www.ijatir.org

International Journal of Advanced Technology and Innovative Research

ISSN 2348–2370
Vol.09,Issue.12,
November-2017,
Pages:2242-2247

Pattern Recognition of Palm Print and Footprint Images for their Effective Storage Management - A Nonlinear Approach

VANI PERUMAL[1], SHIBU SAMUEL[2]

[1]Assistant Professor, Dept of IT, Rustaq College, MOHE, Sultanate of Oman, E-mail: perumal.vani@gmail.com.

[1]Lecturer, Dept of IT, Rustaq College, MOHE, Sultanate of Oman, E-mail: perumal.vani@gmail.com.

Abstract: Obviously, with the mounting applications of biometric systems, there arises a complexity in storing and retrieving those acquired biometric data. Biometric traits are widely used in human authentication and crime tracking systems. The traits includes fingerprint, iris pattern, palm print and footprint. The application areas of these traits are enormous. In all the areas, huge volume of data should be preserved. Hence it is essential to store the data in a memory efficient manner. This paper introduces a non-linear methodology to compress the file size while storing the biometric traits like palm print and footprint. The same methodology can also be used for storing fingerprint and iris pattern. It proposes a new technique to map the ridges of the biometric patterns to a set of x and y coordinates referred as control points. It is sufficient to store these control points alone in the memory instead of storing the entire biometric pattern. Whenever there is a need, the same patterns can be reconstructed from these set of stored control points. Hence the requirement of storage space is reduced considerably, also retrieves the same speedily when needed. These attributes make the methodology a very reliable tool in forensic and similar applications.

Keywords: Palm Print; Footprint; Bezier Curve; Non-Linear; Control Points; Compression.

I. INTRODUCTION

Palm print and footprints are significant physiological biometric traits. These biometric features have several advantages while using in recognition systems. In recognition systems, even low-resolution images can be employed. The line features of these prints are stable. The main issue involved in recognition systems is the storage of such patterns. While storing these patterns, an image based storage method can be deployed. These patterns can alternatively, be represented the collection of curves. A cubic Bezier curve based information storage and retrieval procedure for handling palm and footprint is described in this paper. After the biometric images are captured, the ridge patterns are extracted from the scanned image. Then these patterns are mapped to a cubic Bezier curve and the coordinates of the four control points are extracted. These control points alone are stored as a data file instead of storing the palm prints and footprints as image files. Similarly whenever needed the ridges and patterns can be regenerated using the Bezier equations.

A. Human Palm Print Patterns And Uniqueness

Palm print has proved to be one of the most unique and stable biometric characteristics [1]. Even identical twins have different palm prints [2]. It can be combined with fingerprint and can be used as a bimodal biometric authentication system. Similarly it can also be combined with any biometric trait and used as multi modal authentication system. The rich information of palm print contains distinctive features like principal lines, wrinkles (secondary lines), ridges, valleys, texture, minutiae and delta points. The principal lines that vary from palm to palm are only used for recognition. If there are similarities then the crisscrossing of the secondary lines, wrinkles are also considered for recognition [3].

B. Human Footprint

Footprints are the impressions or images left behind by a person while walking. Sometimes the print left behind at a crime, can also give vital evidence to the crime. Analysis of footprints is a special part in forensic science. Footprint is also an important biometric trait, which is used in biometric authentication [4]. It can also be combined with any other biometric trait and used as a multi modal authentication system [5] for human authentication. Since footprints are not intended to support large-scale high security applications, in electronic banking, it is used for authentication while the person enters. The person's footprint is captured and authentication is ensured while entering inside the ATM itself. This is used to employ a high level of security in ATM. To implement this type of security, a footprint sensor should be installed at the entrance just before the door that enables the person to open the door. It helps to avoid robberies and theft in ATM.

II. THE METHODOLOGY

In Bezier equations, the coordinates of the curve are generated from a set of control points and u value. Bezier curve is a parametric curve, which can be mapped to any open curve. In the biometric images, the patterns composed of curves called ridges. All the ridges fall in the category of a curve. Hence these ridges and patterns can be represented as a Bezier curve [6, 7].

A. The Inverse Problem

Extracting the control points from a Bezier curve is the primary task of the work. Because once the patterns of the biometric ridge are to be treated as a Bezier curve, the representative control points should be extracted for each and every segment. The extraction of the control points from the Bezier curve is actually the inverse problem associated with the Bezier equation. In this paper a methodology is proposed to generate the control points from the Bezier curve. This method is the enhancement of the methodology proposed by Bhuiyan and Hama [8, 9]. They described a methodology to find the control points of a cubic Bezier curve. But it work only for known perfect Bezier curve. They extracted control points from a third order Bezier curve assuming the available x, y coordinates of the curve are corresponding to equal intervals of u values ranging from zero to one. Their approach is based on two stage approximation. There are some limitations in their methodology. Here the second and third control points are assumed based on the tangent of the Bezier curve. Then using the assumed control points, the Bezier curve will be generated. Then the generated Bezier curve's $n/5^{th}$ and $n-(n/5)^{th}$ part will be compared to the original curve's $n/5^{th}$ and $n-(n/5)^{th}$ part. To find these values of the curve, the u value in the curve should be known. But in the particular part of the curve, the x, y coordinates are known and the u value is not predictable. The proposed methodology is for extracting control points to any curve when the available coordinates of the curve are equally spaced along the x-axis. So it is a case of extracting the control points, as accurately as possible, when the u values of the coordinates are not known. The methodology can be generalized for finding the control points for a general Bezier curve of n^{th} order where $n \geq 3$.

B. Formulation of the Inverse Problem

The first control point is the starting point of the Bezier curve. The fourth control point is the end point of the same curve. The second and the third control points should be extracted from the (x, y) coordinates of the Bezier curve. The second control point must be on the tangent of the curve made at the starting point of the curve. The third control point must lie on the tangent made at the end point of the curve. The first step is to find the equation of the tangent which made at the starting point of the curve. For finding the tangent, the first control point and the immediate next point with $x=x\_0+\Delta x$ where $\Delta x \rightarrow 0$, on the curve is taken and equation of the straight line which passes through these points is obtained. For finding the equation of the straight line, the difference between the x coordinate and the y coordinate is computed. The computed difference between the y coordinate is divided by the difference between the x coordinate. This gives the slope of the tangent. Then by substituting the slope of the tangent, x coordinate of the first control point, y coordinate of the first control point in the equation of the straight line, the intercept of the straight line is evaluated. After finding the slope and intercept the equation of the tangent is readily available as T1. The second step is to find the equation of the tangent at the end point of the curve.

For finding the tangent, the third control point and the immediate point which appear before curve with $x=x\_3+\Delta x$ where $\Delta x \rightarrow 0$ where is taken and equation of the straight line which passes through these points is obtained as T2. The next step is to find the second and third control points which are on the tangent. To find the second control point, the assumed x coordinate is initialized to the x coordinate of the first control point i.e. x0 and varied till the x coordinate of the last control point. Then these coordinates are substituted in the above derived equation and the assumed y coordinate is generated for the second control point. Similarly the third control point is obtained by initializing the x coordinate from the last control point's x coordinate to the first control point's x coordinate. The y coordinate of the third control point is generated by substituting these x coordinates one by one in the equation of the tangent. Now the next procedure is verifying whether the assumed control points are the actual control points or not. For verification, the coordinates of the curve are generated by using the new available set of four control points of which the first and the last are known without any ambiguity and the in-between two control points are now computed as potentially possible control points. Now the generated coordinates of the curve should be checked with the physical coordinates of the actual curve. Here comparing the coordinates of the actual curve with the coordinates of the generated curve using presently generated potentially possible control points are not possible.

Hence the present work suggest an acceptable protocol of checking with analytically generated values of (x, y) coordinates of the curve for values of u=u' and u=1-u' where u' could be a value like 0.2. Theoretically speaking this arbitrary value of u' can be any value 0<u^'<1 and u'≠0.5. If the presently generated set of (x, y) coordinates for u=u' and u=1-u' are found be real coordinates on the physical curve each x and y value with an error radius $0<\xi$ where ξ is the maximum allowable error, then the presently computed set of potentially possible control points are the real desired control points. Otherwise an algorithm shifts the potentially possible second control point along T1 and the third control point along T2. An iterative algorithm with suitable boundary conditions is implemented which extracts the control points within the limits of allowable error. A Bezier curve made up of points scattered on the x-y plane can be evaluated in terms of its geometric properties. (i) Movement of the control points around, influences the shape of the curve. (ii) The curve does pass through the two endpoints (P0 and Pn, n may take any integer greater than or equal to 2) and is tangent at the endpoints to the corresponding edge of the polygon of the control points [10]. Here these basic properties are labeled as two proved lemmas.

Lemma 1: In a Bezier curve, with the starting and ending points at P0 and P3, the control point P1 lies on the tangent to the curve at P0 and the control point P2 lies on the tangent at P3.

Proof: P0, P1, P2 and P3 form a regular quadrilateral and hence P0 P1 is a tangent at the starting point of the curve as per the construction of the Bezier curve and likewise P3 P2 is a tangent at P3.

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(0<m<1) 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 ΔSSE and y'a and y"a are actual y coordinates of the given curve. Then $\Delta SSE=(y'a-y')^2+(y''a-y'')^2 \leq \xi$ where ξ is the maximum error radius that is permissible during numerical evaluation around the neighbourhood of y'a and y"a when $\Delta SSE=\xi$ the presently generated Bezier curve matches with the given curve for all values of m.

C. Determination Of The Control Points

Let C be a Bezier Curve, and the actual control points P_0, P_1, P_2 and P_3 have the coordinates (x_0, y_0), (x_1, y_1), (x_2, y_2), and (x_3, y_3) which are to be evaluated. Here (x_0, y_0) and (x_3, y_3) are the starting and the ending points of C and hence they are known. (x_{01}, y_{01}) is assumed as an immediate next point of (x_0, y_0) on the curve C and (x_{31}, y_{31}) is assumed as an immediate previous point of (x_3, y_3) on the curve C. $\triangle x_0$, $\triangle y_0$ and $\triangle x_3$, $\triangle y_3$, as defined by $\triangle x_0 = | x_{01} - x_0 |$, $\triangle y_0 = | y_{01} - y_0 |$ and $\triangle x_3 = | x_{03} - x_3 |$, $\triangle y_3 = | y_{03} - y_3 |$, are evaluated. Now $m_1 = \triangle y_0 / \triangle x_0$, the slope of the tangent T_1 passing through the starting point (x_0, y_0) and hence the equation of T_1 is evaluated. Similarly $m_2 = \triangle y_3/\triangle x_3$, the slope of the tangent T_2 passing through the end point (x_3, y_3) and hence the equation of T_2 is evaluated. In a cubic Bezier curve with the starting and ending points at P_0 and P_3, the control point P_1 lies on the tangent T_1 to the curve at P_0 and the control point P_2 lies on the tangent at P_3. For a certain combination the positions of a point A moving along the tangent T_1 and a point B moving along the tangent T_2 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 as per Lemma 2 described above. Likewise the control points are generated so that the presently generated Bezier curve exactly matches with the given curve for all values of m.

D. Verifying the Control Points

After generating the control points in the first iteration, it should be verified to determine whether they are the real control points are not. The actual first and last control points are (x_0, y_0) and (x_3, y_3) respectively. The assumed intermediate control points are (xc_1, yc_1) and (xc_2, yc_2). For verification, the conjugate property of Bezier equation is used. The Bezier equations at the value u and at the value (1-u) are conjugate equations. For example in the Bezier equations, u if the u value is substituted as 0.2, the following equation is obtained and if it is 0.8 corresponding to (1-u), then the following equation is obtained. The two equations of x at u = 0.2 and u = 0.8, u = 0.3 and u = 0.7 and so on. In the conjugate equations, the assumed control points are substituted and the square error was calculated depending on the following conditions. $x_{\alpha 2}$ lies between $x_{\alpha\beta i}$ and $x_{\alpha\beta i+1}$, where $x_{\alpha 2}$ is the computed x coordinate where u=0.2, $x_{\alpha\beta i}$ is the actual x coordinate of the curve at i. $y_{\alpha 2}$ lies between $y_{\alpha\beta i}$ and $y_{\alpha\beta i+1}$, where $y_{\alpha 2}$ is the computed y coordinate where u=0.2, $y_{\alpha i}$ is the actual y coordinate in the curve at i, where i varies from 1 to n-1. $x_{\alpha 8}$ lies between $x_{\alpha i}$ and $x_{\alpha i+1}$, where $x_{\alpha 8}$ is the computed x coordinate where u=0.8, $x_{\alpha i}$ is the actual x coordinate in the curve at i. $y_{\alpha 8}$ lies between $y_{\alpha i}$ and $y_{\alpha i+1}$, where $y_{\alpha 8}$ is the computed y coordinate where u=0.8, $y_{\alpha i}$ is the actual y coordinate in the curve at i, where i varies from 1 to n-1. The square error is computed by $(y_{\alpha\beta i} - y_{\alpha 2})^2 + (y_{\alpha i} - y_{\alpha 8})^2$. If the square error is less than 0.00001 (minimum permissible error) then the assumed control points are the actual control points. Thus the control points are generated for the ridges of the palm print and footprint. Based on the complexity of a palm print and footprint ridge, each ridge can be divided into two or more segments. Now the next step is to map these individual segments to an unique Bezier curve so that the Bezier control points can be generated for representing that particular pattern without affecting any information. These individual segments fall under two categories.

E. Multiple Y Valued Segments

Normal curve defined in between two x values and a multiple y valued segment. The mathematical difficulty of handling a multiple y valued curves is overcome in this present work by rotating that segment by ninety degrees, so that the segment becomes a single normal curve of the first category, which can be generated using a set of four control points of a cubic Bezier curve.

F. Self-Coiling Segments

Some of the segments in the biometric patterns have the special type of curves called self-coiling segments or curves. These segments are self-folding curves. For these ridges, they can be broken into two or more different portions and each one of them can be represented as a Bezier curve. Closed curves, circles, concentric circles can all fall under this category. These curves have any of the cases or the combination there of: (i) The curves may be closed or very nearly closed curves. So no starting point labeled (x_0, y_0) and ending point labeled (x_3, y_3) distinctly possible. (ii) The assumed starting control point (x_0, y_0) and the assumed ending control point (x_3, y_3) are may be very close to each other. (iii) From one control point, more curves may start. Here one starting point and more ending points may exist. (iv) The ridges may bifurcate at one point.

III. PREPROCESSING AND THE EXPERIMENTAL RESULTS

The captured print must be preprocessed. 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 [11]. Fast Fourier Transform (FFT) is applied separately to each block of the image [11]. Binarisation increases the contrast between the ridges and valleys. The Binarisation results a binary bio metric image containing two levels of information, the foreground ridges and the background valleys [12]. After preprocessing, the patterns from the palm print are extracted, the x and y coordinates of the pattern were read. These x, y coordinates were used for extracting the control points. The next step is to map these patterns to Bezier curves and to extract the control points from these curves. There are a maximum of 20 to 30 prominent patterns in the palm including principal lines, secondary lines and ridges. All the different patterns in the palm print is visualized as a cubic Bezier curve and its Bezier control points including two end points and two control points which does not lie on the curve were evaluated using the proposed scheme. The compressed form of each and every pattern is found to be the determined set of four Bezier control points. For example, if a palm print contains 'n' different patterns, then the compressed file contains 4 x n Bezier control points. Subsequently, to visualize every pattern as a Bezier curve, the palm print can be scaled from the values 0 to 99 and the values are stored in C_r .- coordinates for set of ridges Then from C_r , the two end points are the first and the last points of the curve in C_r . Apart from these points, the two control points should also be determined. These points re determined using the proposed algorithm. The different structures of the extracted patterns present in the palm print image should be treated separately. There are three different special cases as specified above to be considered separately while mapping the patterns in the palm print.

Fig.1. A sample physical palm print.

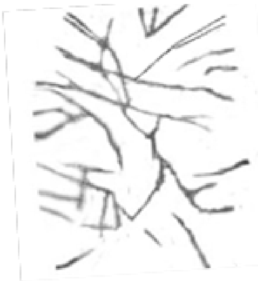

Fig.2. The input pattern of the palm print.

Fig.3. The Reconstructed pattern of the palm print from control points.

Fig.4. The Original Left & Right footprint.

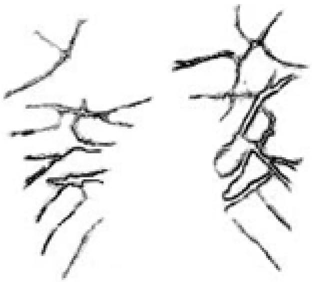

Fig. 5. Extracted Pattern of Left & Right footprint.

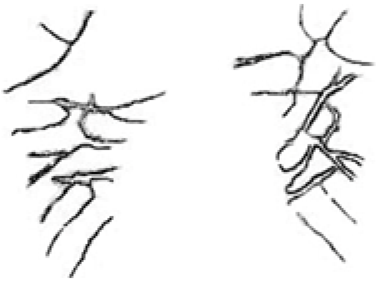

Fig. 6. Regenerated pattern of Left & Right Footprint from Control points.

Thus the control points can be extracted from each and every pattern of the palm print. These points alone can be stored instead of storing the entire palm print. Then these points and the Bezier equation can reconstruct the original palm print. Hence this results in a less-lossy or non-lossy data compression. If a ridge is a multi-segmented ridge, hence it is divided into more than one segment in order to preserve the minutiae points. The figs.1 shows a palm print, figs. 2 and 3 depict the extracted pattern from the palm print and the reconstructed patterns respectively. Hence in this methodology a compressing techniqueis introduced.

Internally all the patterns of the palm prints are stored as numbers. It occupies very less space and whenever there is a need of the palm print, it can be regenerated using the Bezier algorithm. All the Bezier control points of the patterns are stored in a data file. This file is the original palm print file in a compressed form. The control points are stored in the file. The first four control points represent one Bezier curve. The four control points are treated as (x_0, y_0), (x_1, y_1), (x_2, y_2) and (x_3, y_3). Then these points are substituted in the Bezier equation and the x co-ordinate and the y co-ordinate of the curve was computed to plot the palm print pattern. For computing the various coordinates of the palm print pattern, the u values are varied from 0 to 1. By increasing the number of the u values and by decreasing the interval, palm print pattern can be reconstructed in a smoother manner. In the same manner, the control points are extracted from footprint after preprocessing and stored for efficient utilization of memory. Fig.4 shows the two foot prints of a single person. Similarly the fig.5 depicts the extracted pattern of the same footprint. Then these extracted patterns are mapped to Bezier curves and reconstructed from the Bezier control points as exhibited in fig.6. The reproduced footprint pattern is smooth and lucid since the u value is divided into more number of equal intervals.

IV. RELATED WORK

A handful of researchers have presented approaches for representation and recognition of palm print and footprint. Xiang et al. [3] used Wavelet transform for matching two palm prints. In their paper, a novel palm print feature, named wavelet energy features are defined employing wavelet. Similarly methodology introduced by Bong [13], obtain the edge of the palm print image. Then the palm prints are matched using the AND function. Goh and Andrew [14] encode the palm print feature in bit string representation. In addition, a new scheme is presented by them to extract knuckle print feature using ridgelet transform. Struc and Pavesic [15] presented a novel palm-print feature extraction approach which deals the palm print recognition problem by employing the two-dimensional phase congruency model for line-feature extraction. Michael et al. [16] also used Wavelet transform for palm print recognition system. But they adopted wavelet transform to decompose the palm print image into lower resolution. Nirupama and Evangelia [17] developed a palm print recognition system based on three key corners. They extracted, a consistent Region of Interest (ROI) for each palm. Asmita et al. [18] used primary lines of the palm print for recognition. A fuzzy rule is constructed in which the entropy function makes the criterion function for the purpose of learning parameters. Min et al. [19] propose a ridge detection method for palm print identification using low-resolution images.

Young et al. [20] used sub-image based feature extraction and reconstruction approach for palm print recognition. Xiang et al. [21] presents a novel approach for palm print recognition based only on the valley features. This approach uses the Bothat operation to extract the valleys from a low-resolution palm image in different directions to form the valley feature. Jane et al. [22] propose a hierarchical multifeature coding scheme to facilitate coarse-to-fine matching for efficient and effective palm print verification and identification in a large database. Jaspreet et. al [23] extracted dominant spectral features have been such as the major lines instead of using the whole palm print image at a time. Sumalatha et. al [24] reviewed some important algorithm for palm print recognition in their work. They recommended CCD scanner for effective palm print recognition. Regarding footprint and representation and recognition, Kennedy [25] observed that feet have many weight bearing areas, which leave an imprint. This insight leads to the birth of footprint-based recognition that led to the development of a handful of pressure sensing devices, consisting of a matrix for footprint based measures, while some of footprint scanner are designed as sensing mats, such as the Nitta Big Mat [26], TechStorm Foot Analyzer [27] and Nitta F-Scan [28]. Andreas Uhl and Peter Wild [5] presents an approach for personal identification using footprint biometric features based on Eigen feet, local ridge characteristics and shape geometry. Khamael [29] introduced a recognition process, which utilizes the determinant value that produces the features for the footprint. Ambeth Kumar et al [30] applied Sequential modified Haar transform to the footprint image to obtain Modified Haar Energy (MHE) feature.

V. CONCLUSION

Palm print and footprint patterns are used in the recently evolved human authentication systems. The primary limitation of such authentication systems are the size of the file while storing these patterns in memory. Hence a methodology to compress the file size of these patterns is desired always. In this paper, a methodology for data compression was described. This methodology uses Bezier curves for representing such patterns. First these patterns were mapped to Bezier curves and the Bezier control points were extracted, then these control points alone were stored as BCD numbers in the computer memory. This results in the preservation of all the structures - minutiae points and the delta points in the palm print and footprint. Then these can be identified and matched after regeneration.

VI. REFERENCES

[1] Shriram D. Raut, Vikas T. Humbe, 2012. Biometric Palm Prints Feature Matching for Person Identification. I. J. Modern Education and Computer Science, 11, 61-69.

[2] A. Kong et. al., 2006. A study of identical twins palmprint for personal verification. Pattern Recognition, 11, 2149-2156.

[3] Xiang-Qian wu et. al., 2002. Wavelet based Palm Print Recognition, Proceedings of the First International Conference on Machine Learning and Cybernetics, 1253-1257.

[4] Andreas Uhl, Peter Wild, 2008. Footprint-based biometric verification, Journal of Electron and Imaging, 17, 11-16.

[5] Andreas Uhl, Peter Wild, 2007. Personal identification using Eigenfeet, Ballprint and Foot geometry biometrics, Proceedings of the First IEEE International Conference on Biometrics, 1-6.
[6] Vani Perumal, Jagannathan Ramaswamy, 2009. An Innovative Scheme For Effectual Fingerprint Data Compression Using Bezier Curve Representations. I.J. Computer Science and Information Security, 6, 149-157.
[7] Vani Perumal, Jagannathan Ramaswamy, 2010. A Novel Mathematical Algorithm to Represent and Reconstruct some Bio-Metric Patterns. I.J. Computer Applications, 11, 26-33.
[8] Md. Al-Amin Bhuiyan, Hiromitsu Hama, 2002. Identification of actors drawn in Ukiyoe Pictures, Pattern Recognition, 35, 93-102.
[9] Md. Al-Amin Bhuiyan and Hiromitsu Hama, 1997. An Accurate Method for Finding the Control Points of Bezier Curves, Proceedings of Osaka City University, 38, 175-181.
[10] M. Newman, F. Sproull, 1979. Principles of Interactive Computer Graphics, McGraw-Hill book company.
[11] Greenberg et. al., 2000. Fingerprint image enhancement using filtering techniques, Proceedings of the 15th International Conf. on Pattern Recognition III, 326-329.
[12] Raymond Thai, 2003. Fingerprint Image Enhancement and Minutiae Extraction, Technical Report, The University of Western Australia.
[13] D.B.L. Bong et. al., 2010. Palm Print Verification System, Proceedings of the World Congress on Engineering, 1, 110-113.
[14] Goh Kah Ong Michael et. al., 2010. Robust Palm Print and Knuckle Print Recognition System Using a Contactless Approach, Proceedings of the IEEE Conference on Industrial Electronics and Applications, 323- 329.
[15] V. Struc, N. Pavesic, 2009. Phase congruency features for palm-print verification, IET Journal of Signal Processing, 3, 258–268.
[16] Michael KO Goh et. al, 2006. A Fast Palm Print Verification System, Proceedings of the IEEE International Conference on Computer Graphics, Imaging and Visualisation, 306-310.
[17] Nirupama Srinivasan, Evangelia Micheli-Tzanakou, 2006. Palm Print Recognition: A New Algorithm For Corner Detection Using Palm Anatomy Features, Proceedings of IEEE International Workshop Measurement Systems, 6- 9.
[18] Asmita Rajawat et. al., 2009. Fuzzy Modeling Based Palm Print Recognition System, Proceedings of International Conference on Emerging Trends in Electronic and Photonic Devices & Systems - 189-192.
[19] Min. Jae. Park et. al, 2004. Image pre-processing methods for palm print authentication, Proceedings of the 26th Annual International Conference of the IEEE EMBS, 1495- 1496.
[20] Young-Gi Song et. al, 2007. Verification of live palm-print using sub-images reconstruction, Proceedings of the IEEE International Conference on Convergence Information Technology, 2432- 2436.
[21] Xiang-Qian wu et. al., 2005. Palmprint Recognition using Valley Features, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 4881- 4885.
[22] Jane You et. al., 2004. On Hierarchical Palmprint Coding With Multiple Features for Personal Identification in Large Databases, IEEE Transactions on Circuits and Systems for Video Technology, 14, 234 - 243.
[23] Jaspreet Kour et. al., 2013. Palmprint Recognition System, I. J. Innovative Research in Science, Engineering and Technology, 2, 1006-1009.
[24] Sumalatha K.A, Harsha H, 2014. Biometric Palmprint Recognition System: A Review, I. J. Advanced Research in Computer Science and Software Engineering, 4, 429-433.
[25] R. Kennedy, 1996. Uniqueness of bare feet and its use as a possible means of identification, Journal of Forensic Science International, 82, 81-87.
[26] K. Nakajima et. al, 2000. Footprint-based personal recognition, IEEE Transactions on Biomedical Engineering, 47, 1534-1537.
[27] J.W. Jung et. al., 2004. Dynamic Footprint-based Person Recognition Method using Hidden Markov Model and Neural Network, I. J. Intelligent Systems, 19, 1127-1141.
[28] J.W. Jung et. al., 2002. Unconstrained Person Recognition Method using Static and Dynamic Footprint, Proceedings of 18th Hungarian-Korean Seminar, 129-137.
[29] Khamael Abbas Al-Dulaimi, 2013. Using Feature Extraction for Human Footprints Recognition, I.J. Computer Applications, 64, 39-44.
[300V.D.Ambeth Kumar, M. Ramakrishnan, 2010. Footprint Recognition using Modified Sequential Haar Energy Transform (MSHET), I. J. Computer Science Issues,7,47-51.