

# Biometrics

Michael Shell, *Member, IEEE*, John Doe, *Fellow, OSA*, and Jane Doe, *Life Fellow, IEEE*

## I. 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.

### A. Preprocessing

1) *Face location and segmentation*: Face detection and segmentation was performed by OpenCV face detector [1]. 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).

2) *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.



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.

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 [2]. 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.

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.

### B. Face Verification

1) *Feature extraction*: The features extracted were based on the Karhunen-Loeve (KL) expansion, also known as principal component analysis (PCA). The main reasons to use 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 [3]. On the other hand, the main disadvantages of KL methods is its complexity and

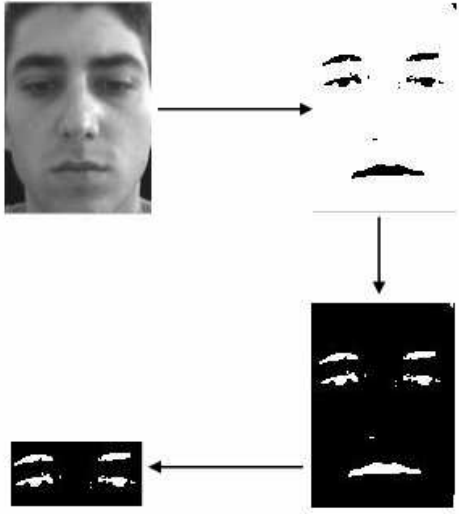


Fig. 2. Binarization, inversion and eye mask selection from detected and segmented face image.



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

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 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 [4], 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,  $F_1, \dots, F_N$ , resulting into a new set of vectors  $f_1, \dots, f_N$ . They can be written as a matrix,



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.

concatenating all images as columns,

$$X = [f_1, \dots, f_N] \quad (1)$$

Hence, removing the mean of the training vectors,  $f_\mu$ , 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

$$y = U^T (f - f_\mu) \quad (2)$$

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 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.



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.



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.

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.

2) *Classification*: For classification purposes, a GMM based classifier was used [5]. 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 ??.

## REFERENCES

- [1] Open Source Computer Vision Library Documentation. <http://www.intel.com/technology/computing/opencv/>
- [2] Seber, G. A. F., Multivariate Observations, Wiley, 1984.
- [3] 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.
- [4] Turk M., Pentland A., Eigenfaces for Recognition. Journal of Cognitive Neuroscience. Volume 3, Number 1. Massachusetts Institute of Technology, 1991.
- [5] Sanderson C., Bengio S., Robust Features for Frontal Face Authentication in Difficult Image Conditions. IDIAP-RR 03-05, January 2003.

**Michael Shell** Biography text here.

PLACE  
PHOTO  
HERE

**John Doe** Biography text here.

**Jane Doe** <http://www.intel.com/technology/computing/opencv/>