

Score Level Fusion of Face and Finger Traits in Multimodal Biometric Authentication System

Utkarsh Gupta
 University of Pune
 P-234/II, D.I.A.T,
 Girinagar, Pune

Jasraj Fukane
 University of Pune
 43/63, Rajyog,
 Karvenagar, Pune

Varshini
 Ramanan
 University of Pune
 10/14/4, N.A.D.P,
 Nagpur, Maharashtra

Rohit Thakur
 University of Pune
 B41, H.A. Colony,
 Pimpri, Pune

ABSTRACT

In many real-world applications, unimodal biometric systems often face significant limitations due to sensitivity to noise, intra class variability, data quality, pressure, dirt, dryness and other factors. Multimodal biometric authentication systems aim to fuse two or more physical or behavioral traits to provide optimal Genuine Acceptance Rate (GAR) Vs Imposter Acceptance Rate (IAR) curve i.e. Receiver's Operating Characteristic (ROC). This paper presents a real time multimodal biometric authentication system integrating finger and face traits based on weighted score level fusion. Each biometric trait produces a varied range of scores i.e. heterogeneous scores. Various scores normalization techniques have been developed for fusion of such scores. Whereas this paper presents a technique for producing compatible scores (homogeneous). We have observed interesting variations in ROC through experimental analysis by changing the number of Eigen Faces in Face Verification Module for considering real time vibrations of input face. The statistical analysis for optimized ROC using fusion of the two traits is also represented.

General Terms

Principle Component Analysis (PCA), Eigenface, Minutiae Vector, Biometric Fusion.

Keywords

Unimodal Biometric Authentication System (UBAS), Multimodal Biometric Authentication System (MBAS), Percentage Confidence (pC) or Accuracy Score, Genuine Acceptance Rate (GAR), Imposter Acceptance Rate (IAR).

1. INTRODUCTION

A Unimodal Biometric Authentication System (UBAS) is usually more cost-efficient than a multimodal biometric system. However, it may not always be applicable in a given domain because of the limitations and problems like skin dryness, disease, data quality, pressure, dirt, oil and high IAR. In a multimodal system (MBAS) that uses different biometric traits, fusion can be done at three different levels of information, (a) Feature extraction level, (b) Matching Score (c) Decision [1]. Our proposed system is based on Matching Score level fusion.

Feature matching or input projection on template generates a score range which varies for different biometric traits. Scores are usually the number of features matched. There are two major challenges in the fusion, first is the heterogeneous nature of scores generated by different biometric traits and second is the overlapping score distribution of genuine and imposter. So, to fuse two or more traits, score normalization (numerical scaling) is performed to overcome the limitation of incompatibility of scores [2, 3]. Overlapping distribution of scores can be transformed to non-overlapping scores using

Quantile Transformation [4].

In a real time MBAS, particularly in the face module, there is a very important factor which has to be given huge consideration and that is face vibrations, both during training and authenticating which greatly affects authentication process and score generation. Our proposed system includes the theory of vibrations, and generation of percentile scores which are fusible with another percentile scores using weighted sum rule without any explicit transformation and normalization [3, 4]. Finger verification module also generates a percentile score using Crossing Number Method [5, 6]. Implementing an authentication based on weighted MBAS gives not only high efficiency and performance but also allows the administrator to adjust ratio of weights as required.

2. FACE VERIFICATION

2.1 Computational Analysis of PCA

We have used High quality 1/4 CMOS sensor- 480K pixels (Interpolated 8M pixels still image) for capturing face input. Face verification is based on the fundamental concept of 2D model i.e. Principal Component Analysis. It is a mathematical procedure that performs dimensionality reduction by extracting the principal components of the multi-dimensional data. It can be used for feature extraction from face. Real time scanning of face input involves natural face vibrations, so to consider vibrations; we take M number of training face images of the genuine person (to be authenticated). A set of M training images is taken; they are processed, represented in the form of matrices: P1, P2, P3... PM (grey scale images). Average image ψ is computed using Eq. 1 and as shown in Fig 1.

$$\psi = \frac{1}{M} \sum_{i=1}^M P_i[x][y] \quad \dots (1)$$

Where "x * y" is the number of pixels (resolution) of each image P_i . Each face differs from the average by the vector $\phi_i = P_i - \psi$, their respective values are put in an array A [N][M]. Where, N is "x * y" and M is the number of training face images. Now, $A^T [N][M]$ (matrix transpose) is calculated. A^T is then multiplied with A to obtain Covariance Matrix 'C' which is used for eigenface generation (features) [7]. For larger values of N (e.g. 14400), $A.A^T$ results in intractable computation of the covariance matrix C, as the time complexity of the multiplication algorithm will be $O(M.N^2)$.

$$\begin{bmatrix} p1 & p2 & p3 & \dots \\ px & px+1 & px+2 & \dots \\ py & py+1 & py+2 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} + \begin{bmatrix} q1 & q2 & q3 & \dots \\ qx & qx+1 & qx+2 & \dots \\ qy & qy+1 & qy+2 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} + \begin{bmatrix} r1 & r2 & r3 & \dots \\ rx & rx+1 & rx+2 & \dots \\ ry & ry+1 & ry+2 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} + \dots + \begin{bmatrix} l1 & l2 & l3 & \dots \\ lx & lx+1 & lx+2 & \dots \\ ly & ly+1 & ly+2 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} a1 & a2 & a3 & \dots \\ ax & ax+1 & ax+2 & \dots \\ ay & ay+1 & ay+2 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Fig 1: Average image matrix formation for calculating Covariance matrix. Each image is 120 x 120 pixels in resolution

The average image as shown in Fig 1 is little distorted indicating the presence of real time input vibrations in face.

Fig 2 shows an example where $M=20$ and $N = 120 * 120$ i.e. 14400 and Eq. 2 represents intractable computation. But if the number of training face images is less than the dimension of the image ($M < N$), it will be only $M-1$ rather than N . Eq. 3 shows the feasible computation where $M \times M$ is the dimension of covariance matrix C .

$$A = \begin{bmatrix} p_{11,1} & p_{11,2} & p_{11,3} & \dots & p_{11,M} \\ p_{12,1} & p_{12,2} & p_{12,3} & \dots & p_{12,M} \\ p_{13,1} & p_{13,2} & p_{13,3} & \dots & p_{13,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{rc,1} & p_{rc,2} & p_{rc,3} & \dots & p_{rc,M} \end{bmatrix} \begin{matrix} 120 \times 120 \\ \text{pixels} \\ (14400 \times 20) \end{matrix}$$

20 Training Images

$$A^T = \begin{bmatrix} p_{11,1} & p_{11,2} & p_{11,3} & \dots & p_{rc,1} \\ p_{12,1} & p_{12,2} & p_{12,3} & \dots & p_{rc,2} \\ p_{13,1} & p_{12,3} & p_{13,3} & \dots & p_{rc,3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{11,M} & p_{12,M} & p_{13,M} & \dots & p_{rc,M} \end{bmatrix} \begin{matrix} 20 \text{ Training} \\ \text{Images} \\ (20 \times 14400) \end{matrix}$$

120 x 120 pixels

Fig 2: Matrix Representation of A and A^T

$$C_{(14400 \times 14400)} = A_{(14400 \times 20)} \cdot A^T_{(20 \times 14400)} \quad \dots(2)$$

$$C_{(20 \times 20)} = A^T_{(20 \times 14400)} \cdot A_{(14400 \times 20)} \quad \dots(3)$$

2.2 Eigen Space Generation

After calculation of C , eigenvectors are computed using Eq.4 and Eq. 5

$$C.v = \lambda.v \quad \dots(4)$$

$$|C - \lambda.I| = 0 \quad \dots(5)$$

Where, v is Eigenvector (Eigen face Component) and λ is Eigen value and I is Identity Matrix. Eigenvector computation is a deterministic polynomial time algorithm which has the time complexity $O(n^3)$.

Selection of M depends on the consideration of face vibrations, we have experimentally derived that $M = 14$ gives best results. So, if $M=14$, it means that we have taken 14 images of the genuine person as training images. Number of

eigenvectors will be $M-1$ i.e. 13. These eigenvectors are also called eigenfaces (ghostly faces), and spanning of these eigenfaces generates "face space" [7, 8].

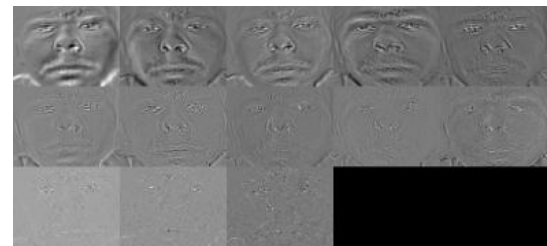


Fig 3: Eigen face space; for $M=14$ (training set) and number of eigenfaces is $M-1$

Note: Remember that, to consider real time input vibrations of face we need a large training set of continuously captured face images, and this applies while login (authentication stage) also. Experimentally we have derived that $M=14$ gives best results.

2.3 Post Training Confidence Check

This phase is carried out just after training the system i.e. after eigen space (face space) generation. This phase is extremely important because it helps in determining threshold for the genuine person.

Verification is performed by projecting a test image into the subspace spanned by the eigenfaces ("face space") [7]. Again to consider vibrations as mentioned in section 2.2, we take a sufficiently large number of test images (e.g. 50). Each test image is projected into each eigenface. In this process scalar product of the test image matrix and each eigenface is stored in a matrix which is called as ProjectedTest Matrix. Similarly ProjectedTrain Matrix is obtained by scalar products of all training images (M) with face space. Difference in the distance of test image and train images is calculated by subtracting ProjectedTrain matrix from ProjectedTest matrix. Now Euclidean distance or mahalanobis distance is calculated [1]. Least distance out of all the distances is used to calculate percentage confidence.

Now we calculate the percentage confidence / percentage accuracy using the Eq. 6

$$pC = (1 - \text{sqrt}(\text{leastDist} / M * (M-1)) / 255) * 100 \quad \dots(6)$$

pC is the percentage accuracy for a test image, leastDist is the least Euclidean/ mahalanobis distance of projected test image matrix from projected train matrices [7, 8]. Constant 255 is for the grey scale value.

So, if test images are 50 then pC for 50 test images are calculated and mean pC is stored as threshold for the person and this threshold is used in authentication (login) module.

2.4 Authentication Stage in Face Module

Authentication phase also starts by continuously capturing a sufficiently large number of face images for considering vibrations (e.g. 50). Following algorithm describes the process of authentication; it assumes some constants which are experimentally derived and yields high efficiency and best results:

1. Capture input face image
2. Process it i.e. convert to grey scale, crop and resize and then enhance (using histogram equalization or FFT (Fast Fourier Transform)).
3. Calculate scalar product of the input image matrix and eigenface space and put it in ProjectedTest matrix.
4. For loop (iTrain=0; iTrain<M)
 - a. distSq=0;
 - b. for loop (i = 0; i < M-1)
 - (i) $d_i = \text{projTest}[i] - \text{projTrainFace}[i\text{Train}*(M-1) + i]$
 - (ii) $\text{distSq} += d_i * d_i$; // Euclidean distance
 - c. if distSq < leastDistSq
 - (i) leastDistSq = distSq
5. $pC = (1 - \sqrt{(\text{leastDist} / M * (M-1)) / 255}) * 100$
6. Repeat steps from 1 to 5 for all input images i.e. 50.

We observed very interesting results, when we found mean of all pC values (50 values) taken in a zig-zag manner i.e. first mean of 1-10 pC is taken, then mean of 5-15 pC values is taken, then 10-20, then 15-25... 35-45, 40-50. In other words, while authentication, 50 input images are taken and for all 50 images, pC is calculated using above algorithm. Now 10 data sets of these 50 pC values are made i.e. (a) 1st 1-10 pC values (b) 5th-15th pC values (c) 10th- 20th pC values (d) 15th-25th and so on upto (j) 40th-50th pC values. After calculating mean percentage accuracy score for 10 data sets, we took standard deviation for each data set, and we found that if the person is genuine then deviation in scores (pC values) is very high as compared to imposter scores deviation which clearly indicates the theory of real time face vibrations i.e. WHENEVER A GENUINE PERSON GIVES LOGIN, DUE TO HIS FACE VIBRATIONS, FOR EACH INPUT IMAGE THE PERCENTAGE ACCURACY VARIES A LOT (i.e. high deviation) AND WHENEVER AN IMPOSTER GIVES LOGIN, EACH TEST IMAGE PRODUCES PERCENTAGE ACCURACY WITH VERY LESS DEVIATION (i.e. same range and very little deviation) AS IMPOSTER TEST FACES ARE GOING TO BE DIFFERENT FROM THE TEMPLATE WITH THE SAME DISTANCE NO MATTER HOW MUCH IT VIBRATES.

Computation of zig-zag mean scores and deviation of scores is elaborated in the following algorithm:

1. Initialize k = 0, counter1 = 1, sum = 0, summation = 0
2. Compute pC
3. scores[k]=(pC)
4. k = k+1
5. if counter1!=5 and counter1%5==0
 - a. for loop (z = counter1 - 10; z < counter1)
 $\text{sum} = \text{sum} + \text{scores}[z];$
 - b. $\text{mean} = \text{sum} / 10$
 - c. for loop (z = counter1 - 10; z < counter1)
 $\text{summation} += \text{Sq}(\text{scores}[z] - \text{mean})$

- d. $\text{std_d} = \text{summation} / 9$
- e. $\text{std_d} = \text{Sqrt}(\text{std_d})$
- f. $\text{sum} = 0$
- g. $\text{summation} = 0$
6. counter1 = counter1 + 1
7. For next input image repeat from 2 to 6

Where scores is an array to store pC values for all input (or test) face images, counter1 is for counting faces i.e. from 1 to 50, mean is the arithmetic mean of zig-zag data sets, std_d is the standard deviation for data set.

Fig 4 and Fig 5 represents mean and deviation results for a genuine person and an imposter.

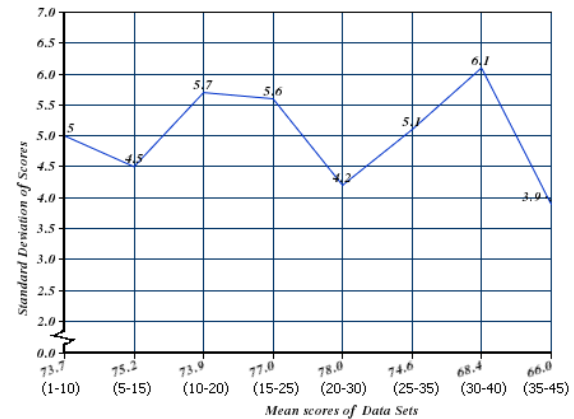


Fig 4: Genuine Person giving input

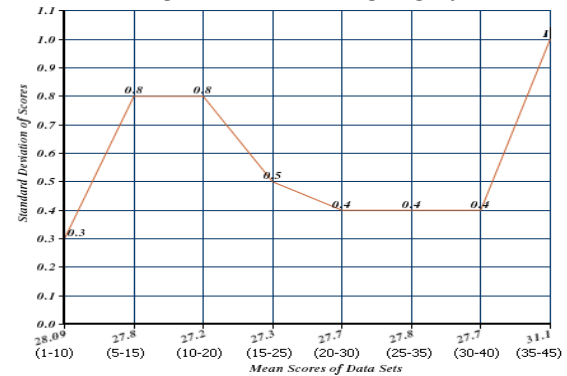


Fig 5: Imposter Person giving input

Fig 4 shows that pC value of genuine is very high as compared to the imposter and deviation of scores is also greater than imposter. So threshold obtained in post training confidence check should also include the average deviation of scores which eliminates the problem of overlapping distribution of genuine and imposter scores.

2.5 Score Statistics based on Vibrations

Theory of vibrations also yields an interesting result in the number of training images, percentage confidence, GAR and IAR.

Table 1. Affect of Number of Eigen Faces on GAR and IAR.

No. of images (M)	Threshold (% Confidence)	GAR (%)	IAR (%)
12	58	100	12
12	50	75	14
14	55	100	0
14	57	100	0
14	67	66	0
14	70	50	0
20	77	33	20
20	70	100	0
20	64	100	33
20	60	100	60

Table 2. Acceptance statistics for Number of images = 14.

Person	Confidence (Threshold) in %	GAR in %	FAR in %
Person A	57	100	0
	52	100	0
	42	100	20
	38	100	60
Person B	65	33	0
	55	66	0
	35	100	25
	30	100	50
Person C	67	66	0
	62	100	0
	58	100	20
	52	100	40
Person D	50	75	14.28
	40	100	14.28
	30	100	28.57
	25	100	57.14
Person E	57	100	0
	55	100	0
	50	100	25
	42	100	75

Table 3. Acceptance statistics for Number of images = 20.

Person	Confidence (Threshold) in %	GAR in %	FAR in %
Person A	75	25	0
	65	50	0
	60	100	40
	56	100	60
Person B	62	100	0
	58	100	0
	42	100	25
	38	100	60
Person C	77	33	20
	70	100	40
	65	100	60
	60	100	60
Person D	73	100	0
	68	100	0
	64	100	33
	60	100	66
Person E	70	100	0
	65	100	0
	60	100	60
	55	100	60

3. Fingerprint Verification

An input image is taken through optical fingerprint scanner. To improve the image quality, Fourier Transformation on the image matrix is done. Then the grey scale is converted to a binary (black and white) image using an adaptive threshold floating point value. If image is of size $N \times N$, then the binarization algorithm needs $O(N^2)$ comparison operations, which is deterministic and solvable in polynomial time. Then all ridges are made one pixel wide by removing redundant pixels [9, 10]. Image preprocessing is then followed by segmentation, now the complete image is assumed to be divided in 3×3 matrices [11]. The process of minutiae selection (marking) starts which is based on pattern identification [11]. The process starts by scanning all the pixels from 0 to N and comparing them with reference points (patterns) using crossing number method [5], as shown in Fig. 6. Seven most common minutiae types are shown in Fig. 7. A minutia is represented by $p(x, y, \theta)$ as in Fig 8.

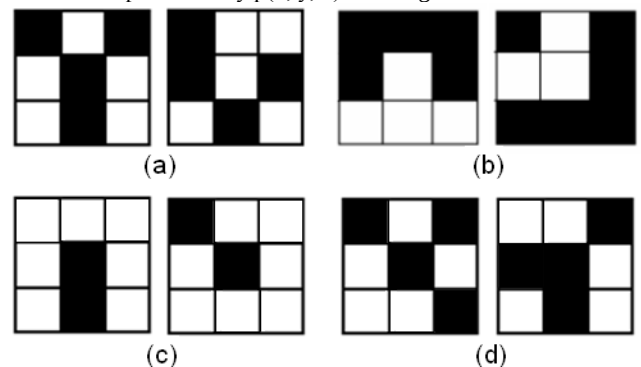


Fig 6: Most common types of minutiae (a) represents bifurcation of ridge, (b) represents center or core, (c) termination and (d) triple branch

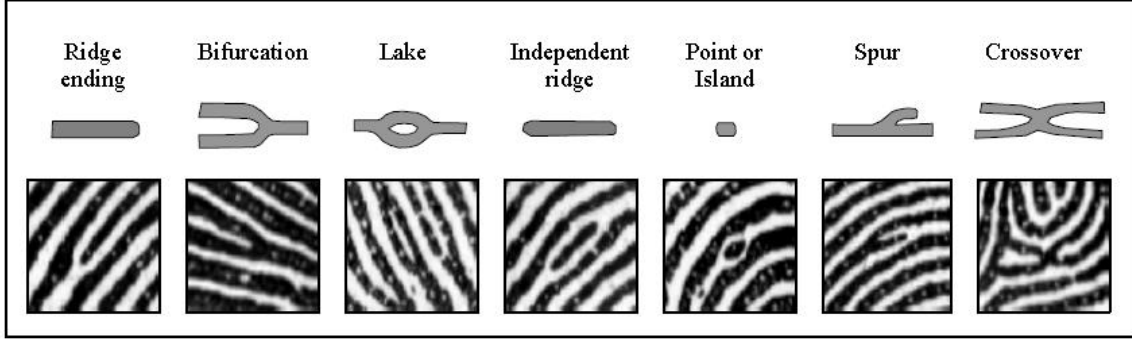


Fig. 7: Seven most common types of Minutia

Depending on the type of minutia i.e. reference type, fingerprint matching is done. All the neighbor minutiae points are determined [5] and their differences are calculated which is used to decide the matching percentage.

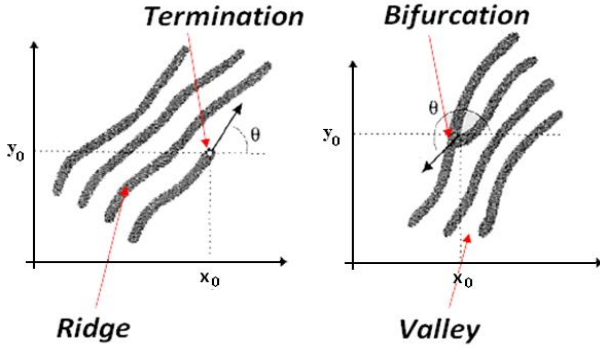


Fig. 8: Minutia Geometry

Minutiae $p(x, y, \theta)$ is converted into polar coordinates using Eq. 7.

$$\begin{bmatrix} \gamma_i \\ \phi_i \\ \alpha_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x)^2 + (y_i - y)^2} \\ \tan^{-1} \left(\frac{y_i - y}{x_i - x} \right) \\ \theta_i - \theta \end{bmatrix} \quad \dots(7)$$

Where γ_i is the polar radius, θ_i is the polar angle and α_i is the difference of direction. Eq. 7 is the general structure of minutiae Eigen vector which describes important information of the feature points.

Minutiae matching algorithm (MMA) calculates percentage accuracy by checking the number of minutiae matched depending on the comparison of difference of two eigen vectors produced in Eq. 7 and threshold values [5]. MMA includes 10 iterations for obtaining 10 accuracy scores (percentile scores).

4. Fusion of Scores

After matching of individual biometric traits, mean scores generated from them are stored in data file. For precise and accurate score fusion, our proposed methodology will take 10 sets of scores (accuracy values) i.e. $i = 10$ and they will be then fused in a weighted sum rule Eq. 8.

$$\sum_{i=1}^{10} ifacescore * (n) + ifingerscore * (1 - n) \quad \dots (8)$$

Where $ifacescore$ are confidence/accuracy (%) from the face scores file. $ifingerscore$ are matching (%) from the finger scores file. n is a floating point number less than 1. Summation of scores in Eq. 8 is then compared with a threshold of range 10^3 derived from the threshold values computed for both face and finger while training the system using same equation Eq. 8, to declare the authenticity of the person. Here the fusion system has time complexity $O(1)$, which is polynomial time deterministic algorithm.

5. Conclusion

We have conducted experiments on 100 students for testing our system. Our main aim was to increase genuine acceptance rate and decrease false acceptance rate and our experiments have shown expected results. Fig. 8 represents genuine acceptance rate (GAR) vs. false acceptance rate (FAR or IAR). Purple line represents ROC of our proposed system which clearly indicates that genuine acceptance rate is higher for MBAS. From sections 2.3, 2.4 and 4 we conclude that our proposed system does not require normalization of scores explicitly.

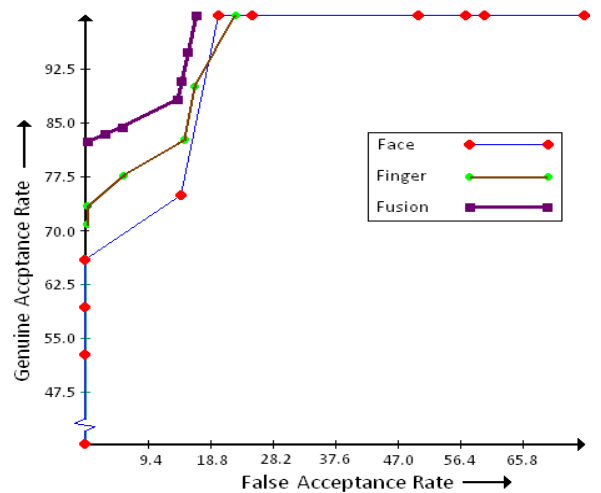


Fig 8: Receiver Operating Characteristics showing GAR Vs FAR graph plot

Face verification module takes an average latency of 3.5 seconds (authentication) and finger verification module takes an average latency of 2.8 seconds. Since these modules run parallel while authentication, the average latency of the system is 3.5 seconds.

6. ACKNOWLEDGMENTS

We would like to express our deep gratitude to Shri Anil M Datar Director Armament R&D Establishment for approving the project research work under ARDE, Pashan, Pune, Maharashtra. We would like to thank Alok Ghosh sir for his significant guidance and support. We are deeply indebted to our supervisor Prof. Y. B. Gurav whose help, stimulating suggestions and encouragement helped us in all the time of the research work.

Especially, we would like to give our special thanks to our parents for their encouragement and support.

7. REFERENCES

- [1] A. Ross and A. Jain (2003), "Information Fusion in Biometrics", *Pattern Recognition Letters* 24 (2003), pp. 2115-2125
- [2] Anil Jain, Karthik Nandakumar, Arun Ross (2005), "Score normalization in multimodal bio metric systems", *Pattern Recognition* 38 (2005) 2270 – 2285
- [3] Shi-Jinn Horng, Kevin Octavius Sentosal, Yuan-Hsin Chen (2009), "An Improved Score Level Fusion in Multimodal Biometric Systems", *2009 International Conference on Parallel and Distributed Computing, Applications and Technologies*, DOI 10.1109/PDCAT.2009.82
- [4] Jayanta Basak, Kiran Kate, Vivek Tyagi and Nalini Ratha (2010), "QPLC: A novel multimodal biometric score fusion method", *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference*, DOI: 10.1109/CVPRW.2010.5543232
- [5] Sudiro, S.A.; Painsavoine, M.; Kusuma, M; *Simple "Fingerprint Minutiae Extraction Algorithm Using Crossing Number On Valley Structure"*, *Automatic Identification Advanced Technologies, 2007 IEEE Workshop*, DOI: 10.1109/AUTOID.2007.380590
- [6] Shashi Kumar D. R., R. K. Chhotaray, K.B. Raja, Sabyasachi Pattanaik, "Fingerprint Verification based on fusion of Minutiae and Ridges using Strength Factors", *International Journal of Computer Applications*, DOI: 10.5120/799-1136
- [7] Matthew Turk, Alex Pentland (1991), "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience Volume 3, Number 1*.
- [8] Matthew A. Turk and Alex P. Pentland (1991) "Face Recognition Using Eigenfaces", *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference*, DOI: 10.1109/CVPR.1991.139758
- [9] Wang Ye-Lin, Ning Xin-Bao, Yin Yi-Long (2003). "Study on the Fingerprint Thinning Algorithm". *Journal of NanJing University (Natural Science)*. 2003, 39(4): 468-475.
- [10] Bin Fang, Huan Wen, Run-Zong Liu, Yuan-Yan Tang (2010), "A New Fingerprint Thinning Algorithm", 978-1-4244-7210-9/10/\$26.00 ©2010 IEEE
- [11] Davide Maltoni, Dario Maio, Anil K. Jain, Salil Prabhakar, "Handbook of Fingerprint Recognition (Second Edition)", ISBN: 978-1-84882-253-5
- [12] A. M. Patil, Dr. Satish R. Kolhe, Dr. Pradeep M. Patil (2009), "Face Recognition by PCA Technique", *Second International Conference on Emerging Trends in Engineering and Technology, ICETET-09*
- [13] F.A. Afsar, M. Arif and M. Hussain (2004), "Fingerprint Identification and Verification System using Minutiae Matching", *National Conference on Emerging Technologies 2004*
- [14] Ellis Horowitz, Sartaj Sahni, S Rajasekaran, "Fundamentals of Computer Algorithms"
- [15] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms (Second Edition)", ISBN 0-262-03293-7 (hc. : alk. paper, MIT Press).—ISBN 0-07-013151-1 (McGraw-Hill).