International Journal of Computer Applications (0975 – 8887)
Volume 11– No.7, December 2010

A Novel Mathematical Algorithm to Represent and Reconstruct some Bio-Metric Patterns

Vani. P
S. A. Engineering College
Chennai, Tamil Nadu
India

Jagannathan. R
Vinayaga Missions University
Chennai, Tamil Nadu
India

ABSTRACT

Fingerprint and Iris are significant biometric tools used for authentication. This paper proposes a novel methodology by which the biometric patterns can be visualized as a set of Bezier curves and hence represented by the corresponding Bezier points, resulting in considerable reduction in the file size. This scheme utilizes the Bezier curve representations for effective compression of all those images. Initially, the ridges and furrows present in the fingerprint and iris image are extracted along with their co-ordinate values. The control points are determined for all the ridges and furrows. The control points of all the ridges determined are stored and are used to represent the fingerprint and iris image. When needed, those can be reconstructed from the stored control points using Bezier curves. The quality of the reconstructed fingerprint and iris is also maintained. Thus the proposed scheme achieves considerable memory reduction in storing the biometric patterns.

General Terms

Pattern Recognition, Biometrics.

Keywords

Bezier Control points, Data Compression, Fingerprint, Iris Pattern.

1. INTRODUCTION

Fingerprints and Iris' are formed in the embryo. They remain the same for an individual during their entire life; they can only be altered in life by accidental causes like burns, injuries, diseases etc. Nonetheless, the fingerprints are formed underneath the skin in a layer called dermal papillae. As long as that layer of papillae is there, the fingerprints will always come back, even after scarring or burning. Similarly except in the event of an injury to the eyeball, iris remains unchanged throughout an individual's lifetime. Iris is a membrane in the eye. The fingerprint patterns are hereditary and they are unique. The pattern extracted from the iris is unique for even genetically identical twins. The iris-scan process begins with a photograph. A specialized camera, typically very close to the subject, no more than three feet, uses an infrared imager to illuminate the eye and capture a very high-resolution photograph. This process takes only one to two seconds and provides the details of the iris that are mapped, recorded and stored for future matching or verification. The fingerprint can be combined with any authentication factor and can be used as a powerful tool against repudiation [1].

If a scanned fingerprint is stored as a JPEG file, it occupies around five to ten Kilo Byte. On the other hand if the scanned data is stored as string or as an index file [2], it occupies a memory space of about two to four KB. Here in the present work, a methodology is presented, by which a fingerprint can be stored with in a memory space of about 400 to 900 bytes only, resulting in considerable reduction in data size and from which an acceptable quality of a fingerprint can be regenerated. The details corresponding to the minutiae points are carefully retained to the maximum extend, so that the loss of information is minimized. Iris recognition technology converts the visible characteristics as a phase sequence into an Iris Code. Usually the size of the template is 600 to 800 bytes. A template stored, is used for future identification attempts. Here a methodology is presented, by which an iris can be stored with in a memory space of about 200 to 500 bytes only, resulting in considerable reduction in data size and from which an acceptable quality of an iris can be regenerated. The details corresponding to the rings, furrows and freckles are carefully retained to the maximum extend, so that the loss of information is minimized.

2. PREPROCESSING

2.1 Histogram Equalization

Histogram equalization describes a mapping of grey levels p into grey levels q in such a way that the distribution of grey level q is uniform. This mapping stretches contrast (expands the range of grey levels) for grey levels near the histogram maxima. As the contrast is extended to most of the image pixels, the transformation increases the detectability of many image features [3].

2.2 FFT Enhancement

The image enhancement techniques are frequently used to decrease the noise and improve the definition of ridges against valleys. In our scheme, to enhance the fingerprint image, Fast Fourier Transform (FFT) is applied separately to each block of the image [3]. The enhanced image is then binarized and fed as input to orientation field estimation.

2.3 Binarization

Binarization increases the contrast between the ridges and valleys in a fingerprint image, and as a result eases minutiae extraction. The binarization results a binary fingerprint image containing two levels of information, the foreground ridges and the background valleys [4].

2.4 Orientation Field Estimation

The orientation estimation is an elementary step in the enhancement process as the succeeding filtering stage depends on the local orientation so as to efficiently enhance the fingerprint image [4]. The proposed scheme for fingerprint compression employ gradient based approach for estimating the orientation field of the fingerprint image. Primarily, the gradient vectors are computed by considering the partial derivatives of image intensity at every pixel. The gradient vectors can be represented as $[g_x, g_y]^T$. With an input image, the gradient vectors signify the highest deviation of gray intensity that lie perpendicular to the edge of ridgelines [5].

2.5 Region of Interest (ROI) Extraction by Morphological operations

The binary morphological operators are applied on the binarized fingerprint image. The primary function of the morphological operators is the elimination of any obstacles and noise from the image. In addition, the morphological operators remove the unnecessary spurs, bridges and line breaks. Then, thinning process is performed to reduce the thickness of the lines (removes redundant pixels) so that the lines become 1-pixel wide and easily distinguishable from the other regions of the image. Clean operator, Hbreak operator, Spur operator and Thinning are the morphological operators utilized in the proposed scheme [6], [7].

3. THE ALGORITHM

The Bezier equation of a curve being

$$P(u) = \sum_{k=0}^{n} P_k J_{k,n}(u), 0 \le u \le 1$$

where $P_k = (X_k, Y_k, Z_k)$, k = 0 to n are used to produce the position vector P(u) on the path of an approximating Bezier polynomial function between P(0) and P(n).

$J_{k,n}$ (u) is Bernstein Polynomial.

The x co-ordinate of any point on a Bezier curve is given by

$$x(u) = \sum_{k=0}^{n} X_k J_{k,n}(u), 0 \le u \le 1 \text{ where } P_k = X_k$$

and similarly the y coordinate of any point is represented by

$$y(u) = \sum_{k=0}^{n} Y_k J_{k,n}(u), 0 \le u \le 1 \quad \text{where } P_k = Y_k$$

But the same x(u) and y(u) can also be obtained, as given below, from a unique set of the four control points (x_0, y_0), (x_1, y_1), (x_2, y_2), and (x_3, y_3) where the first and the last points are the two end points of the given Bezier curve.

$$x = x(u) = x_0(1-u)^3 + 3x_1.u(1-u)^2 + 3x_2.u^2(1-u) + x_3.u^3 \text{ ---- (1)}$$

$$y = y(u) = y_0(1-u)^3 + 3y_1.u(1-u)^2 + 3y_2.u^2(1-u) + y_3.u^3 \text{ ---- (2)}$$

The present work treats each pattern of an iris as a Bezier curve and four control points are extracted. Hence, it is sufficient to store these four control points instead of storing the whole pattern. At the user end the patterns can be reproduced from these control points.

3.1 Obtaining the Control points

To obtain the control points from the from the Bezier equation, various methods are available. Each method is suffered by each limitation. The algorithm discussed in "An Innovative Scheme For Effectual Fingerprint Data Compression Using Bezier Curve Representations" treats in the x,y values and u value is equally distributed along the curve. Hence it doesnot generate the accurate control points ate all the caes. But this methodology suits for all the caes and generate control points for any types of curves.

In the fingerprint and iris, each and every pattern further is considered individually as a Bezier curve to find the control points. It is divided in to n equal intervals. Then the deviation of x_0, $\triangle x_0$ and x_3,$\triangle x_3$ and y_0, $\triangle y_0$ and y_3, $\triangle y_3$ is taken. From these values, the slope of the tangent at the coordinate (x_0, y_0) and (x_3, y_3) is manipulated. Using the slope and y_0 and y_3, the straight-line equations of the tangents in both the endpoints are fitted as shown in Fig.1. Then the control points x_2 and x_3 are initialized to x_0 and x_3 and varied with the step value depending on the number of divisions of the curve. These are the assumed control points for x_1 and x_2. These values are substituted in the tangent line made at the end points and the assumed y values are computed. Because the second and third control points are located at the tangent lines made at the end points of the curve [8],[9]. Now from the above equation (1) and (2), the u values are substituted 0.2 and 0.8; the following conjucate equations are obtained

$$x = x(0.2) = 0.512\,x_0 + 0.384\,x_1 + 0.096\,x_2 + 0.008\,x_3 \text{ ---- (3)}$$

$$y = y(0.2) = 0.512\,y_0 + 0.384\,y_1 + 0.096\,y_2 + 0.008\,y_3 \text{ ---- (4)}$$

$$x = x(0.8) = 0.008\,x_0 + 0.096\,x_1 + 0.384\,x_2 + 0.512\,x_3 \text{ ---- (5)}$$

$$y = y(0.8) = 0.008\,y_0 + 0.096\,y_1 + 0.384\,y_2 + 0.512\,y_3 \text{ ---- (6)}$$

In the above equations (3), (4), (5) and (6), the assumed control points along with x_0, x_3 and y_0, y_3 are substituted. Thus the assumed value at x(0.2), y(0.2), x(0.8) and y(0.8) are manipulated and checked with the original curve coordinates. If there is a match for both the points, the assumed control points are fixed as the actual control points of the curve. The above procedure can be summarized as two lemmas as below:

Lemma 1: In a Bezier curve with the starting and ending points at P_0 and P_3, the control point P_1 lie on the tangent to the curve at P_0

and the control point P_2 lies on the tangent at P_3. Proof: P_0, P_1, P_2 and P_3 form a regular quadrilateral and hence $P_0 P_1$ is a tangent at the starting point of the curve as per the construction of the Bezier curve and likewise P_3P_2 is a tangent at P_3.

Lemma 2: For a certain combination the positions of a point A moving along the tangent at the starting point and a point B moving along the tangent at the end of the given Bezier curve, the presently generated curve will exactly fit in with the given Bezier curve, provided A and B approach each other. Proof: Let for a given combination of the positions of A and B, let x' = x at u = m (0<m<1) and y' = y at u = m be computed. Likewise x'' = x at u = 1-m and y'' = y at u = 1-m be computed. Now since A and B approach each other during every iteration for every combination of their position, as per Lemma 1, there must be a Bezier curve passing through x' and x''. Let the sum of the square error defined as Δ SSE and y'a and y''a are actual y coordinates of the given curve. Then Δ SSE = (y'a - y')2 + (y''a - y'')2 ≥ ξ where ξ is the maximum error radius that is permissible during numerical evaluation around the neighbourhood of y'a and y''a and when Δ SSE = ξ the presently generated Bezier curve exactly matches with the given curve for all values of m. Hence the algorithm for numerical evaluation of the control points P1 and P2, for a set of points forming any curve on the x-y plane can be as below. The following is the algorithm to determine the control points of the Bezier Curve named a function rtocp() is based on four conditions

Condition 1: x_{as02} lies between x_{ai} and x_{ai+1}, where x_{as02} is the computed x coordinate where u=0.2, x_{ai} is the actual x coordinate in the curve at i (i varies from 1 to n-1).

Condition 2: y_{as02} lies between y_{ai} and y_{ai+1}, where y_{as02} is the computed y coordinate where u=0.2, y_{ai} is the actual y coordinate in the curve at i (i varies from 1 to n-1).

Condition 3: x_{as08} lies between x_{aj} and x_{aj+1}, where x_{as08} is the computed x coordinate where u=0.8, x_{aj} is the actual x coordinate in the curve at j (j varies from 1 to n-1).

Condition 4: y_{as08} lies between y_{aj} and y_{aj+1}, where y_{as08} is the computed y coordinate where u=0.8, yaj is the actual y coordinate in the curve at j (j varies from 1 to n-1).

The square error is computed by (yai - yas02)2+ (yaj - yas08)2. If the square error is less than 0.00001 (minimum permissible error) then the assumed control points are the actual control points.

1. Start.
2. Put the x-y coordinates in an array.
3. Find $\triangle x_0$ and $\triangle y_0$ where x_0 and y_0 is the starting point of the curve.
4. Find $\triangle x_3$ and $\triangle y_3$ where x_3 and y_3 is the ending point of the curve.
5. Find the slope m_1 of the tangent T1.
6. Find the slope m_2 of the tangent T2.
7. Obtain the equation of the tangent T1 to the curve passing through the coordinate (x_0, y_0).
8. Obtain the equation of the tangent T2 to the curve passing through the coordinate (x_3,y_3).
9. Initialize the assumed x-coordinate xc_1 from the first coordinate x_0 and vary it to the last coordinate x_3.
10. Use equation of the tangent T1 and compute the equivalent y-coordinate.
11. Initialize the assumed x-coordinate xc_2 from the last coordinate x_3 and vary it to the first coordinate x_0.
12. Use equation of the tangent T2 and compute the equivalent y-coordinate.
13. Compute x, y values corresponding to u = 0.2 and 0.8 using above mentioned equations (1), (2), (3) and (4).
14. Check whether the condition 1 satisfies or not. If condition 1 satisfies then go to step 14; otherwise go to step 18.
15. Check whether the condition 2 satisfies or not. If condition 2 satisfies then go to step 15; otherwise go to step 18.
16. Check whether the condition 3 satisfies or not. If condition 3 satisfies then go to step 16; otherwise go to step 18.
17. Check whether the condition 4 satisfies or not. If condition 4 satisfies then go to step 17; otherwise go to step 18.
18. Find the square error ξ. If ξ is less than or equal to 0.00001 then go to step 20; otherwise go to step 18.
19. Decrement xc_2 to the next value. If xc_2 equals x_0, then go to step 19; otherwise go to Step 9.
20. Increment xc_1 to the next value. If xc_2 equals x_0, then go to step 21; otherwise go to Step 20.
21. Assign (xc_1, yc_1) and (xc_2, yc_2) as the two original control points.
22. Return.

This sequence of evaluations extract the two desired control points numerically from a set of x-y values on a curve, with equally spaced x-values, instead of equally spaced u-values.

3.2 Multi-y-valued Ridges and Furrows

A portion of the patterns may have multiple values for y for the same value of x. For such a many valued curves, a ninety-degree rotation with respect to the coordinate system will make them single valued. Here just the x co-ordinates are changed into y co-ordinates and the y co-ordinates are changed into x co-ordinates provided they are stored after taking in to account this fact of rotation again implemented while storing. Fig.2 shows a sample fingerprint and figures 3,4,5 and 6 depicts the way of storing its control points.

3.3 Multi segmented Ridges and Furrows

In the case of self-folding or non-trivial curves, it will be necessary to break the furrows in to two or more simple curves, each one of which can be represented as a Bezier curve. In general, a ridge or furrow can be visualized as being composed of with many segments depending up on its complexity. The algorithm treats each segment as a Bezier curve. Fig.7 denotes the inner most ridges of an original fingerprint and it cannot be represented by a single Bezier curve. The present ridge is divided two segments at the point when both the ridges are coincide as shown in Fig.8 and Fig.9.

3.4 Combination of multi-segmented and multi-y-valued Ridges and Furrows

A furrow having many multi-y-valued portions and also the self-coiling necessitates the segmentation. This depicts in Fig.10. The ridges and furrows should be divided in to at least three segments, each segment becoming a well-behaved Bezier curve. The ridge should be divided in to at least three segments, each segment becoming a well behaved Bezier curve as shown in Fig.11 and 13. A self coiling or a closed or near-closed ridge is first split into multiple segments and each segment is treated as a Bezier curve. In case any of these segments are multi-y-valued then it is given a ninety degree rotation before extracting the control points. Such curve is shown in Fig.12.

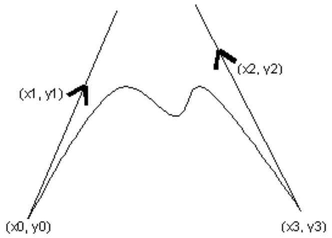

Fig.1. Sample curve along with the tangents made at the end points

3.5 Storing the Data Points

For a typical fingerprint there are about 30 to 80 Bezier curves, each of which can be represented by 4 control points that is by means of 8 coordinates, requiring 12 bytes of memory space since it is stored as a BCD of three digits. Thus the entire information content of the fingerprint can be stored with about 400 to 900 bytes. Similarly an iris pattern can have maximum of 15 to 25 Bezier curves since it can be stored only within 100 to 200 bytes. When it is needed, it can be regenerated precisely using these control points alone. Thus this near non-lossy method is able to store the fingerprint and iris information. The results obtained in compressing two fingerprint and iris images are given in Table 1.

3.6 Regeneration of the Ridges from the Control Points

The stored Bezier co-ordinates were read from the file and then it is substituted in the Bezier equation (1), (2) and the x, y co-ordinates of the curve to plot every ridge in the fingerprint and furrow and freckle of the iris.

4. Experimental Results

The experimental results of the novel and efficient scheme presented for compressing fingerprint and iris images are provided in this section. The proposed scheme is implemented using Matlab (Matlab 7.4) and C++. The ridges and furrows present in the image are first extracted by using Matlab. Then, the process of determining the Bezier control points and the reconstruction phase are performed using C++. First, the ridges in the preprocessed images are separated with their respective co-ordinate values. Subsequently, each ridge or furrow is visualized as a Bezier curve and for every curve, four control points are determined. The set of four control points represent the compressed form of an individual ridge. Consequently, using the Bezier control points, the fingerprint or iris image is reconstructed, which preserves the fine details of the original image. The results obtained from experimentation with two fingerprint images are shown in Figure 14 and 15. Each figure consists of a) the original fingerprint image, b) image constructed from the co-ordinate values of the extracted ridges, c) the reconstructed image using Bezier control points and d) Evaluation result (image (b) superimposed on image (c)). The performance of the presented scheme has been evaluated by superimposing the fingerprint image constructed using the co-ordinate values of the extracted ridges on the reconstructed fingerprint image using Bezier control points. Similarly Figure.16(a) and 17(a) represents the original irises, 16(b) and 17(b) represents the extracted furrows, freckles and rings of the same. Then the regenerated irises are shown in Figures 16(c) and 17(c). Moreover, the proposed compression scheme has achieved an exceptionally good compression ratio. The results obtained in compressing two fingerprint and two iris images are given in Table 1.

Table 1. Compression results of the proposed scheme

Image	Original JPEG image size	Compressed file size
Fingerprint 1	19.4	636 bytes
Fingerprint 2	19.2	888 bytes
Iris 1	18.2	144 bytes
Iris 2	16.7	152 bytes

5. Researches

A handful of researchers have presented approaches for the compression of fingerprint images. With fingerprint image databases, developing schemes for the compression of fingerprint images has emerged as an active and eminent research area. A brief review of some recent and significant researches is presented here.

Awad Kh. Al-Asmari [10] has implemented a progressive fingerprint image compression method by means of edge detection scheme. The image was decomposed into two components, the first component is called as the primary component which encloses the edges and the second component contains the textures and the features. An approximate of the image was reconstructed in the first stage at a bit rate of 0.0223 bpp for one Sample and 0.0245 bpp for another Sample image. The quality of the reconstructed images was competitive to the 0.75 bpp target bit set by FBI standard. The compression ratio for the algorithm is about 45:1 (0.180 bpp).

S. S. Gornale et al. [11] have highlighted different transforms of wavelet packet and their compression ratio for noisy and noiseless fingerprint images. They have also showed that the compression ratio can be increased by selecting appropriate threshold value. The compression ratios of noisy and noiseless fingerprint images are found by considering the number of zeros and the retain energy. Wavelet packet transform certainly has an effect on the Retain Energy (RE) and Number of Zeros (NZ) but the extent of it is dependent on the decomposition level, the type of image, threshold and also the type of transform used. For a maximum threshold value and greater level of decomposition, more energy can be lost, since, at higher levels of decomposition there is a higher proportion of the coefficients in the detail sub-signals. Therefore, it is always crucial to choose an optimal threshold value so as to achieve better compression and minimum loss to images.

Gulzar A. Khuwaja [12] has identified the best design parameters for a data compression scheme designed for fingerprint images. Their method focuses on reducing the transmission cost while maintaining the person's identity.

Fig.2. Original

Fig.3. Multi y-valued

Fig.4. Separated Multi y-valued curve

Fig.5. 90 degree rotated curve

Converted into Bezier Control Points (Co-Ordinates)

Bezier x Co-Ordinates are changed into y co-ordinates and vice versa and then it is stored in a

Fig.6. Reproduced curve from the changed Bezier Control

Fig.7. The example ridge for a multi segment ridge

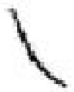

Fig.8. The divided top segment from the original

Converted into Bezier Control Points

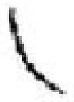

Fig.9. The divided bottom segment from the original

Converted into Bezier Control Points

Fig.10. The example ridge should be divided into three segments

Fig.11. The divided left segment

Converted into Bezier Control Points

Fig.12. The divided middle segment

Converted into Bezier Control Points

Fig.13. The divided right segment

Converted into Bezier Control Points

In choosing the wavelet packet's filters, decomposition level, and sub-bands that are better adapted to the frequency characteristics of the image, one may achieve better image representation in the sense of lower entropy or minimum distortion is considered. Empirical results proved that the selection of the best parameters has a remarkable effect on the data compression rate of fingerprint images. Statistical significance test was conducted on the experimental measures to perform the most suitable wavelet shape for fingerprint images. Image quality measures such as mean square error and peak signal-to-noise ratio are used to estimate the performance of different wavelet filters.

Song Zhao and Xiao-Fei Wang [13] have presented a compression algorithm termed as, Wavelet-Based Contourlet Transform (WBCT), for fingerprint images. It is based on wavelet transform and directional filter banks (DFBs) and can be used for efficiently approximating natural images containing contours and oscillatory patterns. To minimize frequency scrambling, a scheme based on maximally-flat filters which implements the DFBs was proposed. A quadtree sorting procedure, similar to SPIHT, is used to explicitly form classes of WBCT coefficients. The classes are encoded using arithmetic and trellis-coded quantization. The resulting encoding algorithm presents constant improvement over SPIHT performance. Simulations reveal that the new encoding algorithm gives enhanced encoding performance over SPIHT and preserves more fingerprint image details.

Kasaei. S et al. [14] have presented a vector quantization scheme based on an accurate model for the distribution of the wavelet coefficients and a compression algorithm for fingerprint images using wavelet packets and lattice vector quantization. This technique is based on the generalized Gaussian distribution. They also discussed a method for determining the largest radius of the lattice used and its scaling factor, for both uniform and piecewise-uniform pyramidal lattices. The presented algorithm aims to achieve the best rate-distortion function by adapting to the characteristics of the sub-images. In the optimization algorithm, no assumptions about the lattice parameters are made, and no training and multi-quantizing are required. They proved that the wedge region problem encountered with sharply distributed random sources was resolved in the proposed algorithm. The proposed algorithm adjusts to variability in input images and to the specified bit rates. Compared to other available image compression algorithms, the proposed algorithm results in high quality reconstructed images for identical bit rates.

6. Conclusion

In this paper, a methodology is proposed to store fingerprints and irises as a collection of Bezier control points resulting in the finite saving of the storage space. This also results in reduced time overhead to store, retrieve fingerprints and irises. The same algorithm can also be implemented to the other bio-metric features like palm-print and foot-print.


(a) (b) (c) (d)

Figure 14. (a) Original Fingerprint Image (b) Constructed Fingerprint image using co-ordinates of the ridges (c) Reconstructed fingerprint image using Bezier control points (d) Evaluation Result


(a) (b) (c) (d)

Figure 15. (a) Original Fingerprint Image (b) Constructed Fingerprint image using co-ordinates of the ridges (c) Reconstructed fingerprint image using Bezier control points (d) Evaluation Result


(a) (b) (c)

Figure 16. (a) Original Iris Image (b) Extracted Furrows (c) Reconstructed Iris

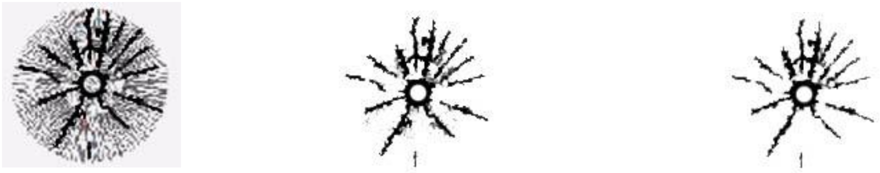
(a) (b) (c)

Figure 17. (a) Original Iris Image (b) Extracted Furrows (c) Reconstructed Iris

7. REFERENCES

[1] Yongjian Hu, Heung-Kyu Lee and Huafei Zeng, Curve Watermarking Technique for Fingerprinting Digital Maps, Proceedings of the IEEE 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Vol, Issue 15-17, pp. 223-226, 2008

[2] N. Krishnamurthy, Introduction to Computer Graphics, Tata McGraw – Hill Publishning Company Ltd., New Delhi (2006)

[3] Greenberg, S., Aladjem, M., Kogan, D., Dimitrov, I., "Fingerprint image enhancement using filtering techniques", In Proc. 15th International Conf. on Pattern Recognition III, pp. 326–329, 2000.

[4] Raymond Thai, Fingerprint Image Enhancement and Minutiae Extraction, Technical Report, The University of Western Australia, 2003.

[5] Yi Wang, Jiankun Hu and Fengling Han, "Enhanced gradient-based algorithm for the estimation of fingerprint orientation fields," Applied Mathematics and Computation, Special Issue on Intelligent Computing Theory and Methodology, Vol. 185, No. 2, pp. 823-833, February 2007.

[6] Manvjeet Kaur, Mukhwinder Singh, Akshay Girdhar, and Parvinder S. Sandhu, “Fingerprint Verification System using Minutiae Extraction Technique", in proc. of World Academy of Science, Engineering and Technology, Vol. 46, pp: 497, 2008.

[7] N. Lalithamani, K.P. Soman, "An Effective Scheme for Generating Irrevocable Cryptographic Key from Cancelable Fingerprint Templates", International Journal of Computer Science and Network Security, Vol.9, No.3, pp: 183- 193, 2009.

[8] Md. Al-Amin Bhuiyan and Hiromitsu Hama, An Accurate Method for Finding the Control Points of Bezier Curves, Proceedings of the Osaka City University, Vol. 38, pp. 175-181, 1997

[9] Md. Al-Amin Bhuiyan and Hiromitsu Hama, Identification of actors drawn in Ukiyoe Pictures, Pattern Recognition, Vol. 35, pp. 93-102, 2002

[10] Awad Kh. Al-Asmari, "Progressive Fingerprint Images Compression Using Edge Detection Technique", International Journal of Images Systems & Technology, John Wiley & Sons, Vol. 12, pp. 211 – 216, 2002.

[11] S. S. Gornale, Vikas T Humbe, R. R. Manza and K.V.Kale, "Fingerprint Image Compression using Retain Energy (RE) and Number of Zeros (NZ) through Wavelet Packet (WP)", International Journal of Computer Science and Security, Vol: 1, No:2, pp: 35-42, 2008.

[12] Gulzar A. Khuwaja, "Best parameter based compression of fingerprints with wavelet packets", International Journal of Computer Applications in Technology, Vol: 19, No: 1, pp: 51-62, 2004.

[13] Song Zhao, Xiao-Fei Wang, "Fingerprint Image Compression Based on Directional Filter Banks and TCQ ", 2009 Second International Workshop on Knowledge Discovery and Data Mining, pp.660-663, 2009.

[14] Shohreh Kasaei, Mohamed Deriche, Boualern Boash, "A Novel Fingerprint Image compression technique using Wavelet packets and Pyramid Lattice Vector Quantization", IEEE Tran. on Image Processing, Vol.11, No.12. , pp: 1365-1378, Dec-2002.

[15] Yuan Huaqiang, Ye Yangdong, Deng Jianguang, Chai Xiaoguang, Li Yong "A fingerprint feature extraction algorithm based on curvature of Bezier curve", Progress In Natural Science, Vol:17, No:11, 2007.

[16] L.Flom and A.Safir, “Iris Recognition System”, U.S. Patent 4641 349, Feb.3, 1987

[17] J.Daugman, “Recognizing persons by their Iris Patterns”, Biometrics : Personal Identification in Networked Society, 1999

[18] L. Ma, Y.Wang and T.Tan, “Iris Recognition based on Multi Channel Gabor Filtering”, Proc. Asian Conference of Computer Vision, pp.23-25, 2002

[19] [19] C.I. Tisse et al., “Person Identification Technique using Human Iris Recognition”, J.Syst. Res., Vol. 4, pp. 67-75, 2003

[20] [20] R.P.Wildes, “Automated Iris Recognition : An Emerging Biometric Technology”, Proc. IEEE, Sep.1997, Vol.85, No.9, pp. 1348-1363

[21] Daniel Schonberg and Darko Kirovski, “EyeCerts”, IEEE Transactions on Information Forensics and Security, June 2006, Vol.1, No.2,pp. 144-153

[22] Vani Perumal and Jagannathan Ramaswamy, “An Innovative Scheme For Effectual Fingerprint Data Compression Using Bezier Curve Representations”, International Journal of Computer Science & Information Security, Vol.6, No.1, October 2009, pp.149-157