elements are mostly less than 16 pixels. Ridge length is taken as the distance between two minutiae points in the input image. This provides more discrimatory information and improves the accuracy since the distance between two points is always constant in the case of misplacement of finger when compared to the template image. The distance between two points is calculated using the Euclidean's distance formula given below.

$$\text{Distance} = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \qquad (4)$$
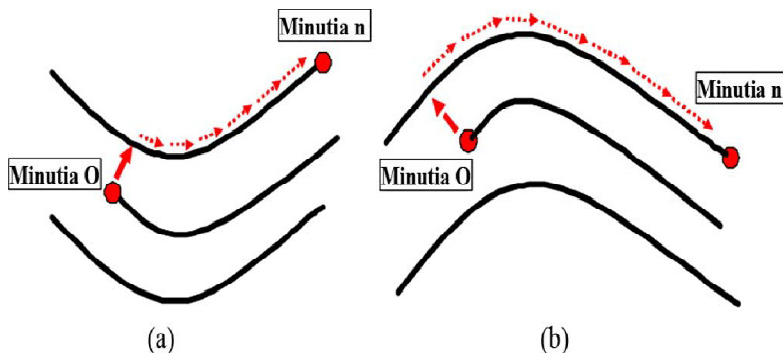

(a)　　　　　　　　　　(b)
Fig. 4. Ridge curvature direction. (a) Concave shape. (b) Convex shape.

To use more topology information in ridge patterns for matching, the ridge curvature direction is also considered. As shown in Fig. 3, even though the ridge count and ridge length values are very

similar, the shapes of the ridge patterns may be different [concave shape—Fig. 3(a); convex shape—Fig. 3(b)].

  The ridge curvature direction is defined as follows:

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

where v_i represents the ith vector between the sampling points along the horizontal axis from the intersection of the vertical and horizontal axes to the minutia n and N represents the number of sampling points.

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 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.

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) Whenever the vertical axis meets with the ridges attached to a minutia, then extract the ridge features such as 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 corresponding minutiae.

  b) Keep traversing all the ridges until one of three   terminating conditions is satisfied.

3) If all minutiae in the image gone through the above steps (1) and (2), terminate the procedure. Otherwise, return to step (2) so that all minutiae can be marked as origin.

The termination conditions include the following three cases:

  1) The vertical axis of each minutiae reaches a background region which has no needed information in the fingerprint image.
  2) The vertical axis reaches a poor quality region with very less minutiae and ridge points in the fingerprint image.
  3) The vertical axis reaches a high circular variance region in the fingerprint image.

3. 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. The graph matching methods are adopted to utilize the ridge feature vectors in fingerprint matching.

Chikkerur *et al.* [12] 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. While matching, the ridge maps of two fingerprints are taken which provides information about ridge features. The similarity between two ridge maps is calculated by choosing one minutiae from each image as origin. Initially the alignment stage is performed. From the origin all possible ridges are taken. The minimum ridge length is selected to compare the similarity certainity for two ridges. The ridge associated with each minutia is represented by using its coordinate values of the points on the ridge. Throughout the average inter-ridge length L, only a few sample points are taken. The overall flow of the proposed fingerprint matching algorithm is as follows:

1) Extract a pair of ridge based coordinate system, one from the enrolled fingerprint image and the other from the input fingerprint image. Match them using dynamic programming.
2) Select the top degree of matched ridge-based coordinate pairs.
3) For every initially matched pair, a breadth-first search is performed to detect the matched ridge- based coordinate pairs. This is done incrementally until all the possible matches are found.
4) Perform validity check and count the number of minutiae points which are matched by using the above three steps.
5) Iterate steps 3) and 4) N times and 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 calculated as follows:

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

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. So the similarity of correlating the two ridges is derived from:

$$S = \sum_{i=0}^{m} x_i X_i / [\sum_{i=0}^{m} x_i^2 X_i^2]^{\wedge} 0.5, \tag{7}$$

where $(x_i\text{-}x_n)$ represents the set of minutiae in first ridge and $(X_i\text{-}X_N)$ represents the set of minutiae in the second ridge. If the similarity value is greater than the threshold, transfer all the minutia in both fingerprints based on the reference pair of minutiae. translate and rotate all other minutia with respect to the reference minutia according to the formula,

$$\begin{pmatrix} xi\_new \\ yi\_new \\ \theta i\_new \end{pmatrix} = TM * \begin{bmatrix} (xi - x) \\ (yi - y) \\ (\theta i - \theta) \end{bmatrix}$$

(8)

where (x, y, θ) is the parameters of the reference minutia, and translation matrix (TM) is given by,

$$TM = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(9)

The rotation angle is based on the minutiae direction obtained in the preprocessing stage.

For the ridge feature vector, 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 using the relative position and orientation of the minutiae. If the relative position and orientation of the minutiae in the coordinate pair are also matched, then it can be sure that these minutiae are correctly matched. Then the number of matched minutiae are counted and stored. Finally, after the execution of the BFS procedure for every initial matched pair, the maximum number of matched minutiae between two fingerprints is found. The final match ratio for two fingerprints is the ratio of the number of minutiae in the input fingerprint image over the template fingerprint image. The matching score is calculated as 100*ratio and it ranges from 0 to 100. If the resultant score is larger than a pre-specified threshold, then the two fingerprints are concluded to be from the same finger.

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:

$$S_m = \frac{L*L}{m0*n0} * \frac{L*L}{N1*N2}$$

(10)

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_0 and n_0 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

4. EXPERIMENTAL RESULTS AND ANALYSIS

The recognition performances of two algorithms (the conventional minutiae-based matching method [13] and the proposed method) is compared. To demonstrate the effect of the proposed ridge features more generally, the conventional minutiae-based method, which is based on popular minutiae features such as minutiae position, minutiae orientation, and minutiae type [13] is chosen 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 genuine matches, each impression of each finger is compared with other impressions of the same finger. It is shown in Fig. 5 that the genuine acceptance rate is higher in the proposed method when compared to that of the conventional minutiae based method.
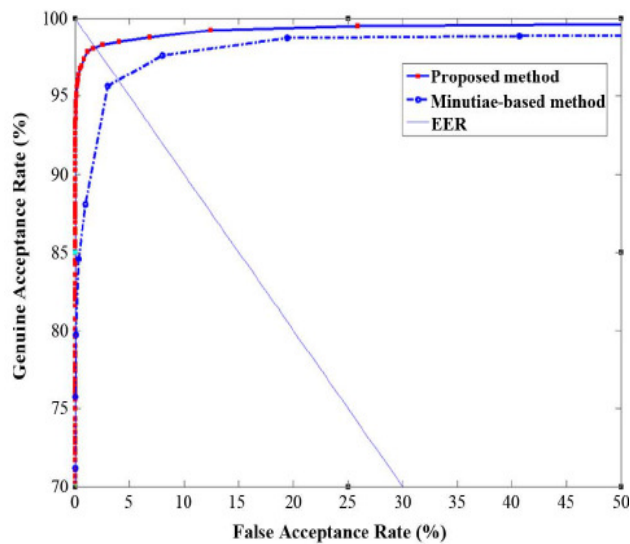

Fig. 5. ROC curve showing the accuracy of proposed method

5. CONCLUSION

In ridge based coordinate system, the ridge count (rc) is calculated by counting the number of ridges crossing 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. Ridge count feature is more robust to skin deformation. Furthermore, to increase the discriminating power of the ridge count (rc) feature, the direction of the ridge count line is also considered. The ridge length value also has a sign and follows the sign of the related horizontal axis to improve the discriminating power. With this approach , no complex data structure is involved and no minutiae alignment algorithm is used. 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 and gives additional information for fingerprint matching with little increment of template size. The proposed fingerprint matching method can be applied to images with small foreground area and images with low quality which contains very

less number of minutiae points than required. This proposed fingerprint matching method can also be enhanced by combining level 3 pores along with ridge features for improved accuracy. Before extraction of the fingerprint features, the fingerprint is enhanced to improve the robustness of the fingerprint. For this enhancement purpose, STFT (Short Time Fourier Transform Analysis) can be used. If such algorithm is implemented, all the intrinsic properties of the fingerprints such as the foreground region mask, local ridge orientation and local ridge frequency can be estimated in a less time.

REFERENCES

[1] Heeseung Choi, Kyoungtaek Choi, and Jaihie Kim, "Fingerprint Matching Incorporating Ridge Features With Minutiae," IEEE Trans. Inf. Forensics Security, vol. 6, no. 2, pp. 338-345, Jun. 2011.

[2] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition. New York: Springer-Verlag, 2003.

[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] X. P. Luo, J. Tian, and Y. Wu, "A minutia matching algorithm in fingerprint verification," in Proc. 15th ICPR, Sep. 2000, vol. 4, pp. 833–836.

[5] 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.

[6] A.Ross, S. Dass, and A. K. Jain, "A deformable model for fingerprint matching," Pattern Recognit., vol. 38, no. 1, pp. 95–103, 2005.

[7] 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. 1038–104.

[8] N. K. Ratha and R. M. Bolle, "Effect of controlled acquisition on fingerprint matching," in Proc. 14th ICPR, 1998, vol. 2, pp. 1659–1661.

[9] Asker M. Bazen and Sabih H. Gerez, "Thin-Plate Spline Modelling Of Elastic Deformations In Fingerprints", Proc. 3rd IEEE Benelux Signal Processing Symposium (SPS-2002), Leuven, Belgium, pp. S02-1 – S02-4, March 21-22, 2002.

[10] 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..

[11] 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.

[12] 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.

[13] 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.

[14] Sharat Chikkerur, Alexander N. Cartwright and Venu Govindaraju, "Fingerprint enhancement using STFT analysis", Pattern Recognition 40 (2007) 198–211.