

State of the Art of finite GMM Based Biometrics Face Authentication Systems

Soltane Mohamed

*Electrical Engineering & Computing Department,
Faculty of Sciences & Technology,
DOCTOR YAHIA FARES UNIVERSITY OF MEDEA,
26000 MEDEA, ALGERIA &
Laboratoire des Systèmes Électroniques Avancées (LSEA)
soltane.mohamed@univ-medea.dz ,
soltane.mohamed.3099@gmail.com &
xor99@hotmail.com*

Abstract

This paper presents a comparative analysis of the performance of three estimation algorithms: Expectation Maximization (EM), Greedy EM Algorithm (GEM) and Figueiredo-Jain Algorithm (FJ) - based on the Gaussian mixture models (GMMs) for a Dynamic Biometrics Face Authentication Systems. An automated biometric systems for human identification measure a “signature” of the human body, compare the resulting characteristic to a database, and render an application dependent decision. A Dynamic Face From eNTERFACE 2005 Database is used and Simulation shows that finite mixture modal (GMM) is quite effective in modelling the genuine and impostor score densities. Hence, the still face information scheme based on dynamic biometrics face is robust and could be explored for identity authentication.

Keywords: *Biometric Authentication, Face Recognition, Gaussian Mixture Modal, Expectation Maximization, Greedy Expectation Maximization and Figueiredo-Jain*

1. Introduction

BIOMETRIC is a Greek composite word stemming from the synthesis of bio and metric, meaning life measurement. In this context, the science of biometrics is concerned with the accurate measurement of unique biological characteristics of an individual in order to securely identify them to a computer or other electronic system. Biological characteristics measured usually include fingerprints, voice patterns, retinal and iris scans, face patterns, and even the chemical composition of an individual's DNA [1]. Biometrics authentication (BA) (*Am I whom I claim I am?*) involves confirming or denying a person's *claimed identity* based on his/her physiological or behavioral characteristics [2]. BA is becoming an important alternative to traditional authentication methods such as keys (“something one has”, *i.e.*, by possession) or PIN numbers (“something one knows”, *i.e.*, by knowledge) because it is essentially “who one is”, *i.e.*, by biometric information. Therefore, it is not susceptible to misplacement or forgetfulness [3]. These biometric systems for personal authentication and identification are based upon physiological or behavioral features which are typically distinctive, although time varying, such as fingerprints, hand geometry, face, voice, lip movement, gait, and iris patterns. An identity verification system has to deal with two kinds of events: either the person claiming a given identity is the one who he claims to be (in which case, he is called a client), or he is not (in which case, he is called an impostor). Moreover, the system may generally take two decisions: either accept the client or reject him and decide he is an impostor. A

generic biometric authentication system [4] shown in Figure 1., given with four important modules (i) the sensor module which captures the biometric trait in the form of raw data; (ii) the feature extraction module which processes the data to extract a feature set that is compact representation of the trait; (iii) the matching module which employs a matcher or classifier to compare the extracted feature set with the templates residing in the database to generate match scores; and (iv) the decision module which uses the matching scores to either determine an identity or validate a claimed identity.

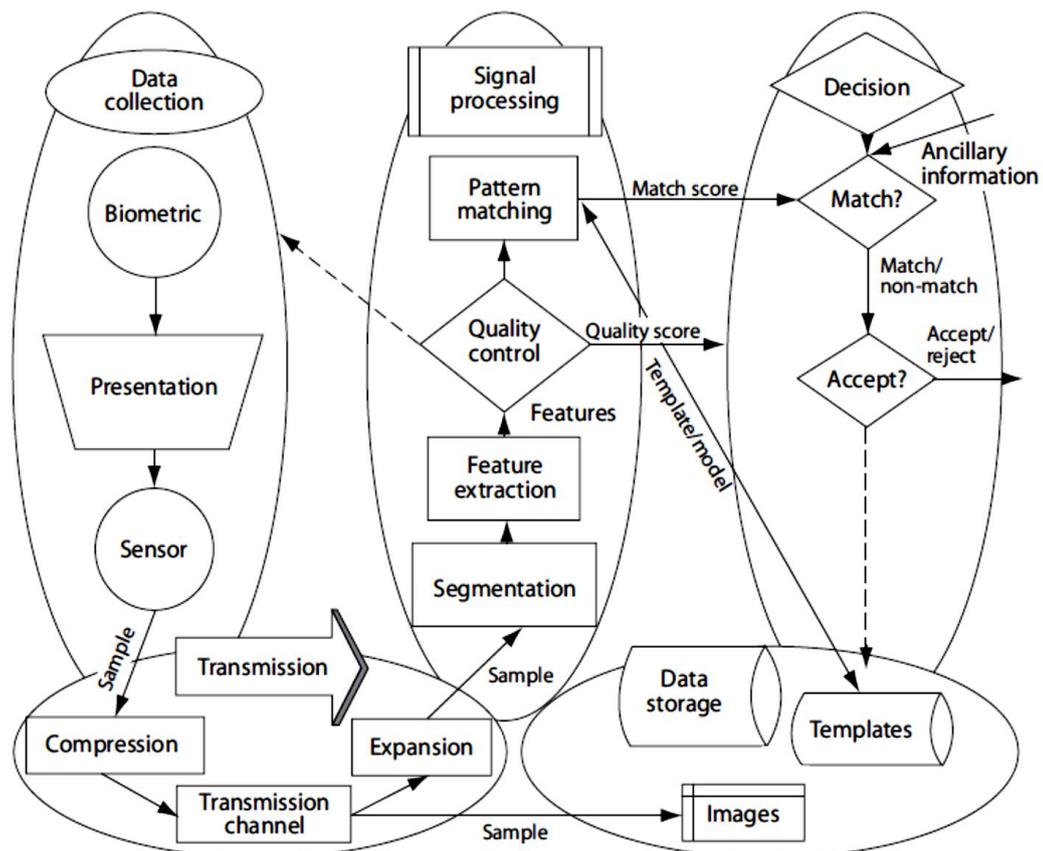


Figure 1. Generic Biometric System [4][5].

A facial recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. Some works based on biometric facial identity verification systems has been reported in literature. **Ben-Yacoub et al.** [6] evaluated five binary classifiers on combinations of faces and voice modalities (XM2VTS database). They found that (i) a support vector machine and bayesian classifier achieved almost the same performances; and (ii) both outperformed Fisher's linear discriminant, a C4.5 decision tree, and a multilayer perceptron. **Korves et al.**[7] compared various parametric techniques on the BSSR1 dataset. That study showed that the Best Linear technique performed consistently well, in sharp contrast to many alternative parametric techniques, including simple sum of z-scores, Fisher's linear discriminant analysis, and an implementation of sum of probabilities based on a normal (Gaussian) assumption. **Jain et al.** [8] propose a framework for optimal combination of match scores that is based on the likelihood ratio test. The distributions of genuine and impostor match scores are modeled as finite Gaussian mixture model. The proposed fusion approach is general in its ability to handle

(i) discrete values in biometric match score distributions, (ii) arbitrary scales and distributions of match scores, (iii) correlation between the scores of multiple matchers and (iv) sample quality of multiple biometric sources. The performance of complete likelihood ratio based fusion rule was evaluated on the three partitions of the NIST-BSSR1 database and the XM2VTS-Benchmark database. As expected, likelihood ratio based fusion leads to significant improvement in the performance compared to the best single modality on all the four databases. At a false accept rate (FAR) of 0:01%. **Jain et al. [9]** applied the sum of scores, max-score, and min-score fusion methods to normalized scores of face, fingerprint and hand geometry biometrics (database of 100 users, based on a fixed TAR). The normalized scores were obtained by using one of the following techniques: simple distance-to-similarity transformation with no change in scale (STrans), min-max, z-score, median-MAD, double sigmoid, tanh, and Parzen. They found that (a) the min-max, z-score, and tanh normalization schemes followed by a simple sum of scores outperformed other methods; (b) tanh is better than min-max and z-score when densities are unknown; and (c) optimizing the weighting of each biometric on a user-by-user basis outperforms generic weightings of biometrics. **Kittler et al. [10]** proposed a multi-modal person verification system, using three experts: frontal face, face profile, and voice. The best combination results are obtained for a simple sum rule.

2. Biometric Face Extraction and Recognition

Face recognition, authentication and identification are often confused. Face recognition is a general topic that includes both face identification and face authentication (also called verification). On one hand, face authentication is concerned with validating a claimed identity based on the image of a face, and either accepting or rejecting the identity claim (one-to-one matching). On the other hand, the goal of face identification is to identify a person based on the image of a face. This face image has to be compared with all the registered persons (one-to-many matching). Thus, the key issue in face recognition is to extract the meaningful features that characterize a human face. Hence there are two major tasks for that: Face detection and face verification.

2.1. Face Detection

Face detection is concerned with finding whether or not there are any faces in a given image (usually in gray scale) and, if present, return the image location and content of each face. This is the first step of any fully automatic system that analyzes the information contained in faces (*e.g.*, identity, gender, expression, age, race and pose). While earlier work dealt mainly with upright frontal faces, several systems have been developed that are able to detect faces fairly accurately with in-plane or out-of-plane rotations in real time. For biometric systems that use faces as non-intrusive input modules, it is imperative to locate faces in a scene before any recognition algorithm can be applied. An intelligent vision based user interface should be able to tell the attention focus of the user (*i.e.*, where the user is looking at) in order to respond accordingly. To detect facial features accurately for applications such as digital cosmetics, faces need to be located and registered first to facilitate further processing. It is evident that face detection plays an important and critical role for the success of any face processing systems.

On the results presented on this paper only size normalization of the extracted faces was used. All face images were resized to 130x150 pixels, applying a bi-cubic interpolation. After this stage, it is also developed a position correction algorithm based on detecting the eyes into the face and applying a rotation and resize to align the eyes of all pictures in the same coordinates. The face detection and segmentation tasks presented in this paper was performed based on 'Face analysis in Polar Frequency Domain' proposed by Yossi Z. *et al.* [11]. First it extract the Fourier-Bessel (FB) coefficients from the images. Next, it compute the Cartesian distance between all the Fourier-Bessel

transformation (FBT) representations and re-define each object by its distance to all other objects. Images were transformed by a FBT up to the 30th Bessel order and 6th root with angular resolution of 3° , thus obtaining to 372 coefficients. These coefficients correspond to a frequency range of up to 30 and 3 cycles/image of angular and radial frequency, respectively. Figure 2., Shows the face and eyes detections for different users from the database, and Figure 3., Shows the face normalization for the same users.

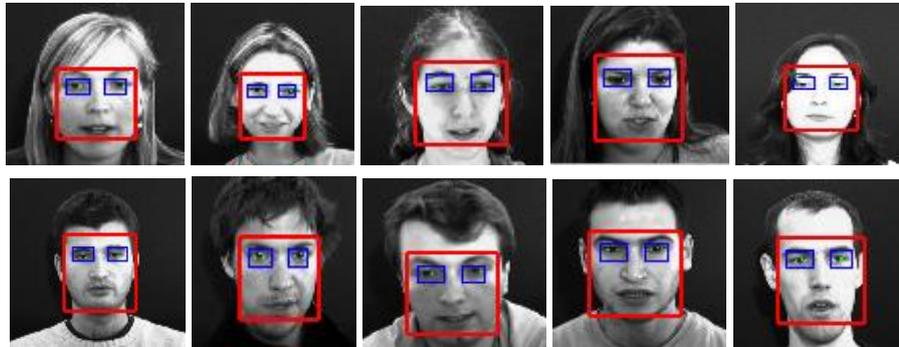


Figure 2. Face & Eyes Detections for Different Users



Figure 3. Face Normalization for the above Users

Polar Frequency Analysis

The FB series is useful to describe the radial and angular components in images [11]. FBT analysis starts by converting the coordinates of a region of interest from Cartesian (x, y) to polar (r, θ) . The $f(r, \theta)$ function is represented by the two-dimensional FB series, defined as:

$$f(r, \theta) = \sum_{i=1}^{\infty} \sum_{n=1}^{\infty} A_{n,i} J_n(\alpha_{n,i} r) \cos(n\theta) + \sum_{i=1}^{\infty} \sum_{n=1}^{\infty} B_{n,i} J_n(\alpha_{n,i} r) \sin(n\theta) \quad (1)$$

Where J_n is the Bessel function of order n , $f(R, \theta) = 0$ and $0 \leq r \leq R$. $\alpha_{n,i}$ is the i^{th} root of the J_n function, *i.e.* the zero crossing value satisfying $J_n(\alpha_{n,i}) = 0$ is the radial distance to the edge of the image. The orthogonal coefficients $A_{n,i}$ and $B_{n,i}$ are given by:

$$A_{0,i} = \frac{1}{\pi R^2 J_1^2(\alpha_{0,i})} \int_{\theta=0}^{\theta=2\pi} \int_{r=0}^{r=R} f(r, \theta) r J_0\left(\frac{\alpha_{0,i}}{R} r\right) dr d\theta \quad (2)$$

if $B_{0,i} = 0$ and $n = 0$;

$$\begin{bmatrix} A_{n,i} \\ B_{n,i} \end{bmatrix} = \frac{2}{\pi R^2 J_{n+1}^2(\alpha_{n,i})} \int_{\theta=0}^{\theta=2\pi} \int_{r=0}^{r=R} f(r, \theta) r J_n\left(\frac{\alpha_{n,i} r}{R}\right) \begin{bmatrix} \cos(n\theta) \\ \sin(n\theta) \end{bmatrix} dr d\theta \quad (3)$$

if $n > 0$.

An alternative method to polar frequency analysis is to represent images by polar Fourier transform descriptors. The polar Fourier transform is a well-known mathematical operation where, after converting the image coordinates from Cartesian to polar, as described above; a conventional Fourier transformation is applied. These descriptors are directly related to radial and angular components, but are not identical to the coefficients extracted by the FBT.

2.2 Face Verification:

Feature Extraction

The so-called ‘‘Eigen faces’’ method [12] is one of the most popular methods for face recognition. It is based on the Principal Components Analysis (PCA) of the face images in a training set. The main idea is that since all human faces share certain common characteristics, pixels in a set of face images will be highly correlated. The K-L (Karhunen-Loeve) transform can be used to project face images to a different vector space that is of reduced dimensionality where features will be uncorrelated. In the new space nearest neighbor classifiers can be used for classification. Euclidean distances d in the projection space are mapped into the [0,1] interval of the real line using the mapping function: $f = d / (1+d)$. It is easily seen that f is also a metric with distance values in [0,1]. Thus, the decomposition of a face image into an Eigen face space provides a set of features. The maximum number of features is restricted to the number of images used to compute the KL transform, although usually only the more relevant features are selected, removing the ones associated with the smallest eigenvalues. Two different approaches, database training stage and the operational stage [12]. The concept verification system is illustrated in Figure 5..

The Training Stage

Faces spaces are eigenvectors of the covariance matrix corresponding to the original face images, and since they are face-like in appearance, they are so are called Eigen faces.

Consider the training set of face images be i_1, i_2, \dots, i_m ; the average face of the set is defined as:

$$\bar{i} = \frac{1}{M} \sum_{j=1}^M i_j \quad (4)$$

Where M is the total number of images.

Each face differs from the average by the vector $\phi_n = i_n - \bar{i}$. A covariance matrix is constructed where:

$$C = \sum_{j=1}^M \phi_j \phi_j^T = AA^T \quad (5)$$

Where $A = [\phi_1 \ \phi_2 \ \dots \ \phi_M]$.

Then, the eigenvectors v_k and the eigenvalues λ_k with a symmetric matrix C are calculated. v_k Determines the linear combination of M difference images with ϕ to form the Eigen faces:

$$u_l = \sum_{k=1}^M v_{lk} \phi_k \quad l = 1, \dots, M \quad (6)$$

From these Eigen faces, $K (< M)$ Eigenfaces are selected corresponding to the K highest eigenvalues.

At the training stage, a set of normalized face images, $\{i\}$, that best describe the distribution of the raining facial images in a lower dimensional subspace (Eigen face) is computed by the following operation:

$$\omega_k = u_k(i_n - \bar{i}) \quad (7)$$

Where $n = 1, \dots, M$ and $k = 1, \dots, K$.

After that, the training facial images are projected onto the Eigen space, Ω_i , to generate representations of the facial images in Eigenface:

$$\Omega_i = (\omega_{n1}, \omega_{n2}, \dots, \omega_{nk}) \quad (8)$$

Where $n = 1, 2, \dots, M$.

The Figure 4. Show a version of a face space illustrating the four results of the projection of an image onto the face space [12].

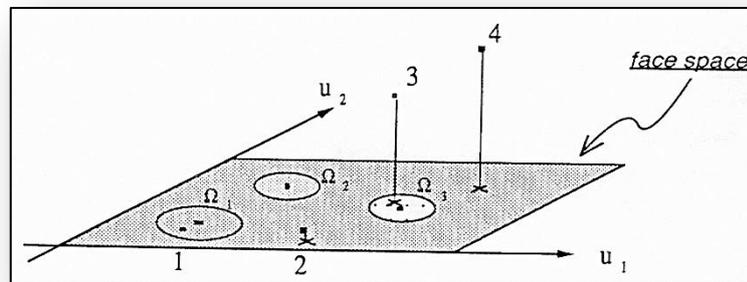


Figure 4. Simplified Version of the Face Space Illustrating the Four Results of the Projection of an Image onto the Face Space. In this Case there are Two Eigen Faces, u_1 and u_2 [12].

The Operational Stage

This approach is based on the same principles as standard PCA, explained in the training stage. The difference is that an Eigen face space is extracted for each user. Thus, when a claimant wants to verify its identity, its vectored face image is projected exclusively into the claimed user Eigen face space and the corresponding likelihood is computed. The advantage of this approach is that it allows a more accurate model of the user's most relevant information, where the first Eigen faces are directly the most representative user's face information. Another interesting point of this method is its scalability in terms of the number of users. Adding a new user or new pictures of an already registered user only requires computing or recomputed the specific Eigen face space, but not the whole dataset base as in the standard approach. For verification systems, the computation of the claimant's likelihood to be a specific user is independent on the number of users in the dataset. On the contrary, for identification systems, the number of operations increases in a proportional way with the number of users, because as many projections as different users are required. In the verification system described in this article, the independent user Eigen face approach has been chosen. Each user's Eigen face space was computed which 24 frames extracted from the database still faces.

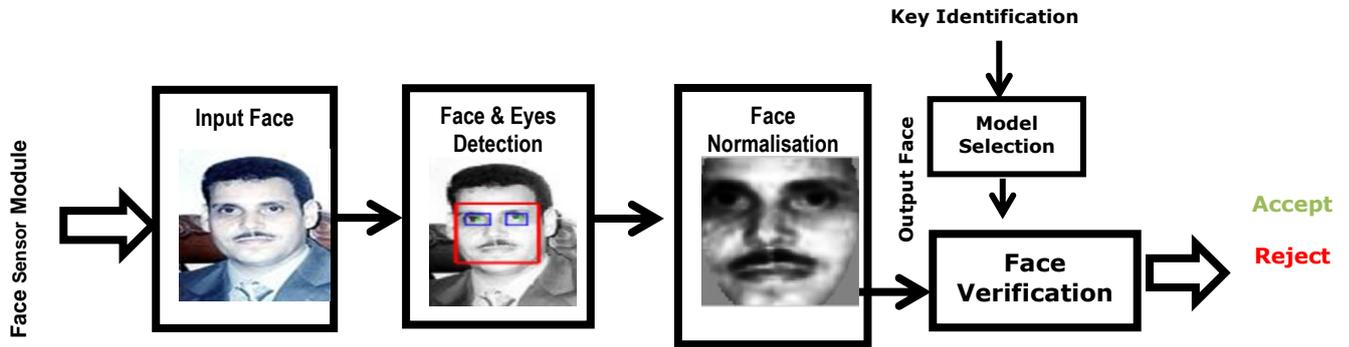


Figure 5. Face Verification Concept System.

3. Estimation of Match Score Densities

Gaussian mixture model (GMM) has been successfully used to estimate arbitrary densities and it is used for estimating the genuine and impostor score densities [13][14].

Let $\Phi^K(x; \mu, \Sigma)$ be the K-variate Gaussian density with mean vector μ and covariance matrix Σ , i.e.,

$$\Phi^K(x; \mu, \Sigma) = (2\pi)^{-K/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right).$$

The estimates of $f_{gen}(x)$ and $f_{imp}(x)$ are obtained as a mixture of Gaussians as follows.

$$\hat{f}_{gen}(x) = \sum_{j=1}^{M_{gen}} P_{gen,j} \Phi^K(x; \mu_{gen,j}, \Sigma_{gen,j}) \quad (9)$$

$$\hat{f}_{imp}(x) = \sum_{j=1}^{M_{imp}} P_{imp,j} \Phi^K(x; \mu_{imp,j}, \Sigma_{imp,j}) \quad (10)$$

Where M_{gen} (M_{imp}) is the number of mixture components used to model the density of the genuine (impostor) scores, $p_{gen,j}$ ($p_{imp,j}$) is the weight assigned to the j^{th} mixture component in $\hat{f}_{imp}(x)$ ($\hat{f}_{gen}(x)$), $\sum_{j=1}^{M_{gen}} P_{gen,j} = \sum_{j=1}^{M_{imp}} P_{imp,j} = 1$. Selecting the appropriate number of components is one of the most challenging issues in mixture density estimation; while a mixture with too many components may result in over-fitting, a mixture with too few components may not approximate the true density well. The GMM fitting algorithm automatically estimates the number of components and the component parameters using an EM, GEM and FJ algorithms and the minimum message length criterion [13,14].

Maximum Likelihood Parameter Estimation

Given a set of observation data in a matrix X and a set of observation parameters θ the ML parameter estimation aims at maximizing the likelihood $L(\theta)$ or log likelihood of the observation data $X = \{X_1, \dots, X_n\}$

$$\hat{\theta} = \arg \max_{\theta} L(\theta). \quad (11)$$

Assuming that it has independent, identically distributed data, it can write the above equations as:

$$L(\theta) = p(X|\theta) = p(X_1, \dots, X_n|\theta) = \prod_{i=1}^n p(X_i|\theta). \quad (12)$$

The maximum for this function can be find by taking the derivative and set it equal to zero, assuming an analytical function.

$$\frac{\partial}{\partial \theta} L(\theta) = 0. \quad (13)$$

The incomplete-data log-likelihood of the data for the mixture model is given by:

$$L(\theta) = \log p(X|\theta) = \sum_{i=1}^N \log p(x_i|\theta) \quad (14)$$

Which is difficult to optimize because it contains the log of the sum. If it considers X as incomplete, however, and posits the existence of unobserved data items $Y = \{y_i\}_{i=1}^N$ whose values inform us which component density generated each data item, the likelihood expression is significantly simplified. That is, it assume that $y_i \in \{1 \dots K\}$ for each i , and $y_i = k$ if the i -th sample was generated by the k -th mixture component. If it knows the values of Y , it obtains the complete-data log-likelihood, given by:

$$L(\theta, Y) = \log p(X, Y|\theta) \quad (15)$$

$$= \sum_{i=1}^N \log p(x_i, y_i|\theta) \quad (16)$$

$$= \sum_{i=1}^N \log(p(y_i|\theta)p(x_i|y_i, \theta)) \quad (17)$$

$$= \sum_{i=1}^N (\log p_{y_i} + \log g(x_i|\mu_{y_i}, \Sigma_{y_i})) \quad (18)$$

Which, given a particular form of the component densities, can be optimized using a variety of techniques [23].

EM Algorithm

The expectation-maximization (EM) algorithm [14][15][16][17] is a procedure for maximum-likelihood (ML) estimation in the cases where a closed form expression for the optimal parameters is hard to obtain. This iterative algorithm guarantees the monotonic increase in the likelihood L when the algorithm is run on the same training database.

The probability density of the Gaussian mixture of k components in \mathbb{R}^d can be described as follows:

$$\Phi(x) = \sum_{i=1}^k \pi_i \phi(x|\theta_i) \quad \forall x \in \mathbb{R}^d, \quad (19)$$

Where $\phi(x|\theta_i)$ is a Gaussian probability density with the parameters $\theta_i = (m_i, \Sigma_i)$, m_i is the mean vector and Σ_i is the covariance matrix which is assumed positive definite given by:

$$\phi(x|\theta_i) = \phi(x|m_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-m_i)^T \Sigma_i^{-1}(x-m_i)}, \quad (20)$$

And $\pi_i \in [0, 1]$ ($i = 1, 2, \dots, k$) are the mixing proportions under the constraint $\sum_{i=1}^k \pi_i = 1$. if it encapsulates all the parameters into one vector: $\theta_k = (\pi_1, \pi_2, \dots, \pi_k, \theta_1, \theta_2, \dots, \theta_k)$, then, according to Eq. (18), the density of Gaussian mixture can be rewritten as:

$$\Phi(x|\theta_k) = \sum_{i=1}^k \pi_i \phi(x|\theta_i) = \sum_{i=1}^k \pi_i \phi(x|m_i, \Sigma_i). \quad (21)$$

For the Gaussian mixture modeling, there are many learning algorithms. But the EM algorithm may be the most well-known one. By alternatively implementing the E-step to estimate the probability distribution of the unobservable random variable and the M-step to increase the log-likelihood function, the EM algorithm can finally lead to a local maximum of the log-likelihood function of the model. For the Gaussian mixture model, given a sample data set $S = \{x_1, x_2, \dots, x_N\}$ as a special incomplete data set, the log-likelihood function can be expressed as follows:

$$\log p(S|\theta_k) = \log \prod_{t=1}^N \phi(x_t|\theta_k) = \sum_{t=1}^N \log \sum_{i=1}^k \pi_i \phi(x_t|\theta_i), \quad (22)$$

This can be optimized iteratively via the EM algorithm as follows:

$$P(j|x_t) = \frac{\pi_j \phi(x_t|\theta_j)}{\sum_{i=1}^k \pi_i \phi(x_t|\theta_i)}, \quad (23)$$

$$\pi_j^+ = \frac{1}{N} \sum_{t=1}^N P(j|x_t), \quad (24)$$

$$\mu_j^+ = \frac{1}{\sum_{t=1}^N P(j|x_t)} \sum_{t=1}^N P(j|x_t) x_t, \quad (25)$$

$$\Sigma_j^+ = \frac{1}{\sum_{t=1}^N P(j|x_t)} \sum_{t=1}^N P(j|x_t) (x_t - \mu_j^+)(x_t - \mu_j^+)^T. \quad (26)$$

Although the EM algorithm can have some good convergence properties in certain situations, it certainly has no ability to determine the proper number of the components for a sample data set because it is based on the maximization of the likelihood.

Greedy EM Algorithm

The greedy algorithm (GEM) [15-18] starts with a single component and then adds components into the mixture one by one. The optimal starting component for a Gaussian mixture is trivially computed, optimal meaning the highest training data likelihood. The algorithm repeats two steps: insert a component into the mixture, and run EM until convergence. Inserting a component that increases the likelihood the most is thought to be an easier problem than initializing a whole near-optimal distribution. Component insertion involves searching for the parameters for only one component at a time. Recall that EM finds a local optimum for the distribution parameters, not necessarily the global optimum which makes it initialization dependent method. Given p_c a C-component Gaussian mixture with parameters θ_c . The general greedy algorithm for Gaussian mixture is as follows:

1. Compute the optimal (in the ML sense) one-component mixture p_1 and set $C \leftarrow 1$.
2. Find a new component $\mathcal{N}(x; \mu', \Sigma')$ and corresponding mixing weight α' that increase the likelihood the most:

$$\{\mu', \Sigma', \alpha'\} = \arg \max_{\{\mu, \Sigma, \alpha\}} \sum_{i=1}^N \ln[(1 - \alpha)p_c(x_i) + \alpha \mathcal{N}(x_i; \mu, \Sigma)] \quad (27)$$

while keeping p_c fixed.

3. Set $p_{C+1}(x) \leftarrow (1 - \alpha')p_c(x) + \alpha' \mathcal{N}(x; \mu', \Sigma')$ and then $C \leftarrow C + 1$.
4. Update p_c using **EM** (or more other method) until convergence.
5. Evaluate some stopping criterion; go to step 2 or quit.

The stopping criterion in Step 5 can be for example any kind of model selection criterion, wanted number of components, or the minimum message length criterion. The crucial point is of course Step 2. Finding the optimal new component requires a global search, which is performed by creating CN_{cand} candidate components. The number of candidates will increase linearly with the number of components C , having N_{cand} candidates per each existing component. The candidate resulting in the highest likelihood when inserted into the (previous) mixture is selected. The parameters and weight of the best candidate are then used in Step 3 instead of the truly optimal values.

The candidates for executing Step 2 are initialized as follows: the training data set X is partitioned into C disjoint data sets $\{A_c\}, c = 1 \dots C$, according to the posterior probabilities of individual components; the data set is Bayesian classified by the mixture components. From each A_c number of N_{cand} candidates are initialized by picking uniformly randomly two data points x_l and x_r in A_c . The set A_c is then partitioned into two using the smallest distance selection with respect to x_l and x_r . The mean and covariance of these two new subsets are the parameters for two new candidates. The candidate weights are set to half of the weight of the component that produced the set A_c . Then new x_l and x_r are drawn until N_{cand} candidates are initialized with A_c . The partial **EM** algorithm is then used on each of the candidates. The partial **EM** differs from the **EM** and **CEM** algorithms by optimizing (updating) only one component of a mixture; it does not change any other components. In order to reduce the time complexity of the algorithm a lower bound on the log-likelihood is used instead of the true log-likelihood. The lower-bound log-likelihood is calculated with only the points in the respective set A_c . The partial **EM** update equations are as follows:

$$w_{i,C+1} = \frac{\alpha N(x_i, \mu, \Sigma)}{(1-\alpha)p_C(x) + \alpha N(x_i, \mu, \Sigma)} \quad (28)$$

$$\alpha = \frac{1}{N(A_c)} \sum_{i \in A_c} w_{i,C+1} \quad (29)$$

$$\mu = \frac{\sum_{i \in A_c} w_{i,C+1} x_i}{\sum_{i \in A_c} w_{i,C+1}} \quad (30)$$

$$\Sigma = \frac{\sum_{i \in A_c} w_{i,C+1} (x_i - \mu)(x_i - \mu)^T}{\sum_{i \in A_c} w_{i,C+1}} \quad (31)$$

Where $N(A_c)$ is the number of training samples in the set A_c . These equations are much like the basic **EM** update equations in Eqs. (16) - (18). The partial **EM** iterations are stopped when the relative change in log-likelihood of the resulting $C + 1$ -component mixture drops below threshold or maximum number of iterations is reached. When the partial **EM** has converged the candidate is ready to be evaluated.

Figueiredo-Jain Algorithm

The Figueiredo-Jain (FJ) [15][16][17][18] algorithm tries to overcome three major weaknesses of the basic EM algorithm. The EM algorithm presented previous section requires the user to set the number of components and the number will be fixed during the estimation process. The FJ algorithm adjusts the number of components during estimation by annihilating components that are not supported by the data. This leads to the other EM failure point, the boundary of the parameter space. FJ avoids the boundary when it annihilates components that are becoming singular. FJ also allows starting with an arbitrarily large number of components, which tackles the initialization issue with the EM algorithm. The initial guesses for component means can be distributed into the whole

space occupied by training samples, even setting one component for every single training sample.

The classical way to select the number of mixture components is to adopt the "model-class/model" hierarchy, where some candidate models (mixture pdf's) are computed for each model-class (number of components), and then select the "best" model. The idea behind the FJ algorithm is to abandon such hierarchy and to find the "best" overall model directly. Using the minimum message length criterion and applying it to mixture models leads to the objective function:

$$\Lambda(\theta, X) = \frac{V}{2} \sum_{c: \alpha_c > 0} \ln \left(\frac{N\alpha_c}{12} \right) + \frac{C_{nz}}{2} \ln \frac{N}{12} + \frac{C_{nz}(V+1)}{2} - \ln \mathcal{L}(X, \theta) \quad (32)$$

Where N is the number of training points, V is the number of free parameters specifying a component, and C_{nz} is the number of components with nonzero weight in the mixture ($\alpha_c > 0$). the last term $\ln \mathcal{L}(X, \theta)$ is the log-likelihood of the training data given the distribution parameters Equation 18.

The EM algorithm can be used to minimize Equation 32, with a fixed C_{nz} . It leads to the M-step with component weight updating formula:

$$\alpha_c^{i+1} = \frac{\max \left\{ 0, \left(\sum_{n=1}^N w_{n,c} \right) - \frac{V}{2} \right\}}{\sum_{j=1}^C \max \left\{ 0, \left(\sum_{n=1}^N w_{n,c} \right) - \frac{V}{2} \right\}} \quad (33)$$

This formula contains an explicit rule of annihilating components by setting their weights to zero.

The above M-steps are not suitable for the basic EM algorithm though. When initial C is high, it can happen that all weights become zero because none of the components have enough support from the data. Therefore a component-wise EM algorithm (CEM) is adopted. CEM updates the components one by one, computing the E-step (updating W) after each component update, where the basic EM updates all components "simultaneously". When a component is annihilated its probability mass is immediately redistributed strengthening the remaining components.

When CEM converges, it is not guaranteed that the minimum of $\Lambda(\theta, X)$ is found, because the annihilation rule (Eq. 33) does not take into account the decrease caused by decreasing C_{nz} . After convergence the component with the smallest weight is removed and the CEM is run again, repeating until $C_{nz} = 1$. Then the estimate with the smallest $\Lambda(\theta, X)$ is chosen. The implementation of the FJ algorithm uses a modified cost function instead of $\Lambda(\theta, X)$.

$$\Lambda'(\theta, X) = \frac{V}{2} \sum_{c: \alpha_c > 0} \ln \alpha_c + \frac{C_{nz}(V+1)}{2} \ln N - \ln \mathcal{L}(X, \theta). \quad (34)$$

4. Experiments and Results

The experiments were performed using still face database extracted from video, which is encoded in raw UYVY. AVI 640 x 480. Uncompressed PNG files are extracted from the video files for feeding the face detection algorithms. The database obtained from eINTERFACE 2005 [19]. Thirty subjects were used for the experiments in which twenty-five are males and five are females. For face experts, ninety-six face images from a subject were randomly selected to be trained and projected into Eigen space, and the other twenty-four samples were used for the subsequent validation and testing. Three sessions of the face database were used separately. Session one was used for training the face experts. Each expert used ten mixture client models. To find the performance, Sessions two and three were used for obtaining expert opinions of known impostor and true claims.

Performance Criteria

The basic error measure of a verification system is false rejection rate (FRR) and false acceptance rate (FAR) as defined in the following equations:

False Rejection Rate (FRR_i) is an average of number of falsely rejected transactions. If n is a transaction and $x(n)$ is the verification result where 1 is falsely rejected and 0 is accepted and N is the total number of transactions then the personal False Rejection Rate for user i is

$$FRR_i = \frac{1}{N} \sum_{n=1}^N x(n) \quad (35)$$

False Acceptance rate (FAR_i) is an average of number of falsely accepted transactions. If n is a transaction and $x(n)$ is the verification result where 1 is a falsely accepted transaction and 0 is genuinely accepted transaction and N is the total number of transactions then the personal False Acceptance Rate for user i is

$$FAR_i = \frac{1}{N} \sum_{n=1}^N x(n) \quad (36)$$

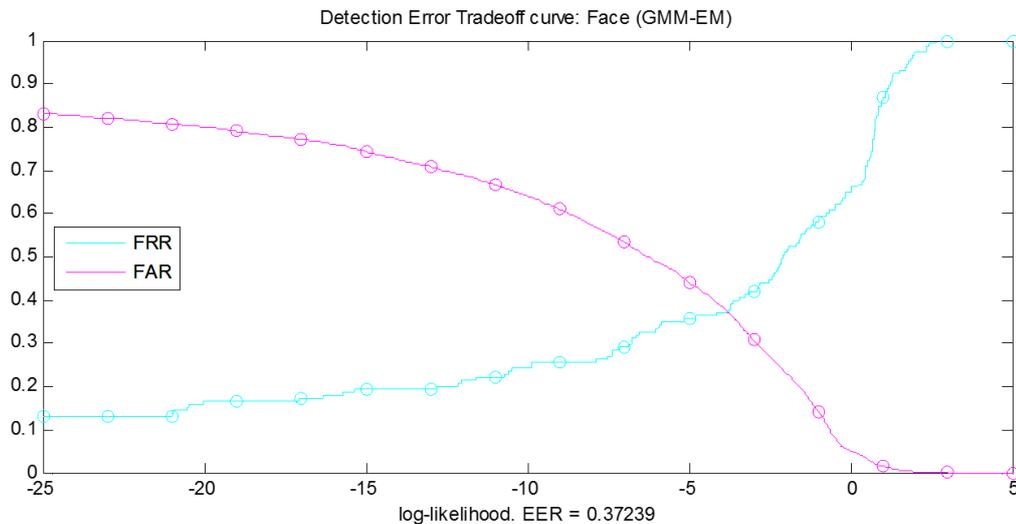
Both FRR_i and FAR_i are usually calculated as averages over an entire population in a test. If P is the size of populations then these averages are

$$FRR = \frac{1}{P} \sum_i FRR_i \quad (37)$$

$$FAR = \frac{1}{P} \sum_i FAR_i \quad (38)$$

Equal Error Rate (EER), is an intersection where FAR and FRR are equal at an optimal threshold value. This threshold value shows where the system performs at its best.

As a common starting point, classifier parameters were selected to obtain performance as close as possible to EER on clean test data (following the standard practice in the face verification area of using EER as a measure of expected performance). A good decision is to choose the decision threshold such as the false accept equal to the false reject rate. In this paper it uses the Detection Error Tradeoff (DET) curve to visualize and compare the performance of the system see Figure 6..



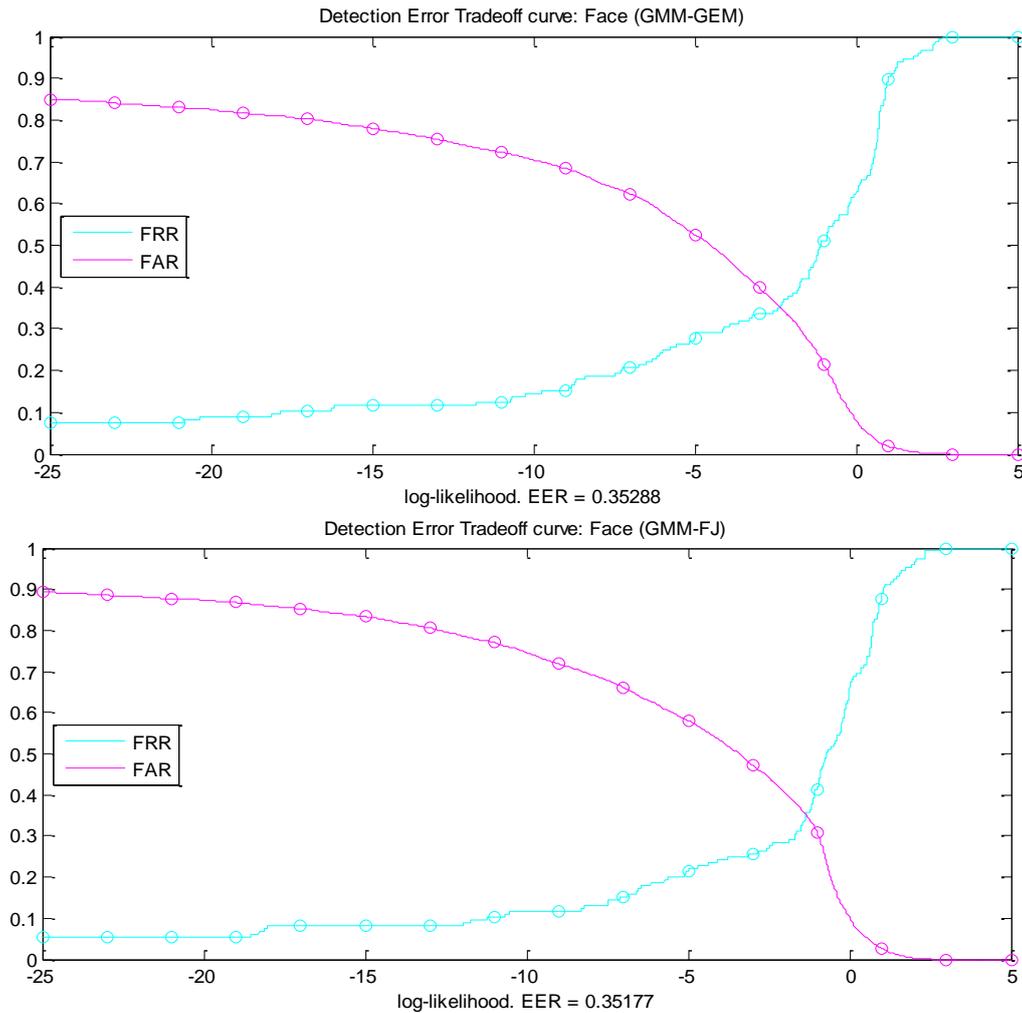


Figure 6. Detection Error Tradeoff Curves

5. Conclusions

The paper has presented a human authentication method of dynamic biometrics face information. Simulation results show that state-of-the-art finite mixture modal (GMM) is quite effective in modeling the genuine and impostor score densities. The (EM), (GEM) and (FJ) estimation algorithms achieve a significant performance rates, the log-likelihood of EER = 0.37239 for (GMM-EM), log-likelihood of EER = 0.35288 for (GMM-GEM) and the log-likelihood of EER = 0.35177 for (GMM-FJ). Hence, the still face information scheme based on dynamic biometrics face is robust and could be explored for identity authentication.

References

- [1] S. Gleni and P. Petratos, "DNA Smart Card for Financial Transactions", The ACM Student Magazine (2004), <http://www.acm.org>
- [2] G. Chetty and M. Wagner, "Audio-Visual Multimodal Fusion for Biometric Person Authentication and Liveness Verification", Australian Computer Society, Inc. (2006), This paper appeared at the NICTA-HCSNet Multimodal User Interaction Workshop (MMUI2005), Sydney, Australia.
- [3] N. Poh and S. Bengio, "Database, Protocol and Tools for Evaluating Score-Level Fusion Algorithms in Biometric Authentication", IDIAP RR 04-44, August (2004), a IDIAP, CP 592, 1920 Martigny, Switzerland.
- [4] J. Wayman, A. Jain, D. Maltoni and D. Maio, "An Introduction to Biometric Authentication Systems",

- Biometric Systems Technology, Design and Performance Evaluation Wayman, (2005), <http://www.springer.com/978-1-85233-596-0>
- [5] J. Duane M. Blackburn, "BIOMETRICS 101" Version 3.1, Federal Bureau of Investigation. March (2004). <http://www.biometriccatalog.org>
- [6] S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz, "Fusion of Face and Speech Data for Person Identity Verification", IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 10, NO. 5, SEPTEMBER (1999).
- [7] H. J. Korves, L. D. Nadel, B. T. Ulery, D. M. Bevilacqua Masi, "Multi-biometric Fusion: From Research to Operations", MTS MitreTek Systems, sigma summer (2005), pp. 39-48, <http://www.mitretek.org/home.nsf/Publications/SigmaSummer2005>
- [8] K. Nandakumar, Y. Chen, S. C. Dass and Anil K. Jain, "Biometric Score Fusion: Likelihood Ratio, Matcher Correlation and Image Quality", IEEE Transactions on Pattern Analysis and Machine Intelligence, (2007).
- [9] A. Jain, K. Nandakumar, A. Ross, "Score normalization in multimodal biometric systems", THE JOURNAL OF PATTERN RECOGNITION SOCIETY, ELSEVIER (2005).
- [10] Kittler, J., Hatef, M., Duin, R. P. W. and Matas, J., "On combining classifiers", IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3): 226–239. (1998).
- [11] Y. Zana, Roberto M. Cesar-Jr, Rogerio S. Feris and M. Turk, "Face Verification in Polar Frequency Domain: A Biologically Motivated Approach", G. Bebis *et al.* (Eds.): ISVC 2005, LNCS 3804, pp. 183–190, 2005. C. Springer-Verlag Berlin Heidelberg (2005).
- [12] M. Turk and A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, vol.3, no. 1, pp. 71-86, (1991).
- [13] Pekka Paalanen, "Bayesian classification using gaussian mixture model and EM estimation: implementation and comparisons", Information Technology Project, 2004, Lappeenranta, June 23, (2004) <http://www.it.lut.fi/project/gmmbayes/>
- [14] P. Paalanen, J.-K. Kamarainen, J. Ilonen, H. Kälviäinen, "Feature Representation and Discrimination Based on Gaussian Mixture Model Probability Densities: Practices and Algorithms", Department of Information Technology, Lappeenranta University of Technology, P.O.Box 20, FI-53851 Lappeenranta, Finland (2005).
- [15] K. Veeramachaneni, L. Ann Osadciw, and P. K. Varshney, "An Adaptive Multimodal Biometric Management Algorithm", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART C: APPLICATIONS AND REVIEWS, VOL. 35, NO. 3, AUGUST (2005).
- [16] V. Trees and Harry L., "Detection, Estimation, and Modulation Theory", Part I, John Wiley and Sons, (1968).
- [17] J. Q. Yan and Rick S. Blum, "Distributed Signal Detection under the Neyman-Pearson Criterion", IEEE Transactions on Information Theory, vol.47, no.4, (2001).
- [18] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", IEEE Transactions on Acoustics, Speech and Signal Processing, (4):357–366, (1980).
- [19] Y. Stylianou, Y. Pantazis, F. e Calderero, P. Larroy, F. Severin, S. Schimke, R. Bonal, F. Matta, and A. Valsamakis, "GMM-Based Multimodal Biometric Verification", eINTERFACE 2005 The summer Workshop on Multimodal Interfaces July 18th – August 12th, Faculté Polytechnique de Mons, Belgium (2005).

Author



Mohamed SOLTANE (Assoc. Prof. Dr.), he received the M.Eng. degree in Electronics from Badji-Mokhtar University of Annaba-Algeria, in 1995 and the M.Sc. degree in Electrical and Electronics Engineering from UKM Malaysia in 2005, and the Ph.D. degrees in Electronics from Badji-Mokhtar University of Annaba-Algeria, in 2010. He is currently an Associate Professor at Electrical Engineering & Computing Department, Faculty of Sciences & Technology, DOCTOR YAHIA FARES UNIVERSITY OF MEDEA, ALGERIA. His research interests include statistical pattern recognition, biometric authentication, cryptography and quantum computing, computer vision and machine learning and microcomputer based system design...