An Efficient Technique For Iris Data Compression An Algorithm By Bezier Curve Approach

Piyush Tripathi[1] , Aparna Shukla[2]

GJCST Classification (FOR) E.4

***Abstract*-Iris is a strong biometric tool used for human authentication. In this methodology, the patterns in the iris such as rings, furrows and freckles can be envisaged a set of Bezier curves and hence represented by the corresponding Bezier points, resulting in considerable reduction in the file size. After the iris is captured using scanner, the patterns are extracted from the scanned image. Then they are treated as a Bezier curve and the coordinated of the characteristics four control points are determined. These set of coordinates of the control points are stored as a data file representing the iris, resulting in a considerable memory saving. Whenever the iris, is needed for recognition the retinal blood vessels can be regenerated by drawing the corresponding Bezier curves using the control points. Truthfulness of this regenerated iris is ascertained mathematically.**

Keywords-Iris; Mapping and Regeneration; Bezier curves; compression, cross correlation coefficient

I. Introduction

Iris‘ are composed before birth and, except in the event of an injury to the eyeball, remain unchanged throughout an individual‘s lifetime. It is a membrane in the eye. Iris is a biometric attribute which can be used for authenticating and distinguishing people. The pattern extracted from the iris is unique for even genetically identical twins.Iris patterns are extremely complex; carry an astonishing amount of information. 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 iris can be combined with any authentication factor and can be used as a powerful tool against repudiation.

1) *Data Reduction*

Iris recognition technology converts the visible characteristics as a phase sequence into an Iris code. Usually the size of the template is 512 bytes. A template stored, is used for future identification attempts. Here in this work, a methodology is presented, by which an iris can be stored with in a memory space of about 100 to 200 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) *Bezier Representation*

In an attempt to achieve data reduction in storing the patterns of an iris, each rings, furrows and freckles are treated as a Bezier curve. A Bezier curve is a parametric curve important in computer graphics. Bezier curves were widely used to designed automobile bodies. The curves can conventionally be represented by de Casteljau‘s algorithm. A Bezier curve is a function of four control points, of which two will be the two end points lying outside the curve. These four points completely specify the entire curve [11]. The curve can be regenerated uniquely, from the control points. Each and every pattern of the iris being treated as a Bezier curve and by using the Bezier equation, the end points and the control points are determined. Thus every pattern in the iris gives raise to four Bezier points. Thus all are represented as Bezier points. So that instead of storing the entire iris in template, only the collections of Bezier points are stored. Whenever the iris is needed, it can be regenerated as a set of Bezier curves using this set of control points.

II. 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 approximation Bezier polynomial function between P(0) and P(n).
The x co-ordinate of any point on a Bezier curve is given by

$$x\ u\ = \sum_{k=0}^{n} X_k J_{k,n}\ u\ , 0 \le u \le 1$$

where $P_k = X_k$

and similarly the y coordinate of any point is represented by

$$y\ u\ = \sum_{k=0}^{n} Y_k J_{k,n}\ u\ , 0 \le u \le 1$$

where $P_k = Y_k$

About[1]- Amity School of Engineering and TechnologyViraj Khand-5, Lucknow, UP(India) Email:-piyush.tripathi2007@gmail.com
About[2]- Birla Institute of Technology B-7, Industrial Area Post-TSL, Naini-Allahabad,UP(India)Email:-meet2aparna@gmail.com

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.

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

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

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 thewhole pattern. At the user end the patterns can be reproduced from these control points.

1) *Obtaining the control points*

In the 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 $x_0, \Delta x_0$ and $x_3, \Delta x_3$ and $y_0, \Delta y_0$ and $y_3, \Delta y_3$ 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 tangent in both the endpoints are fitted as shown in figure 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 [1,2]. Now from the above equation (1) and (2), the u values are substituted 0.2 and 0.8; the following conjucate equations are obtained

$$x = x(0.2) = 0.512x_0 + 0.384x_1$$
$$+0.096x_2 + 0.008x_3 \quad .....(3)$$
$$y = y(0.2) = 0.512y_0 + 0.384y_1$$
$$+0.096y_2 + 0.008y_3 \quad .....(4)$$

$$x = x(0.8) = 0.008x_0 + 0.096x_1$$
$$+0.384x_2 + 0.512x_3 \quad .....(5)$$
$$y = y(0.8) = 0.008y_0 + 0.096y_1$$
$$+0.384y_2 + 0.512y_3 \quad .....(6)$$

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

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. 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 ΔSSE = 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 P_1 and P_2, for a set of points forming any curve on the x-y plane can be as below:

Step 1. Put the x-y coordinates of the points on a furrow in an array

Step 2. Evaluate the equation-1 of the tangent to the curve passing through (x_0, y_0)

Step 3. Evaluate the equation-2 of the tangent to the curve passing through (x_3,y_3)

Step 4. Set the assumed x-coordinate xc_1 of the first control point at x_0

Step 5. Use equation-1, and compute the y-coordinate of the assumed first control point

Step 6. Set the assumed x-coordinate xc_2 of the second control point at x_3

Step 7. Use equation-2, and compute the y-coordinate of the assumed second control point

Step 8. Compute x, y values corresponding to u = 0.2 and 0.8 using Bezier‘s conjucate equations

Step 9. If corresponding to the actual x value of the curve equal to the computed value of the x at u=0.2, the actual y value agrees to the computed y-value with an error radius of e, then step 10 otherwise step 13

Step 10. If corresponding to the actual x value of the curve equal to the computed value of the x at u=0.8, the actual y value agrees to the computed y-value with an error radius of e, then step 11 otherwise step 13

Step 11. Assign (xc_1,yc_1) and (xc_2,yc_2) as the two original control points
Step 12. End
Step 13. Decrement xc_2 to the next value
Step 14. Go to Step 7
Step 15. Increment xc_1 to the next value
Step 16. Go to Step 5

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.

2) *Multi-y-valued 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.

3) *Multi segmented 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 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.

4) *Combination of multi-segmented and multi-y-valued*

A furrow having many multi-y-valued portions and also the self-coiling necessitates the segmentation. The furrows and freckles should be divided in to at least three segments, each segment becoming a well-behaved Bezier curve. 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.0

III. Storing the Data Points

For a typical iris pattern, as shown in Figure.2(a), there are about 20 Bezier curves, each of which can be represented by 4 control points that is by means of 8 coordinates, requiring 8 bytes of memory space since it is stored as a BCD. Thus the entire information content of the iris can be stored with about 20 X 8 = 160 bytes. When want to regenerate the fingerprint by any user or application, it can be regenerated precisely using these control points alone. Thus this near non-lossy method is able to store the iris information in about 200 bytes.

IV. 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 furrows and freckles of the iris. Samples of two irises are shown below. Figure.2(a) and 3(a) represents the original irises, 2(b) and 3(b) represents the extracted furrows, freckles and rings of the same. Then the regenerated irises are shown in Figures 2(c) and 3(c). Thus the patterns are extracted from the original iris shown in Figure 2(a). Then the patterns of the original iris as shown in Figure.2(b) is thus construed as Bezier curves as per the above discussed methodology, their corresponding control points were computed and the collection of these points are stored as the iris file. Then the regeneration of the irises are done using this file having the control points and the regenerated iris is shown in Figure 2(c).

V. Correlation of the Regenerated Iris Withoriginals

In order to evaluate to what extent these regenerated patterns truthfully represent original iris, cross correlation coefficient is evaluated. If an iris recognition algorithm is developed then to find the accuracy of the algorithm, Equal Error Rates of ID accuracy can be used. But this algorithm is for reduction of memory storage in iris and thus the cross-correlation coefficient is used. The cross-correlation coefficient is presented in Table.I.
These values suggest that the regenerated patterns agree very well with the original ones. The correlation is strongly positive as seen in the Table I. All the irises are chosen to have 130x120 pixels for the sake of uniformity. Then both the original and the regenerated files were read using Matlab6.5 and cross-correlated for verification. Depending upon the value of the cross correlation coefficient, the acceptability of the reproduced one is decided. As a general hypothesis, taking more number of ‗u' intervals yield better correlation. A complete examination should always be carried out to identify multi y-valued or multi segmented furrows and their combination, so that the deviations in the values of the cross-correlation coefficient are minimized. The Figure.4 depicts the graph, which represents the relationship between two original irises with the reconstructed irises.

VI. Related Work

The template methodology is used for representing irises. Early ideas are by Flom and Safir in the year 1987[10]. From that numerous techniques have been developed. Among that the most advanced system has been developed by Daugman[9]. His work involves a technique based on wavelet like technique. Similar technologies have been developed by performing pattern matching via Gabor filter banks[7] by combining the Hough transform for iris localization[6] and by matching multiscale iris representations[8]. Daniel and Kirovski introduced EyeCerts. It converts the iris using a modified Fourier-

Mellin transform into a standard domain where the common radial patterns are conscisely.

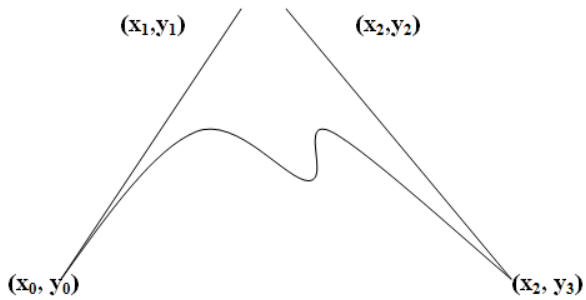

Fig 1. Sample curve along with the tangents made at the end points



Fig.2(a).Original Iris Fig.2(b). Extracted Furrows Fig.2(c). Reproduced Iris



Fig.3(a). Original Iris Fig.3(b). Extracted Furrows Fig.3(c). Reproduced Iris

Table I. The normalized cross correlation coefficients between the original and the reconstructed fingerprints

	Regenerated fingerprint from the Bezier control points of	
	Fingerprint 1 (R1)	Fingerprint 2 (R2)
Original 1	0.9961	0.3401
Original 2	0.2470	0.9999

represented[5].Basit represents eigen-irises after determining the centre of each iris and recognition is based on Euclidean distances[12]. Iris image is convolved with a blurring function which is a 2D Gaussian operator[11]. All the above algorithms require considerable memory space for storing an iris file. Similarly Bezier Curve is used in fingerprint technology for feature extraction and thus fingerprints matching [3]. For water marking the curves, the Bezier Blending function is used [4].

VII. Review of the Result

A C++ package is developed to generate Bezier control points using the (x,y) inputs of the iris furrows. In order to read the co-ordinates of each every furrow in the iris, XY-it software is used. The collections of all the Bezier control points are stored as a file. The file size is noted. A large reduction in the file size is observed. For the above-mentioned iris1, if it is stored in a JPEG format, it occupies 24.7KB. The extracted JPEG file occupies 18.5KB. When if it is stored as Bezier points, after dividing in to the necessary furrows, totally it has only 22 curves to be stored. The entire file size is exactly 176 bytes. Similarly fro the second iris, the original file size is 25.4KB in JPEG format. The extracted JPEG file occupies 18.5KB. But in Bezier representation, totally it has only 19 curves and it occupies exactly 152 bytes. The iris generation from the control points program is also written in C++. The graphics output of this particular program can be converted in to JPEG file if needed. In order to check for its truthfulness, the cross correlation coefficient is estimated. For that a program was developed using Mat Lab6.5.

VIII. Conclusion

In this paper, a methodology is proposed to store 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 and compare irises.

IX. References

1) Md. Al-Amin Bhuiyan and Hiromitsu Hama, An Accurate Method for Finding the Control Points of Bezier Curves, Proceedings of the Osaka City University, 1997, Vol. 38, pp. 175-181
2) Md. Al-Amin Bhuiyan and Hiromitsu Hama, Identification of actors drawn in Ukiyoe Pictures, Pattern Recognition, 2002, Vol. 35, pp. 93102
3) Yuvan Huaqiang, Ye Yangdong, Deng Jianguang, Chai Xiaoguang and Li Yong, A Fingerprint feature extraction algorithm based on curvature of Bezier curve, Progress in Natural Science, 2007, Vol. 17, pp. 13761381
4) Yongjian Hu, Heung-Kyu Lee and Huafei Zeng, Curve Watermarking Technique for Fingerprinting Digital Maps, Proceedings of the IEEE 2008 International Conference on ntelligent Information Hiding and Multimedia Signal Processing, 2008, Issue 15-17, pp. 223-226
5) Daniel Schonberg and Darko Kirovski, EyeCerts, IEEE Transactions on Information Forensics and Security, June 2006, Vol.1, No.2,pp. 144-153
6) C.I. Tisse et al., Person Identification Technique using Human Iris Recognition",J.Syst. Res., 2003, Vol. 4, pp. 67-75
7) L. Ma, Y.Wang and T.Tan, ―Iris Recognition based on Multi Channel Gabor Filtering", Proc. Asian Conference of Computer Vision, 2002, pp.23-25
8) R.P.Wildes, ―Automated Iris Recognition : An Emerging Biometric Technology", Proc. IEEE, Sep.1997, Vol.85, No.9, pp. 1348-1363
9) J.Daugman, ―Recognizing persons by their Iris Patterns", Biometrics : Personal Identification in Networked Society, 1999
10) L.Flom and A.Safir, ―Iris Recognition System", U.S. Patent 4641 349, Feb.3, 1987
11) Y.P.Huang, S.W.Luo and E.Y.Chen, An efficient iris recognition system", Proc.1st Int. Conf. Mach. Learning and Cybermetics, Beijing, Nov 2002, pp.4-5
12) A.Basit, M. Y. Javed and M. A. Anjum, Efficient Iris Recognition Method for Human Identification", World Academy of Science, Engineering and Technology, 2005, pp. 24-26
13) L.Ma, T.Tan, Y.Wang, D.Zhang, Efficient iris recognition by characterizing key local variations" IEEE transactions on Image processing, June 2004, vol. 13, no. 6
14) Steven Harrington, Computer Graphics-A Programming Approach, McGraw – Hill International Editions, New York (1987)
15) Amos Gilat. Mat Lab An Introduction with Applications, Wiley India Pvt. Ltd, New Delhi (2007)
16) N. Krishnamurthy, Introduction to Computer Graphics, Tata McGraw – Hill Publishning Company Ltd., New Delhi (2006)