GMM-Based Multimodal Biometric Verification

Yannis Stylianou, Yannis Pantazis, Felipe Calderero, Pedro Larroy, Francois Severin,
Sascha Schimke, Rolando Bonal, Federico Matta, and Athanasios Valsamakis.

***Abstract*— In this work, we describe how biometric data can be used for person identification and verification. We rely on three categories of traits, that is speech, signature, and face. These distinguishing features or characteristics of a person, on their own, do not provide satisfactory results using well-known techniques. This is the case especially when the number of enrolled persons is large. For this reason, we develop techniques for making good use of all the three traits. In particular, we choose to follow late fusion of the scores of each single trait. The results of these techniques are quite better than using only one trait. Another goal of this work is the creation of a high quality multilingual database with video, audio, and signatures from forty seven persons.**

***Index Terms*— biometrics, speaker recognition, on-line signature authentication, eigenfaces, fusion, multilingual database.**

I. Introduction

OVER the past years, the need for secure transactions using biometric data has attracted a lot of attention. Knowledge-based techniques such as passwords suffer from various shortcomings as they can be forgotten or stolen. Biometric-based features promise easier interaction and potentially high security level.

The use of only one trait for person identification has been proved that is not enough for real life applications such as banking access. This problem is more evident when the number of enrolled persons is increasing. To meet real life applications demands, it is required to take advantage of not only one trait but of two or potentially three. In the current work we decided to make use of three easily acquired traits, that is speech, signature, and face. For this reason and in order to test our algorithms we have created a database of 47 persons. The database contains high quality video of approximately 4 min, speech and signature data of each person.

Algorithms that use biometrics for person identification/verification rely on two categories of traits: physiological and behavioral. Speech and signature can be put under the category of behavioral traits. In a more theoretical context, behavioral traits can be thought of as being different realizations of a stochastic (random) process. These kind of traits have the advantage that are not easily copied. Physiological traits are constant for each person, for example fingerprint face and iris. These kind of features can also be use for identification purposes. In general, physiological features provide better identification results but they suffer from various shortcomings as they can be duplicated.

For the current work, we have decided to make use of both behavioral and physiological traits. This is because we believe that a combination of both will provide better identification results therefore higher security for a potential application. The specific biometrics we use are speech and signature as behavioral traits and face as physiological trait. This particular decision has initiated from the fact that these kind of traits can be easily obtained by prevalent devices such as PDAs or mobile phones.

The whole system is divided in three agents (subsystems) one for every trait. An important point that must be addressed is the fusion of the results from the different subsystems. The procedure we follow here is by adding the different likelihoods from the three different agents (speech, signature and face agent) and picking the largest one.

The remainder of this report is organized as follows. Section II describes the different biometric traits, we used. Section III describes the fusion procedure in detail. In section IV, details about multimodal database are given. Section V shows the results of each agent and of the fusion. Finally, future directions are given in section VI.

II. Biometric Traits

A. Speech

Based on results of previous studies for automatic speaker recognition systems, we have used Mel-cepstral features. Which are one the most successful feature representations in speech recognition tasks.

The feature extraction consists of the following steps. Every 10 ms the speech signal is multiplied by a Hamming window $w[n]$ with a duration of 20 ms to produce a short time speech segment $x[n]$. The discrete Fourier spectrum is obtained via a fast Fourier transform from which the magnitude squared spectrum is computed. The magnitude spectrum $X[n]$ is put through a bank of filters. The filter bank used simulates critical band filtering with a set of triangular bandpass filters. The critical band warping is done following an approximation to the Mel-frequency scale which is linear up to 1000 Hz and logarithmic above 1000 Hz. The center frequency of the triangular filters follow a uniform 100 Hz Mel-scale spacing and the bandwidths are set so the lower and upper passband frequencies of a filter lie on the center frequencies of the adjacent filters, giving equal bandwidths on the Mel-scale but increasing bandwidths on the linear scale. The Mel-scale cepstral coefficients are computed from the filter bank outputs. The first coefficient $c[0]$ reflects the average log energy in the speech frame and is discarded as a form of amplitude normalization.

This report, as well as the source code for the software developed during the project, is available on-line from the eNTERFACE'05 web site: www.enterface.net.

This research was partly funded by SIMILAR, the European Network of Excellence on Multimodal Interfaces, during the eNTERFACE05 Workshop in Mons, Belgium.

B. Face

In our daily life, one of the most important and human-friendly biometrics to identify people is face recognition. Almost all recognition systems including human actors incorporate this modality, based on photographs or video sequences. For more than 20 years, understanding and developing face recognition systems has become a challenge able to seduce people from a wide range of research areas, from pattern recognition and computer vision to cognitive and perception sciences.

The main problem in face recognition is its high interclass variability. On one hand, it suffers from extrinsic variability, for instance the mapping from 2D to 3D or changes on illumination conditions cause that different views provide highly different realizations of the same face. On the other, intrinsic variability due to non-permanent face parameters, as skin color or facial hair length, adds information that is not useful into the recognition process. Thus, the key issue in face recognition is to extract only the meaningful features that characterize a human face, discarding all irrelevant attributes.

Generally speaking, a face recognition system for verification can be divided in the following stages:

1) **Preprocessing**
   - *Localization and segmentation*
   - *Normalization*
2) **Face verification**
   - *Feature extraction*
   - *Classification*

In the following sections, the implementation details for our frontal-view face recognition system are explained.

1) Preprocessing:

a) Face location and segmentation: Face detection and segmentation was performed by OpenCV face detector [3]. Based on cascade Haar classifiers, it provides excellent results in our scenario: a single user in front of a camera. It returns a bounding box centered on the detected face (see Figure 1).

b) Normalization: On the results presented on this paper only size normalization of the extracted faces was used. All face images were resized to 150x150 pixels, applying a bicubic interpolation if needed. After this stage, the image was cut on the borders (30 pixels on the upper and lower borders, and 10 into the left and the right ones), resulting into 90x130 pixel images, to discard most of the hair (a highly variant part of the face) and the picture background.

Although not integrated in the final system, we 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 eye detection proposed in this work is based on a k-means clustering method in a bidimensional space [13]. Initially, the face is binarized and inverted, and the algorithm is not applied to the whole image but to an eye mask including only the upper half part. After that, the pixels are grouped into four clusters, using k-means method. Selecting the lower clusters of each side of the face the position of the eyes is estimated, as can be seen in Figure 3. Some results from different users are shown in Figure 4.

Fig. 1. Face extraction example from our database video sequences performed by OpenCV face detector. The gray scale size-normalized extracted face is shown on the upper left corner of the image.

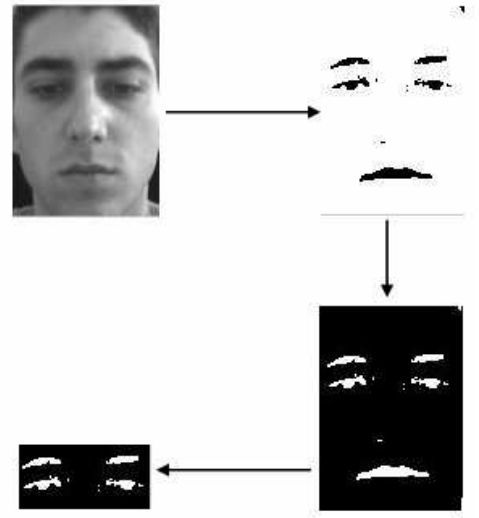


Fig. 2. Binarization, inversion and eye mask selection from detected and segmented face image.

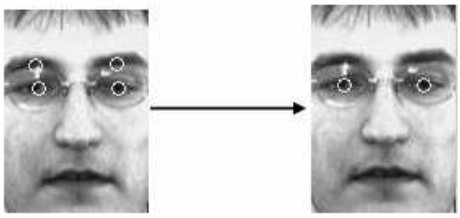


Fig. 3. Detecting and selecting clusters for eye detection.

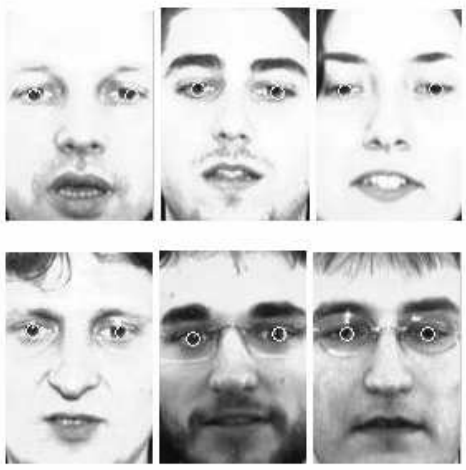


Fig. 4. Eye detection examples for different users.

Fig. 5. Left: Mean of two different face images from the same user without position correction. Right: Mean of the same two images after position correction based on eye detection and location.

The orientation and size correction minimizes the diffusion in the eigenface conformation (see Feature Extraction section) and we believe that improves the verification rate. To illustrate the advantages of further normalization, the average of two images from the same user without and with position correction is shown in Figure 5.

Other normalization schemes would include removing luminance inhomogeneities. In our database, luminance conditions were approximately constant, hence no method was used for this purpose.

2) Face Verification:

a) Feature extraction: The features extracted were based on the Karhunen-Loeve (KL) expansion, also known as principal component analysis (PCA). The main reasons to used KL expansion was that it has been exhaustively studied and have proved to be quite invariant and robust when proper normalization is applied over the faces [1]. On the other hand, the main disadvantages of KL methods is its complexity and that the extracted base is data-dependent: if new images are added to the database the KL base need to be recomputed.

The main idea is to decompose a face picture as a weighted combination of the orthonormal base provided by the KL transform. The base corresponds to the eigenvectors of the covariance matrix of the data, known as eigenfaces (see Figures 6 and 7). This expansion is optimal in a MSE sense, meaning that the image reconstruction that minimizes the MSE, on a dimensional reduced space, is obtained removing

Fig. 6. Upper left corner: mean face image from the whole face database. From left to right, the whole database eigenfaces associated with the 7th largest eigenvalues are shown in decreasing order.

the eigenfaces associated with the smallest eigenvalues of the covariance matrix.

Thus, the decomposition of a face image into an eigenface 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 common eigenfaces and independent user eigenface space are detailed in the next sections.

Common Eigenface Space

In the classic eigenface method, proposed by Turk and Pentland [14], the PCA is performed on a dataset of face images from all users to be recognized.

The first step is to vectorize the set of N face images from different users in the database, $F_1, \ldots, F_N$, resulting into a new set of vectors $f_1, \ldots, f_N$. They can be written as a matrix, concatenating all images as columns,

$$X = [f_1, \ldots, f_N] \tag{1}$$

Hence, removing the mean of the training vectors, f_μ, the data covariance matrix, $X^T X$, can be computed. Grouping as columns the k eigenvectors associated with the first largest eigenvalues into the matrix U, a k-dimensional feature vector for each image can be obtained as

$$y = U^T (f - f_\mu) \tag{2}$$

The feature vector y describes the contribution of each eigenface in representing the input face. Consequently, an image can be projected into the common eigenface space, generating a k-dimensional point.

User Eigenface Space

This approach is based on the same principles as standard PCA, explained in the previous section. The difference is that an eigenface space is extracted for each user. Thus, when a

Fig. 7. Two different examples of individual user eigenfaces. In each row, the first 4 eigenfaces for the same user are shown, the first one including the mean face of the user.

claimant wants to verify its identity, its vectorized face image is projected exclusively into the claimed user eigenface space and the corresponding likelihood is computed.

The advantage of this new approach is that it allows a more accurate model of the user's most relevant information, where the first eigenfaces 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 to compute or recompute the specific eigenface space, but not the whole dataset base as in the standard approach. For verification systems, the computation of the claimant's likelihood to be an 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 eigenface approach has been chosen. Each user's eigenface space was computed which 200 non-consecutive frames extracted from the described database videos.

b) Classification: For classification purposes, a GMM based classifier was used [12]. A total number of 10 non-consecutive images, not previously included into the training database, were used in each claim to compute the average log-likelihood of the claimant being the claimed user. Further details in GMM models and log-likelihood can be found in Section III-A.

C. Signature

Following Plamondon and Lorette [5], the methods of handwriting processing can be classified regarding the type of data acquisition – off-line vs. on-line. For off-line processing, the data acquisition is carried out from the Writing surface (e.g. paper) after the writing process. In the normal case, this off-line acquisition is done with an optical scanner device and the resulting data are a kind of 2-dimensional image. In contrast, in the on-line approach, the data acquisition occurs during the writing process itself. The resulting data of this approach are signals, which describe the pen motion on the writing surface. For gathering of on-line handwriting data, special devices are used, for example graphic digitizer tablets, Tablet PCs or PDA-like computers with pressure sensitive screens. In the following we will concentrate on on-line handwriting processing, recorded by digitizer tablet devices.

The device we used for data acquisition is able to output the pen tip position on the active writing surface with a high resolution. Additionally it measures the pen pressure. The sampling rate is about 100Hz. (For details, see section IV.)

The raw sample point, captured at time t_i, is the following: $s_i = (x_t, y_t, p_t, t_i)$, where x_i, y_i and p_i are the pen tip position and the pressure, respectively. Figure 8 shows x-, y- and p-signals of an example signature.

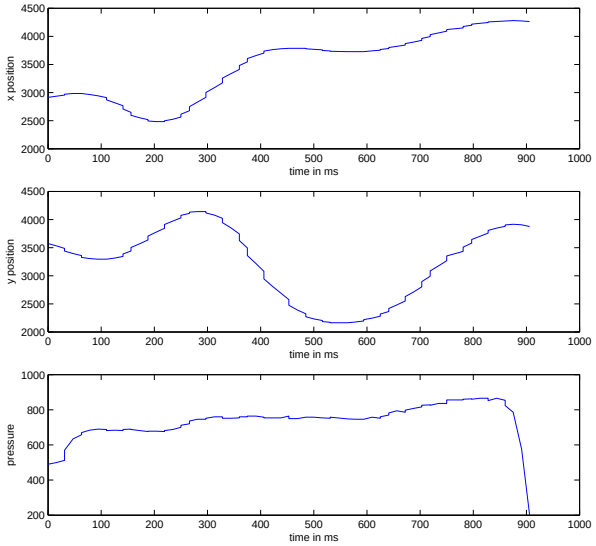


Fig. 8. Signals (x-, y-position and pen pressure) of one signature fragment.

In addition to these raw data, the writing velocity v_i as well as the tangent angle θ_i at every time t_i is computed:

$$v_i = \sqrt{\dot{x}_i^2 + \dot{y}_i^2} \qquad \theta_i = \arctan(\dot{y}_i, \dot{x}_i)$$

with $\dot{x}_i = x_i - x_{i-1}$ and $\dot{y}_i = y_i - y_{i-1}$ (see [9]). These five dimensional feature vectors are used for GMM processing (see section III-A).

III. Fusion

It is well documented that multiple modalities are necessary for high performance in user verification and identification systems [2] [10]. As a consequence of this, a generic biometric system has four substantial modules

(a) *sensor module* where raw biometric data are captured from the devices;
(b) *feature module* in which a feature set is extracted from the raw data of each modality;
(c) *matching module* where a classifier is utilized to compare the features extracted from the previous module with the trained patterns;

(d) *decision module* in which the outputs of the classifiers are combined in order to make a decision.

In the following subsections the matching and decision modules are discussed.

Because of the use of multiple modalities, fusion techniques should be established for coupling the different modalities. Integration of information in a Multimodal biometric system can occur in different levels

(a) *feature level* where the feature sets of different modalities are combined. Fusion at this level provides the highest flexibility but classification problems may arise due to the large dimension of the combined(concatenated) feature vectors.
(b) *score (matching) level* is the most common level where the fusion take place. The scores of the classifiers are usually normalized and then they are combined in a consistent manner.
(c) *decision level* where the output of the classifiers establish the decision via techniques such as majority voting. Fusion at the decision level is considered to be rigid for information integration.

The fusion of our system is done at the score level.

A. Matching Module

In the feature module, a feature set is extracted from each modality. The feature vectors of each modality constitute an D-dimensional feature space. Feature vectors with class labels —in our case one user constitute one class— can be used to estimate a model describing a class.

We propose a method similar to Bayesian classification for the determination of users' identification(or verification). The scores of each modality will be the posterior probabilities or decision risks calculated from the probabilities of the model. The posterior probability of pattern $\mathbf{x}$ to belong in class ω_k can be computed with the Bayes rule

$$P(\omega_k|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_k)P(\omega_k)}{p(\mathbf{x})}$$

where $p(\mathbf{x}|\omega_k)$ is the probability density function of class ω_k, $P(\omega_k)$ is the prior probability and $p(\mathbf{x})$ is merely a scaling signatures factor. The major problem in Bayesian classifier is the determination of $p(\mathbf{x}|\omega_k)$. Some assumptions have to be made about the structure of the class-conditional probabilities $p(\mathbf{x}|\omega_k)$.

One very common approach for approximating the unknown class-conditional probabilities $p(\mathbf{x}|\omega_k)$ is by using Gaussian Mixture Models(GMMs). A GMM is defined as

$$p(\mathbf{x}|\omega_k;\Theta) = \sum_{k=1}^{C} \alpha_k \mathcal{N}(\mathbf{x};\mu_k,\Sigma_k)$$

where $\mathcal{N}(\mathbf{x};\mu_\mathbf{k},\Sigma_k)$ is the Gaussian probability function with mean value μ_k and covariance matrix Σ_k, α_k are positive weights of the component k and $\sum_{k=1}^{C}\alpha_k = 1$. The parameter list

$$\Theta = \{\alpha_1,\mu_1,\Sigma_1,...,\alpha_C,\mu_C,\Sigma_C\}$$

defines a particular Gaussian mixture probability density function.

The parameters of the Gaussian mixture probability density functions are estimated with Expectation Maximization(EM) algorithm [4]. EM algorithm is an iterative method for calculating maximum likelihood distribution parameters. It can also be used to handle cases where an analytical approach for maximum likelihoods estimation is infeasible, such as GMMs with unknown and unrestricted covariance matrices and means.

The training vectors used in EM are first normalized making the standard deviation of each class equal to unity. Given a pattern $\mathbf{x} = [x_1\ x_2\ ...\ x_D]$ the new pattern is defined as

$$\mathbf{x}' = [x'_1\ x'_2\ ...\ x'_D] = [x_1/\sigma_1\ x_2/\sigma_2\ ...\ x_D/\sigma_D]$$

Moreover a Universal Background Model (UBM) [7] is applied in order to model the user-independent distribution of the features. The class-conditional probabilities(or likelihoods) computed by the UBM used for the normalization of the users' class-conditional probabilities. The normalization is done by dividing the likelihood of the UBM from the likelihood of the user.

B. Decision Module

Several techniques have been used to consolidate the matching scores and arrive at a decision. There are two major categories

(a) *classification techniques* where a feature vector is constructed using the matching scores output by the individual classifier. Typical examples are Neural Networks, Decision Trees and Support Vector Machines;
(b) *combination techniques* where the output of the classifiers are combined accordingly. Simple yet considerable examples are Sum or Product Rules and Linear combination of the scores.

In this work we concentrate on the combination techniques.

The advantage of using GMMs for obtaining the matching scores for all the modalities is that they are homogeneous. Applying Bayes rule all the scores are the class-conditional probabilities of the models. If we assume that the a priori probabilities are equiprobable for each user and apply a normalization scheme then the scores are the posterior probabilities. To obtain the posterior probabilities is sufficient to divide the likelihood of each model with the sum of the likelihoods of all the models. In mathematical terms,

$$P(\omega_k|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_k)}{\sum_{i=1}^{C} p(\mathbf{x}|\omega_i)} \qquad k = 1,...,C$$

This operation actually makes the likelihoods $p(\mathbf{x}|\omega_k)$ a distribution ,i.e. likelihoods are transformed in posterior probabilities $P(\omega_k|\mathbf{x})$.

After normalizing the scores of each modality with the above method a simple product rule is applied. This rule is based on the assumption of independence of the modalities. In general, different biometric traits of an individual are mutually independent. There are also other normalization methods as well as combination techniques that were tested but they did not perform better.

IV. DATABASE

A. Database of signatures

The device used for recording the handwriting data was a Wacom Graphire3 digitizing tablet. Size of sensing surface is 127.6mm x 92.8mm. With spatial resolution of 2032 lpi (lines per inch), able to measure 512 degrees of pressure. Data is acquired with a non-fixed sampling rate of about 100Hz.

Altogether, the new database consists of 1641 signatures of 47 persons. For each person, at least 30 signatures are available. The structure of the database is as follows:

```
signatures-+-user01-+-2005-08-08-12-00-00.dat
           |        +-2005-08-08-12-01-00.dat
           |        +- ...
           |
           +-user02-+-2005-08-08-13-01-00.dat
           |        +-2005-08-08-13-04-00.dat
           |        +- ...
           |
           +-user03-+- ...
           |
           +- ...
           |
           +-user47-+- ...
```

Each `.dat` file represents one signature. Each line of a `.dat` file consists of four comma separated integer values for the sampled x- and y-position of the pen tip, the pen pressure and the timestamp (in ms). Those lines with values of -1 for x, y and pressure represent a pen-up/pen-down event.

Because of hypothetical legal and privacy concerns, the definitive acquired handwritten inputs were not real signatures. At an initial stage, experiments were done with a preliminary database composed of real signatures from our team members; then, test subjects were asked to write an arbitrary word as *fake signature*, other subjects chose to do a modification of their true signature. Every subject had to repeat the writing at least thirty times. They were able to see their writings on the screen.

Forged signatures

As a test for the robustness of the identity verification system, *skilled forgeries* of the real signatures of the preliminary signature database were created. The choice of the considered modalities for this database was done admitting that speech and face are very hard to reproduce, in comparison to signatures. For time constraints, *skilled forgeries* were not added to the definitive database of *fake signatures*.

For helping the imitators to reproduce the signatures, an application was developed. It consists of a user interface written in Matlab. Its first task is to reproduce the image of the signature the user wants to be imitate. Then, the user can play a movie representing the exact way the signature has been drawn, in function of time. The speed of the signature is so conserved; and the user can modify it as a parameter for playing the movie. The second parameter of this application is the number of frames per second. Once these parameters are set by the user, the movie is created with linear interpolation between successive samples.

B. Database of audio and video

Audio and still pictures are extracted from the video, which is encoded in raw UYVY. AVI 640 x 480, 15.00 fps with uncompressed 16bit PCM audio; mono, 32000 Hz little endian. A few videos are with uncompressed PCM audio; stereo, 44100 Hz little endian.

We provide Perl scripts for extraction of audio and still pictures from the videos, extraction of audio takes significantly less time than picture extraction. These scripts use `mplayer` and `sox`.

Uncompressed PNG files are extracted from the video files for feeding the face detection algorithms.

Audio is extracted as 16 bit PCM WAV file (with wav header), sampled at 16000 Hz, mono little endian.

Capturing Devices

- Allied Vision Technologies AVT marlin MF-046C 10 bit ADC, 1/2" (8mm) Progressive scan SONY IT CCD.
- Shure SM58 microphone. Frequency response 50 Hz to 15000 Hz. Unidirectional (Cardiod) dynamic vocal microphone.

Bugs

There was a problem with mplayer not writing the `byte_alignment` of the wav header correctly, which caused files not being read correctly on Matlab. A patch fixing the bug was sent and merged in the mplayer CVS. We include the patch with the database. Against CVS revision: 1.29 of `/cvsroot/mplayer/main/libao2/ao_pcm.c` The patch should also work against MPlayer-1.0pre7, which was the latest official release of mplayer which was unpatched. So using the CVS version is recommended, until a patched official release is made. If not, `--fixheader` option can be used for `separate audio from video stream*` scripts.

V. RESULTS

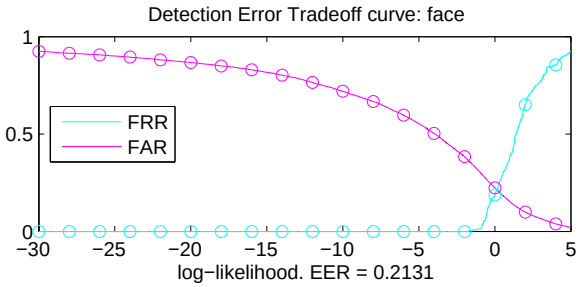
In a verification system we often have a tradeoff between the ratio of impostors accepted on the system, which is denoted as False Acceptance Rate (FAR) or false positives, and the ratio of rejected genuines, denoted as False Rejection Rate (FRR) or false negatives. When choosing the decision threshold θ, choosing a too low value, would let too many impostors in, while choosing a very high value would cause too many rejections to genuines, and the performance of the system would be unacceptable for practical uses.

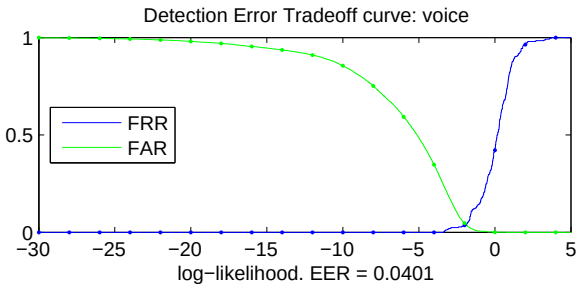
A good decision is to choose θ_{EER} such as $FAR(\theta) = FRR(\theta)$, but since FAR and FRR are discrete, an option is to choose:

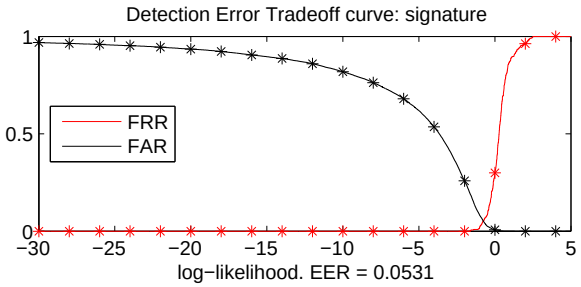
$$\theta_{EER} = argmin_{\theta}|FRR(\theta) - FAR(\theta)|$$

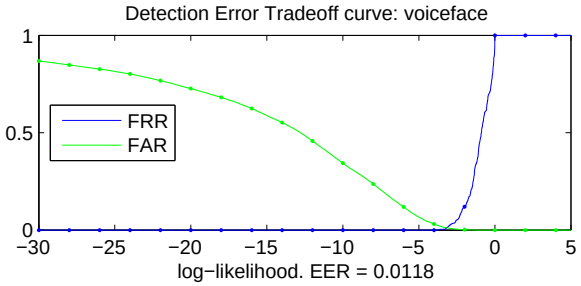
$$\theta_{EER} = \frac{FRR(\theta) + FAR(\theta)}{2}$$

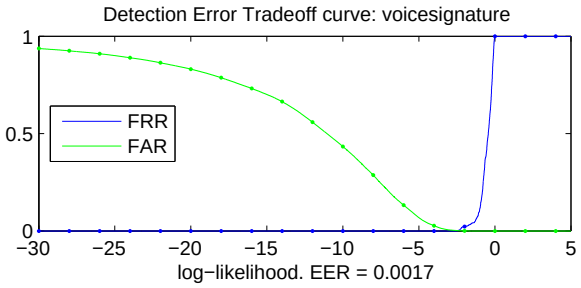
To compare the performance of our system we use the Detection Error Tradeoff (DET) curve and the definitions above for the calculation of the Equal error rate (EER):

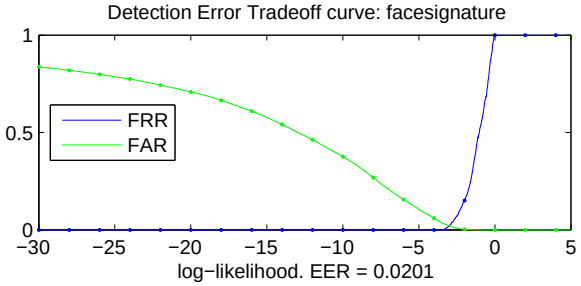


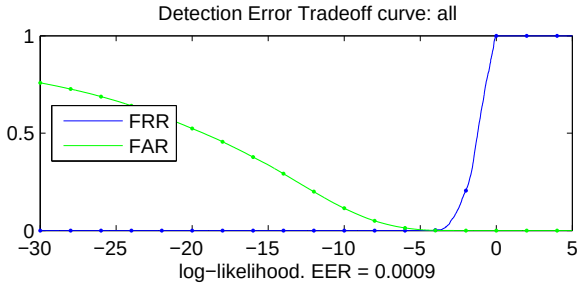












VI. Conclusion and Future Work

We can see how combining all the modalities allows us to achieve an EER of 0.09% which is much better than those of the modalities taken separately.

Different methods of fusion could be tested for cases in which modalities could not be considered independent. Also new feature extraction methods could be tested.

Acknowledgment

The authors would like to thank eNTERFACE and its sponsors for providing the means which made this project possible. Also thanks to previous research articles and the free/open source software we have used, which has allowed us to *stand on the shoulders of giants*.

References

[1] Chellappa R., Wilson C.L., Sirohey S., Human and Machine Recognition of Faces: A Survey. Proceedings of the IEEE. Volume 83. Number 5. May 1995.
[2] A.K. Jain, A. Ross, S. Prabhakar, *An introduction to biometric recognition*, IEEE Trans. Circuits Systems Video Technology, pp. 4–20, 2004.
[3] Open Source Computer Vision Library Documentation. http://www.intel.com/technology/computing/opencv/
[4] Pekka Paalanen, *Bayesian classification using gaussian mixcute model and EM estimation: implementation and comparisons*, Information Technology Project, 2004, http://www.it.lut.fi/project/gmmbayes/
[5] R. Plamondon, G. Lorette,: *Automatic Signature Verification and Writer Identification – The State of the Art*, Pattern Recognition, Vol. 22, No. 2, pp. 107–131, 1989.
[6] D. Reynolds *Speaker identification and verification using Gaussian Mixture Models*, Speech Communication, 1995.
[7] D.A. Reynolds, T.F. Quatieri and R.B. Dunn, *Speaker verification using adapted gaussian mixture models*, Digital Signal Processing, pp. 19–41, 2000.
[8] D. Reynolds, T. Quatieri and R. Dunn, *Speaker Verification Using Adapted Gaussian Mixture Models*, Digital Signal Processing 19-41, 2000.
[9] J. Richiardi and A. Drygajlo, *Gaussian Mixture Models for Online Signature Verification*, ACM Press, 2003.
[10] A. Ross, A.K. Jain, *Information Fusion in Biometrics*, Pattern Recognition Letters, 2003.
[11] A. Ross, and K. Jain, *Multimodal Biometrics: An Overview*, Prom. of 12th European Signal Processing Conf (EUSIPCO), pp. 1221-1224, 2004.
[12] Sanderson C., Bengio S., Robust Features for Frontal Face Authentification in Difficult Image Conditions. IDIAP-RR 03-05, January 2003.
[13] Seber, G. A. F., Multivariate Observations, Wiley, 1984.
[14] Turk M., Pentland A., Eigenfaces for Recognition. Journal of Cognitive Neuroscience. Volume 3, Number 1. Massachusetts Institute of Technology, 1991.

Yannis Stylianou Associate professor in the Computer Science Department, University of Crete. email: yannis@csd.uoc.gr

Yannis Pantazis Postgraduate student in the Computer Science Department, University of Crete. email: pantazis@csd.uoc.gr

Felipe Calderero Technical University of Catalonia (UPC). Superior Telecommunication Engineering. email: felipe@gps.tsc.upc.edu

Pedro Larroy Technical University of Catalonia (UPC). Superior Telecommunication Engineering. email: pedro@larroy.com

Francois Severin Faculte Polytechnique de Mons. email: francois.severin@tcts.fpms.ac.be

Sascha Schimke Otto-von-Guericke University Magdeburg, Germany. email: sschimke@iti.cs.uni-magdeburg.de

Rolando Bonal Universidad de las Ciencias Informaticas (UCI), Habana, Cuba. email: rolandobonal@gmail.com

Federico Matta Institut Eurecom (CNRS) France. email: Federico.Matta@eurecom.fr

Athanasios Valsamakis Postgraduate student in the Computer Science Department, University of Crete. email: valsamak@csd.uoc.gr