Person Identification from Video using Facial and Speaker Features

Andrei Mihaila

University of Joensuu

Department of Computer Science

Master's Thesis

28th January 2005

Table of contents

Abstract

Demands for reliable person identification systems have increased significantly due to highly security risks in our everyday life. Biometric systems have been a major research focus area since the traditional means of security are not reliable and convenient enough. A recent trend in biometrics is to combine different modalities by using multiple human characteristics to increase the system's effectiveness.

This thesis addresses the problem of a bimodal biometric system for person identification from audio-video shots. The solution consists in fusion of the face and speaker recognition classifiers results. We cover both modalities with the emphasis on face recognition and the fusion techniques.

For face modality, we first address the face detection task and then we study statistical methods for recognition such as Eigenfaces and Fisherfaces. We describe the face space and we present one method for selection of the most representative faces from a video sequence by clustering the face space.

We give a short introduction to the speaker recognition technology. Then, we introduce the information fusion theory and we discuss different methods for classifier integration. Our work deals with multi-modal multi-expert fusion strategy. We regard only the offline fusion approach when we combine the results after both classifiers processed the test sequence. We perform experiments for both classifiers and then we integrate the results at score level.

Keywords: *audio-video person recognition, face recognition from video sequence, face detection, speaker recognition, information fusion, multi-modality, biometrics, audio-video databases.*

1. Introduction

Demands for reliable person identification systems have increased significantly due to highly security risks in our everyday life. Application areas for person identification are broad, including low enforcement, control access to financial transactions, to computer networks and to secured locations. Most of the current systems are based on identity claims (e.g. tokens, cards, keys) or recognizing passwords or personal identification numbers. The weaknesses of these systems are the possible fake or loss of the identity claims and the discovery of passwords and numbers, and using them without detection.

Biometric identification systems have been a major research focus area since the traditional means of security are not reliable and convenient enough. A biometric is a measurable physical characteristic or personal behavioral trait used to recognize the identity, or to verify the claimed identity of an enrollee [24]. Moreover, they are more difficult to fake, and thus they ensure much greater security than traditional identification methods.

Although relatively high recognition rates have been achieved using a single biometric, a recent trend is to combine different modalities by using multiple human characteristics. The goal is to complement one modality with another when one of them performs poorly, so it will not affect the final decision.

1.1 Problem Definition

In this thesis, we address the problem of a bimodal biometric system that recognizes people from audio-video sequences based on the fusion of *facial* and *speech* features that are already stored in a database of known individuals. More specifically, our interest is in recognition from shots where only one person appears in the image while talking, and his or her identity does not change during the session. Our aim is to study different data fusion techniques of the two recognition modules to achieve higher recognition rate than using only one of the modalities alone.

Although most of the literature in audio-visual based biometrics includes only *dynamic visual features* of faces such as lip movement, opening a new chapter of *visual speech and speaker recognition* [9, 16, 26, 58, 80], we approach the visual recognition from the whole face region combined with different methods for speaker recognition. We are not aware of a general statement of this problem [15, 38, 58, 81]. We consider an offline approach where we perform recognition and integration of both classifiers after the audio-video sequence has been processed.

1.2 Description of the System

Figure 1-1 presents the layout of the system that consists of a *face recognition module*, a *speaker recognition module* and a *fusion module*. The two recognition modules represent unimodal biometric systems based on a common *sensor,* an audio-video camera, and a database of facial and speaker features assigned for each known individual. The audio and visual streams are separated and both of them are processed in parallel by its corresponding module for *feature extraction.* Both modules output a *matching* score reflecting their confidence in the presumed identity. The fusion module combines the two scores to make the final recognition decision. This is the basic structure of most multimodal biometric systems, in the same way we could combine many other recognition classifiers by the fusion module.

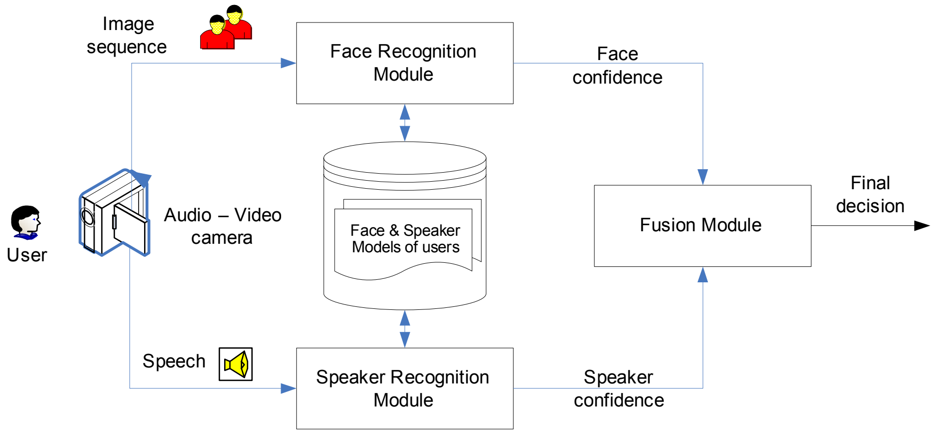


Figure 1-1: Overview of a bimodal biometric system.

Figure 1-2 depicts the components of any person recognition module while Figure 1-3 depicts the related tasks involved for face recognition, in particular. In a realistic environment for face recognition, shots do not have always the same background, pose or facial expression and that is why we need to detect faces in any kind of background.

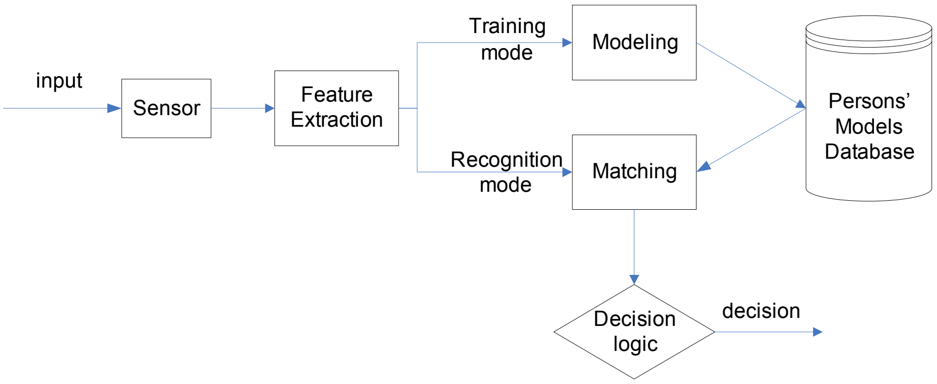


Figure 1-2: Components of any person recognition system.

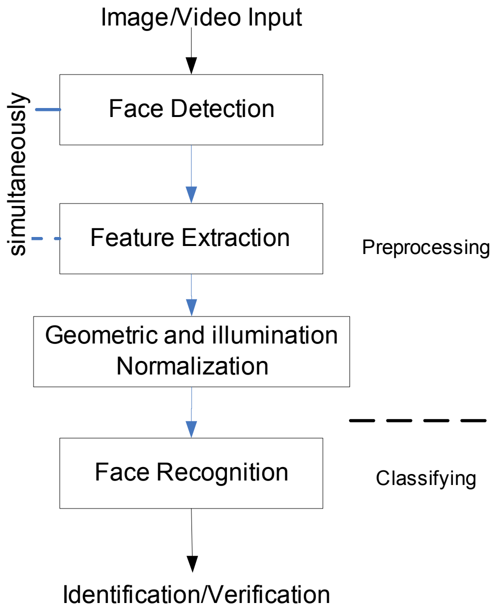


Figure 1-3: Configuration of a typical face recognition system.

1.3 Scenarios

Person recognition scenarios can be classified in two types: *identification* and *verification.* Person identification refers the process that given an unknown subject, the system reports his or her identity by looking up a database of known individuals whereas in verification problems, the system confirms or rejects the claimed identity of the person from the input shot (Figure 1-4).

In order to achieve these tasks, we distinguish two operating modes (Figure 1-2): *training* mode *(enrollment)* and *recognition* mode. In training mode, the system employs a *feature extraction* step, in which it acquires face and voice information of all users, and stores it in the database with labeled identity. They represent the *models* or *templates* of the registered persons in the database. In recognition mode, face and voice are acquired from an unknown person. Finally, a *decision* is taken based on the *matching* score between the models of the unknown and the already registered individuals.

While identification involves comparing the acquired biometric information against face and voice templates corresponding to all users in the database, verification involves comparison with only those templates corresponding to the claimed identity [39, 60]. Moreover, if the user is already modeled, then the recognition is a *closed-set* problem, while if the testing person's model might not be stored, then it is an *open-set* task.

A visual representation of recognition tasks is shown in Figure 1-4, where identification and verification engines have the structure from Figure 1-1. The identification and verification scenarios as well as the training and recognition operating modes are applicable for each of the biometric modules.

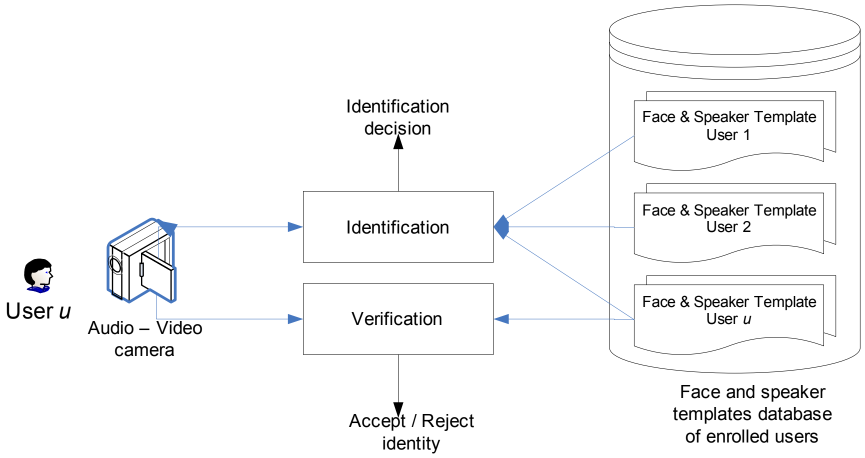

Identification decision
Identification
Verification
Accept / Reject identity
User *u*
Audio – Video camera
Face & Speaker Template User 1
Face & Speaker Template User 2
Face & Speaker Template User *u*
Face and speaker templates database of enrolled users

Figure 1-4: Identification and verification subtasks

1.4 General terms

Face detection and recognition deal with the main following problems (Figure 1-5):

a) Orientation (tilted faces) = subjects rotate their heads having different orientation of faces.

b) Illumination = shots taken when source of light is not uniform, or from different angles

c) Face expression = subject does not have a neutral face expression.

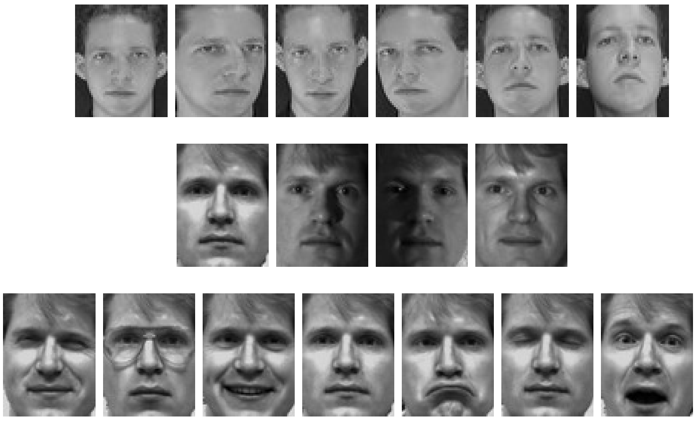


Figure 1-5: Examples of problems for the face recognition and detection tasks: orientation (top row), illumination (middle row) and face expression (bottom row)

Vector Representation of Images = an image *I(rows, cols)* can be represented as a *N*-dimensional vector by lexicographic ordering of the pixel elements that is concatenating each row or column of the image, where $N = rows \times cols$.

Image Space = all image vectors establish an image space, where each axis corresponds to a pixel of the image, and the coordinate is its gray intensity level. By this representation, each image is mapped to a point in a high dimensional vector space.

Train Image = an image of an individual that is stored in the database. It is also known as *gallery image.*

Test image = an image that needs to be classified as belonging to a known individual from the database or not. It is also known as *probe image.*

1.5 Outline of the Thesis

For a concise presentation of the chosen topic, we organized the paper as follows. Section 2 includes description of various methods for face detection. Section 3 is reserved for the face recognition task from still images, while the video sequences are regarded in Section 4. Section 5 gives a short introduction to the speaker recognition technology and we reserved Section 6 for presenting different fusion strategies. We perform experiments for separate and combined modalities and we discuss the results in section 7. We draw the conclusions in the last section.

2. Face Detection

2.1 Introduction

Many of the current face recognition methods assume that the faces in an image have been already localized, are frontal and they have similar sizes [33, 81]. In realistic application scenarios, however, facial images are acquired under natural conditions and it is common that faces occur in many different positions and scales, within a complex background. These conditions represent an uncontrolled environment. These factors might affect the performance of a face recognition system, and in order to correct this, we should perform first an accurate face detection process with the purpose of localizing and extracting the face regions from the background. Normalization ensures that the training and testing faces have all the same illumination, orientation and sizes.

In this section, we will make a summary of this topic, highlighting the basic approaches and detail some algorithms for a better understanding. Detailed surveys of face detection algorithms can be found in [33, 75].

The problem of face detection can be defined as following: given a still image, the goal is to determine whether there are faces in the image. If faces are present, localize each of them regardless of their positions, scales, orientations, poses and lighting conditions [75, 80] (Figure 2-1).

This is a challenging problem because human faces are highly non-rigid objects with a high degree of variability in size, shape, color and texture. The solution of the problem involves segmentation, face and feature extraction, verification of faces and possibly extraction of facial features [33]. Face detection and feature extraction used in face recognition can be achieved simultaneously because the facial features such as eyes, nose and mouth are often used in face detection process (Figure 1-3).

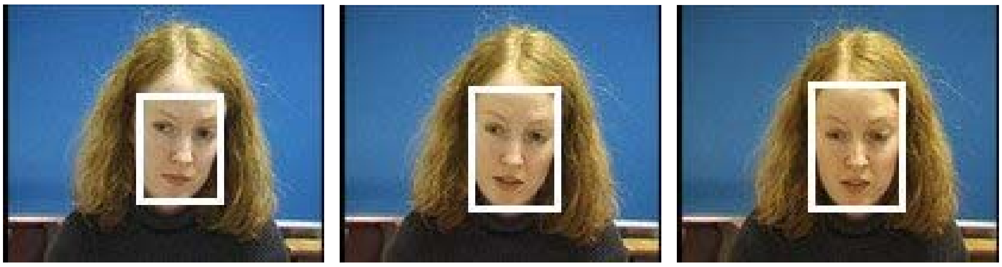


Figure 2-1: Visual representation of face detection.

Existing face detection methods use two main approaches: the *geometrical* and *appearance-based* approach (*image-based*). The first makes explicit use of face knowledge for deriving low-level features. It exploits face geometry by manipulating distances, angles and area measurement of the visual features derived from the scene. In the second approach, the models are learned from a set of training images that should capture the representative variability of facial appearance [33, 75]. A preview over the approaches regarded in this section can be found in Figure 2-2.

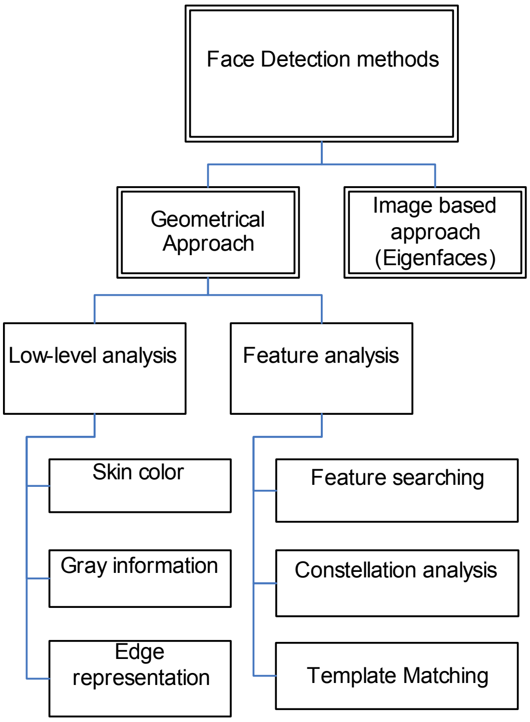


Figure 2-2: Classification of face detection methods described in this section.

2.2 Geometrical Approach

We can divide the techniques further into two areas: *low-level analysis* extracts facial features by segmenting the images using pixel-level properties such as color and gray-scale. *Feature analysis* is based on simple rules about face geometry. In general, a face appears in one image with two eyes that are symmetric to each other, a nose and a mouth; their relative distances and positions can represent the relationship between features. First, facial features in an input image are extracted by low-level analysis. Then, by feature analysis, possible face candidates are identified based on the rules created and usually verified to reduce false detections [75].

2.2.1 Low-Level Analysis

Edge Representation

Edge representation is a primitive feature often used in computer vision applications. It can be also used to find facial features defined by the edges: eyebrows, eyes, nose, mouth and hairline. Edge features within the head outline are then subject to feature analysis using shape and position information of the face [33].

This approach is based on *edge detection.* Many different types of *edge operators* have been applied but the most commonly used are *Sobel* [10] and *Marr-Hildreth* [48]. Varieties of first and second derivatives (*Laplacian*) of Gaussians have also been used [31, 32, 33, 61].

In this approach, edges need to be labeled and matched to a face model in order to verify correct detections. Govindaraju et al. presented in [28] a method for face localization from a cluttered background, in which face hypothesis are generated and tested. The Marr-Hildreth operator is used to detect the curves of the left side, the hairline and the right side of a frontal face and to obtain an edge map of the input image (Figure 2-3b). A filter is used to remove the components whose contours are unlikely to be part of a face. Pairs of fragmented contours are linked based on their proximity and relative orientation. Corners are detected to split the contours into features curves, which will be labeled by checking their geometric properties and relative positions in the neighborhood. Pairs of three feature curves (Figure 2-3c) are joined by edges to form possible face candidate locations to check if they match against a face model using the *golden ratio*[1] for an ideal face [28, 33, 75]:

$$\frac{height}{width} = \frac{1+\sqrt{5}}{2}$$

The same approach was also used in [66] with the difference that they make use of the elliptical structure of the human head. The edge map is created by using *Canny* edge detector [33]. After the segments have been labeled and linked, the algorithm takes pairs of segments and tries to fit an ellipse to them (Figure 2-3d). Facial features can be found by finding first the horizontal and vertical edges of the image (Figure 2-3e).

The problem of segmenting faces from a uniform background is not very difficult in this approach. The edge map of the image can give a good outline of the image containing the face region. In this case, searching for a candidate is limited because all edges in such images represent the face region, which is not the case of a cluttered background.

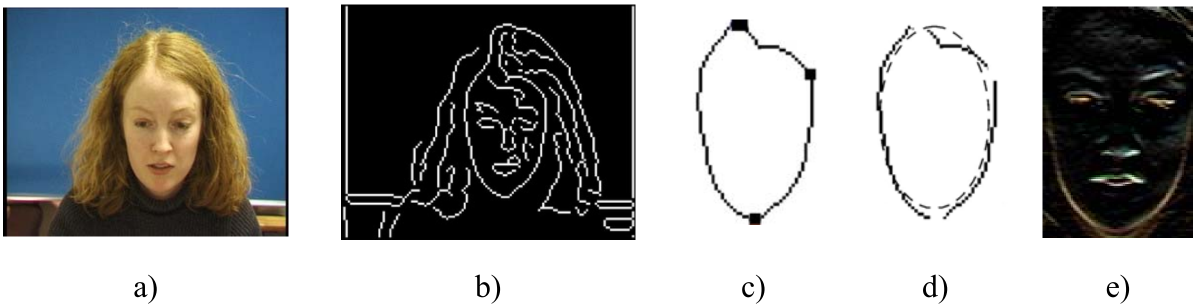


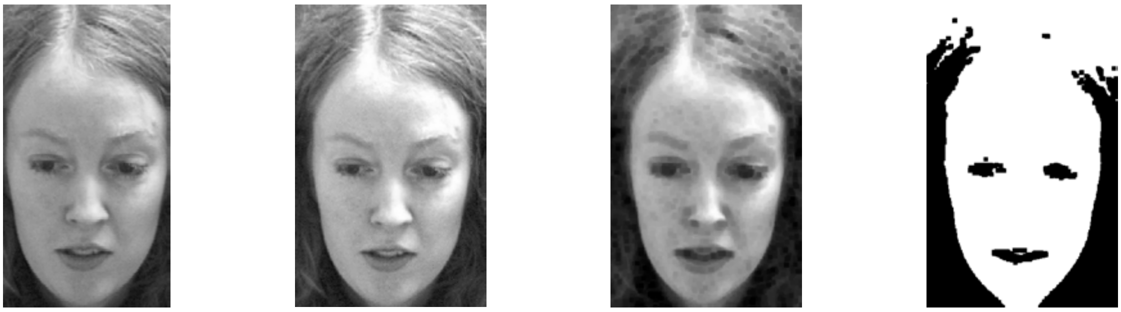
Figure 2-3: Edge detection approach.

Gray Information

The gray information within a face can be used for finding features because eyebrows, pupils and lips appear generally darker than their surroundings facial regions. Based on this, several facial feature extraction algorithms search for local gray minima within segmented facial regions [7, 29, 41]. In order to make detection easier, the input images are first enhanced by several image processing techniques such as contrast-stretching and gray-scale morphological

[1] An aesthetically proportioned rectangle used by artists

routines to improve the quality of local dark patches, which are further extracted by gray-scale thresholding [33] (Figure 2-4).



Original Image　Contrast-stretched　Eroded 3 times　Threshold = 90

Figure 2-4: Face detection based on gray-scale thresholding.

Another method [33] consists in finding possible face candidates by horizontal and vertical projections of the gray level pixels. Let *I(x, y)* be the intensity value of an $m \times n$ image at position *(x, y).* We define the two projections as:

$$HI(x)=\sum_{y=1}^{n} I(x,y) \qquad VI(y)=\sum_{x=1}^{m} I(x,y) \tag{1}$$

Two local minima of the horizontal projection (*HI*) correspond to the left and right side of the head, while the local minima of the vertical projection (*VI*) determine the locations of chin, mouth lips, nose tip, eyes and forehead (Figure 2-5a).

Images can also be preprocessed by applying two gradient operators to obtain horizontal and vertical edges. This approach is suitable for frontal and uniform background images but it cannot detect multiple faces in one image because it cannot distinguish between two faces that are on the same coordinate as in Figure 2-5b where local minima of horizontal projection for the chin corresponds to the both faces. Despite its simplicity at a first glance, this method could lead to false features candidates because of many local-minima in the projection values and that is why we have to strengthen them according to some other feature-searching techniques (see next section).

A *coarse-to-fine* or *focus-of-attention* strategy for face detection creates a multi-resolution hierarchy of images of three levels by reducing the resolution gradually by either sub-sampling or averaging as shown in Figure 2-6. The face candidates are found at the lowest resolution (Level 1) by scanning a window over the input image and searching for uniform regions by locally thresholding. At Level 2, histogram equalization is performed on the face candidates received from Level 1, followed by edge detection [75]. Surviving candidates regions are verified at Level 3 by the existence of prominent features such as eyes and mouth using local gray minima, similar to the first method in this category.

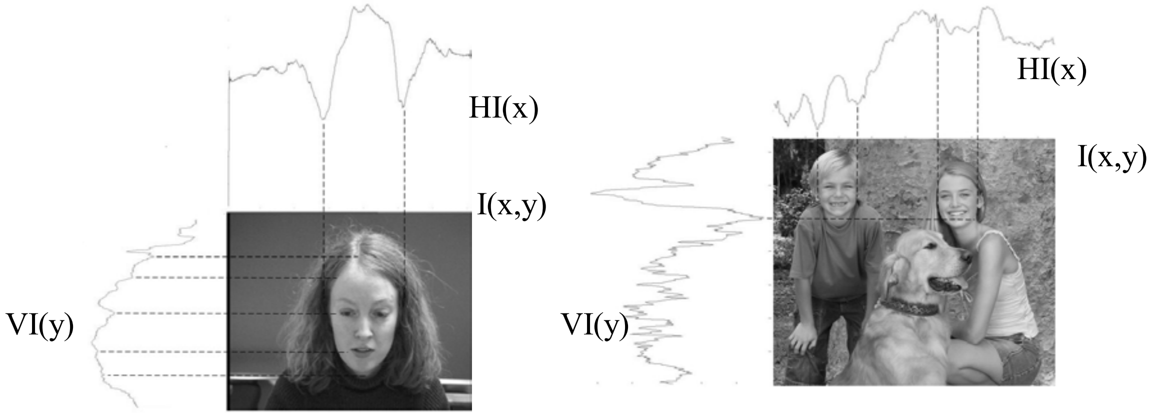
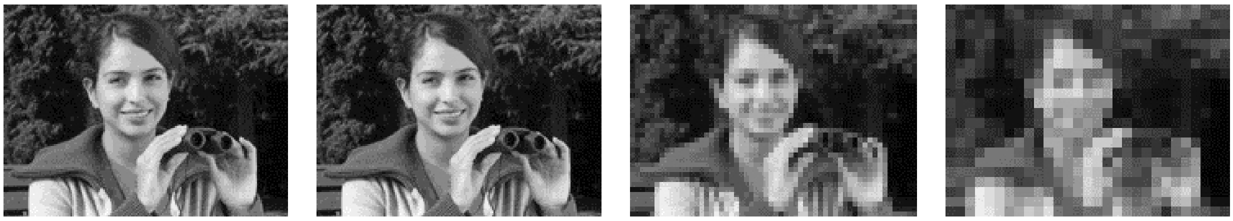


Figure 2-5: Face detection and locating features by vertical and horizontal projections.



original image *(n = 1)* Level 3 *(n = 8)* Level 2 *(n = 16)* Level 1 *(n = 32)*

Figure 2-6: Multi-resolution hierarchy of images. Each square cell consists of n x n pixels, in which is pixel is replaced by the average intensity of the pixels in that cell.

Skin Color

Human skin color is an efficient feature for identifying facial areas but fragile in the same time due to the variation of illumination in uncontrolled environments. The basic ideas that this approach relies on are that color composition of human skin differs little across individuals and the difference between different skin colors is because of their intensity (brightness) rather than their chrominance [29, 33, 75, 77].

Current algorithms use for skin representation a variety of color models such as *RGB*, *normalized RGB*, *HSI*, *YCrCb*, *YIQ*, *CIE-xyz*, *CIE-L*a*b** and *CIE-L*u*v**. From all these, RGB representation is one of the most widely used color model, though normalized RGB colors should be preferred over RGB for canceling the luminance effect [33, 75]. In general, the apparent color of objects depends on the illumination conditions, so the intensity or brightness of the color is discarded in order to obtain a high level of invariance to the intensity of ambient illumination.

One intuitive way for color segmentation is to use skin color thresholds in YCrCb color system using only the *Cr* and *Cb* components[2] [14]. We chose the thresholds [Cr_{min}, Cr_{max}] and [Cb_{min}, Cb_{max}] from samples of skin color pixels from a training set, and in order to

[2] *Y* is luminance component, *Cr* is chrominance of red and *Cb* is chrominance of blue from YCrCb color space

classify one pixel as being a skin region, its red and blue component must be within the interval (see Figure 2-7 for an example).

More complex, statistical methods use Gaussian density functions [75, 78, 12] and a mixture of Gaussians [79] to model the skin color variance within a wide set of samples of faces. The histogram models are superior in accuracy, and computational cost is lower than for the mixture models [75], but the last ones have the advantage of adapting new color variation of the new users by a learning approach [33].

Every color model has some advantages over the others. For example, HSI color model is useful for extracting facial features as lips, eyes and eyebrows [42] while the YIQ suppress the background of other colors as well [33]. The most common drawbacks of this approach are their sensitivity to variation of illumination and that we can detect background objects of similar color to human faces as well (Figure 2-8). They are not efficient alone and that is why combination of shape analysis, color segmentation and motion information are preferred. Despite this, the advantage of using skin color is that it is rotation invariant and is one of the fastest detection methods with application in video-sequence face segmentation.

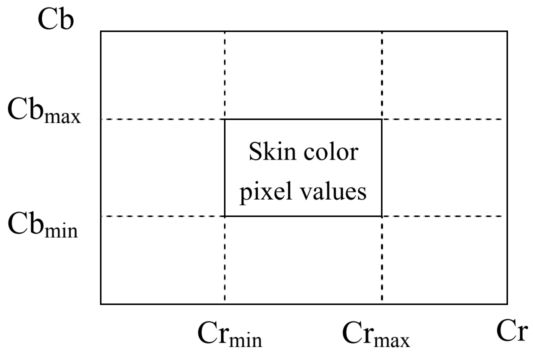


Figure 2-7: Skin color pixel values in CbCr color space.

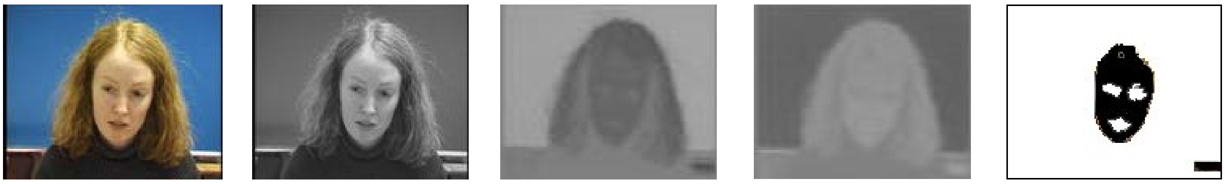


Figure 2-8: Original image, luminance, red and blue chrominance from YCrCb and picture thresholded.

2.2.2 Feature Analysis

If low-level analysis can provide ambiguous features, methods based on feature-analysis use knowledge of face geometry to enhance their confidence. In general, face detection is a two-stage process: first face features candidates are generated and feature analysis is then employed for verification of the candidates. The second step performs actually the facial *feature extraction*. We will present two approaches including *feature searching* based on relative positioning of individual facial features and *constellations* that use different face models [33].

Feature Searching

Feature searching techniques aim at finding out the prominent facial features as a pair of eyes [76, 19, 34, 29], main face axes [18], outline of the head [18, 65] and body [68]. Feature candidates are detected by the methods previously mentioned in Section 2.2.1 and then their confidence is enhanced by the existence of the near-by features based on the relative positioning and anthropometric distances (Figure 2-9). The pair of eyes is the most applied feature because of its symmetry.

Constellation Analysis

Probabilistic face models have been proposed by grouping facial features in face-like constellations. A set of local feature detectors identify candidate locations for facial features as eyes, nose and nostrils. For a better accuracy in finding features, the method makes use of relative distances between features, which are modeled by a Gaussian distribution. The arrangement of facial features is reffered as *constellation*. It can be viewed as a graph, in which the nodes correspond to the features and the edge lengths represent the distances between the features (Figure 2-9) [43]. Finding the best constellation becomes a of *graph matching* problem [33] in which we try to match the graph model of the face candidate to the graph model of the template face.

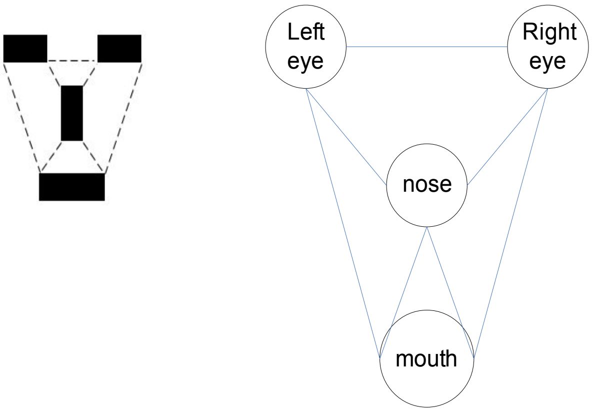


Figure 2-9: Examples of feature-analysis. Facial features and distances (left), and facial features graph model (right).

Template Matching

Predefined or *parameterized templates* can model face outlines and facial features on the basis of a priori information about the expected shapes. Templates are built using manually determined fiducially[3] points from a set of training images, and variations of these points include objects of different sizes and poses. Face evidence is determined based on the cumulative correlation values between the standard templates and the patterns computed from the facial features of the input image. Templates are defined in terms of line segments or their contour can be discretized into a set of labeled points (Figure 2-10).

Deformable templates describe a higher level of appearance of features by employing an *elastic model* for facial features. Once initialized near a feature, they will interact with local image features (edges, brightness) and gradually deform themselves by translation, scale or rotate operations to take the shape of the feature but they constrained in the same time to be similar to the shapes of the training set [33] (Figure 2-10). The best fit of the elastic model is found by minimizing an energy function of the parameters [75], and the final parameter values are used as descriptors of the features.

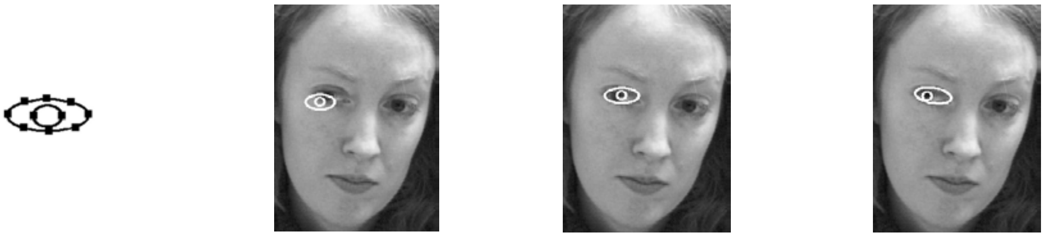


Figure 2-10: Deformable template. Label points from the eye template. Initialization of the template. Steps in the minimization process for eye detection.

2.3 Image-based Approach

The geometrical approach implies knowledge about face for creating models and if these models are not properly designed, the face detection process may fail due to cluttered-intensive background and unpredictability of the face appearance. In image-based approach, face templates are learned from a set of example images by a training process and, in this way, possible modeling errors because of incomplete or inaccurate face knowledge will be eliminated.

Face detection can be treated as a pattern recognition problem, in which the image is considered as a 2-D intensity array. In general, these methods find out statistical characteristics of face and non-face images in order to classify the examples. The learned characteristics are in the form of distribution models or discriminant functions (threshold function, decision surface).

[3] Important point from the face, e.g. eye and nose corners

Most of the methods based on this approach apply a multi-resolution window scanning technique, which is just an exhaustive search of the input image for all possible face locations at all scales [33]. *Face likelihood* for each observation in the window are computed for all locations in the image and they are used as a measure of *faceness* [68] to build a *face map* (distance map). Then, a face can be detected from finding the global minima of the face map and comparing to a threshold value, which is chosen experimentally [33 , 75].

These algorithms are in general computational expensive but window scanning can be avoided by combining the image-based approach with feature-based methods with the purpose of guiding the search based on visual clues such as skin color. Perhaps the most popular algorithm is the one based on *Principal Component Analysis* [67] but we will detail it Section 3.3.2 since it is commonly used for face recognition as well. Other methods are based on *Linear Discriminant Analysis, Neural Networks* and *Hidden Markov Models* [33 , 75].

2.4 Face Detection Algorithm

Since face detection was not the main goal of our work, we present here a naïve algorithm we have used for extracting frontal faces from an image sequence from Cuave database [20]. We apply low-level and feature analysis of images such as edges and gray levels, skin color, and knowledge about facial features, all described in Section 2.2. We summarize all the steps in Figure 2-11. We constrain our algorithm to work only *one face* present in the image, *simple background* and *no profile* faces.

The algorithm consists of the following main phases:

- Detect face region
- Detect facial features
- Face normalization.

For *detecting the face region* from image *img*, first we model the widths of the faces (d_1) and heights of the foreheads (d_2) by two Gaussian distributions $N_1(\mu_1, \sigma_1)$ and $N_2(\mu_2, \sigma_2)$. For this, we label manually an arbitrary set of faces to get the distances. We calculate the means μ_1, μ_2 and variances σ_1, σ_2 and we choose the maximum width of face and range for the forehead's height as:

$$d_{1\,max} = \mu_1 + 2\sigma_1 = 202 + 2 * 13 = 228$$

$$[d_{2,min}\ d_{2,max}] = [\mu_2 - 2\sigma_2 , \mu_2 + 2\sigma_2] = [129 - 2 * 15, 129 + 2 * 15] = [99, 159]$$

Next, we construct one face space *FS* with the faces cropped manually. We choose experimentally the threshold θ such as:

$$\theta = \max d(f_i, FS), \forall f_i = face$$

$$d(f,FS)=d,\begin{cases} d \le \theta & f \text{ is face} \\ d > \theta & f \text{ is not face} \end{cases}$$

We separate the two color planes *Cr* and *Cb* from *img* (Figure 2-11.1). We threshold *Cr* and *Cb* by *Otsu* method[4] (Figure 2-11.2), and then combine the resulting images by OR function into image *binImg* (Figure 2-11.3).

We label *binImg* and extract the biggest object as being the body. Get the center of mass vertical position x_{center} and limit (crop) the original image to the object's boundaries (Figure 2-11.4). We calculate *vertical projection* (Equation 1). Then, we find *left and right edge* of face as being the *local minima* within the range $[x_{center} - d_{1,\ max} / 2]$ and $[x_{center} + d_{1,max} / 2]$ (Figure 2-11.6, 7). We find horizontal edges by *Sobel* operator (Figure 2-11.8) and then find the *top of the head* as being the first horizontal line from top (Figure 2-11.9).

For *detecting facial features*, we calculate *horizontal projection* and find horizontal position of the eyes as being the local minima in range $[d_{2,min}\ d_{2,max}]$ (Figure 2-11.10). Eyebrows are the first horizontal edges starting from $d_{2,min}$. Then, find the beginning of the left and ending of the right one to get rid of the hair (Figure 2-11.11). Next, we erode and threshold, and then label the objects, sort them in descendent order of the area size (Figure 2-11.12). Set restrictions for area, position and length. Centroids are centre of the eyes. Get the coordinates *x, y* of each eye.

For *normalization*, we calculate the rotation angle of the eyes axis.

$$tg\alpha = \frac{\left| y_{eye1} - y_{eye2} \right|}{\left| x_{eye1} - x_{eye2} \right|},$$

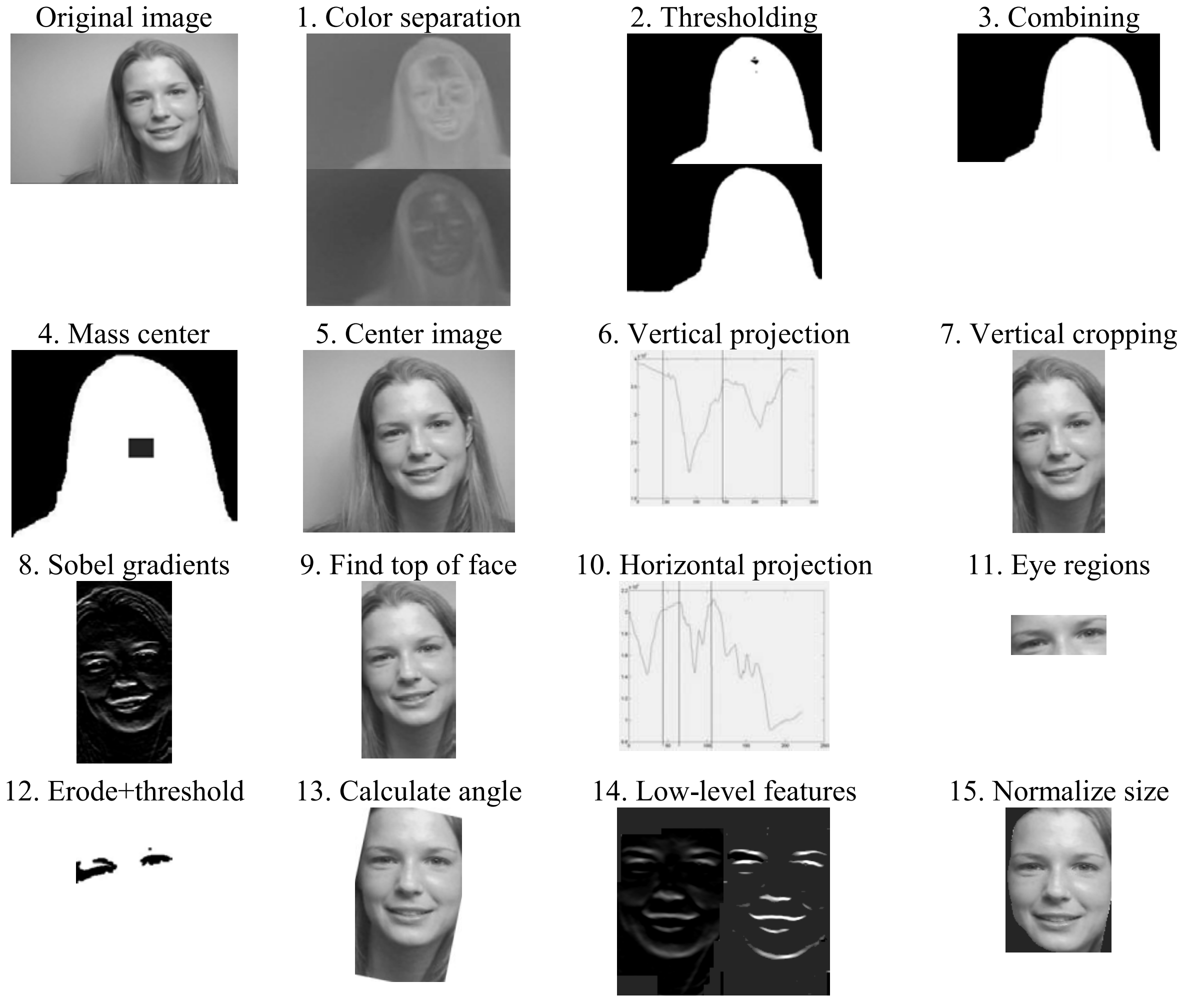
$$\alpha = arctg(tg\alpha)$$

Next, we rotate the image by angle α such that eyes axis becomes horizontal in the image plane (Figure 2-11.13).

Further, we are looking for the rest of the facial features. We use low pass filter to remove small edges. Get horizontal edges by Sobel and enhance the edges. Starting from eyes position, we find horizontal edges for: nostrils, upper and lower lip. Find chin by horizontal edge and horizontal projection.

We normalize faces *f* to have same size: *63 x 74* and in the end we verify the presence of the face by calculating *distance(f, F_s)* to be less than threshold θ.

[4] Otsu is a threshold selection method from gray level histograms (http://iul.cs.byu.edu/morse/550-F95/node25.html)



Original image

1. Color separation

2. Thresholding

3. Combining

4. Mass center

5. Center image

6. Vertical projection

7. Vertical cropping

8. Sobel gradients

9. Find top of face

10. Horizontal projection

11. Eye regions

12. Erode+threshold

13. Calculate angle

14. Low-level features

15. Normalize size

Figure 2-11: Illustration of the steps of face detection.

3. Face Recognition from Still Images

3.1 Introduction

With a wide range of successful applications such as commercial, information security, law enforcement and surveillance, the area of face recognition attracted researchers from disciplines as image processing, pattern recognition, neural networks, computer vision, computer graphics and psychology. Face recognition proved to be perhaps the most natural way of identification. Comparing to other intrusive physiological characteristics such as fingerprints, retina and iris patterns, hand geometry and voice, the face analysis is often effective even without cooperation or knowledge of the participants [38]. Although over than 30 years of extensive research has been conducted in this area, there still exist open research issues, the performance of the current algorithms being still far from that of human perception.

The goal of the face recognition is to identify or verify the persons present in the shots based on their facial features, despite of wide variations in pose, facial expressions and illumination changes [81, 38]. This topic has led to the new branches such as recognition from *still* and *video* images, while they can be applied for *frontal* or *profile* shots.

Although numerous literatures exists in the field, only main techniques in face recognition are briefly summarized in this thesis. Detailed surveys are recommended in [38, 81]. We will regard in the current section only several important methods for designing a face recognition module from frontal still images.

A typical face recognition problem involves the sub tasks shown in Figure 1-3. In a realistic environment, shots do not have always the same background, pose or facial expression and that is why we need reliable *face detection* in any kind of background. The next step would be to *extract features* from face regions but we might achieve this simultaneously with detection of faces. Finally, the *matching* module makes the classification decision.

Existing face recognition approaches can be classified into two broad categories: *analytic* and *holistic* methods, and they can be combined into *hybrid* approaches. Figure 3-1 presents an outlook over this classification and the approaches regarded in this section. We will shortly overview basic methods for analytic approach, and then focus more on the traditional algorithms for holistic approach, such as PCA (*Eigenfaces*) and LDA (*Fisherfaces*). Finally, we review the hybrid methods. Here we assume that an accurate face detection process has normalized all faces beforehand.

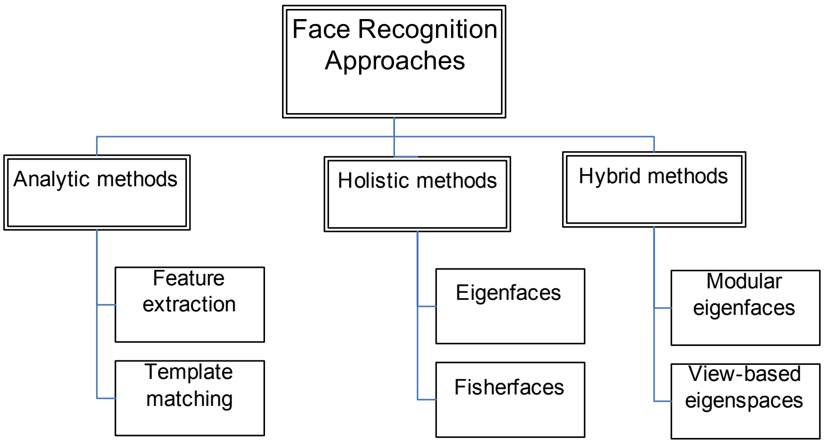


Figure 3-1: Classification of face recognition methods.

3.2 Analytic Methods

Early research on face recognition focused on *analytic* (or *feature-based*) methods that employ geometry of the facial features such as: eyes, nose, mouth, and eyebrows. First, we need to perform a *feature extraction* step as described in Section 2.2.2. Properties and relations between facial features (areas, distances and angles) form feature vectors that are used for discriminating the test face from the training samples. Typically, a high number of geometrical features are extracted in [10], in total 35 features including eyebrow thickness, vertical position of the eye center, a coarse description of the left eyebrow arches, nose vertical position and width, mouth vertical position, width, height of the upper and lower lips, radial description of chin shape, face width at nose position (Figure 3-2). It is clear that the performance depends on the feature extraction precision, and it would be rotation invariant because all distances are relative to other features. For a good accuracy, the features should be geometrically normalized, independent of the position, scale and rotation of the face in the image [54].

Recognition can be employed by *Bayesian* or *nearest neighbor* classifiers [10]. For the latter case, *Principal Component Analysis* is first employed to reduce the 35-dimensional feature vector space and then Euclidean distance is the most commonly used metric. When face detection was done by *template matching*, recognition is performed by computing the correlation between the templates from the test image and the trained templates from the face database.

Different algorithms have been compared in one of the most comprehensive survey about face recognition technology, at the time of writing [81]. Thus, the experimental results denote *Elastic Bunch Graph Matching* [72] approach as being one of the most successful. It is based on *Gabor wavelets* and *Dynamic Link Architecture* [40]. In this method, each face is represented by a graph whose nodes are local feature vectors calculated at different fiducially points from the face. Actually, these points correspond to the nodes of a coarse, rectangular grid placed over the image. The features are called *jets* and they consist of Gabor wavelet coefficients for different scales and rotation, which makes them robust to illumination and geometrical changes. Elasticity is required to be able to fit the graph to whichever face and thus, the problem of comparing two faces resumes to matching and adapting a grid over one image to the features of the other image.

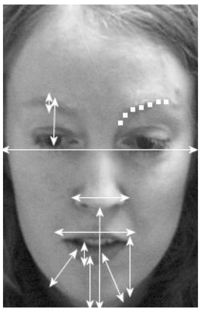


Figure 3-2: Relations between facial features

3.3 Holistic Methods

Recently research in face recognition focused more on *holistic*[5] methods that do not depend on detailed geometry of the faces. The main difficulty in recognizing faces by feature-based approach is that it is not easy to design proper face models due to large variability of face appearance, e.g. pose and light variance. Rather than constructing face models based on geometry, a face recognition system should learn the models automatically from a collection of training images, that is to learn what attributes of appearance will be the most effective in recognition [69]. Of course, enough training data should be available in order the system to account for variations in images.

These methods use global representation of faces by treating images as a vector of gray level intensities. Hence, this approach can be referred as *image* or *appearance-based*.

3.3.1 Motivation

It is obvious that exhaustive searching for visually similar images in the image space demand very large number of distances calculations because the size of the vectors is the number of pixels in the image. As the resolution of images increases, the dimensionality o the image space increases also. For example, only for 8x8 binary images, the whole image space has 2^{64} points.

We can treat a collection of *M* possible face images under different illumination, scale and position as a set of *M* points in the image space, which define a *manifold*[6] within the whole image space [69]. The problem of matching would be relatively easy if all images of faces are *clustered* in the image space and if this cluster is well separated of other objects clusters. In this case, a simple metric such as *Euclidean distance* could be used to determine the nearest face (*nearest neighbor*) within the face cluster:

[5] Emphasizing the functional relation between parts and the whole

[6] A collection, a multiple set of points, a group

$$\min_i d\left(x, \bar{x}_i\right), \forall i = 1,...,k \qquad (2)$$

where x is the test image, $\bar{x}_i$ are the existing training faces and k is the size of the training set. Although we limited the search to the face space, many calculations are still needed because of the high dimensionality vectors.

Face images, even with several transformations, are similar in general and it is expected that they are not randomly distributed in this huge space, but they will occupy a relatively small and distinct region in the space. Moreover, it is assumed that different people occupy different regions in the space [69] (Figure 3-3).

The holistic approach uses *statistical methods* to analyze the distribution of these points in the whole image space, and to derive an effective representation of them. This representation is in sense of features, which are not related to facial features. This way, face recognition becomes a matching problem between the extracted features of the test image and the ones extracted from the training set. Most common, *linear analysis* methods aim to find a lower dimensionality representation of the human faces subspace.

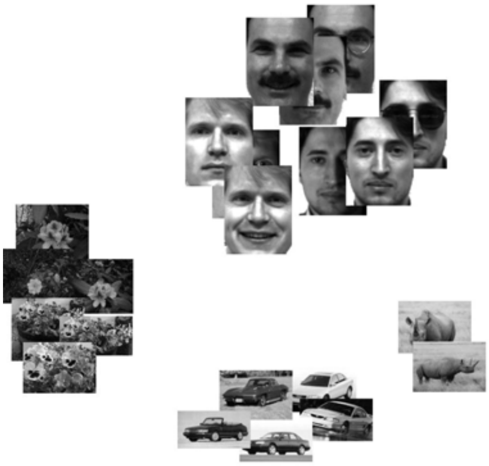
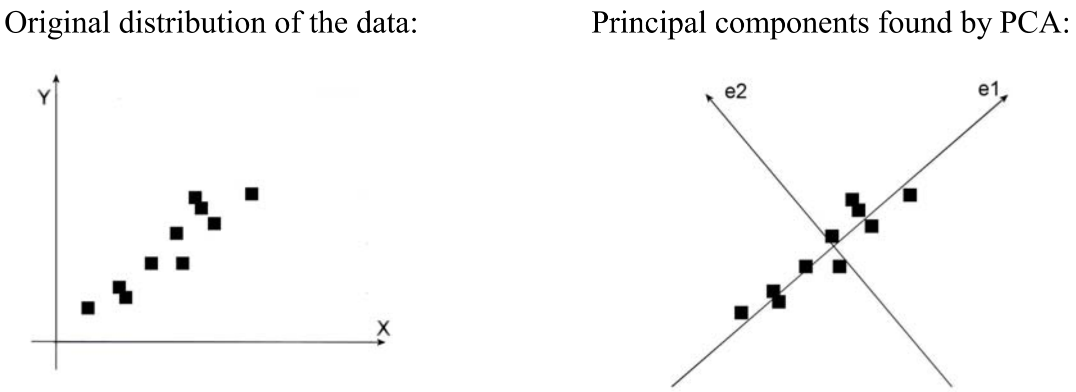


Figure 3-3: A representation of the image space.

3.3.2 Eigenfaces Method

Eigenfaces method [68] is perhaps the most common method based on holistic approach for face recognition. It employs *Principal Component Analysis* (PCA, also known as *Karhunen-Loève transform*) in order to analyze the distribution of the points in the image space, and to express their variation in a number of *principal components,* which is an orthonormal set of axes (Figure 3-4). For a thorough understanding of the method, we recommend a tutorial about PCA in [67].


Original distribution of the data:

Principal components found by PCA:

Figure 3-4: Principal Components in a 2D space.

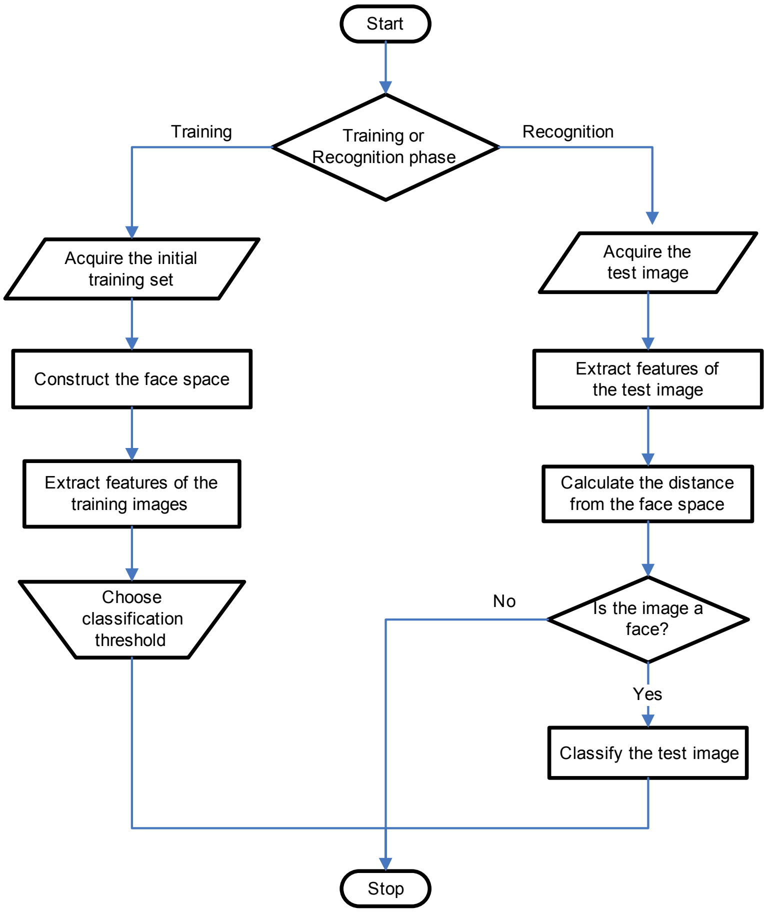

Start
Training or Recognition phase
Training
Recognition
Acquire the initial training set
Construct the face space
Extract features of the training images
Choose classification threshold
Acquire the test image
Extract features of the test image
Calculate the distance from the face space
Is the image a face?
No
Yes
Classify the test image
Stop

Figure 3-5: Flowchart of the Eigenfaces algorithm.

Figure 3-5 represents a flow chart of the method. We distinct two main phases: *training* and *recognition.* Training is an *offline* initialization procedure when we construct the *face space* for the training images by calculating their eigenfaces. The face space needs to be reconstructed whenever the train set changes because the covariation between images will also change. We extract *features* from the training images as being the projection coefficients onto the face space. They are needed in order to classify the test images as belonging to the persons present the training set. Next, we choose the classification threshold.

Recognition is an online step, which is performed for every test image. To recognize a face, first we extract the features with the respect to the face space. We calculate distance from the face space to verify if the test image is a face, in general, and if not, the input image is classified as "unknown". We match the images according to the extracted features of the test image and the training set. We will detail these steps in the following, but first we will define the face space.

Face Space

Let us consider

$$\{x_i\},\ i = 1 \ldots M$$

as being a set of M training images (Figure 3-6) represented by a matrix X where

$$X = [x_1 x_2 \ldots x_M]$$

and X is of dimension N×M, where x_i is the vector representation of the image and N is the number of the pixels from the image.

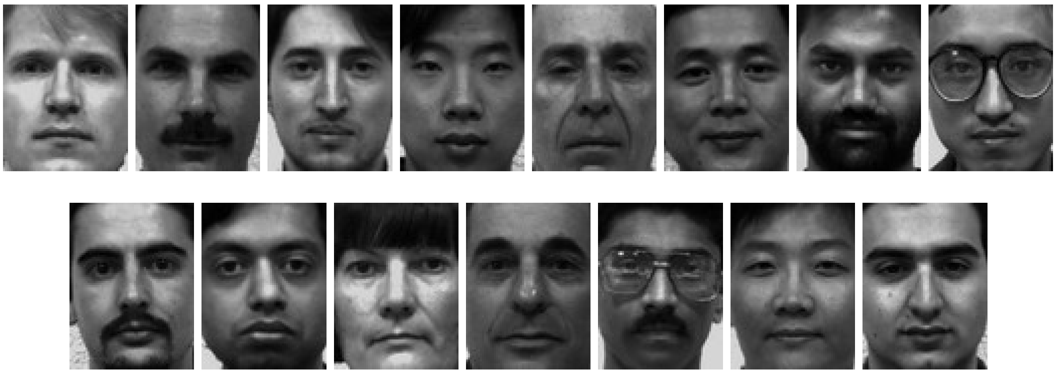


Figure 3-6: A set of training images from Yale database [74].

In mathematical terms, the principal components of a distribution of a set of faces X are the *eigenvectors* of the *covariance matrix* of that set. We recall from statistical analysis that if the standard *deviation* or *variance* measure the spread of data in one dimension, then the *covariance* reflects the variances of the dimensions from the mean with respect to each other. It is always measured between two dimensions and the covariance between one dimension to itself is the variance itself [67].

For face recognition, the dimensions are the training face images represented by a column vector of size *N*, and therefore, the *covariance matrix* represents all the different covariances between all the face images. The matrix is order *N* and square, and since this measure is symmetric, then the matrix is symmetric with respect to the main diagonal, which is zero. We can calculate in total *N* number of eigenvectors, each having size *N*.

Because the eigenvectors have the same number of pixels from the images, they can be reshaped in a 2-dimensional array to have a visual representation. They are linear combinations of the face images, thus they have a face-like appearance as shown in Figure 3-7, and they are therefore called *Eigenfaces* [68].

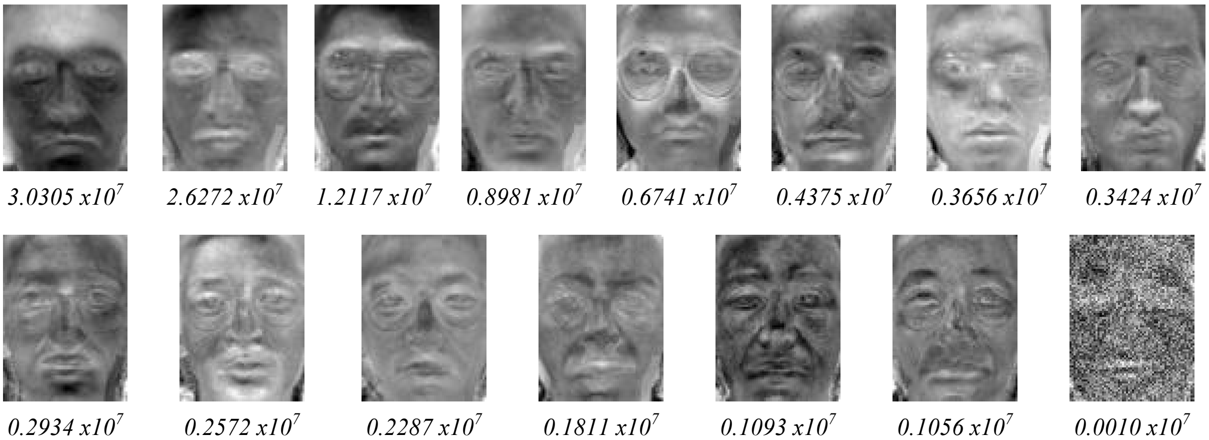


Figure 3-7: Fifteen eigenfaces of the training set from Figure 3-6 sorted in descending according to their eigenvalue.

One eigenface shows how the face images are related to it, by calculating the significant variation among faces with respect to it. Their associated eigenvalues reflect the significance of their encoding and therefore we sort the eigenvectors descendant according to their eigenvalue in order to ignore the eigenfaces of less importance. Thus, the first eigenface encodes to the most variation between faces in one direction, while the rest of them correspond to the remaining variations. We consider only those eigenvectors associated with the largest eigenvalues to be the axes (principal components) of the image subspace, which is called *face space* or *eigenspace* [69]. Rest of the eigenvectors is discarded.

The face space spans onto the eigenfaces calculated from the training set, and becomes the new *feature space*. Moreover, it has significantly smaller dimensionality than the original image space because of considering only the largest eigenvectors.

Next, all images are linearly transformed to the face space by projecting them, aiming to *minimize the mean squared projection error.* Thus, each image from the whole image space is mapped to a point in the face space. Therefore, we will represent the faces by a *lower dimensionality feature vector* that contains the projection coefficients (weights) on each axe of the face space (Figure 3-8).

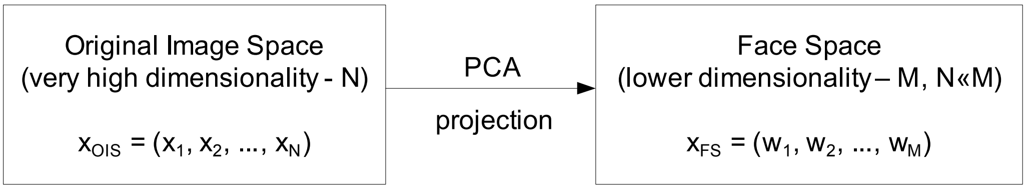

Original Image Space
(very high dimensionality - N)
$x_{OIS} = (x_1, x_2, ..., x_N)$
PCA
projection
Face Space
(lower dimensionality– M, N«M)
$x_{FS} = (w_1, w_2, ..., w_M)$

Figure 3-8: Mapping of one point from the original image space into a point from the face space.

Construct the Face Space

We calculate the mean face image from the set *X* as (Figure 3-9)

$$\bar{x} = \frac{1}{M}\sum_{i=1}^{M} x_i$$

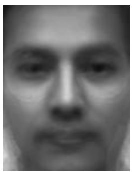


Figure 3-9: The mean image of the set from Figure 3-6.

We denote by *Y* the difference matrix between *X* and the mean face image:

$$Y = X - \bar{x} = \left[\left(x_1 - \bar{x}\right)\left(x_2 - \bar{x}\right)\ldots\left(x_M - \bar{x}\right)\right] = \left[y_1 y_2 \ldots y_M\right]$$

The *covariance matrix* of the distribution of faces is order *N*×*N*:

$$C = \sum_{i=1}^{M}\left(x_i - \bar{x}\right)\left(x_i - \bar{x}\right)^T = \sum_{i=1}^{M} y_i\, y_i^{\;T} = YY^T$$

In PCA, the projection U_{opt} is chosen to maximize the determinant of the total scatter matrix after applying the linear transformation U^T. The eigenvectors are the column vectors of the matrix U_{opt} where

$$U_{opt} = \arg\max_{U}\left|U^T C U\right|$$

and satisfy the property:

$$Cu_i = \lambda_i u_i \Leftrightarrow \left(Y Y^T\right)u_i = \lambda_i u_i \quad (3)$$

where λ_i are the eigenvalues corresponding to the eigenvectors u_i. The matrix *C* has an order of *N,* and therefore can have *N* eigenvectors. In practice, we have to calculate a very large number of big dimension eigenvectors, which is an intractable problem [69]. For example, if we consider images of size 100×100 we have to calculate 10,000 eigenvectors. It was proved in [68] that there are only *M-1* non-zero eigenvalues of an *M*×*M* matrix, where *M* is the number of the faces in the image space, and in general, the size of the training set is significantly smaller than *N*:

Let us consider the eigenvectors v_i of the matrix $D = Y^T Y$ so that:

$$Y^T Y v_i = \mu_i v_i$$

If we multiply both sides at left by matrix *Y*, we obtain:

$$Y Y^T Y v_i = Y \mu_i v_i \Leftrightarrow \left(YY^T\right) Y v_i = \mu_i \left(Y v_i\right)$$

From this we observe that $Y v_i$ are in fact the eigenvectors of matrix $C = Y Y^T$ from (3), because μ_i are scalars.

We showed that, first we can calculate *M* number of eigenvectors of matrix $D = Y^T Y$ and then get the eigenvectors u_i of the covariance matrix $C = Y Y^T$ by a linear combination of the eigenvectors v_i:

$$u = Yv \quad \Leftrightarrow u_i = Y v_i, \forall i = 1 \ldots M$$

The eigenvectors of a large matrix *C* are equal to the eigenvectors of a smaller matrix *D,* pre-multiplied by *Y*. By this observation, the complexity of the algorithm reduces significantly, from calculating *M* eigenvectors instead of *N*.

We reduced the dimensionality of the image space from *N* to *M*, in other words, from the number of pixels in the image to the number of eigenfaces. The *M* eigenvectors u_i are column vectors of dimension *N*.

Next, we sort the eigenvectors in descending order according to their associated eigenvalues. Although *M* is much smaller than *N*, for a large training set we will still have large feature vectors, thus a sparse feature space. PCA can encode the data by choosing a smaller number of the components (eigenvectors), ignoring the components with less significance (eigenvalues) [67]. Of course, the problem is how to select the optimal number of components. We will see later in the experiments results in Section 7 that a relatively small number of eigenfaces is enough for recognition, since the exact reconstruction of the test image is not necessary for classification purpose.

Figure 3-7 shows also the eigenvalues associated to their eigenfaces obtained during experiments. We can see that the first three eigenvalues are the biggest, while the last one is even 1000 times less than the previous ones. This means that the first three eigenfaces encode most of the variation, while the last one is arbitrary and irrelevant, reflected by its eigenvalue and appearance. We can ignore the last eigenface so the face space will span onto *M-1* dimensions. The first three eigenfaces are the most discriminant and we found in our experiments that they are enough for recognition on specific test conditions.

In principle, if all eigenfaces were selected, then the reconstruction of the original image would be lossless, that is SNR (*Signal-to-Noise Ratio*) should be infinite. Moreover, the smaller number of eigenfaces chosen, the bigger would be the reconstruction error (Figure

3-12). In practice, during the experiments, we obtained a SNR of 65dB between one image from the training set and its reconstructed image using all the set of eigenfaces. We explain this by the rounding and gray levels normalization errors.

Feature Extraction

We represent the training set images in the face space by calculating the projections of all images onto all eigenfaces (axes) as:

$$w = u^T Y$$

The training faces can be reconstructed by the formula:

$$x_i^R = \overline{x} + uw$$

$$\Leftrightarrow x_i^R = \overline{x} + \sum_{i=1}^{M'} u_i w_i = \overline{x} + u_1 w_1 + u_2 w_2 + \ldots + u_{M'} w_{M'}$$

The procedure is similar for any test image to be recognized. First, we reshape the image as a column vector, denoted by x_T and extract the mean image of the training set $\overline{x}$:

$$y_T = x_T - \overline{x}$$

We project y_T on the face space by:

$$w_T = u^T y_T$$

And it can be reconstructed by:

$$x_T^R = \overline{x} + uw_T$$

$$\Leftrightarrow x_T^R = \overline{x} + \sum_{i=1}^{M'} u_i w_T^i = \overline{x} + u_1 w_T^1 + u_2 w_T^2 + \ldots + u_{M'} w_T^{M'}$$

where w_T^i are the projections of the test image x_T in the face space.

Figure 3-10 shows an intuitive representation of the face space. The projection coefficients for all face images represent their coordinates in the face space and they are considered the features for classification. These holistic features are in contrast to the features extracted by the geometric-based approaches for face recognition, as they store the relevant information about the geometry of the face. The weight w_i encodes how far are the images from the mean face on the dimension u_i, while the eigenfaces encode the way to morph the mean face into specific faces [49].

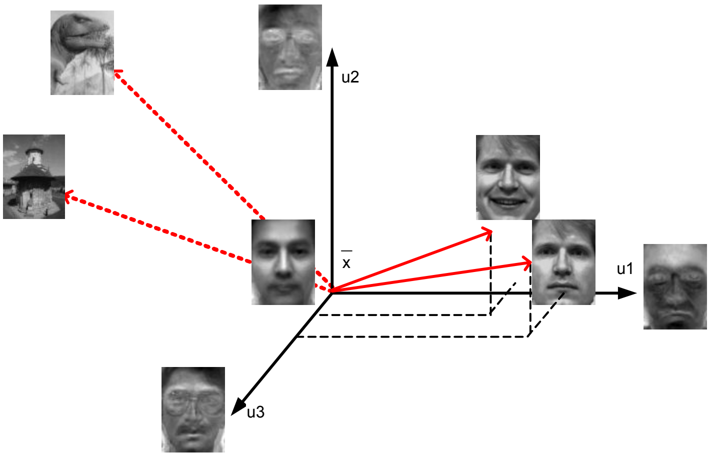
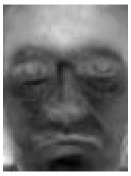
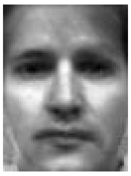


Figure 3-10: Representation of a face space spanned on first three eigenfaces, few sample projected faces, and two random images.

Let us consider an example of a test image x_T as shown in Figure 3-11, and the face space spanned over the first three eigenfaces. We can reconstruct the image as:

$x_T =$

$x_T^R =$ $+ w_T^1$  $+ w_T^2$ $+ w_T^3$ $=$ 

For all set of eigenfaces:

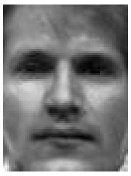
$x_T^R = \overline{x} + u_1 w_T^1 + u_2 w_T^2 + \ldots u_{14} w_T^{14} =$ 

Figure 3-11: Reconstruction of a face image as a linear combination of eigenfaces.

The weights describe the contribution of each eigenface in representing the input face image and the more information we use for decoding the images, the smaller the reconstruction errors would be, because more eigenfaces contribute for the facial features to become more evident. Figure 3-12 shows the losses during the encoding of the first face from the training set, reconstructed with a variable number of eigenfaces.

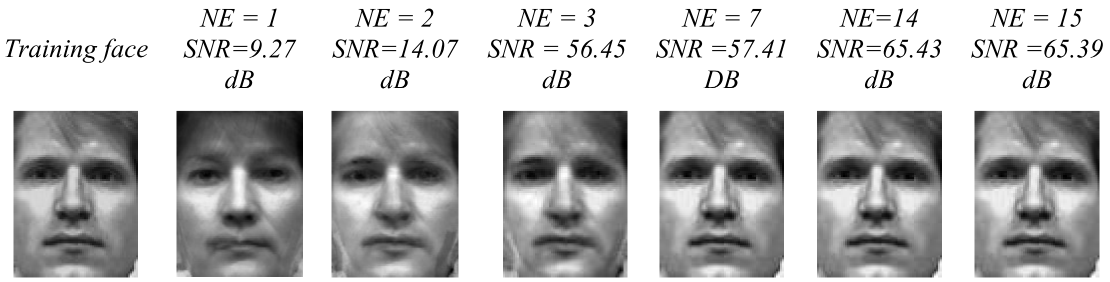


Figure 3-12: Reconstruction errors of a face from the training set using different number of eigenfaces (NE).

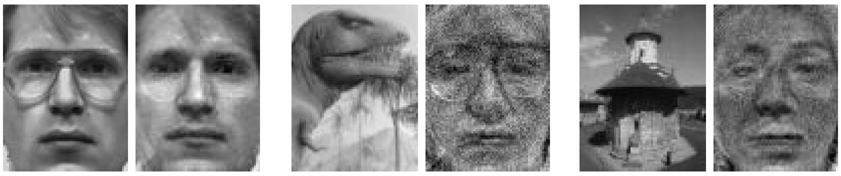


Figure 3-13: Different images reconstructed from the face space.

As we can see from Figure 3-13, face images do not change radically when projected into the face space, while the projection of random non-face images appear different from the originals and even have eigenface-appearance because they are reconstructed as a linear combination of them, which encode the most variation, so the result should have a face appearance. We emphasize that the face space is a representation of the manifold of points from image space, which correspond to the training face images.

If test images are very different from the training set then they do not belong to this manifold, moreover, they are far from it and the face space does not reflect their distribution. This can be described by calculating the distance from the face space, which is in fact the distance from the mean image in the center of the space, and represents how different is the test image from the mean face. In other words, it is a measure of “faceness” of one image (Figure 3-10).

Eigenfaces approach has also been used for *face detection* [68] by applying a window scanning to detect the presence of a face in the local image (see Section 2.3). If the distance is greater than a threshold experimentally chosen, then it does not represent a face and we can classify the image as *unknown*.

Matching

Considering the image containing a face and having feature vectors, face recognition process becomes a typical pattern recognition task of matching images, which is to find out the class in which the face image belongs to. A common approach is to apply *nearest neighbor* criterion to find the training face from the face space that is closest to the test image or, in mathematical terms, to find the closest vector from feature space that minimizes the distance:

$$d = \min_{i=1\ldots M} d\left(\overline{w_i}, \overline{w_T}\right)$$

where $\overline{w_T}$ and $\overline{w_i}$ are the feature vectors of test and training face images and M is the size of training set. During the training phase, each image and its features were assigned to some known individual. The test face is classified as belonging to person i if the minimum distance d_i is below a given *threshold* θ, otherwise as being an "unknown" person (open-set identification). This classification threshold is experimentally chosen according to the *false acceptance/rejection* rates imposed by a specific application.

The most common metric used for calculating the similarity between feature vectors is the *Euclidian distance* [81, 68, 38]:

$$d\left(\overline{w_i}, \overline{w_T}\right) = \left\|\overline{w_i} - \overline{w_T}\right\| = \sqrt{\sum_{k=1}^{n}\left(w_i^k - w_T^k\right)^2} \text{ where } n = \left|\overline{w_i}\right| = \left|\overline{w_T}\right|$$

Conclusions

The eigenfaces method is simple and recognition rates as high as 96% have been reported in [68] for lighting variation, 85% for orientation variation, and 64% for size variation. The common drawbacks of Eigenfaces method are the poor recognition rates for illumination and pose variation conditions. An intuitive workaround for this would be to train the system with different views of faces that are similar to the test conditions: faces at different orientation and illuminated from different angles [56]. However, for a large number of points for each individual class in the face space, PCA may not provide a good discrimination between classes after the projection, so we have to find another representation of the face space that separates intra-class and inter-class variations of faces.

The eigenfaces approach has been the basis for several other algorithms, from which we mention only few. *Second-order eigenfaces* [71] uses not only one set of eigenfaces for the original image but also the set of second-order eigenfaces of the residual images defined by the differences between the original and the reconstructed images obtained from the set of eigenfaces.

Eigenphases method [64] performs PCA in the frequency domain on the phase spectrum of the images. In *Self-Eigenfaces* approach, each person is modeled with a different set of eigenfaces. In *Composite PCA* [27], face images of size *NxN* are divided into L^2 blocks of size *N/L x N/L* before performing eigenfaces method. *Probabilistic eigenfaces* [52] employ probability density estimation by decomposing the input space into two mutually exclusive subspaces, the principal and its orthogonal subspaces, leading to dual eigenfaces (extrapersonal and intrapersonal). This approach is more robust to variations in lighting and facial expressions.

Discussion

The idea to use multiple views for each individual in the training set leads to bad discrimination in Eigenfaces method because the dimensionality reduction is performed by PCA, which finds the directions that maximize the variance across all images. As Moses et al. [53] stated, the biggest drawbacks of the appearance-based methods is that

“the variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to change in face identity”

This means that the eigenfaces encode the variation due to lighting and pose [8], whereas our goal is to encode variation across classes of individuals.

We recall that the basic assumption is that the set of all face images forms a cluster in the image space. Starting from this idea, we are interested to see if the face images of the same individual form clusters inside the face space as well. In order to demonstrate the PCA’s drawback about poor discrimination, we performed experiments to see how a collection of face images clusterizes in the eigenspace. We considered a set of cropped images from Yale database [74] as training with C=15 subjects and M=11 different image conditions for each pose, such as facial expression and illumination variation (left and right side light sources). We constructed the eigenspace corresponding to the whole C×M=165 training set and we considered all number of eigenfaces for an accurate discrimination. We represented the sample set by their projections onto face space and applied *K-means clustering* [47] algorithm using Euclidean distance for the feature vectors and considering 15 clusters (the number of individuals).

We introduce the notion of *face-centroid* as being the center of a set of faces that form a cluster in the face space, or in other words, the most representative sample from that set. As expected, the poses for each subject under different facial expressions were clustered well in a face-centroid very similar to individual’s normal face (Figure 3-14). However, samples from the three specific light conditions clusterized in 3 different clusters and have the appereance of the mean image under the same light conditions.

Experiments on ATT database [4] with 40 subjects on 10 different head positions performed good clusters for face images. Some of them were very close to one of the training images and some were an average between individual’s faces.

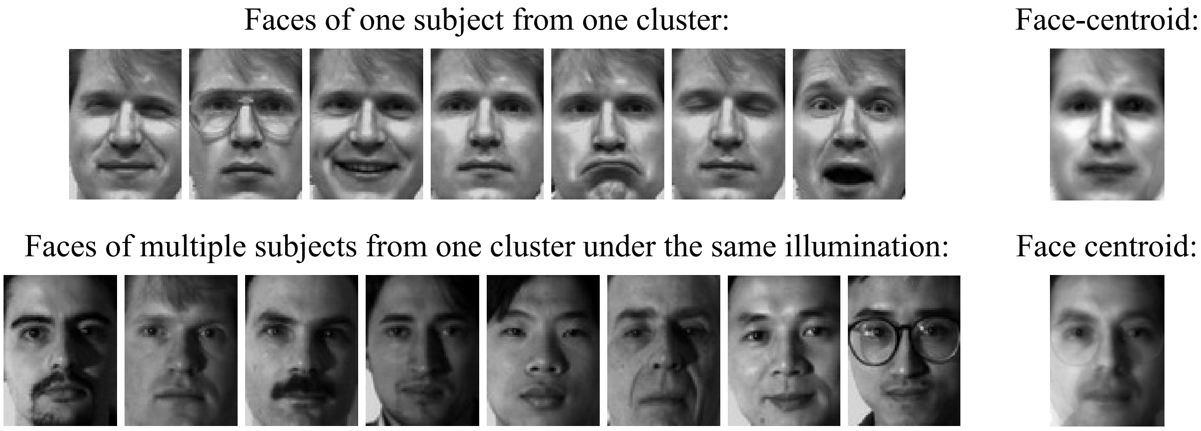

Faces of one subject from one cluster:
Face-centroid:
Faces of multiple subjects from one cluster under the same illumination:
Face centroid:

Figure 3-14: Samples from the Yale database and clusters obtained by k-means algorithm from the face space for Eigenfaces method.

If we consider different instances of the same person's face with variations in lighting, pose or facial expressions, to be in the same class, and faces of different subjects to be from different classes, after we build the face space we need to preserve this classification. PCA does not take into account the underlying class structure (i.e. class membership of each image) and it finds the projections as all the samples are from one class of images.

3.3.3 Fisherfaces Method

Fisherfaces method [8] derives from *Fisher's Linear Discriminant Analysis* (FLD or LDA) [25] and it works on the same principle as the Eigenfaces method: it provides a linear description of the face subspace by reducing the dimensionality of the image space.

The objective of LDA is to perform dimensionality reduction while preserving as much of the class discriminatory information as possible by finding directions along which the classes are best separated. In contrary, PCA is aimed at representation (encodes the maximum variance along one component) because a PCA projection does not create an optimal discrimination for different classes, as shown in Figure 3-15. In our case, LDA distinguishes better the variation due to identity, from variation due to other sources such as illumination and expression.

PCA finds the projections that maximize the determinant of the total scatter matrix of the data. LDA calculates a set of optimal projections that maximize the ratio of the determinants of the *between-class* over *within-class* scatter matrices. The between-class scatter matrix, also called the *inter-personal*, represents variations in appearance due to differences in identity while the within-class scatter matrix, also called *intra-personal*, represents variations in appearance of the same individual due to different poses [54].

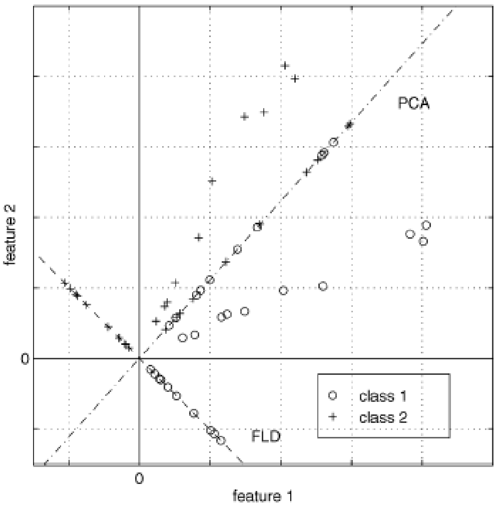


Figure 3-15: A comparison of the projections found by PCA and LDA [8].

Let us consider a training set of *c* classes of individuals. We compute the between-class (S_B) and within-class (S_W) scatter matrices as follows:

$$S_B = \sum_{i=1}^{c} N_i \left(\overline{x_i} - \overline{x}\right)\left(\overline{x_i} - \overline{x}\right)^T$$

$$S_W = \sum_{i=1}^{c} \sum_{x_K \in X_i} \left(x_K - \overline{x_i}\right)\left(x_K - \overline{x_i}\right)^T$$

where $\overline{x}$ is the mean of all images, $\overline{x_i}$ is the mean image of class X_i, , and N_i is the number of samples from class X_i.

If S_W is non-singular, then LDA determines the projections that are the columns of the transformation matrix W_{opt}, which maximizes the ratio of the determinant of the S_B to the determinant of S_W as:

$$W_{opt} = \arg\max_W \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} = \left[w_1 \;\; w_2 \; \ldots w_{c-1}\right]$$

where $\{w_i | i = 1,\, 2, \ldots, c-1\}$ are the generalized eigenvectors with non-zero eigenvalues of S_B and S_W that satisfy:

$$S_B w_i = \lambda_i S_W w_i \Leftrightarrow S_W^{-1} S_B w_i = \lambda_i w_i,\; i = 1,\, 2,\, \ldots,\, c-1$$

We note that there are only *c*-1 eigenvectors [23]. If we consider the example of two-dimensional space from Figure 3-15, only one projection is enough to discriminate between the two classes of samples. By this transformation, we reduce the dimensionality to minimum *c*-1 but as in the Eigenfaces method, we select only the *m* eigenvectors that have the largest eigenvalues associated, where $m < c$-1. Matrix

$$S_W \in \Re^{nxn},$$

where n is the number of pixels in the image.

$$rank(S_W) \leq N - c$$

because there can be only N-c columns that are linearly independent (one image from one class is independent on the other N-c images), and therefore the within-class scatter matrix is always singular .

In order to avoid this, *Fisherfaces* method applies first PCA to reduce the dimensionality to N-c, thus the new within-class scatter matrix is $S_W \in \Re^{(N-c)\ x\ (N-c)}$ and non-singular. The second step reduces the dimensionality to c-1 by performing standard FLD. Here, W_{opt} can be calculated as:

$$W_{opt} = W_{FLD}^T W_{PCA}^T$$

where $W_{PCA} \in \Re^{n\ x\ (N-c)}$ are the eigenvectors of the total-scatter matrix of data, from which we select only the first N-c eigenvectors. The term $W_{FLD} \in \Re^{(N-c) x m}$ contains the generalized eigenvectors of the matrices $W_{PCA}^T S_B W_{PCA}$ and $W_{PCA}^T S_W W_{PCA}$:

$$W_{FLD} = \arg\max_{W} \frac{\left| W^T W_{PCA}^T S_B W_{PCA} W \right|}{\left| W^T W_{PCA}^T S_W W_{PCA} W \right|}$$

The eigenvectors of this method are called *Fisherfaces* and we show them in Figure 3-16. Reconstruction of faces, feature extraction and classification can be done in the same way as in the eigenfaces method.

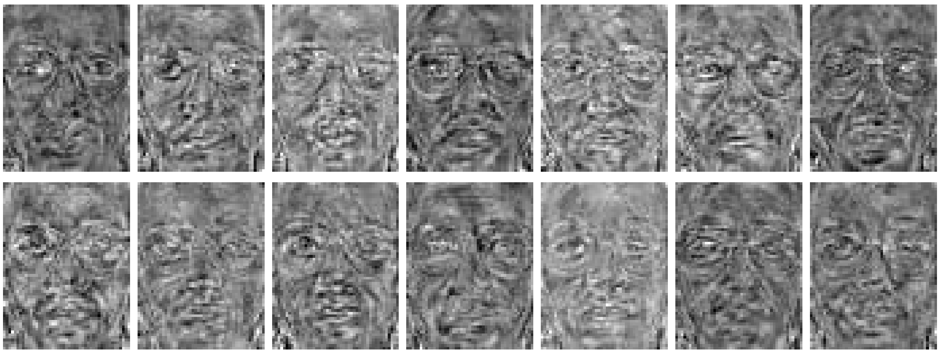


Figure 3-16: Fourteen Fisherfaces constructed from 15 subjects from Yale database each with 11 poses.

Discussion

In order to prove that Fisherfaces method finds a better representation of the face space by preserving the class discriminatory information of the training images, we performed again K-means clustering algorithm. We considered Yale database [74], which consists in 15 subjects with 11 different shots per class, in different illumination and facial expression variation. Obvious, we considered the number of clusters to be the same as the number of subjects, and we represent the resulting codebook in Figure 3-17. If the person from Figure 3-14 was classified into many clusters by PCA representation, we observe in this case that the subject is clearly separated from the other classes.

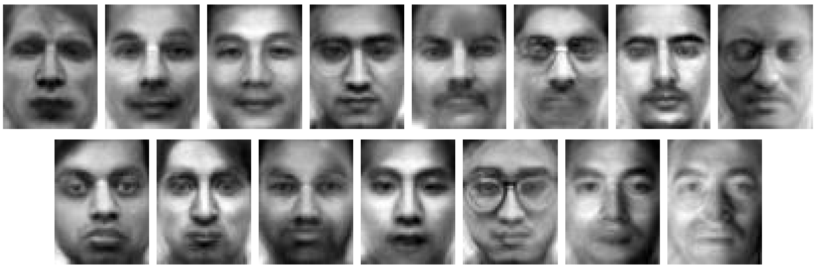


Figure 3-17: Codebook resulted from clustering the face space represented by Fisherfaces

One drawback of the LDA method is its high computational complexity and large memory requirements to calculate the between-class and within-class scatter matrices. Furthermore, it also needs many images for each class in the training process for good discrimination. Nevertheless, when the number of samples is large and representative for each class, LDA provides better recognition rates than PCA under illumination variance in [8] and in our experiments. We will make a more comparison between Eigenfaces and Fisherfaces methods in the experiments section.

Another approach used in face recognition is *Independent Component Analysis* (ICA) [6] that is a generalization of PCA. It is a technique for extracting statistically independent variables from a mixture of them by separating the high-order moments of the images in addition to the second-order moments.

3.4 Hybrid Methods

The holistic methods consider the global information of images and treat them as simple vectors of gray level intensities, thus they do not use any knowledge about the geometry of the face. In fact, these methods are commonly used in computer vision for object recognition in general as well.

We noticed in the previous sections that the Eigenfaces method performs badly in pose and illumination changes, and the Fisherfaces method improved the discrimination but it is still not robust enough. In order to increase the performance under the mentioned test conditions, hybrid methods have been proposed to combine the strengths of the analytic and holistic methods by using facial features and the whole face region for recognition, at the same time.

The Eigenface method was extended in [56] to *view-based eigenspaces* and *modular eigenfaces*. As we saw in Section 3.3.2, PCA will encode also the variation due to head positions while Eigenfaces method works well when frontal images are used. Based on this observation, the view-based approach tries to eliminate variation between poses by constructing separate eigenspaces for each different view, so that the collection of images on a different head position has their own eigenspace to represent only the identity variation (Figure 3-18).

Figure 3-18: Sample images for two different eigenfaces.

During training, all images have to be classified according to their orientation in order to assign them to the corresponding eigenspace. In testing phase, this can be done by determining the distance from each eigenspace and only the eigenspace from the same viewpoint will be used for recognition.

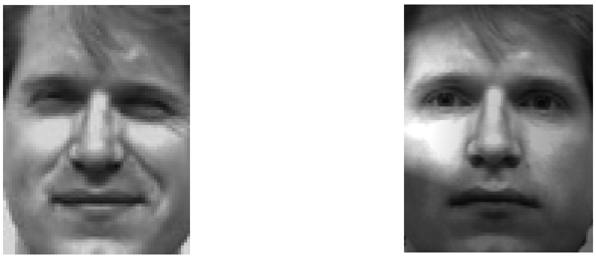


Figure 3-19: Example of changes in facial expression (left) and illumination (right).

When there is variation in images due to illumination or facial expressions, only some of the face regions will vary and the rest of the regions will remain the same as in the neutral expression and normal illumination images. For example, in Figure 3-19 we can recognize that person only by his nose and mouth because eyes are closed, or we can still distinguish his nose and eyes. As people have distinct facial features, it might be a good idea to match faces only by those regions under normal conditions, not by the whole faces. In other words, we would make a *local* and *modular* recognition of faces by using only the most discriminant features between people such as eyes, nose and mouth, which are not altered in the image. First, it detects the facial features to select the region of interest as shown in Figure 3-20, and then constructs eigenspaces to encode each feature from all the training set by creating *eigeneyes, eigennoses* and *eigenmouths* [56].

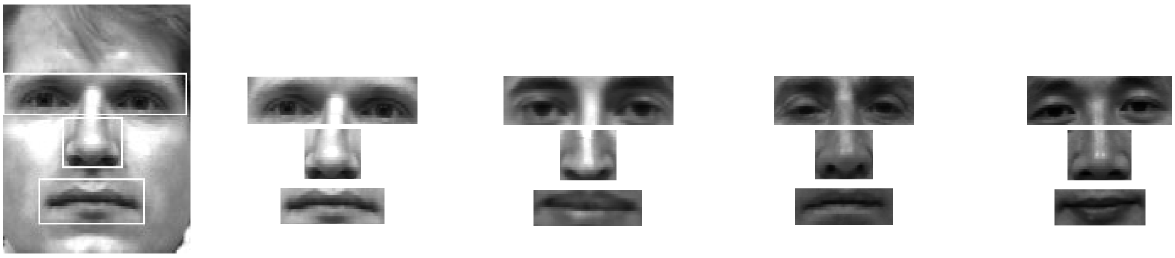


Figure 3-20: Example of facial features regions for modular eigenspaces.

After training, facial features can be detected by common geometrical methods but they can also be combined with the image-based methods by calculating the distance from each feature space. Recognition is similar to template matching from the analytic approach (Section 3.2), but this time they are *eigentemplates* [56]. Brunelli and Poggio found out in [10] the following discrimination power between facial features, sorted by decreasing performance: eyes, nose and mouth. The matching is performed for every specific feature aside that can be extracted from the test image.

Different techniques can be used to integrate all the similarity scores for each features to obtain a global matching score. In *voting,* the identity is assigned to the person for whom it was found the most similar feature, adding all the features scores, or adding but using a different weight for each feature that is the same for all people.

This method has the advantage that the recognition can be performed only on a single feature, even the rest are occluded, changed or altered by light conditions. It does not need so accurate fiducially points to be detected, and at the same time, it eliminates the difficulties from correlation of the analytical methods based on parameterized templates matching. Nevertheless, if changes are too significant in the image, facial features cannot be detected so the method would fail.

4. Face Recognition from Video Sequences

4.1 Introduction

Face recognition from video originated from still images. Video sequences provide for face recognition algorithms a large amount of faces covering a variety of pose conditions. This thing is benefic for the holistic and hybrid methods because they have high recognition rates only if geometrical and light conditions of test images match to those in the training set.

One approach for face recognition from video is to detect first the faces and then to employ still-image based recognition methods for each frame separately. Confidence scores are calculated for each frame and then we make the overall decision from the individual recognition results, e.g. by majority voting (Figure 4-1). Other approaches include *spatiotemporal* methods that exploit also temporal information such as *trajectories* of facial features [81].

Video sequences are preferred over the still images because they facilitate the face detection issue using motion as cue. One of the early attempts used pixel-based change detection procedures based on difference images [81]. This section presents only still-image methods.

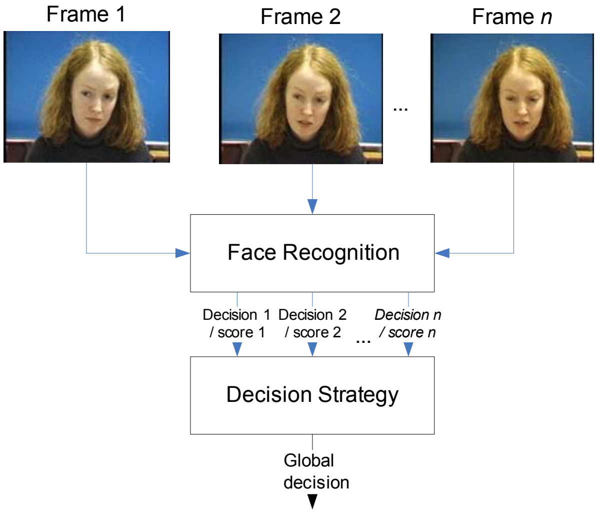


Figure 4-1: Face recognition from video sequence.

Still-image methods deal with huge amounts of frames that imply large memory requirements. For a MPEG sequence of 2 minutes sampled at 25 frames per second (fps), the total number of frames is 3,000. Moreover, there are no significant changes in the same face from two consecutive frames because human face motion is not so fast, so the system would be over-trained, with a high degree of redundancy. It is therefore better to prune the video sequence by sampling, that is processing only every N^{th} frame, or by detecting only the frames that represent changes in facial expression and head position. For the latter, one method is based on pixel change detection from the difference between two consecutive frames. If the difference is greater than a threshold, then the frame is processed. The difference should be calculated only for the face region because otherwise we will detect also the changes in the background.

If the person in the video is speaking, changes in facial expression and especially in lips will appear, which might worsen the face recognition performance. On the other hand, lip movement can also characterize an individual as well as the content of the individual is speaking. Information inherent in lip movement has therefore been exploited by another area of biometrics: *visual speech and speaker recognition.* Here, the lips contour motion represents the visual features. This is also called *lip-reading* and it requires location estimation and tracking of speaker's mouth or lips. We recommend references [26, 58, 9, 16, 44] for more information on this topic.

Comparing to still image recognition, the main drawbacks of video recognition are the *low image quality* and *small sizes of the faces* captured from video [81]. In addition, only fast algorithms for detection and matching are suitable, due to the large number of frames to be processed.

4.2 Face Sequence Clustering

Pruning the video sequence still results in a redundant amount of data. By *most representative faces* from a video sequence, we understand those faces that cover all their possible variations along the video. This sub section deals with the selection of them.

The face images form a manifold in the whole image space and the individual faces clusterize in the global face space, see Section 3.3. Based on this idea, a straightforward approach is to clusterize the initial set of faces in individual's own space, similar to [21, 30, 39].

Let us consider a set of *N=87* training faces for an individual acquired from a video sequence, as shown in Figure 4-2:

$$F = \{F_1, F_2, \dots, F_N\}$$

where $F_j, j = 1 \dots N$ are the vector representations of the faces in the high dimensional image space.

The distribution of these vectors in the space is very sparse and clustering would be impracticable because of the large dimensionality of the vectors[7]. Because of this, we need to find a more suitable representation of the face space, as the one from the *Eigenfaces* method. PCA gives an optimal representation in terms of mean square error, by encoding the variation between different poses of the same person. LDA is aimed for discrimination and we cannot apply it here because we do not know all the possible views of the faces that the subject may have.

[7] size of the vectors is the number of pixels from the image

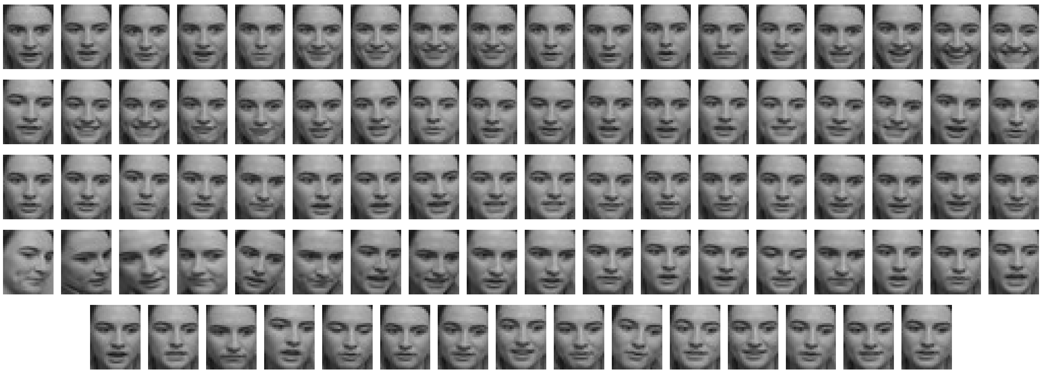


Figure 4-2: Face sequence extracted from Cuave database [20].

We construct a separate low-dimensional face space (SE = self eigenspace) for each subject in which we map all their faces from the training video-sequence F into F^{SE}.

$$F^{SE} = \left\{ F_1^{SE}, F_2^{SE}, \ldots, F_N^{SE} \right\}$$

where $F_j^{SE}, j = 1 \ldots N$ are the representation of the faces in the eigenspace, much smaller size vectors[8]. Authors of [2] call this as *self-eigenspace* and use the reconstruction error of the test image projected on the spaces of all subjects as the classifier. We do not consider this suitable for video-sequences recognition because of the expensive reconstruction errors computations (calculate MSE between each image and its reconstruction for all frames).

We clusterize the points in the self-face space by K-Means clustering algorithm and we consider the *face codebook* as being the centers of the clusters that form in the self-eigenspace:

$$C = \left\{ C_1, C_2, \ldots C_K \right\}$$

where K is the number of the selected face samples for training, and C_i are the vector representation of images in face space. The codebook C should describe best the training set, by minimizing the *average distortion* between F and C:

$$D\left(F^{SE}, C\right) = \frac{1}{N} \sum_{i=1}^{N} \min_{j=1\ldots K} d\left(F_i^{SE}, C_j\right)$$

where d is a metric defined over the face space, and usually is the Euclidean distance. They represent "means" of different classes of views for each individual. Furthermore, we can select the training set T_j as being the K closest images from the sequence that are closest to the codebook (Figure 4-4):

$$T_j = \min_{i=1\ldots N} d\left(F_i^{SE}, C_j\right), \forall j = 1 \ldots K$$

[8] size of the vectors is the number of the eigenfaces calculated from that face space

In this way, we train the system only with the most significant faces and we expect higher recognition rates. One more detail is about the dimensionality of the self-space. The longer the sequence it is, the bigger the maximum number of eigenfaces, thus the larger the space dimensionality. We do not afford clustering in such a big space and since we are not interested in reconstruction of the images, we consider a 25% amount of eigenfaces in our experiments. Thus, clustering will be performed in a space having 25% of the frames dimensions, but the percent of the selected eigenfaces can be reduced experimentally for no large variations of the faces in the set (no need to encode much variation of the faces for the same individual).

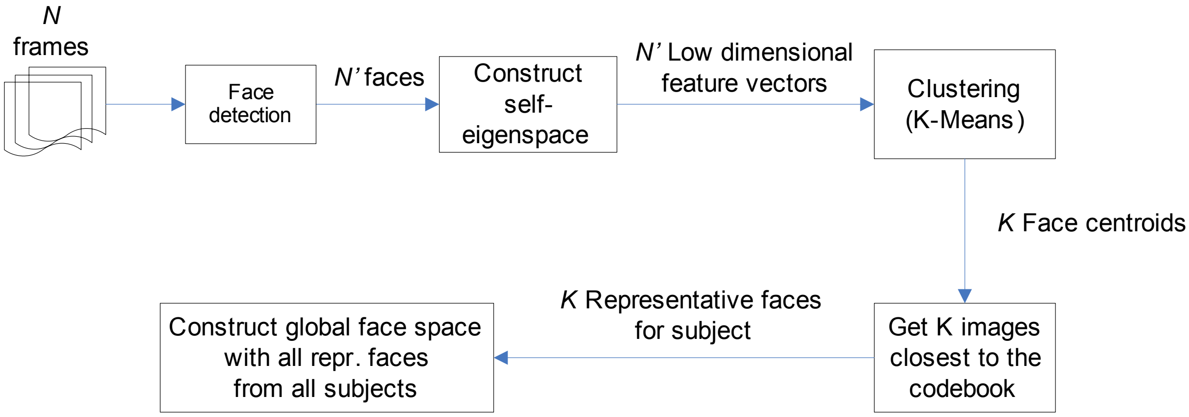

N frames
Face detection
N' faces
Construct self-eigenspace
N' Low dimensional feature vectors
Clustering (K-Means)
K Face centroids
Get K images closest to the codebook
K Representative faces for subject
Construct global face space with all repr. faces from all subjects

Figure 4-3: Selecting the most representative faces of one subject from a video sequence by clustering the self-eigenspaces.

We mention that we cannot use the resulting centroids as a model for the user because there will not be any discrimination between other self-eigenspaces. We cannot use nearest neighbors with distances calculated in each space because that space encodes only its own individual variations. The distance from one face from its self-eigenspace to centre of it might be the same with another face from another self-eigenspace. The self-eigenspace encodes only the variation between different views of each individual, in contrast to the universal eigenspace of all subjects, where PCA represents inter-class variations of individuals and intra-variations across different views of the same subject [21].

This is the reason why we need to get the representative faces and then construct a global face space where all subjects can discriminate. We note we do not reconstruct the centroids from each self-eigenspace. Due to the reduced number of the eigenfaces, the reconstruction will not be precise for the constructing the global face space. For the face sequence from Figure 4-2, for K=15 the representative faces are shown in Figure 4-4.

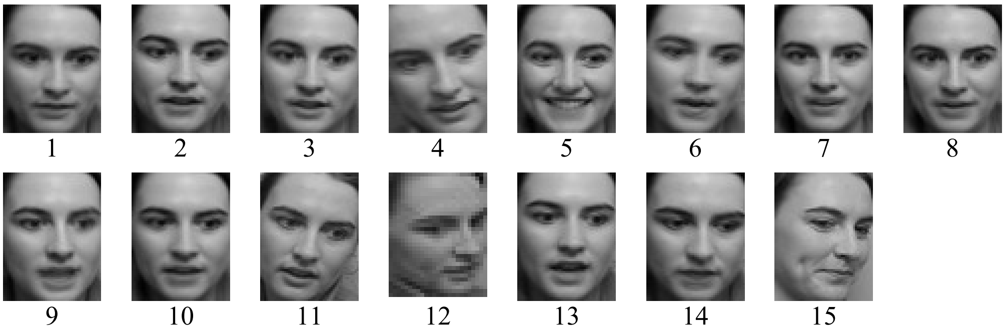


Figure 4-4: Fifteen most representative faces of the face sequence from Figure 4-2.

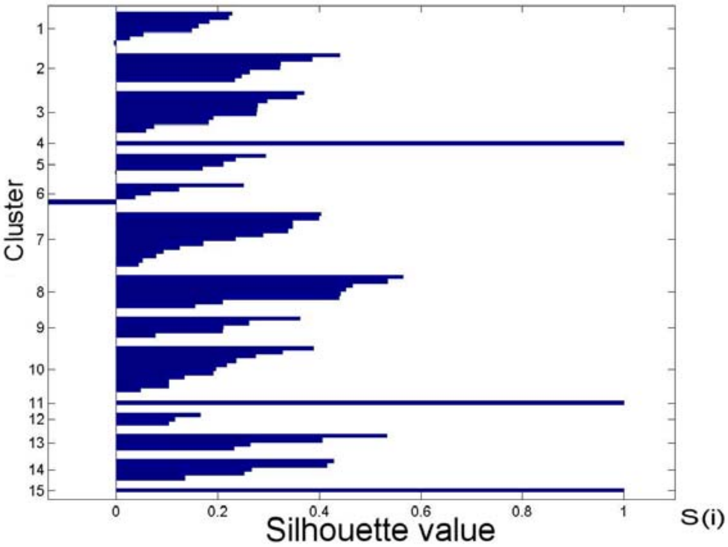


Figure 4-5: Silhouette plot for the clustered faces.

Figure 4-5 provides a representation of the separation between the clusters. The *Matlab silhouette* function displays a measure *S(i)* of how close each face in one cluster is to the faces in the neighboring clusters.

$$S(i) = \frac{\min\limits_{k=1..K}\left(b(i,k)\right) - a(i)}{\max\left(a(i), \min\limits_{k=1...K}\left(b(i,k)\right)\right)}, \ \forall i = 1..N'$$

where *a(i)* is the average distance from the i^{th} point to the other points in its own cluster, and *b(i, k)* is the average distance from the i^{th} point to points in another cluster *k*.

This measure ranges from +1, indicating faces that are very distant from neighboring clusters, through 0, indicating points that are not distinctly in one cluster or another, to -1, indicating points that are probably assigned to the wrong cluster. The chosen face sequence does not contain many frames with the rotated head so we increased the number of clusters until K-means separated those frames. We can see that clusters 4, 11 and 15 contain only one significant face. The rest of the selected faces have a mean appearance of the faces from that cluster.

Face sequence clustering is a similar approach to clustering of the feature vectors extracted from the speech frames in speech related recognition problems. The main drawback of this approach is the arbitrary selection of the number of clusters *K,* we cannot know all possible variation in faces before. Randomized initialization of the codebook in K-means algorithm and possible local minima in the total distortion can provide wrong clusters as well. Randomness can be partially avoided by running clustering many times and then choose the codebook that gives the minimum quantization error from all the iterations.

5. Speaker Recognition

Beside face cues, another natural way to recognize people is by their voices, which leads to the *speaker recognition* task. Comparing to other biometrics, face and speech are not intrusive, that is they do not suppose physical contact between the person and the system. Furthermore, they do not require dedicated sensors as for fingerprints and iris. A photographic sensor is required for face recognition, but for speech, we only need a microphone to acquire the evidence, which makes it convenient and frequently used among the other biometrics.

Early research in face recognition date 30 years ago, but the speaker recognition technology is more mature, starting already from 1950's [37]. However, even though more and more speaker recognition systems are started to be used in practice and much research and progress have been done in the area, it still does not provide exact recognition rates yet. Moreover, even though the technology would be errorless, speech itself is not a fully reliable biometric. Face is a *static* or *passive* biometric, i.e. the facial outlook of a person remains rather constant due to the course of time. Speech, on the other hand, is a *dynamic* or *behavioral* biometric, based on the speaking process, and the acoustic speech waves of the same utterance are not exactly the same [37]. The unreliability of speech has led to multimodal recognition, in which speech is integrated with other technologies [1 , 2, 11, 17 , 22 , 24, 60, 57].

The current section gives a short introduction to the speaker recognition technology. We recommend literature from [13, 37, 56] to be consulted for comprehensive information about this area. For classifier fusion, we consider the speaker recognition from the audio stream as a "black-box" (Figure 1-1), and use algorithms already implemented in our department.

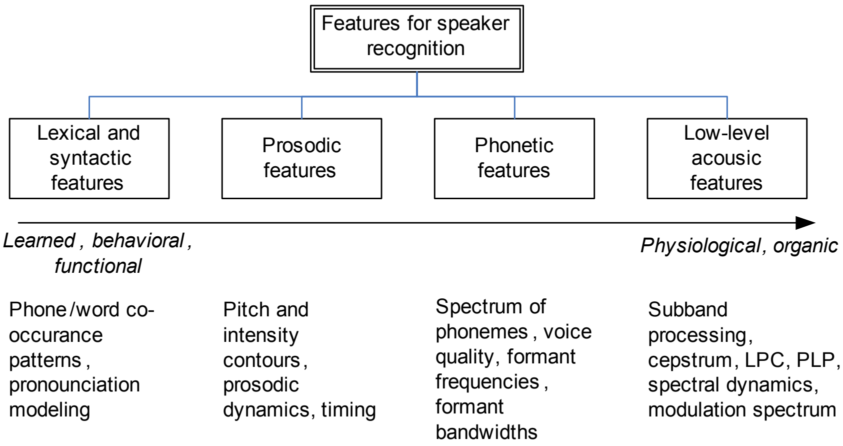
Any speaker recognition system involves either identification or identity verification, and can work in the enrollment and recognition modes, as any other biometric system as shown in Figure 1-4. We recall here the typical steps for performing these tasks: *feature extraction*, *speaker modeling* and *matching*, and *decision-making*.

Speaker identification depends on the text utterance. Therefore, if the utterance is known beforehand and it has been modeled beforehand, then the task is called *text-dependent* speaker identification, whereas the opposite is called *text-independent* identification. In this work, we are concerned with the text-dependent case.

5.1 Feature Extraction

Speaker identity is correlated with the physiological and behavioral characteristics of the speaker [1] and these characteristics are derived both from the vocal tract and the source of speech (vocal chords). The feature extraction process refers to the measuring of those properties of the speech waveform that best characterize an individual. Thus, this step models the speakers by *feature vectors.*

Author of [37] suggests a possible taxonomy of the features used for speaker recognition as shown in Figure 5-1. Among them, the most commonly used methods are the *linear predictive cepstral coefficients* (LPCC) and the *mel-frequency cepstral coefficients* (MFCC).



Features for speaker recognition
Lexical and syntactic features
Prosodic features
Phonetic features
Low-level acousic features
Learned, behavioral, functional
Physiological, organic
Phone/word co-occurance patterns, pronounciation modeling
Pitch and intensity contours, prosodic dynamics, timing
Spectrum of phonemes, voice quality, formant frequencies, formant bandwidths
Subband processing, cepstrum, LPC, PLP, spectral dynamics, modulation spectrum

Figure 5-1: Possible features used in speaker recognition.

Different from face recognition that is based on still samples, speech is a continuous time-dependent signal. The signal is typically divided into short time segments called *frames*, which preserve the local stationary property of the features. The frames are typically around 10-30 milliseconds long and they overlap around 30 to 75% of their length (Figure 5-2). The discontinuities effect is suppressed by the *window function* and the most commonly used is the *Hamming function* [37].

Framing and windowing provide a thorough analysis of the input utterance because each speech sound is approximately centered within the frame. Fourier analysis of the local waveform assumes that the signal is periodic, and the windowing function supports this assumption because the discontinues at the frame edges are interpreted as being part from a signal with an infinite period [37]. As a parenthesis, the short-term spectral analysis of the speech signal leads to similarities between speaker and face recognition from image sequence, in which each frame is processed separately, as shown in Figure 5-3.

From the source-filter model [13, 37] of the speech production, it is generally known that the vocal tract acts a low-emphasis filter with -12dB/octave boost for voiced speech, while lips introduce a +6dB/octave boost to the spectrum. In order to cancel these side effects to preserve the original frequency spectrum that describes better the vocal tract characteristics, a *pre-emphasis* filter of +6dB/octave boost is selected (Figure 5-5).

Each time window is subject to spectral analysis by *Fast Fourier Transform* (FFT). Usually only the magnitude spectrum is used. Thus, feature extraction can be done either in *time* or *frequency* domains. The former method analyzes the full spectrum band of each frame (Figure 5-5a) while the latter processes each sub-band of the spectrum in a time window, making use of *filterbanks* (Figure 5-5b). These last ones provide a smoothed version of the original spectrum, as shown in the second example from Figure 5-4. Extracted features from every sub-band can be combined using fusion techniques that we will present in the following section.

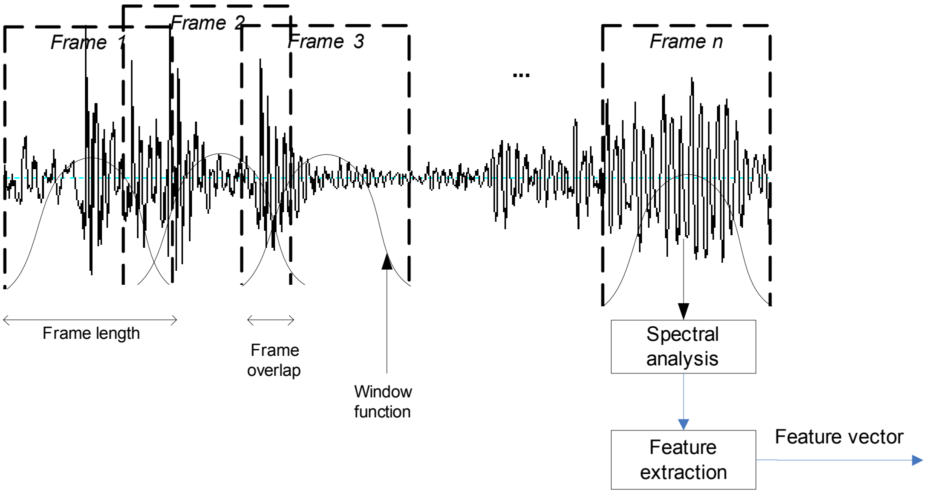


Figure 5-2: Short-term spectral analysis for speech.

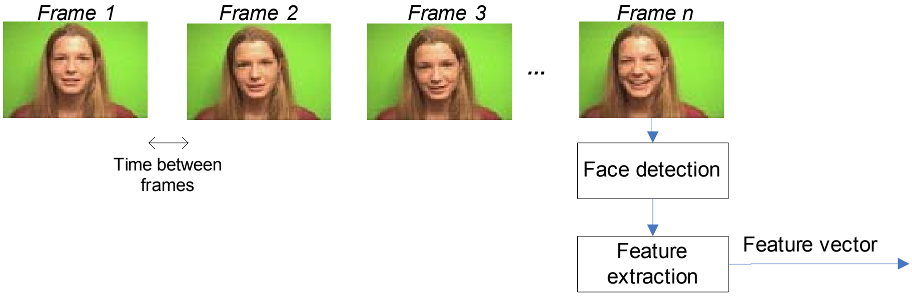


Figure 5-3: Frame analysis for face recognition from video sequence.

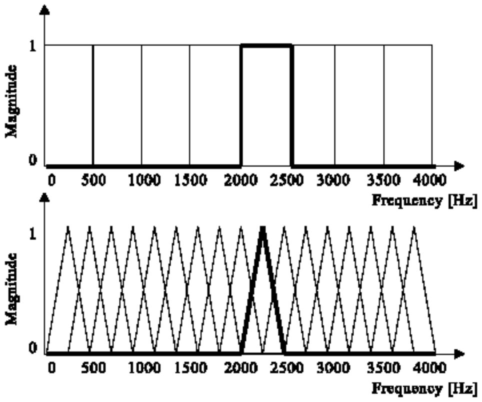
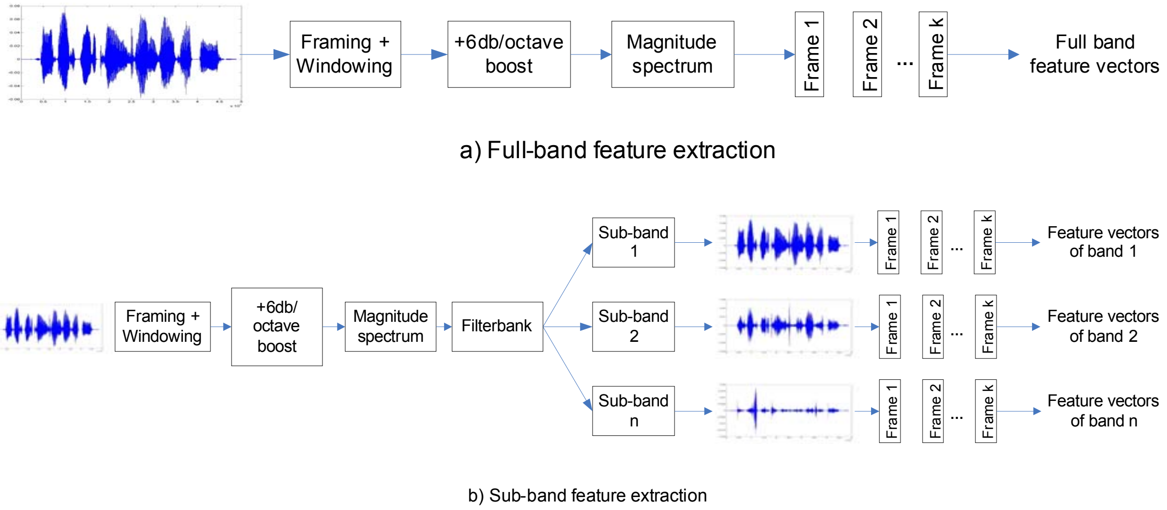


Figure 5-4: Examples of filterbanks [37].



a) Full-band feature extraction

b) Sub-band feature extraction

Figure 5-5: Full-band and sub-band feature extraction.

Beside the *static* features that are extracted from every frame, there exists another type of features, which characterize *dynamic* information. These are so called *delta-features* [37] and they represent the time derivative of the static features calculated over a number of frames by differentiating or by fitting a polynomial expansion. Moreover, the time derivative of the delta features will represent in their turn the *delta-delta features*.

5.2 Speaker Modeling and Matching

Before calculating identity similarities between the unknown and the enrolled speakers, we need to create models of the speakers based on the extracted features, and then perform matching of them. Models are an abstraction or characterization of each individual from the database and represent actually an estimate of the feature probability distributions. Thus, matching of two models outputs a degree of similarity between them, and this represents the input in the decision-making module.

We distinguish two types of modeling techniques: *parametric* (stochastic) and *non-parametric* (templates) [37]. Perhaps the most common approaches used in text-independent speaker recognition are *Vector Quantization* (VQ) from the second class, and *Gaussian Mixture Models* (GMM), which belong to the former one.

In the vector quantization approach, the model of the speaker is represented by the *codebook* of the clusters that form in the distribution of the all feature vectors extracted along the training utterances. In this way, the codebook preserves only the most representative information about the speakers. Matching is performed by finding the codebook C belonging the enrolled speakers, which minimizes the *average quantization distortion* function between these two models:

$$D_Q(X,C)=\frac{1}{T}\sum_{i=1}^{T} d_q(x_i,C) \quad (4)$$

$$d_q(x_i,C)=\min_{c_j\in C} d(x_i,c_j)$$

where $d(\cdot,\cdot)$ is a distance metric; the most common distance function used is the *Euclidean metric.* An illustrative example of VQ-modeling and matching is given in Figure 5-6 and Figure 5-7.

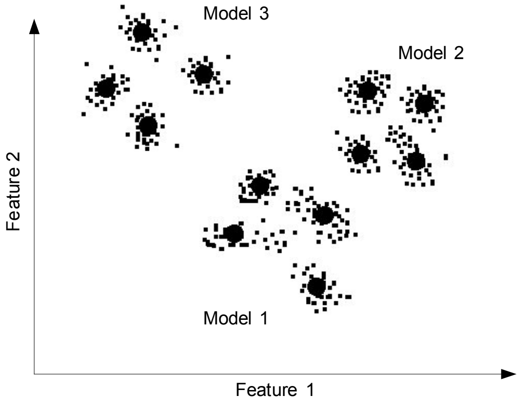


Figure 5-6: Example of VQ modeling for codebook of size K=4 in a two-dimensional feature space.

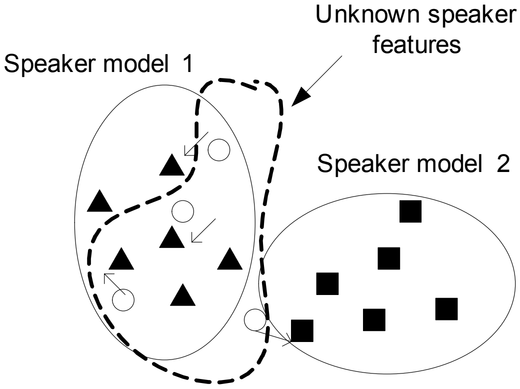


Figure 5-7: Example of matching between T=4 features from an unknown speaker (circles) and codebook of size K=6.

In the second approach, a mixture of Gaussian functions models the distribution of the feature vectors, as shown in Figure 5-8. A speaker model $\lambda=\{\lambda_1,\lambda_2,\ldots\lambda_K\}$ is given by a set of three parameters for all *K* distributions, denoted by λ_i:

$$\lambda_i=(P_i,\mu_i,\Sigma_i),\ i=1\ldots K$$

where P_i are their a priori probabilities, μ_i are the mean vectors and Σ_i are the covariance matrices of the distributions. These parameters are typically estimated by maximum likelihood estimation, using the *Expectation-Maximization* (EM) algorithm [37]. Matching between two models X and λ is calculated in terms of *likelihood,* such as:

$$L=\log p(X|\lambda)=\log\prod_{j=1}^{T} p(x_j|\lambda)=\sum_{j=1}^{T}\log p(x_j|\lambda),$$

where $p(x_j|\lambda)$ is the Gaussian mixture density:

$$p(x_j|\lambda)=\sum_{i=1}^{K} P_i N_i(x_j),$$

and $N_i(x_j)$ is the n-variate normal probability density function [13]:

$$N_i(x_j) = \left(\sqrt{2\pi}\right)^{-\frac{n}{2}} \left|\Sigma_i\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(x_j - \mu_i\right)^T \Sigma_i^{-1} \left(x_j - \mu_i\right)\right\}$$

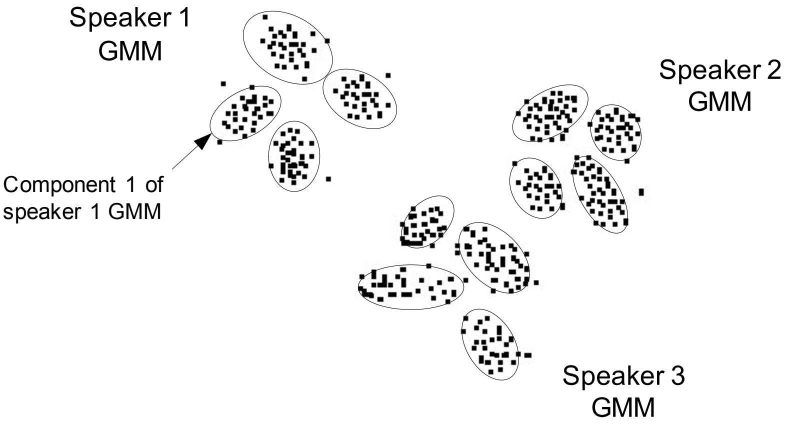


Figure 5-8: Example of modeling by GMM of size 4.

6. Fusion of Face and Speaker Recognition

6.1 Introduction

Although a lot of work has been done in the last years in the fields of face and speaker recognition, the current techniques do not provide a fully reliable recognition yet and they are about to reach a degree of saturation in performance in the close future. In general, the common drawbacks are from the incorrect modeling or from the difference between the training and test conditions. As much as natural they seem to be, face and speech biometric measures tend to vary in time [70] and thus algorithms provide decreasing recognition rates.

While many efforts concentrate to improve the current methods, a recent trend in biometrics is to combine different modalities by using multiple human characteristics. The goal is to complement one modality with another when one of them performs poorly, so it will not affect the final decision. A series of research papers have already proved that the joint use of several biometrics provide a higher accuracy than single modalities [11, 36, 58, 70, 73].

Multimodality is a subject of *information fusion.* This area deals with combination of different sources of information, either to generate one representational format, or to reach a decision [62]. We reserved the current section to present important aspects from the theory of combination of classifiers and to analyze which strategy should be adopted in fusion of face and speaker cues for person recognition.

An important aspect related to information fusion lies on the independence of different classifiers, which characterizes the amount of extra information they bring in for the global system. Using complementary information such as face and voice can reduce error rates, while using multiple sensors can increase reliability [62], by providing redundant cues. We admit to believe that fusion introduces also a benefit because multiple simple sensors might be cheaper than one multimodal sensor, and they can acquire data from multiple points of view.

Fusion can be employed at three different levels:

- Mono-modal mono-expert fusion
- Mono-modal multi-expert fusion
- Multi-modal multi-expert fusion

Mono-modal mono-expert fusion combines the results obtained from a single modality by a single expert over multiple instances of test data acquired over a period. It is also called *temporal* or *horizontal* fusion. This strategy offers the advantage of the availability of more training and test data that can reduce their variances.

Mono-modal multi-expert fusion uses multiple experts based on the same or even different features extracted from a single modality. This leads to the idea that the scores from different experts for the same person are correlated. Suitable examples for this strategy are: the hybrid methods that combine geometric and image-based approaches for face modality, and for speech, fusion of static and dynamic features, or fusion of features extracted from different sub-bands.

Multi-modal multi-expert fusion involves multiple experts that are based on multiple modalities and it may include the previous levels.

Different fusion techniques are needed for the above strategies, see Figure 6-1 [62]. In *pre-mapping fusion*, the information is combined before any use of classifiers; *post-mapping fusion* combines the information after mapping from feature-space into opinion/decision space. The former one is also called fusion at *input level* or *early fusion*, while the latter refers to fusion at *classifier level* or *late fusion*.

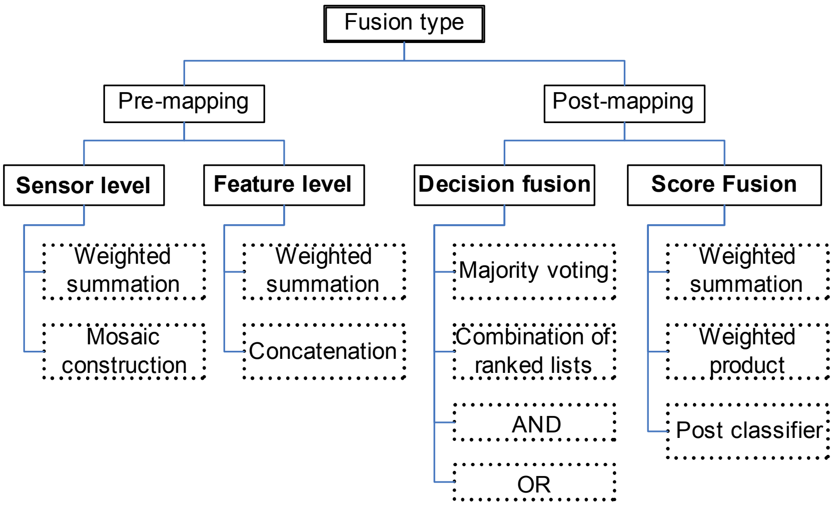


Figure 6-1: Different fusion techniques.

6.2 Sensor–level Fusion

The *sensor level* fusion combines homogenous raw data acquired from multiple sensors and forms a new input data for the classifiers. There are two approaches: *weighted summation* and *mosaic construction.*

The first uses weighted summation of signals such as two speech waveforms from two microphones in order to reduce noise. We note that all the measures must be first normalized to have values in the same range. The second approach includes in the new signal parts from multiple input signals, such as one new image containing different viewpoints. For our case, the source of information is a common sensor (audio-video camera), and the streams are already fused. We need to separate them in order to analyze them independently.

6.3 Feature–level Fusion

Fusion at the feature level uses multiple vectors extracted from different sensors or even multiple vectors extracted from the same sensor but in a different way. A new classifier should be used then for the new training features.

Weighted summation can be used if the features are extracted from homogenous sensors (two microphones) and they are commensurate. Otherwise, all the feature vectors can be concatenated into a single feature vector, which represent the person's identity into a new feature space (Figure 6-2).

The concatenated vector is obviously larger and the new feature space becomes much sparser than the individual feature spaces. More training data is needed or dimensionality reduction techniques such as PCA and LDA can be used. One drawback of this fusion is that it cannot distinguish which of the sensors contribute more or less to the final decision [62].

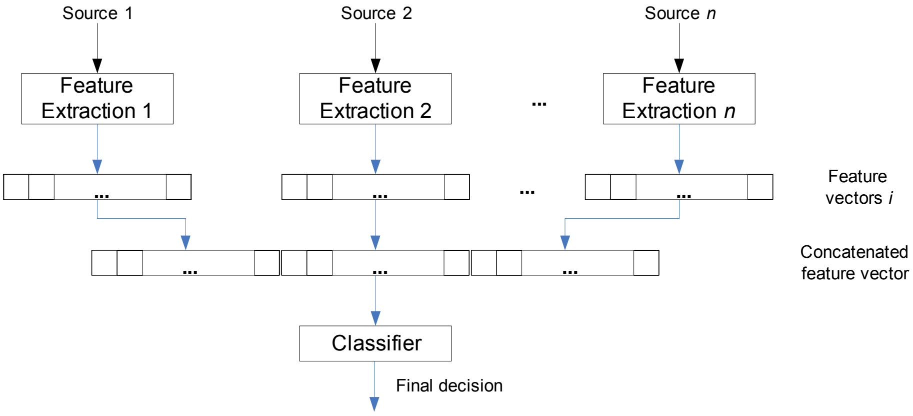


Figure 6-2: Fusion at feature extraction level.

Discussion

Fusion at feature level for stream cues prove to be the most complicated due to the frame synchronization issues. The sensors might be asynchronous, so not all feature vectors might be available at the same time. An example is the continuous recognition from audio and video for example, where both streams have different rate. Typically, video frame rates are between 25fps (PAL) to 30fps (NTSC) (40-33.33ms) while the speech features are extracted in frames between 100fps to 33fps (10-30ms) (Figure 6-3). For synchronization, speech frames should be 40 ms to 33.33 ms long for PAL and NTSC respectively.

For video streams, we remember that we perform frame sampling by selecting each N^{th} frame (Section 4.2), so the frame rate decreases more actually. Moreover, no face can be detected within a wide range of frames because of their variety in videos (wrong subject framing or tilted faces).

We present several methods for frame synchronization in Figure 6-3. The first image shows the typical case between different frame rates from video and speech streams. One workaround is to pad the missing features by zero values, as by considering zero features for the video frames corresponding to speech frames S2, S3, S4 etc.

Buffering can be also used for synchronization. We copy always the previous existing frame, increasing artificially the video frame rate, as in Figure 6-3 b). The next case shows *linear interpolation* between two frames (Figure 6-3 c). When faces are not detected within a wide range, we consider that interpolation decreases the accuracy for significant changes in the faces (Figure 6-3 d).

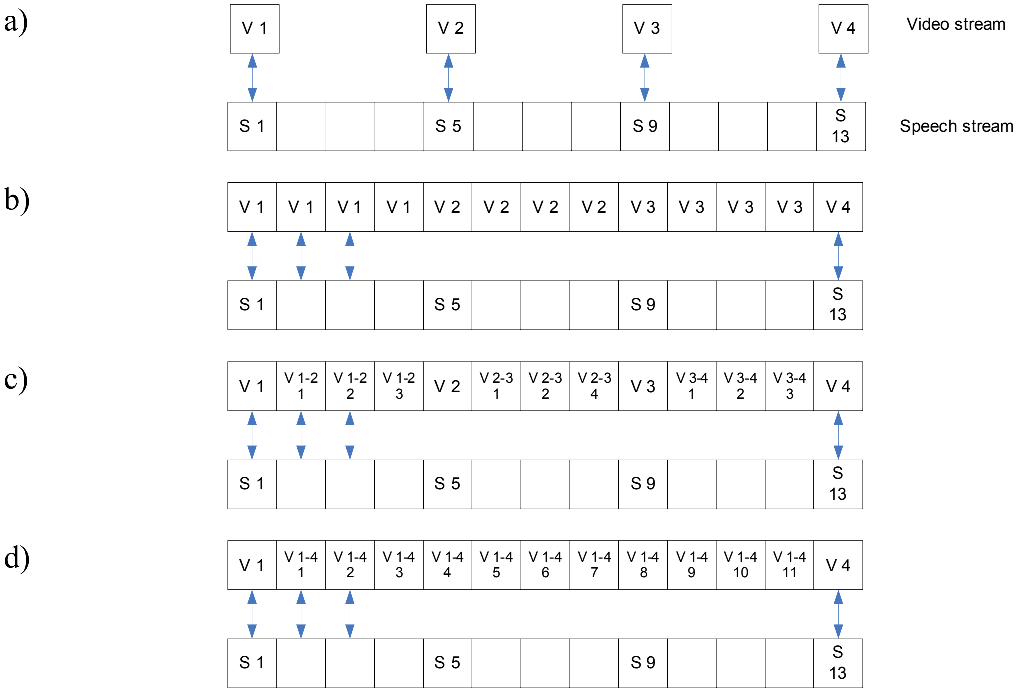


Figure 6-3: Synchronization between frames from audio and video streams: a) Frames at different rate; b) Previuos frame copy; c) Interpolation between two adiacent video frames; d) Interpolation when multiple frames are missing.

Continuous Fusion

Feature level integration provides support for *continuous fusion.* If until now we considered the recognition after the shot was analyzed, continuous biometrics makes the recognition an ongoing process. It requires *temporal (horizontal) integration* to estimate the authenticity of the user at any time from the shot based on the previous estimations [3]. However, this type of integration preserves the temporal information between the both streams.

We mention here one approach for continuous audio-visual speech recognition from [45]. It is based on cooperative HMMs for continuous fusion from asynchronous streams. The input streams are processed independently of each other up to certain temporal anchor points (Figure 6-4). Here, the models have to synchronize and recombine their partial segment-based likelihoods.

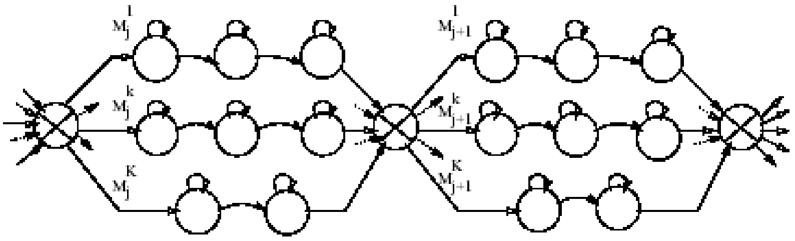


Figure 6-4: General form of a K-stream model with anchor points [45].

An observation sequence is assumed to be composed of *K* streams. The parallel HMMs associated to each stream do not need to have the same topology. We note that the recombination state from Figure 6-4 is not a regular HMM state, it just recombines the scores accumulated over the same temporal segment for all streams. A similar approach of fused-HMMs is presented in [55].

Conclusion

In order to concatenate multiple feature vectors extracted from different sources of information, we need to assume the *conditional dependence* between them. Otherwise, the new vector will not have any meaning. We consider the acoustic and visual data as being independent, so we cannot "link" one particular phoneme to a particular face position or expression. In other words, it does not mean that some utterance can occur only for some specific faces. This is more a subject of audio-visual speech recognition when the shape of the lips is analyzed.

Based on this argument, we did not include this type of fusion in our experiments. We propose the feature level and the continuous fusion for future research. The latter addresses also the problem of the missing data from different biometrics, such as fingerprints and retina. These sensors do not provide continuous output or they do not have similar frame rate, as audio or video do. They are available occasionally, as we do not expect the users to scan their fingerprints or retina all the time. The missing features problem regards also the face recognition when faces cannot be detected.

6.4 Decision–level Fusion

Each classifier provides a decision of *acceptance* or *rejection* based on its corresponding feature vector, and the decisions can be further combined by different architectures and methods. Possible architectures are: *serial*, *parallel* and *hybrid*.

A *serial* architecture consists in many classifiers whose decisions are combined in series or in cascade [70]. This fits for the cases when each decision is trivalent, such as accept, reject or *undecided*. If one expert cannot decide about the identity, it transfers the problem together with the information it has about that identity to the next classifier in the chain and so on. This architecture assumes that the further classifiers from the sequence are more effective than the previous ones and thus they can solve the decision problem. It is also possible that the final result to be indecision.

In the *parallel* architecture, all classifiers process in parallel the feature vectors and output a decision. The final decision can be reached by:

- Majority voting
- Ranked list combination
- AND/OR fusion

This architecture is the most commonly used and it fits when different biometrics are independent of each other. Thus, a stronger biometric can achieve better accuracy alone than combined with weaker biometric. The *hybrid* architecture uses a combination of serial and parallel classifiers. We will refer next only to the parallel model.

The final decision in *majority voting* is the decision taken by the majority of the classifiers. Moreover, an odd number of classifiers are necessary in order to prevent tie decision. This method does not suit to the bimodal system addressed in our work.

For *ranked list combination*, every classifier outputs a ranked list of individuals according to their degree of confidence. These lists are combined considering the reliability and discrimination power of each classifier, and then the final decision consists in the top entry of the combined list.

The final decision in *AND* fusion is the decision taken by all classifiers, while in an *OR* fusion, this is the decision of only one classifier that reached a conclusion about the identity. The former operator leads to a more restrictive decision because all the classifiers must agree about the same identity, so the fused system will have lower *False Acceptance Rate* (FA). Opposite, the latter operator concludes a relaxed decision, so a lower *False Rejection Rate* (FR) [62]. Moreover, the OR operator can lead to more matching decisions for one testing model and thus it can be used only for verification purposes.

6.5 Score–level Fusion

The fusion at score level assumes that all classifiers output a matching score indicating the degree of confidence for an individual. Then all the scores are combined to verify the supposed identity (Figure 6-5).

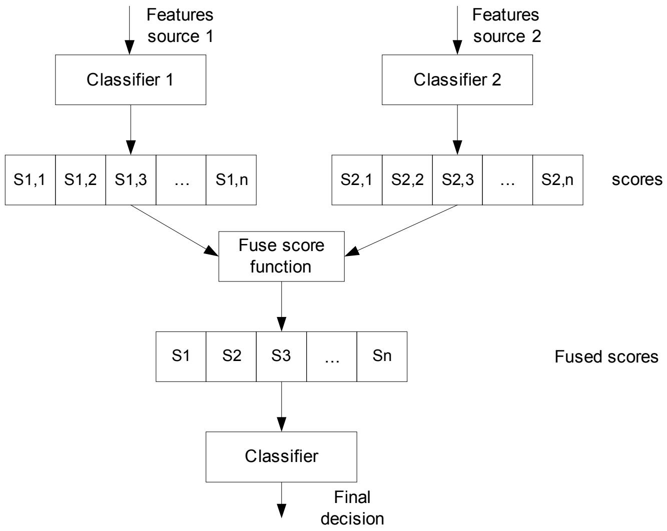


Figure 6-5: Fusion at score level for two classifiers and n classes.

An important aspect for this integration is the measurement *normalization*. Since different classifiers for various traits can be employed, the individual matching scores could be non-homogenous (different scales), expressed such as distances in their own feature space or likelihood ratios. So, a score normalization step shall be a prerequisite.

A simple way to normalize scores is to convert them first to a normal distribution. Then, the values can be mapped into [0,1] interval by *sigmoid function*:

$$d_{norm} = \frac{1}{1 + \exp(-\tau(d))}$$

$$\text{where } \tau(d) = \frac{d - \mu}{\sigma}$$

The parameters for the normalization function can *fixed* or *adaptive*. The former ones are estimated from a certain number of observations, while the latter ones are estimated from the distribution of scores from the current observation. Using only the mapping function is not enough because *exp(-d)* function will decrease to zero even for *d* being order *10e3*. That is why a conversion to standard normal distribution by function $\tau(d)$ is used. We need to estimate average values and standard deviations for scores so that distributions can be translated and rescaled in order to have zero average and unit variance.

Scores can be also in terms of probabilities. We propose the following method for mapping distances scores into probabilities.

Let d_{norm} be a distance $\in [0, 1]$.

$$p = \frac{1}{d_{norm}}$$

Normalize p to belong to [0, 1]:

$$p' = \frac{p}{\sum_{i=1}^{nr\ classes} p_i} \qquad p'' = (p')^k,\ k \geq 1$$

The term $p' \in [0, 1]$ so we raise p' to power k in order to decrease more the smaller probabilities values and to separate more the smaller from the larger values.

The typical approaches to combine the normalized scores are:

- Weighted summation (Sum rule)
- Weighted product (Product rule)
- Post-classifiers

The first two approaches weight the opinions of each classifier considering their reliability and discrimination ability. This is an advantage over the feature or decision level fusion.

In *weighted summation* fusion, the final score s_j for the class j is given by:

$$s_j = \sum_{i=1}^{N_C} w_i s_{i,j}$$

where N_C is the number of classifiers involved in the fusion, $s_{i,j}$ are the scores of the classifier i for the class j, and w_i are their corresponding weights in interval [0,1]. Sum of all weights is constrained to be equal to one:

$$\sum_{i=1}^{N_C} w_i = 1$$

If all the weights are equal to $1/N_C$, then the weighted summation represents the arithmetic mean of all scores.

For *weighted product* fusion, if we assume that the classifiers are independent of each other, then the scores can be regarded as *a posteriori* probabilities [62]. Thus, the final decision for classifying class j is calculated as the product of probabilities of each expert to identify that class:

$$s_j = \prod_{i=1}^{N_C} s_{i,j}$$

We can differentiate each classifier's reliability by introducing weights:

$$s_j = \prod_{i=1}^{N_C} \left(s_{i,j}\right)^{w_i} \tag{5}$$

Weights have the same constrains as for weighted summation. When all the weights are equal to1/N_C, then the product rule becomes the geometric mean of all scores. Sometimes it is more convenient to work with summations, so logarithm function is used over the product. Equation (5) becomes:

$$\log s_j = \sum_{i=1}^{N_C} w_i \log\left(s_{i,j}\right)$$

The main drawback of this fusion is that a very small score of one classifier decreases the overall score. Comparing to feature level fusion, the temporal information between the audio and video streams is lost in the score and decision fusion. On the other hand, the latter two fusion strategies support an alternative approach for continuous fusion at feature level from the asynchronous streams and missing data. The solution is to perform continuous recognition from each individual stream and then fuse at decision or score level for the segments between common features, as shown in Figure 6-6.

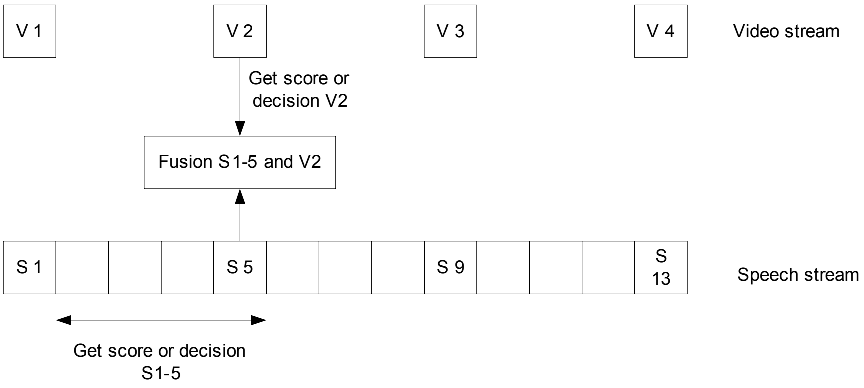


Figure 6-6: Continuous fusion at decision or score level for asynchronous audio and video streams.

In *post-classifier* fusion, the scores from N_C classifiers for N classes can be concatenated to form one $N_C{\cdot}N$-dimensional feature vector. A new classifier, called *post-classifier*, makes the final decision. The scores do not need to be normalized.

Commonly used approaches for post-classifiers are: *Bayesian networks*, *Support Vector Machines* (SVM), *Multi-Layer Perceptrons* (MLP), *Decision Trees*, *Logistic Regression* (LR) and various forms of the *k-Nearest Neighbor*. We do not intend to review all these methods, but we recommend [70]. The paper details each method and performs a comparison of them.

Due to the large number of classes, the new dimensionality of the feature vector can be huge (size of training set). Thus, it makes the method more practical for verification tasks. For these scenarios, there can be maximum one number of target models for each classifier (N=1*)*, so the combined vector is N_C –dimensional only. Each classifier outputs one score that classifies a given claimant as being a true claimant or an impostor. Then, the post-classifier employs a decision surface in N_C –dimensional space, separating the impostors of the true claimant as in Figure 6-7, where N_C=2.

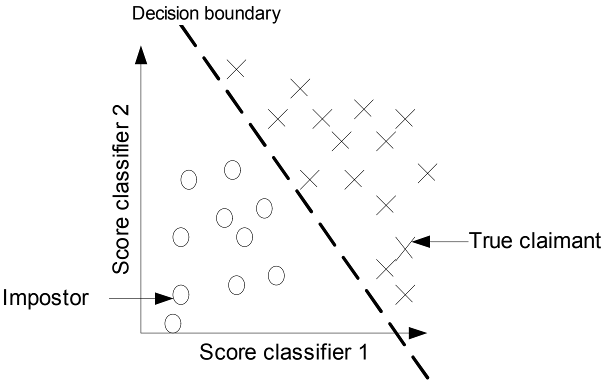


Figure 6-7. Decision surface between true claimant and impostors for post-classifiers.

6.6 Stream Reliability

Weighted summation or product rules from the score fusion takes into account the reliability of each classifier by introducing weights so each classifier will contribute more or less to the fusion result. There are two ways of fixing these weights: *non-adaptive* or *adaptive*.

The non-adaptive methods assume that each contribution is fixed *a priori*, by calculating stream dependent constant weights for a particular audio-visual environment and database, based on the available training data. The weights can be set experimentally and based on preliminary information about each classifier's results. Then, these weights are used for sum or product fusion rules.

In real application scenarios, the test conditions may change significantly in time comparing to training. The adaptive methods vary the contribution of at least one expert according to its reliability and discrimination ability in the presence of some local environmental conditions [62] e.g. noise, face occlusion or face detection failures. If the *a priori* weight for the one classifier is dominant, then the overall result decreases, although the other classifiers involved in fusion could bring more information about the identity. Intuitively, the adaptive methods are suitable for continuous fusion, but we will not approach them in this work.

We used in our experiments one statistical approach for *a priori* weight selection from [62]. For two classifiers case, only one weight w_1 is enough to calculate. The other one is

$$w_2 = 1 - w_1$$

For first classifier:

$$w_1 = \frac{\xi_2}{\xi_1 + \xi_2}$$

where ξ_i represents the *standard error* defined as:

$$\xi_i = \sqrt{\frac{\sigma_{i,true}^2}{N_{true}} + \frac{\sigma_{i,impostor}^2}{N_{impostor}}}$$

where N_{true} and $N_{impostor}$ are the number of correct claims and impostor tests respectively, and $\sigma_{i,true}$ and $\sigma_{i,impostor}$ are the corresponding score variances. Tests run over an arbitrary face and speech data set from database.

We interpret the standard error as a measure of correctness of one classifier, by computing how much it classified wrong or right and with which scores. For example, if it classified impostors with big scores, then the variation is high. When the variation in true claimant and impostor scores is small, then the standard error is small, and the weight *w* is large.

7. Experiments

For a comprehensive understanding of the topic covered in our work, we organized experiments using a couple of algorithms for integrating face and speaker recognition. We implemented first methods such as *Eigenfaces* and *Fisherfaces* in *Matlab,* and then we used *Sprofiler* for speaker recognition experiments, which is software developed at the University of Joensuu within *PUMS* project[9]. The last step of our experiments was to adopt the multi-modal multi-expert fusion strategy by integrating the results of the two classifiers for both modalities. We implemented fusion at score level by weighted sum and product rules.

In the first part of the section, we overview multi-modality audio-visual databases available for research, while in the following sub sections we detail each step from our experiments procedure, and present the results.

7.1 Audio-Visual Databases

In contrast to the abundance of uni-modal databases, the multi-modal audio-video databases are very sparse for research purposes, which make algorithms difficult for testing. This is because the field is relatively young, but also due to the high storage requirements of the video shots, and for the availability of the users as well. Because of these reasons, the databases usually contain a small number of subjects. They have also smaller duration and do not cover a wide variations about subjects' situations. In principle, multi-modality can be achieved simply by assigning multiple biometric evidences to one subject, but in our case both cues (speech and audio) comes from the same video sequences. We review next the audio-visual databases we are aware at the time of writing, with the focus on the database we are going to use in our experiments. The databases are summarized in Table 7-1.

CUAVE

The database included in our experiments is the *Clemson University Audio Visual Experiments* (CUAVE) [20] corpus, see Figure 7-1. It includes realistic test conditions such as movement and different visual features of speakers such as glasses, facial hair and hats. One negative aspect of this database is that the shots do not cover variation of illumination.

The database consists of two major sections. The first one includes 36 individual speakers consisting of 17 females and 19 males with different skin color. The second part consists of 20 pairs of them but we do not use this part in our experiments because we the multispeaker problem has not been covered in this thesis. Visual features such as glasses, facial hair and hats are present in the videos, and this makes the database more difficult for testing.

[9] http://cs.joensuu.fi/pages/pums/index.html

Table 7-1: Overview of audio-visual databases.

Name	Reference	Subjects	Video	Sound
CUAVE	[20]	36	MPEG2, 720×480, NTSC DV	Stereo, 16 bit 44 KHz
VidTIMIT	[63]	43	JPEG, 384×512, PAL DV	Mono,16 bit, 32 KHz WAV
BANCA	[5]	52	PNG, MPEG7 720×576, PAL DV	16/12 bit, 32 KHz
XM2VTS	[51]	295	PPM, 720×576 DV	16 bit 32KHz
DAVID	[50]	124	18 SVHS video tapes	-

There are two kinds of recordings for the individuals. In the first one, each subject speaks 50 connected digits while standing still. This includes only small natural movement. In the second task, they move on purpose while speaking 30 connected digits. Movements include nodding the head in different directions, moving back-and-forth and side-to-side, both profile views, and in some cases rotation of the head.

The database was recorded at a resolution of 720 × 480 with a NTSC standard of 29.97 fps using 1-megapixel-CCD MiniDV camera in controlled conditions, such as uniform background of green color, uniform lighting, and noiseless sound. The data is compressed into individual MPEG2 files for each speaker at 5000 kbps, with stereo sound channel at 16 bit and 44 KHz sampling rate. The database includes also a separately extracted sound channel that is down sampled at 16 bit, mono, 16 KHz. The average length of one shot is 2 minutes.

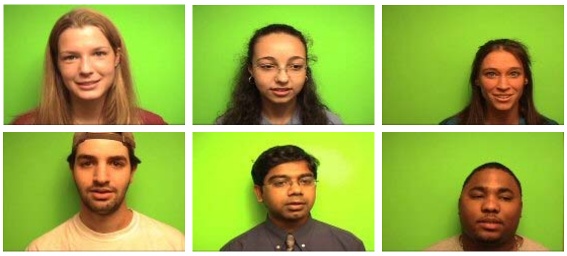


Figure 7-1: Examples from CUAVE database.

VidTIMIT

The VidTIMIT database [63] includes audio-video recordings of 43 subjects divided into 19 females and 24 males; they are speaking 10 short sentences from the NTIMIT corpus [35]. The recordings consist of three delayed shooting sessions with the purpose to allow for changes in voice, hairstyle, make-up, clothing and mood. Each session embraces head rotation sequences such as turn left and right for getting both profiles, up and down.

The shots were taken in a noisy office environment (computer fans) by a PAL digital video camera. The video and audio signals are split into JPEG image sequences at resolution 384 × 512 pixels (Figure 7-2), while the sound channel is stored mono at 16 bit, 32 KHz WAV files. The duration of each sentence is 4.25 seconds on average, which includes 106 video frames per each.

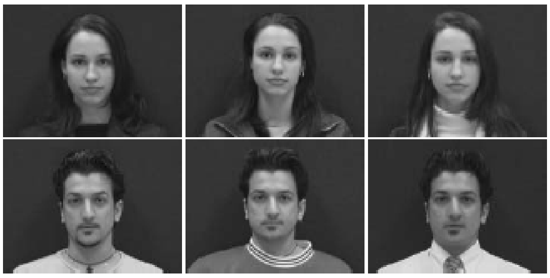


Figure 7-2: Examples from VidTIMIT database.

BANCA

The BANCA database [5] includes several realistic recording scenarios such as controlled, degraded and adverse, using different kinds of material in four different European languages (English, French, Italian, and Spanish). Data was collected from 52 subjects (26 males and 26 females) on 12 different occasions; there are 208 subjects in total. Each session contains two recordings, one for true client access and one for an impostor attack, in which the subject knew the text the claimed identity was supposed to speak.

Recording was done in PAL DV system using a cheap analog web cam and a high quality digital video camera, and both poor and good quality microphones for speech. The audio is uncompressed at 16 and 12 bits at 32 KHz while the video was encoded at 5:1 scale. The web cam was used for the degraded scenario, while the better camera was used for the other two scenarios.

Subjects were recorded while they say random digit numbers, their names, addresses and dates of birth, for about twenty seconds. Examples of shooting sessions are depicted in Figure 7-3.

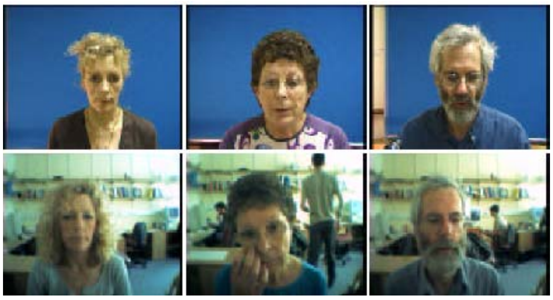


Figure 7-3: Examples from BANCA database in controlled (first row), degraded (middle) scenarios.

XM2VTS

One of the first and the most comprehensive audio-video database is XM2VTS [51], which is an extension of another multi-modal database called M2VTS [46] containing only 37 subjects. The number of users was not considered large enough for impostor tests, and that is why a new database (XM2VTS) of 295 users was collected.

The recording scenarios include 30 seconds dialogs, in which the subjects uttered a predefined sentence. Extreme head movements are also included, such as from centre to the left and then right to extract both sides profiles, then up and down. Variations in physical condition have been included, such as hairstyle, dress and mood. Shooting was done in four separate sessions uniformly distributed over a period of 5 months. Instances of wearing or not wearing eyeglasses are also present.

Equipment included a digital camcorder that provided video data compressed at 5:1 ratio in DV format, while a high quality microphone provided speech at 16-bit audio at a frequency of 32 KHz. The light was placed in both left and right sides and a blue background was used to ease the head segmentation.

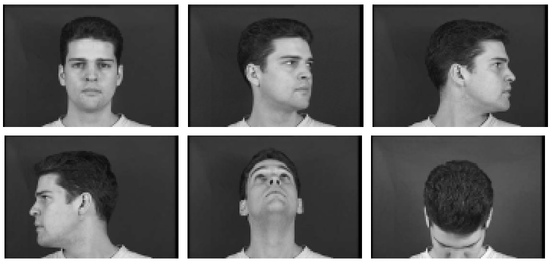


Figure 7-4: Examples from XM2VTS database.

DAVID

DAVID [50] is another large audio-video database that consists of 124 subjects, of which 31 were recorded during 6 months, while the rest were recorded in one session. Shots include full-face combined with side-view on a plain background. Lip highlighting is present in two subsets of the database to aid lip segmentation. The speech material comprises isolated digits, the English-alphabet E-set, and some ‘VCVCV’[10] nonsense utterance.

The database was recorded on 18 SVHS video tapes and we are not aware if it is digitized yet. Sample from the database are show in Figure 7-5.

Figure 7-5: Examples from DAVID database.

7.2 Experimental Setup

We selected the CUAVE database for our experiments because it is free for research purposes and easy to distribute. Even though it does not define any evaluation protocol to follow, we measure *identification rates* for face and speaker classifiers, when we used them individually and combined:

$$IR = \frac{N_{true}}{N_{total}} \cdot 100 \ (\%)$$

where N_{true} is the number of correct matched subjects, and N_{total} is the total number of subjects. We adopt the *Nearest Neighbor* classifier; for each tested subject, we select the trained model that is at the shortest distance from the model of the test subject.

We train and test the system with different shots lengths, see Table 7-2. Due to huge computational and storage requirements, we selected only 29 subjects from the 36 available subjects in database, divided into 16 males and 13 females.

Table 7-2: Amount of train and test shots.

Train			**Test**		
Average shot length		Average amount of images	Average shot length		Average amount of images
%	Sec		%	Sec	
30	40	120	70	84	252
10	12	36	90	108	324
			30	40	120
			1	1.2	3
-	-	1	100	120	360

The first step was to detect, extract and normalize the faces from the video sequence for applying face recognition from the still images (Figure 7-6). Because we did not find support

[10] Sequence of vocals and consonants

in Matlab 6.5 for analyzing the videos, we first extracted all the frames into JPEG format with the use of *OneStopSoft Video Decompiler*[11]. Although the shots are color, we converted them to 256 gray levels for simplicity. For computational reasons, we sampled the face images sequence by selecting only every 10th frame. Statistics about the number of subjects, frames and face detection rate are shown in Table 7-3. We consider the number of faces to be enough for our experiments, taking into account we ignored profile views.

We used the already extracted speech WAV files, which we down sampled from 16 KHz to 12 KHz for compatibility with the speaker software (original in video was at 44 KHz). We considered tests with different lengths for training and testing shots, and we simulated different test conditions for both classifiers, such as illumination changes and noise. We fused the results at score level by weighted sum and weighted product, using several combinations of test conditions and methods.

Table 7-3: Face detection rate from image sequence.

Number of subjects (videos)	29
Total number of frames in all videos	127766
Frame sampling	1:10
Total extracted faces	9992
Average number of faces for subject	345
Face detection rate	78.27 %
Image size	63 × 74

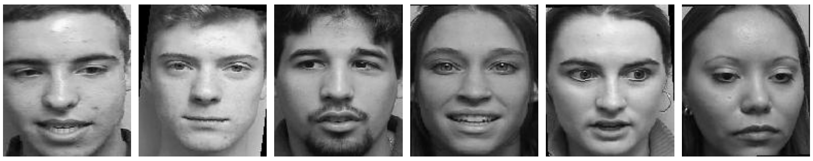


Figure 7-6: Example of detected faces.

We underline the importance of the face detection step. This step is mandatory for a successful recognition because all faces must be extracted and then normalized to have the same orientation and size. It is also hard to decide the accuracy of the localization of the face; in general, the area should contain the mouth and the eyes, and background should be reduced as much as possible. The normalization step highly depends on the precise localization of eyes for example, and involves rotating the faces to the vertical position and scaling in order to have the same sizes.

Large amount of faces have been dropped in our experiments because we miss detected the eyes and we rotated wrong. We also had difficulties caused by wrong subject framing,

[11] The software can be found at www.onestopsoft.com

eyeglasses, beard and a high degree of variability of faces (Figure 7-1). In case of false detection or absent faces, decision is made at this module level.

7.3 Face Recognition

For both the Eigenfaces and Fisherfaces methods, we used the same test conditions. As we saw in Section 2.2, the main drawbacks of the image-based approaches are the changes in the head position and variation in illumination. The face detection process normalized the faces by all having the same size and rotation angle in the image plane[12], wherever we detected successfully the face outline and the eyes. Rotations in 3D were not considered. Although this database does not include variations in illumination, we processed the test images by simulating a source of light from the left, as shown in Figure 7-7

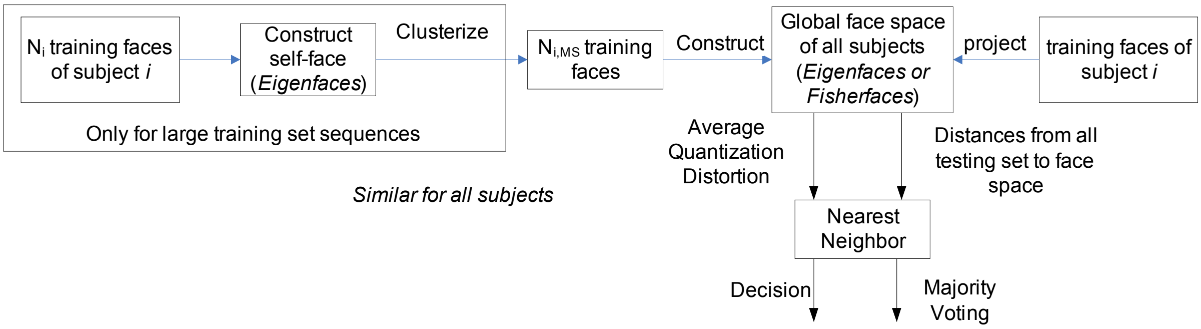
Original image Processed image

Figure 7-7: Example of original image (left), and an artificially generated illumination (right).

Figure 7-8 shows the two approaches we used according to the size of the training set. When we consider large training set, we clusterize first the face sequence as we explained in Section 4.2, and then we model the training faces for each subject by vector quantization. In this way, we extract the most significant faces to be the training set from which we build the face space of all subjects. However, face sequence clustering is not useful for a small training set. For classifying test faces, we first apply Nearest Neighbor method in terms of the distance from a testing face to the training model for one subject, calculated as the *average quantization distortion*. Second, we adopted *majority voting* strategy after we got the best match for each face in the testing set.

We show the identification rates calculated over the whole shots and at frame level. By shot level we refer to the number of corrected identified shots, while for frame level, the overall correct identified frames.

[12] Axis of eyes is horizontal



N_i training faces of subject *i*
Construct self-face (*Eigenfaces*)
Clusterize
$N_{i,MS}$ training faces
Construct
Global face space of all subjects (*Eigenfaces or Fisherfaces*)
project
training faces of subject *i*
Only for large training set sequences
Similar for all subjects
Average Quantization Distortion
Distances from all testing set to face space
Nearest Neighbor
Decision
Majority Voting

Figure 7-8: Different approaches for experiments on face recognition.

We summarize the parameters used in testing as shown in *Table 7-4*.

Table 7-4: Parameters used for testing face recognition.

Percent eigenfaces: • for self-space • for global face space	 25% from the whole sequence 25% = 29
Train model size:	4
Test Condition:	Train normal – test normal Train normal – test in changes conditions

We present in Table 7-5 and Table 7-6 the performance results for the Eigenfaces method according to the number of eigenfaces used for a train sequence equal to 10% time of the shot, and 30% for testing.

Table 7-5: Identification rate (%) by Eigenfaces method with respect to the number of eigenfaces for train model size = 4, and 29 × 4 = 116 training images.

Conditions / Number of Eigenfaces	Normal Light		Light changes	
	Shot	Frame	Shot	Frame
2	79.31	49.42	6.90	12.61
5	96.55	72.30	27.56	23.89
29 (25%)	100	84.06	75.86	60.63
58 (50%)	100	84.67	79.31	63.29
92 (80%)	100	85.33	75.86	62.89

Table 7-6: Identification rate (%) by Eigenfaces method with respect to the number of eigenfaces for MS = 8, and 29 × 8 = 232 training images.

conditions / Number Eigenfaces	Normal Light		Light changes	
	Shot	Frame	Shot	Frame
2	89.66	50.12	10.34	13.04
5	100	74.31	31.03	28.59
29 (12.5%)	100	83.84	72.41	59.25
58 (25%)	100	85.61	79.31	65.30
116 (50%)	100	86.29	79.31	66.13
185 (80%)	100	86.29	79.31	66.77

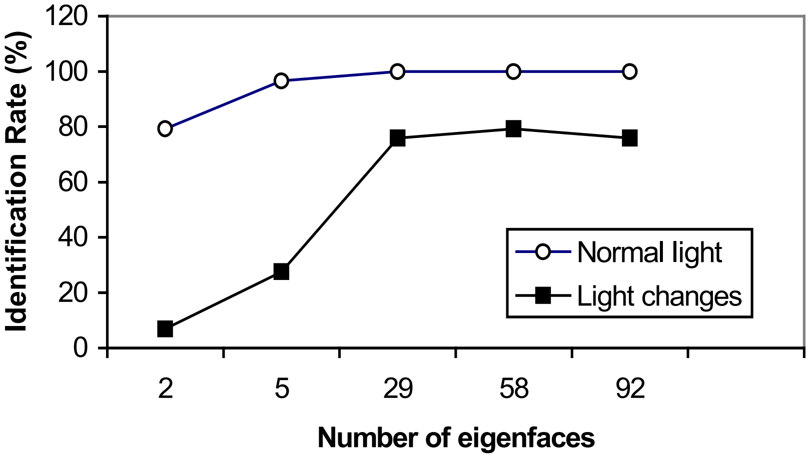


Figure 7-9: Identification rate related to the number of eigenfaces for N=4 training images for subject.

We show in Table 7-7 the identification rates for Eigenfaces and Fisherfaces methods in different test conditions, at shot and frame level. We use for training either one image or four images obtained by clustering the training shot.

Table 7-7: Identification rate (%) for Eigenfaces and Fisherfaces methods in different test conditions. The first row in cells shows majority voting results and the second shows the results by average distances.

Train shot size (%)	Test shot size (%)	Normal Light				Illumination variation			
		Eigenfaces		Fisherfaces		Eigenfaces		Fisherfaces	
		Shot	Frame	Shot	Frame	Shot	Frame	Shot	Frame
1 face	1	65.51 72.41	64.63	79.31 82.76	80.27	48.28 55.17	44.22	75.86 72.41	70.07
	30	82.76 75.86	60.54	86.21 93.1	67.01	34.48 37.93	32.96	72.41 65.51	57.97
10	1	100 100	99.32	100 100	100	82.76 82.76	76.2	96.55 96.55	93.20
	30	100 100	84.06	100 100	86.53	75.86 65.52	60.63	93.1 86.21	75.35
	90	100 100	78.79	100 100	80.16	75.86 68.97	57.00	93.1 79.31	69.54
30	70	100 100	88.30	100 100	89.56	51.72 48.27	44.38	100 96.55	81.16

We show in Table 7-8 the running times for the Eigenfaces and Fisherfaces methods. Training step includes constructing the self-eigenspaces and clustering, building the global face space by the two methods. Testing step consists in extracting features and matching. Experiments ran on an AMD Athlon XP 1600+, 256MB RAM computer.

Table 7-8: Running times for the experiments shown in Table 7-7.

Training				**Test**		
Size (%)	Time (sec)			Size (%)	Time (sec)	
	Clustering	Eigenfaces	Fisherfaces		Eigenfaces	Fisherfaces
10	53	6	180	90	178	209
				30	58	100
				1	8	40
30	204	7	207	70	118	179

Training video shot should include face variability as much as possible so that we can get distinct faces by clustering the sequence. Nevertheless, rotations in the image plane are performed by the face detection normalization, but 3D rotations are out of discussion without considering a 3D model of the face. Although we modeled each subject by selecting only their representative faces, the testing sequence does not include all the training states and thus we deal with one of the biggest drawback: the non-invariance to face rotations (tilted faces).

For clustering of face sequence one can argue that the self-space is too sparse. Despite this, the clustering performs well because the nearest faces to the centroids prove to be representative, as shown in Figure 4-4. We conclude that small variations in face expressions group together as we assumed. We could also try to reconstruct the centroids by using a small number of eigenfaces but we infer that reconstruction will not be so accurate to be used for training.

As depicted in Table 7-5 and Table 7-6 for the same number of eigenfaces, four images for training prove to be enough since there are small differences in results when using eight. Moreover, tuning the number of eigenfaces remains one of the most sensitive aspects since it implies the dimensionality of the space. We showed that a smaller set of eigenfaces is suitable for discrimination because we are not interested in reconstruction. Moreover, we need to optimize the trade-off between the performance and the computational costs. Of course, we tend to believe that the larger the dimensionality the better precision it is, but there are no significant improvements of the results, so a face space spanned over 25% eigenfaces from the whole training set seems to be sufficient in our case. The problem remains when a very large training set is used and decreasing the dimensionality will affect the classification. In general, tuning is done by performing experiments.

As we expected, when variations of light are present the results significantly decrease for Eigenfaces method but not so much for the Fisherfaces (Table 7-7). LDA proves to be much more accurate than PCA method in noisy conditions with the cost of computational time and resources for the very big dimensionality within- and between-classes scatter matrices, as shown in Table 7-8.

Definitely, choosing more than one training faces for each subject improves the accuracy in normal and noisy conditions, by getting even perfect results in the former scenarios. We considered the performance calculated at frame level and over the whole shots. The latter one is an example of mono-modal mono-expert fusion, and we implemented it by use of Majority Voting and by Nearest Neighbor after we calculated the average distance from the testing to the train model. The latter classification criterion performs worse than the former one when larger training and testing set are used.

A thorough look at majority voting results shows there are very less probable candidates while most of the models got zero probability, different from distances approach where all the models have some distances. We can see from Table 7-7 that the video sequences provide an advantage over recognition from only one image.

7.4 Speaker Recognition

Parameters used for speaker recognition experiments are summarized in *Table 7-9*. We used the same training and testing shot sizes as for face recognition in order to fuse the results from the same conditions. Speech stream is noiseless but we added white noise with SNR = 20dB to simulate noisy conditions. The identification results are shown in Table 7-10.

Table 7-9: Parameters used for speaker identification task.

Features type	Mel Frequency Cepstral Coefficients (MFCC)
Window size	30 ms
Window overlapping	33% (10 ms)
Window function	Hamming
Feature vector size	12
Modeling	Vector Quantization
Model size	64
Mel filters	30
Test conditions	train normal – test normal Train normal – noisy test, (SNR = 20dB; white noise)

Table 7-10: Identification rate (%) for speaker recognition classifier.

Train / test length	**SNR=∞**	**SNR = 20 dB**
30% / 70%	100.00	72.41
10% / 90%	100.00	62.07
10% / 30%	96.55	79.31
10% / 1%	51.72	17.24

Results of the speaker recognition classifier in noiseless conditions are good (Table 7-10), even for test shots of 40 seconds long (30%). Shots of 1.2 seconds length (1%) prove to be too short due to a large number of outliers.

Similar to the face expert, we experimented in noisy conditions so we could have several fusion scenarios, because it was no use to combine two reliable modalities. Results significantly degrade in this case. Shots of 84 (70%) and 108 (90%) seconds are too long for testing in noiseless speech but they even decrease the results in noise because of the many miss-classified features.

7.5 Fusion of Face and Speaker Recognition

We fused the speaker classifier results with eigenfaces results obtained by majority voting and average distances. For face classifier, the former results are in terms of *probabilities*, while the latter are in terms of *distances*. The speaker classifier provides results only in terms of distances. We normalized distances scores to range [0, 1] and then we mapped into probabilities as described in Section 6.5.

We used the same shot lengths in different test conditions for each biometric. We considered three different weights for both classifiers (Table 7-11) as well as a non-adaptive stream reliability function presented in Section 6.6.

Table 7-11: Weights used in Sum and Product rules.

	Face (%)	Speaker (%)
I	30	70
II	50	50
III	70	30
IV	Non-adaptive	

Table 7-12: Fusion results for 10% training and 30% testing shots. First row shows the fusion of the probabilities scores and the second the fusion of the distances scores.

Noise		**Individual**		**Fusion**							
F	**S**	**FR**	**SR**	**Weighted Sum**				**Weighted Product**			
				I	**II**	**III**	**IV**	**I**	**II**	**III**	**IV**
0	**0**	100	96.55	100	100	100	100	100	100	100	100
		100	96.55	100	100	100	100	100	100	100	100
0	**1**	100	79.31	93.1	96.55	100	100	100	100	100	100
		100	79.31	100	100	100	100	100	100	100	100
1	**0**	75.86	96.55	100	93.1	79.31	100	96.55	93.1	86.21	96.55
		65.52	96.55	93.1	89.66	86.21	100	96.55	93.1	86.21	100
1	**1**	75.86	79.31	89.66	82.76	79.31	82.76	93.1	93.1	86.21	93.1
		65.52	79.31	89.66	86.21	82.76	89.66	89.66	89.66	86.21	86.21

Table 7-13: Fusion results for 1 image and 10% speech long training, and 1% testing shots. First row shows the fusion of the probabilities scores and the second the fusion of the distances scores.

Noise		Individual		Fusion							
F	S	FR	SR	Weighted Sum				Weighted Product			
				I	II	III	IV	I	II	III	IV
0	0	65.51	51.72	75.86	75.86	72.41	79.31	75.86	72.41	72.41	72.41
		72.41	51.72	72.41	72.41	79.31	75.86	82.76	86.21	86.21	86.21
0	1	65.51	17.24	37.93	72.41	68.97	72.41	68.97	68.97	68.97	68.97
		72.41	17.24	48.27	72.41	79.31	72.41	68.97	82.76	79.31	82.76
1	0	48.28	51.72	65.51	58.62	48.28	51.72	51.72	51.72	51.72	51.72
		55.17	51.72	62.07	62.07	68.97	44.82	72.86	68.97	62.07	55.17
1	1	48.28	17.24	65.52	58.62	48.28	55.17	51.72	51.72	51.72	51.72
		55.17	17.24	62.07	62.07	68.97	62.07	75.86	68.97	62.07	68.97

Table 7-14: Fusion results for 1 image and 10% speech long training, and 30% testing shots. First row shows the fusion of the probabilities scores and the second the fusion of the distances scores.

Noise		Individual		Fusion							
F	S	FR	SR	Weighted Sum				Weighted Product			
				I	II	III	IV	I	II	III	IV
0	0	82.76	96.55	100	100	96.55	100	100	100	96.55	100
		75.86	96.55	100	100	100	100	100	100	100	100
0	1	82.76	76.31	89.66	89.66	86.21	86.21	96.56	96.56	96.56	96.56
		75.86	76.31	96.56	96.56	96.56	96.56	96.56	96.56	96.56	93.1
1	0	34.48	96.55	100	89.66	48.28	100	86.21	82.76	65.52	82.76
		37.93	96.55	93.1	86.21	72.41	93.1	96.56	82.76	75.86	100
1	1	34.48	76.31	86.21	72.41	44.82	44.82	79.31	75.86	68.97	68.97
		37.93	76.31	79.31	72.41	62.07	86.21	86.21	79.31	58.62	86.21

We chose to integrate the Eigenfaces results because this method is less reliable than Fisherfaces, and we wanted to show the improvements for low recognition rates of the classifiers. Simple rule of thumb takes its place in our results: *if one stream is more reliable and if it contributes more to the fusion, then the result is better*.

At a first glance, when used multiple images for training, the majority voting outperforms average distance criterion for individual classifier, but let us see what happens after fusion. We observe in the case of worse results for both classifiers, fusion distance scores outperform the one using probabilities scores. We argue this due to the mapping from distances to probabilities for the speaker classifier, because post-mapping is a heuristic method in practice.

Overall, fusion increases the performance compared to individual classifiers in most of the cases, even choosing equal weights for both experts. The non-adaptive weighting proved to be successful since it reached the maximum result from the three weights and even increased in some cases. Stream reliability has been calculated before fusion, over an arbitrary testing set.

8. Conclusions

In this work, we have addressed the problem of a bimodal biometric system to identify people from audio-video shots. The key aspect consisted of the fusion of face and speaker recognition. We covered both modalities with the emphasis on face recognition and the fusion techniques.

For face recognition, we first addressed the face detection task, which turned out to be very complex problem because of the high degree of variability of faces. For recognition task, we studied statistical methods such as Eigenfaces and Fisherfaces because they are straightforward to implement and we could present notions such as face space as well. The main drawbacks of this approach are the tilted faces and pose illumination changes. We relied on image sequences to increase the accuracy of the recognition and we presented an appropriate method for face modeling based on clustering of the video sequence.

We briefly introduced the field of speaker recognition and performed experiments using software developed at the University of Joensuu. We adopted the score level fusion strategy and showed that the joint use of both face and voice biometrics provides higher accuracy than the single modalities. Moreover, every small amount of information brought by one of the experts still counts for the overall system. The main problem consists of choosing appropriate weights according to the reliability of the individual streams, but we showed that a non-adaptive stream reliability approach performs well.

In our work, we also encountered the problem of continuous fusion, which by our knowledge has not been seriously treated until now. We propose this problem as well as algorithms for adaptive stream reliability for further research.

9. References

1 K. Aas, "Audio-Visual Person Recognition: a Survey", December 1996, http://www2.nr.no/documents/samba/research_areas/BAMG/Publications/FACE_96.ps&e=747

2 A. Albiol, L. Torres, E. J. Delp, Two are better than one: when audio comes to the rescue of video, WIAMIS 04

3 A. Altinok and M. Turk, Temporal Integration For Continuous Multimodal Biometrics, http://ilab.cs.ucsb.edu/projects/turk/AltinokTurk%202003.pdf

4 ATT Face database, formerly "The ORL Database of Faces", AT&T Laboratories, Engineering Department, Cambridge University, http://www.uk.research.att.com/facedatabase.html

5 E. Bailly-Baillière, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariéthoz, J. Matas, K. Messer, V. Popovici, F. Porée, B. Ruiz, and J.-P. Thiran. The BANCA database and evaluation protocol. In 4th International Conference on Audio- and Video-Based Biometric Person Authentication, AVBPA. Springer-Verlag, 2003. http://www.ee.surrey.ac.uk/banca/

6 Barlett, M. S., Lades, H. M. and Sejnowski, T.. Independent component representation for face recognition. In *Proceedings, SPIE Symposium on Electronic Imaging: Science and Technology*. 1998, pp. 528–539.

7 P. J. L. Van Beek, M. J. T. Reinders, B. Sankur, and J. C. A. Van Der Lubbe, Semantic segmentation of videophone image sequences, in *Proc. of SPIE Int. Conf. on Visual Communications and Image Processing, 1992*, pp. 1182–1193.

8 P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, IEEE Transactions On Pattern Analysis And Machine Intelligence, VOL. 19, NO. 7, JULY 1997, pp. 711-720

9 Berger, K. W., *Speechreading: Principles and Methods,* National Educational Press, Baltimore, pp. 73-107, 1972.

10 R. Brunelli and T. Poggio, Face recognition: Feature versus templates, *IEEE Trans. Pattern Anal. Mach. Intell.* 15, 1993, 1042–1052.

11 R. Brunelli, D. Falavigna, *Person Identification Using Multiple Cues*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-17, No. 10 (1995).

12 J. Cai, A. Goshtasby, and C. Yu, "Detecting Human Faces in Color Images," Proc. 1998 Int'l Workshop Multi-Media Database Management Systems, pp. 124-131, 1998.

13 Campbell, J.P., Speaker recognition: a tutorial, Proceedings of the IEEE , Volume: 85 , Issue: 9 , Sept. 1997, pp. 1437 – 1462

14 D. Chai and K.N. Ngan, "Locating Facial Region of a Head-and- Shoulders Color Image," Proc. Third Int'l Conf. Automatic Face and Gesture Recognition, pp. 124-129, 1998

15 R. Chellappa, C. L. Wilson, S. Sirohey, Human and Machine Recognition of Faces: A Survey, *Proc. of the IEEE,* Vol. 83, No. 5, May 1995, pp. 705-740

16 T. Chen, and R.R. Rao, "Audio-Visual Integration in Multimodal Communications," Proceedings of the IEEE, vol. 86, no. 5, pp. 837--852, May 1998.

17 T. Choudhury, B. Clarkson, T. Jebara, and A. Pentland. Multimodal Person Recognition using Unconstrained Audio and Video. Technical Report TR-472, MIT Media-Lab., 1998

18 I. Craw, H. Ellis, and J. R. Lishman, Automatic extraction of face-feature, *Pattern Recog. Lett.* Feb. 1987, 183–187.

19 J.L. Crowley and F. Berard, "Multi-Modal Tracking of Faces for Video Communications," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 640-645, 1997.

20 Cuave audio video database, Clemson University, http://ece.clemson.edu/speech/cuave.htm

21 C. Czirjek, N. O'Connor, S. Marlow, N. Murphy, Face Detection And Clustering For Video Indexing Applications, Centre for Digital Video Processing, Dublin University, Ireland

22 SESAM: A Biometric Person Identification system Using Sensor Fusion, Pattern Recognition Letters 18(1997), pp. 827-833

23 R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

24 J. L. Dugelay, J. C. Jungua, C. Kotropoulus, R. Kuhn, F. Perronnin, I. Pitas, Recent Advances in Biometric Person Authentication, ICASSP 2002

25 R.A. Fisher, The Use of Multiple Measures in Taxonomic Problems, *Ann. Eugenics*, vol. 7, pp. 179-188, 1936

26 Niall A. Fox, Ralph Gross Philip de Chazal Jeffery F. Cohn Richard B. Reilly Person identification using automatic integration of speech, lip, and face experts, Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications, Berkley, California, 2003, pp. 25-32

27 Rajkiran Gottumukkal and K. Vijayan Asari, "A robust face authentication technique based on composite PCA method," *Proceedings of the International Conference on Imaging Science, Systems,*

and Technology - CISST'03, Monte Carlo Resort, Las Vegas, Nevada, USA, vol. 1, pp. 207-213, June 23-26, 2003

28 V. Govindaraju, Locating human faces in photographs, *Int. J. Comput. Vision* 19, 1996.

29 H. P. Graf, T. Chen, E. Petajan, and E. Cosatto, Locating faces and facial parts, in *IEEE Proc. of Int.Workshop on Automatic Face-and Gesture-Recognition, Zurich, Switzerland, Jun. 1995*, pp. 41–45.

30 A. Hadid, M. Pietikäinen, Selecting Models from Videos for Appearance-Based Face Recognition, Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1, August 23 - 26, 2004, pp. 304-308

31 R. Herpers, H. Kattner, H. Rodax, and G. Sommer, Gaze: An attentive processing strategy to detect and analyze the prominent facial regions, in *IEEE Proc. of Int. Workshop on Automatic Face- and Gesture-Recognition, Zurich, Switzerland, Jun. 1995*, pp. 214–220.

32 R. Herpers, M. Michaelis, K.-H. Lichtenauer, and G. Sommer, Edge and keypoint detection in facial regions, in *IEEE Proc. of 2nd Int. Conf. on Automatic Face and Gesture Recognition*, Vermont, Oct. 1996, pp. 212–217.

33 E. Hjelmås, B. K. Low, Face Detection: A Survey, *Computer Vision and Image Understanding*, Vol. 83, 2001, pp. 236-274

34 E. Hjelmås and J. Wroldsen, Recognizing faces from the eyes only, in *Proceedings of the 11th Scandinavian Conference on Image Analysis, 1999.*

35 C. Jankowski, A. Kalyanswamy, S. Basson and J. Spitz, "NTIMIT: A Phonetically Balanced, Continuous Speech Telephone Bandwidth Speech Database", *Proc. International Conf. Acoustics, Speech and Signal Processing*, Albuquerque, 1990, Vol. 1, pp. 109-112.

36 A. Kanak, E. Erzin, Y. Yemez and A. M. Tekalp, Joint Audio-Video Processing for Biometric Speaker Identification, *IEEE Int. Conf. on Acoustic, Speech and Signal Processing*, Hong Kong, China, April 2003.

37 T. Kinnunen, Spectral Features for Automatic Text-Independent Speaker Recognition, Licentiate's Thesis, University of Joensuu, Finland, December 2003

38 S. G. Kong, J. Heo, B. R. Abidi, J. Paik, M. A. Abidi, Recent advances in visual and infrared face recognition – a review, to appear in Computer Vision and Image Understanding, 2004

39 Kuang-Chih Lee Ho, J. Ming-Hsuan Yang Kriegman, D., Video-based face recognition using probabilistic appearance manifolds, Proc. IEEE Computer Vision and Pattern Recognition, 2003, Vol. 1, pp. 313-320

40 M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. Malsburg, R. Wurtz , W. Konen, Distortion invariant object recognition in the dynamic link architecture, *IEEE Trans. Comput. 42*, 300–311, 1993

41 K. M. Lam and H. Yan, Facial feature location and extraction for computerised human face recognition, in *Int. Symposium on information Theory and Its Applications, Sydney, Australia, Nov. 1994.*

42 C. H. Lee, J. S. Kim, and K. H. Park, Automatic human face location in a complex background, *Pattern Recognition* 29, 1996, 1877–1889.

43 T.K. Leung, M.C. Burl, and P. Perona, "Finding Faces in Cluttered, Scenes Using Random Labeled Graph Matching," Proc. Fifth IEEE, Int'l Conf. Computer Vision, pp. 637-644, 1995.

44 J. Luettin. Visual Speech and Speaker Recognition. PhD thesis, University of Sheffield, 1997

45 J. Luettin and S. Dupont, Continuous Audio-Visual Speech Recognition, IDIAP-RR 98-02, 1998

46 M2VTS database, Internet URL: http://www.tele.ucl.ac.be/PROJECTS/M2VTS/m2fdb.html

47 J. MacQueen, Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1, pp. 281-296, 1967

48 D. Marr and E. Hildreth, Theory of edge detection, in *Proc. of the Royal Society of London, 1980.*

49 D. R. Martin, MC374 Topics in Computer Science: An Introduction to Computer Vision, Fall 2003, Boston College, Internet WWW-page, URL: http://www.cs.bc.edu/~dmartin/teaching/mc374f03/assignments/as6/ (01.17.2004)

50 J. S. D. Mason, F. Deravi, C. C. Chibelushi, S. Gandon, DAVID (Digital Audio Visual Integrated Database) - Final Report, Department of Electrical and Electronic Engineering, University of Wales Swansea, 26 Sept. 1996

51 K. Messer, J. Matas, J. Kittler, XM2VTSDB: The Extended M2VTS Database, Second International Conference on Audio and Video-based Biometric Person Authentication (AVBPA '99), Washington D.C, 1999, http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb/

52 Moghaddam, B. And Pentland, A.. Probabilistic visual learning for object representation, IEEE Trans. Patt. Anal. Mach. Intell. 19, 1997 696–710

53 Y. Moses, Y. Adini, S. Ullman, Face Recognition: The Problem of Compensating for Changes in Illumination Direction, *European Conf. Computer Vision*, 1994, pp. 286-296.

54 A. Nes, Hybrid Systems for Face Recognition, MSc. Thesis, University of Science and Technology, Norwegia, 2003

55 H. Pan and Ahi-Pei Liang and T. S. Huang, A New Approach to Integrate Audio and Visual Features of Speech, IEEE International Conference on Multimedia and Expo II, 2000, pp.1093-1096

56 Pentland, A., Moghaddam, B, Starner, T., View-based and modular eigenspaces for face recognition. In

Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, 1994

57 N. Poh and J. Korczak, Hybrid Biometric Person Authentication Using Face and Voice Features, Proc. 3rd International Conference AVBPA 2001, Sweden, June 2001, pp. 348-253

58 G. Potamianos, G. Gravier, A. Garg, W. Andrew, Recent Advances in the Automatic Recognition of Audio-Visual Speech, Proc. of the IEEE , vol. 91, no. 9, September 2003

59 Reynolds, D.A, An overview of automatic speaker recognition technology, Acoustics, Speech, and Signal Processing, 2002. Proceedings, IEEE International Conference on , Volume: 4 , 13-17 May 2002, Pages:IV-4072 - IV-4075

60 A. Ross, A. Jain, Information Fusion in Biometrics, Pattern Recognition Letters 24, 2003, pp. 2115-2125

61 T. Sakai, M. Nagao, and T. Kanade, Computer analysis and classification of photographs of human faces, in *Proc. First USA—Japan Computer Conference, 1972*, p. 2.7.

62 C. Sanderson and K. K. Paliwal. Information Fusion and Person Verification Using Speech & Face Information. IDIAP Research Report 02-33, Martigny, Switzerland, 2002

63 C. Sanderson, “The VidTIMIT Database”, IDIAP Communication 02-06, Martigny, Switzerland, 2002, http://www.eleceng.adelaide.edu.au/Personal/csanders/vidtimit/welcome.html

64 M. Savvides, B.V.K. Vijaya Kumar, P.K. Khosla, Eigenphases vs. Eigenfaces, ICPR 2004, III, pp. 810-813

65 L. C. De Silva, K. Aizawa, and M. Hatori, Detection and tracking of facial features by using a facial feature model and deformable circular template, *IEICE Trans. Inform. Systems* E78–D(9), 1995, 1195–1207.

66 S. A. Sirohey, Human face segmentation and identification, Technical Report CAR-TR-695, Center for automation research, University of Maryland, College Park, Nov. 1993

67 Lindsay I Smith, A tutorial on Principal Components Analysis, February 26, 2002, Internet WWW-page, URL: http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA-tutorial.pdf (04.10.2004)

68 M. Turk and A. Pentland, Eigenfaces for recognition, *J. Cog. Neurosci.* 3, 1991, 71–86.

69 M. Turk, A Random Walk through Eigenspace, IEICE Trans. Inf. & Syst., Vol. E84-D, No. 12, December 2001, pp. 1586-1595

70 P. Verlinde, and M. Acheroy, A Contribution to Multi-Modal Identity Verification Using Decision Fusion, Phd Thesis, Royal Military Academy, Signal and Image Centre, Belgium

71 L. Wang, T. K. Tan, Experimental results of face description based on 2nd-order eigenface method, ISO/MPEG 6001, Geneva, May 2000.

72 L. Wiskott, J.M Fellous , C. Malsburg,. Face recognition by elastic bunch graph matching. *IEEE Trans. Patt. Anal. Mach. Intell. 19*, 775–779, 1997

73 S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz, "Fusion of Face And Speech Data For Person Identity Verification," *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 1065-1074, Sept. 1999.

74 Yale Face Database, http://cvc.yale.edu/projects/yalefaces/yalefaces.html

75 M. Yang, D. J. Kriegman, N. Ahuja, Detecting Faces in Images: A Survey, *IEEE Transactions on Pattern Analysis And Machine Intelligence*, Vol. 24, No. 1, January 2002, pp. 34-58

76 G. Yang and T. S. Huang, “Human Face Detection in Complex Background*,” Pattern Recognition*, vol. 27, no. 1, pp. 53-63, 1994.

77 J. Yang and A. Waibel, A real-time face tracker, in *IEEE Proc. of the 3rd Workshop on Applications of Computer Vision, Florida, 1996.*

78 M.-H. Yang and N. Ahuja, “Detecting Human Faces in Color Images,” Proc. IEEE Int’l Conf. Image Processing, vol. 1, pp. 127-130, 1998

79 M.-H. Yang and N. Ahuja, “Gaussian Mixture Model for Human Skin Color and Its Application in Image and Video Databases”, Proc. SPIE: Storage and Retrieval for Image and Video Databases VII, vol. 3656, pp. 458-466, 1999.

80 M.H. Yang, N. Ahuja, and D. Kriegman, “Mixtures of Linear Subspaces for Face Detection,” Proc. Fourth Int’l Conf. Automatic Face and Gesture Recognition, pp. 70-76, 2000.

81 W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld, Face Recognition: A Literature Survey, *ACM Computing Surveys*, Vol. 35, No. 4, December 2003, pp. 399–458.