IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-5, May 2018 E-ISSN: 2347-2693

A Fingerprint Representation Technique based on Minutiae Quadrilateral Structure

L. Menaria[1*], K. Jain[2]

[1*,2] Dept. of Computer Science and Engineering, College of Technology and Engineering, Maharana Pratap University of Agriculture and Technology, Udaipur, Rajasthan, India

[*]*Corresponding Author: laxmimenaria1@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 21/May/2018, Published: 31/May/2018

***Abstract*—** Fingerprint is a popular biometric trait which is used extensively in several applications for person authentication, providing high uniqueness and acceptable performance. The goal of any fingerprint representation is to capture as much of the unique information available in a fingerprint while discarding the variations between multiple impressions of the same finger. In this paper, we propose a new alignment-free fingerprint representation using the quadrilateral structure. We construct the feature set from the quadrilateral structure. The constructed feature set is quantised and mapped to a pre-defined 3D array. By sequentially visiting the cells of the 3D array, a fixed length 1D bit string is generated. This bit string is applied with DFT to generate a complex vector and the complex vector is multiplied with a random matrix generated by user's key, to generate a final cancellable template. The proposed method FS_QUAD is tested on FVC 2002 databases and results show satisfactory performance.

***Keywords*—** Fingerprint, representation, quadrilateral, cancellable template

I. INTRODUCTION

Biometric systems are extensively used to identify or authenticate persons reliably in many applications. Biometric recognition performs identification of a person using physiological or behavioral characteristics including fingerprint, face, palmprint, hand/finger geometry, iris, retina, signature, gait, voice pattern, ear, hand vein, odor or the DNA information of an individual [1]. These characteristics are referred to as traits, indicators, identifiers or modalities in biometric literature.

Fingerprints provide better distinctiveness and stability as compared to other biometric modalities [2]. Fingerprint-based authentication has been widely used in several applications such as forensic applications (criminal investigation) and non-forensic applications (passport control, distance learning). A fingerprint presents three different types of features when analyzed at different scales. These features are represented at three different levels in a hierarchical order, namely, global level, local level, very local level [2]. Level 1(Global level) includes the macro details of fingerprint such as singular points (core, delta, whorl etc.), level 2 (Local level) includes small details of ridge flow pattern such as minutiae points (ridge ending and bifurcation), level 3 (Very local level) includes all dimensional attributes of ridge such as width, pores, shape etc.

Fingerprint-based authentication provides several advantages over traditional authentication system, but it is not foolproof. Fingerprint-based authentication systems are vulnerable to several security breaches and privacy threats. In knowledge or token-based system, if a token is lost it can be replaced and password can be reset, but if a biometric data is lost it cannot be revoked or replaced [3]. Therefore, biometric template protection schemes are required to secure the original biometric data. Due to large intraclass variations at each fingerprint image acquisition, fingerprint template protection is not an easy task. An ideal template protection scheme must fulfil the following requirements [2]:

- Irreversibility**:** It should be computationally infeasible to obtain the unprotected template from the protected template.

- Diversity**:** For two different applications, the same transformation function cannot be used. It should not be possible to match protected template from two different applications.

- Accuracy**:** Accuracy must be preserved when matching is performed on the transformed template.

• Revocability**:** If a template is compromised, it should be possible to generate a new template for the same finger by using different transformation function.

Biometric template protection methods are broadly classified into two categories: biometric cryptosystem and feature transformation [4]. Biometric cryptosystem [5] protects a template by binding a key to the biometric features or generating a key from the biometric features. Feature transformation [3] approach applies a non-invertible transformation to generate the protected template at enrolment phase. The same transformation is applied at recognition phase to generate query template. Matching is performed in the transformed domain. Hybrid system combines both basic (biometric cryptosystem and feature transformation) approaches.

The rest of the paper is structured as follows. Section II provides a review of existing fingerprint representation and cancellable template generation methods. Section III presents the proposed method of new fingerprint representation. Section IV gives the experimental results of the proposed method and comparison with existing methods. Conclusion is presented in section V.

II. RELATED WORK

In literature, many minutiae based template protection schemes are proposed and can be categorised into alignment based and alignment-free methods. In alignment-based method [3], a registration point (core or delta) is required to align the fingerprint image. In alignment-free approach, no singular point is required for alignment.

Ratha et al. [3] presented three feature transformation approaches namely, Cartesian, polar and functional transformation to generate a cancellable fingerprint template. Moujahdi et al. [6] proposed a fingerprint shell based secure representation of fingerprint templates. For fingerprint shell, spiral curves are constructed using the information provided by minutiae points. Matching is performed using the spiral curves. Wang and Hu [7] proposed an alignment-free cancellable template generation scheme. The quantised pair minutiae vectors are protected by densely infinite-to-one mapping (DITOM) based transformation. Prasad and Santosh Kumar [8] presented an alignment-free cancellable template generation method using multiline neighbouring relation. M rectangles are constructed around the reference minutia point to generate the multiline neighbouring relation. The minutiae points, which are fall in the rectangles are selected and rotation invariant distance and orientation angle of the selected minutiae with respect to reference minutia are calculated.

Lee and Kim [9] proposed an alignment-free cancellable bit string generation method from minutiae points. For this, one minutia is chosen as reference minutia and other minutiae are translated and rotated based on the position and orientation of reference minutia. These minutiae points are mapped onto a pre-defined 3D array. By sequentially visiting the cells of the 3D array, a 1D bit string is generated and the order of string is permuted by using the type of reference minutia and user's PIN. Jin et al. [10] developed a technique to generate a revocable template in terms of bit string created from a set of minutiae points via a polar grid based 3-tuple quantisation technique. Yang et al. [11] developed a Delaunay quadrangle-based fingerprint authentication system and constructed a unique topology code from each quadrangle which enhances the system security and provides accurate local registration under distortion. Sandhya and Prasad [12] formed the K nearest neighbour structure (k-NNS) around each minutiae point. The k-NNS is quantised and mapped to a 2D array to generate a binary string. The bit string is applied with DFT to create a complex vector and the complex vector is multiplied with a user-specific random matrix to generate a cancellable template.

Yang et al. [13] proposed an alignment-free fingerprint bio-cryptosystem based on modified Voronoi neighbour structures. All the Voronoi neighbour structures are quantised and mapped to a predefined 3D array and generate a fixed length binary string. Zhe and Jin [14] proposed a fingerprint representation using minutia vicinity decomposition (MVD). The minutia vicinity is decomposed into four triplets and a set of invariant features are extracted from these triplets. Sandhya et al. [15] proposed two methods namely FS_INCIR and FS_AVGLO to construct a feature set from the Delaunay triangles. The constructed feature sets are quantised and mapped to a 3D array to generate a fixed length binary string. The bit string is applied with DFT to create a complex vector and the complex vector is multiplied with a user-specific random matrix to generate a cancellable template. Vij and Namboodiri [16] presented a fingerprint representation by using the local structure called arrangement structure that captured the complete information about the geometry of neighboring points around a central minutia.

III. METHODOLOGY

The generic framework of the proposed fingerprint representation and protection method contains the following steps:

1. Constructing minutiae quadrilateral structure for each minutia point and computing the feature set (FS) from the quadrilateral structure using method FS_QUAD.

2. Quantising the extracted feature set and mapping it to a 3D array.

3. Generating 1D bit string.

4. Generating the cancellable template.

5. Perform matching.

The flow diagram of the proposed method is shown in figure 1.

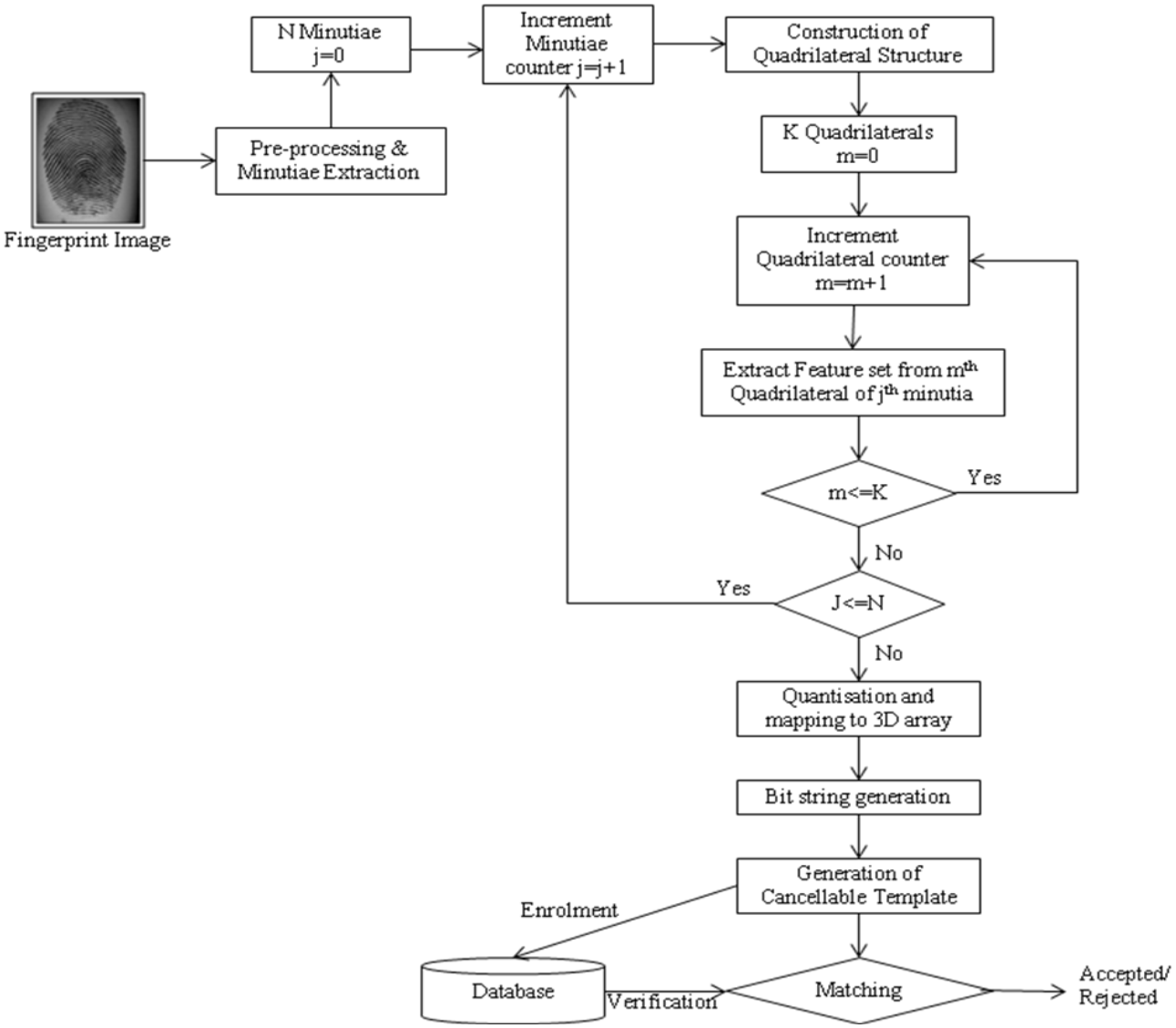

Figure 1. Flow diagram of the proposed method (FS_QUAD)

3.1 Constructing minutiae quadrilateral structure and computing the feature set

Minutia point, extracted from the fingerprint image, is represented by its location (x and y coordinates) and orientation (θ). A set of minutiae points is represented as $M_i=(x_i, y_i, \theta_i)^j$ where j is the number of minutiae points in a fingerprint.

From the extracted minutiae points, quadrilateral structures are constructed for each minutia. The process of constructing quadrilateral structure is shown in figure 2, which is as follows:

• For each minutia point X, we calculate its k nearest neighbors based on their Euclidean distance from the minutia point X. In figure 2, let k=5 and the nearest neighbors n1, n2, n3, n4 and n5 are selected for minutia point X.

• The nearest neighbors are arranged in clockwise order. Let after clockwise arrangement, nearest neighbors are n1, n2, n3, n4 and n5.

• From the clockwise arranged minutiae points, quadrilaterals are constructed. Let, for minutia point X and nearest neighbors n1, n2, n3, n4 and n5 quadrilaterals P, Q, R, S and T are constructed.

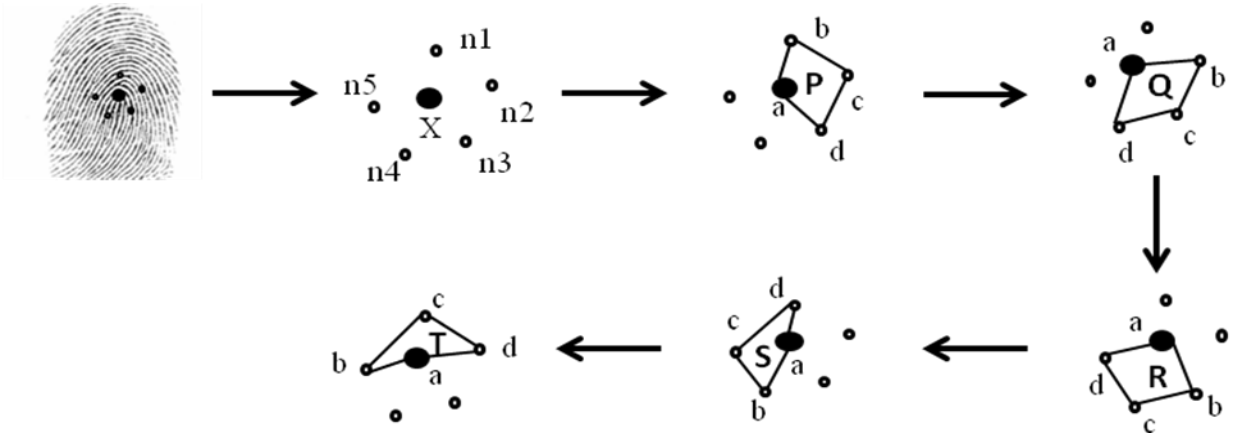

Figure 2. The process of constructing quadrilaterals for minutia X. First, we find the nearest neighbors around minutia point, then arrange neighbors in clockwise order and construct the quadrilateral structure

From each quadrilateral, we compute the feature set which comprises twelve features. Consider a quadrilateral abcd as shown in figure 3, features extracted from this quadrilateral are:

• Rotation invariant distance from each vertex of a quadrilateral to the centroid of the quadrilateral (d_a, d_b, d_c, d_d). Let the centroid of quadrilateral be (x_{cent},y_{cent}). Then distance d_a can be calculated as:

$$\chi = (x_{cent} - x_a)\cos\theta_a + (y_{cent} - y_a)\sin\theta_a$$

$$\gamma = (x_{cent} - x_a)\sin\theta_a - (y_{cent} - y_a)\cos\theta_a$$

$$d_a = \sqrt{\chi^2 + \gamma^2} \quad (1)$$

In the same way d_b, d_c and d_d can be calculated using (1) by considering vertices (x_b, y_b), (x_c, y_c), (x_d, y_d) and orientation θ_b, θ_c, θ_d respectively.

• The orientation of vertices which are the original orientation of minutiae points (θ_a, θ_b, θ_c, θ_d).

• The internal angles of quadrilateral at each vertex (α_a, α_b, α_c, α_d).

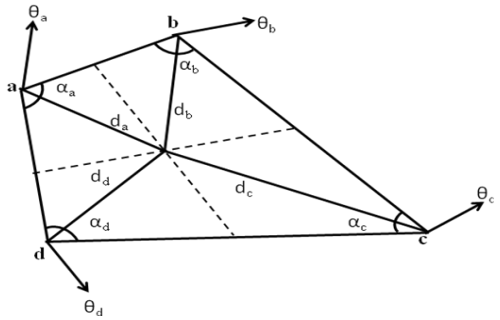

Figure 3. Sample quadrilateral considered for feature extraction

For each minutia point X, we compute the feature set FS_X consisting of k×12 features. The resultant feature set for N minutiae points is given by FS= {FS_1, FS_2, FS_3...FS_N} which consists of N×k×12 features. Thus, quadrilateral provides more discriminative information as compared to the triangle.

3.2 Quantising and mapping the feature set FS

The feature set is quantised by taking d_i on x-axis that ranges from 0 to maximum of distance computed (max (d_i)), orientation θ_i on y-axis that ranges from 0 to 2π and internal angles α_i on z-axis that ranges from 0 to 2π. The feature set is mapped to a 3D array after quantisation. The 3D array is defined and divided into cells of size C_x, C_y, C_z as in [15]. The size of 3D array is $W_x \times W_y \times W_z$ where W_x=floor (max(d_i)/C_x), Wy=floor ($2\pi/C_y$) and W_z=floor($2\pi/C_z$). W_x, W_y, W_z represents the number of cells in the 3D array. Figure 4 shows length, width and height of cells in a 3D array. The advantage of mapping FS to the 3D array is whatever may be the size of feature set, it produces a fixed length bit string.

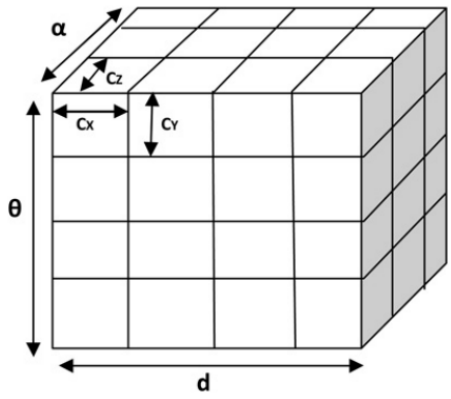

Figure 4. 3D array on which FS is mapped with cell size Cx, Cy, Cz

3.3 Generating the 1D bit string

After mapping the feature set to 3D array, we generate 1D bit string by sequentially visiting the cells of the 3D array. For this, we find which cell of the array includes a feature (d_i, θ_i, α_i) by calculating (x_i, y_i, z_i) as:

$$x_i=\text{floor}(d_i/C_x)$$
$$y_i=\text{floor}(\theta_i/C_y) \quad (2)$$
$$z_i=\text{floor}(\alpha_i/C_z)$$

Where x_i, y_i, z_i represents the indices of the 3D array. The value of a cell is set to one if one or more than one feature falls in a cell otherwise it is set to 0. This shows many to one mapping on a cell of the 3D array and ensures non-invertibility. The size of the bit string (n) is equal to the size of the 3D array, that is, $n= W_x \times W_y \times W_z$.

3.4 Generating the cancellable template

To protect the 1D bit string, we apply a two-step process as in [15]:

(i) A discrete Fourier transform (DFT) is applied to bit string B_s to generate a complex vector V. The complex vector V of size n×1 is generated by performing an n-point DFT as in (3).

$$V = \sum_{s=0}^{n-1} B_s \, e^{-j2\pi is/n}, \quad i = 0, 1, \ldots, n-1 \quad (3)$$

(ii) The complex vector is secured by applying a non-invertible transformation. A user-specific random matrix (A) is generated using the user's key. The dimension of A should be $u \times v$ where $v=n$. Now, to generate a final cancellable template of size $u \times 1$, this random matrix is multiplied by the complex vector V as in (4)

$$A \times V = T \quad (4)$$

The template T is stored in the database during enrolment and provide the irreversibility property because there are infinitely many solutions for V in (4). During verification, the same transformation is applied using the same random matrix used at enrolment, to generate the query template.

3.5 Matching

A fingerprint matching algorithm compares two fingerprint images and output either a matching score (similarity score between 0 and 1) or a binary decision (match/ non-match). We compute the matching score between enrolled and query template as in [15]. The distance between enrolled template Te and query template Tq is given by

$$d(Te, Tq) = \frac{\| Te - Tq \|_2}{\| Te \|_2 + \| Tq \|_2} \quad (5)$$

Where $\|\|_2$ denotes the 2-norm.

From this, the matching score is given by

$$S(Te, Tq)=1- d(Te, Tq) \quad (6)$$

The matching score generates between 1 and 0 where matching score 1 shows the perfect match and 0 shows a total mismatch between query and enrolled template.

IV. RESULTS AND DISCUSSION

4 Experimental results

4.1 Experiment setup

The proposed method is tested using the three databases of Fingerprint Verification Competition (FVC) 2002, namely, FVC 2002 DB1_B, DB2_B, DB3_B [17]. Each database consists of 10 fingers with 8 samples per finger i.e. 80 fingerprints. Out of 8 samples, we considered two samples of each finger for the performance evaluation. The trial version of Neurotechnology Verifinger 10.0 SDK [18] is used to extract the minutiae points from the fingerprints.

4.2 Performance measures

The performance measures used in our experiment are: False acceptance rate (FAR) which is defined as the ratio of successful impostor attempts to the total impostor attempts, False rejection rate (FRR) which is defined as the ratio of unsuccessful genuine attempts to the total genuine attempts, Genuine acceptance rate (GAR) which is defined as the ratio of successful genuine attempts to the total genuine attempts and Equal error rate (EER) which is defined as the error rate when the FRR and FAR are equal. A lower value of EER indicates better recognition performance. The genuine and imposter scores are used to compute the performance measures.

4.3 Accuracy

The proposed method (FS_QUAD) is evaluated in terms of EER by tuning the parameters k (number of nearest neighbours) and C_x, C_y, C_z (cell size of the 3D array in the quantization step). The proposed method is examined in the same key scenario. In the same key scenario, same user's key is used to generate a random matrix in (4) for all the users enrol into the system. Table 1 shows the EER obtained for the proposed method in the same key scenario for different cell size (C_x, C_y, C_z) and nearest neighbours (k). From table, it is observed that our proposed method gives optimal result for k=4, C_x=10, C_y=10 and C_z=30 i.e. EER 9.50%, 1.12%, 10.06% for database FVC 2002 DB1, DB2, DB3, respectively.

Figure 5 shows the ROC curve for the proposed method FS_QUAD evaluated on databases FVC 2002 DB1, DB2, DB3 in the same key scenario.

Table 1. EER% for different k and cell sizes in same key scenario

K	Cell Size			FVC 2002		
	C_x	C_y	C_z	*DB1_B*	*DB2_B*	*DB3_B*
4	10	10	20	10.06	1.05	9.62
4	**10**	**10**	**30**	**9.50**	**1.12**	**10.06**
4	15	15	40	10.12	5.60	10.31
5	10	10	20	10.18	3.50	15.0
5	10	10	30	9.43	2.25	12.23
6	10	10	40	9.75	3.3	10.18

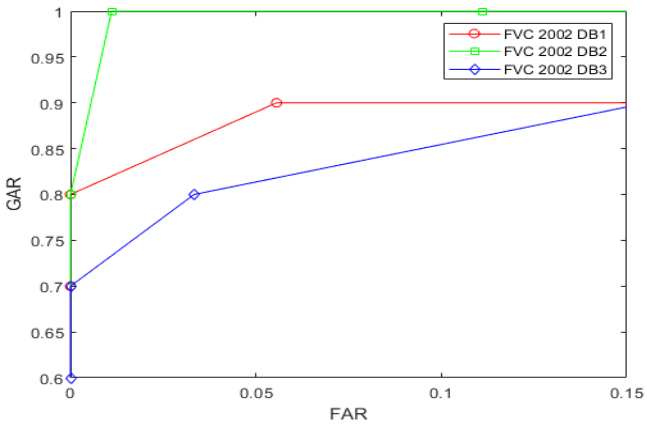

Figure 5. ROC curve for databases FVC 2002 DB1, DB2, DB3 in same key scenario for method FS_QUAD

4.4 Comparison with FS_INCIR [15] and FS_AVGLO [15] methods

We compared our proposed method (FS_QUAD) with other existing methods (FS_INCIR [15] and FS_AVGLO [15]) in terms of EER% in the same key scenario for database FVC 2002 DB1, DB2 and DB3. Table 2 shows the comparison between the proposed method and existing methods (FS_INCIR [15] and FS_AVGLO [15]) in terms of EER%. From the results, it is observed that the proposed method shows lower EER value as compared to methods FS_INCIR [15] and FS_AVGLO [15]. Thus, the FS_QUAD method has better recognition performance than existing methods (FS_INCIR and FS_AVGLO). Figure 6 plots the False Acceptance Rate (FAR) / False Rejection Rate (FRR) and shows the point corresponding to EER obtained for FVC 2002 DB2 in the same key scenario for the proposed method (FS_QUAD) and existing methods (FS_INCIR and FS_AVGLO).

Table 2. Performance comparison in terms of EER%

EER%	FVC 2002		
	DB1_B	*DB2_B*	*DB3_B*
FS_INCIR	9.93	5.39	10.37
FS_AVGLO	9.75	4.40	10.16
FS_QUAD (proposed)	**9.50**	**1.12**	**10.06**

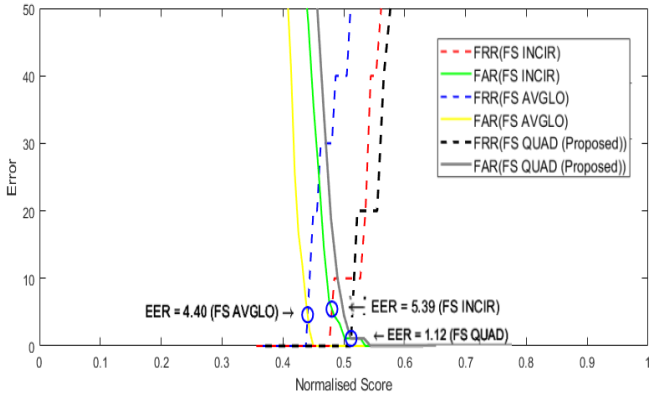

Figure 6. Comparison of EER obtained in same key scenario for database FVC 2002 DB2 for method FS_INCIR [15], FS_AVGLO [15] and FS_QUAD (proposed method)

V. CONCLUSION

A primary requirement in fingerprint template protection is to extract features from the fingerprint which are robust, contain more discriminative information and can be represented in a simplified form (binary) so that it can be directly applied with existing template protection schemes while maintaining the recognition rate. This paper presented a new fingerprint representation technique based on the quadrilateral structure. The invariant features extracted from the quadrilateral structures are alignment-free, fixed-length bit string and provide more discriminative information. The generated bit string is protected using the existing template protection scheme. The experimental results show that the proposed method depicts better performance as compared to the existing methods (FS_INCIR [15] and FS_AVGLO [15]).

References

[1] A.K. Jain, A. Ross, S. Prabhakar, "*An introduction to biometric recognition*", IEEE Transactions on Circuits System for Video Technology, Vol.14, Issue.1, pp. 4–20, 2004.

[2] D. Maltoni, D. Maio, A.K. Jain, S.Prabhakar, "*Handbook of Fingerprint Recognition*", Springer, 2009.

[3] N. Ratha, S. Chikkerur, J. Connell, R.M. Bolle, "*Generating cancelable fingerprint templates*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.29, Issue.4, pp.561-572, 2007.

[4] A.K. Jain, K. Nandakumar, A. Nagar, "*Biometric template security*", EURASIP Journal on Advances in Signal Processing archive, Vol.2008, pp. 1 –17, 2008.

[5] U. Uludag, S. Pankanti, S. Prabhakar, A.K. Jain, "*Biometric Cryptosystems: Issues and Challenges*," Vol.92, Issue.6, pp.948–960, 2004.

[6] C. Moujahdi, G. Bebis, S. Ghouzali, M. Rziza, "*Fingerprint shell: secure representation of fingerprint template*", Pattern Recognition Letters, Vol.45, pp. 189 –196, 2014.

[7] S. Wang, J. Hu, "*Alignment-free cancelable fingerprint template design: a densely infinite-to-one mapping (DITOM) approach*", Pattern Recognition, Vol.45, Issue.12, pp. 4129 – 4137, 2012

[8] M.V.N.K. Prasad, C. Santhosh Kumar, "*Fingerprint template protection using multiline neighboring relation*", Expert Systems with Applications, Vol.41, Issue.14, pp. 6114 –6122, 2014.

[9] C. Lee, J. Kim, "*Cancelable fingerprint templates using minutiae-based bit-strings*", Journal of Network and Computer Applications, Vol.33, Issue.3, pp. 236 –246, 2010.

[10] Z. Jin, A.B.J. Teoh, T.S. Ong, C, Tee, "*Fingerprint template protection with minutiae-based bit-string for security and privacy preserving*", Expert Systems with Applications, Vol.39, Issue.6, pp.6157–6167, 2012.

[11] W. Yang, J. Hu, S. Wang, "*A Delaunay quadrangle-based fingerprint authentication system with template protection using topology code for local registration and security enhancement*", IEEE Transactions on Information Forensics and Security, Vol.9, Issue.7, pp. 1179 –1192, 2014.

[12] M. Sandhya, M.V.N.K. Prasad, "*k-Nearest Neighborhood Structure (k-NNS) based alignment-free method for fingerprint template protection*". In the Proceedings of the 2015 International Conference On Biometrics (ICB), Thailand, pp.386–393, 2015.

[13] W. Yang, J. Hu, S. Wang, M. Stojmenovic, "*An alignment-free finger-print bio-cryptosystem based on modified Voronoi neighbor structures*", Pattern Recognition, Vol.47, Issue.3, pp.1309–1320, 2014.

[14] J. Zhe, A.T.B. Jin, "*Fingerprint template protection with minutia vicinity decomposition*", In the Proceedings of 2011 International Joint Conference on Biometrics (IJCB 2011), USA, pp.1-7, 2011.

[15] M. Sandhya, M.V.N.K. Prasad, R.R. Chillarige, "*Generating cancellable Fingerprint templates based on Delaunay triangle feature set construction*", IET Biometrics, Vol.5, Issue.2, pp. 131–139, 2016.

[16] A. Vij, A. Namboodiri,"*Learning Minutiae Neighborhoods: A New Binary Representation for Matching Fingerprints*", In the Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, USA, pp.64-69, 2014.

[17] D.Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain, "*Fvc 2000: fingerprint verification competition*", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.24, Issue.3, pp.402–412, 2002

[18] "*VeriFinger SDK*", Neurotechnology, 2017.

