# *AAL-2009-2-049, ALIAS*
# *D3.3*
# *Documented identification system basing on face and voice*


| | |
|---|---|
| **Due Date of Deliverable** | 2011-01-31 |
| **Actual Submission Date** | 2011-01-31 |
| **Workpackage:** | 3 |
| **Dissemination Level:** | Public |
| **Nature:** | Report |
| **Approval Status:** | Final |
| **Version:** | v1.0 |
| **Total Number of Pages:** | 40 |
| **Filename:** | D3.3.pdf |
| **Keyword list:** | Biometrics, User identification, Face detection, Face recognition, Speaker recognition, Speaker diarization, ALIAS |

**Abstract**

This document describes the user identification modules that will be developed for the ALIAS Robot in order to distinguish between different enrolled users. The chosen biometric modes involve face recogntion and speaker recognition. This document outlines the general identification algorithms and related enrollment procedures that will be implemented within the ALIAS project.

The information in this document is subject to change without notice. Company or product names mentioned in this document may be trademarks or registered trademarks of their respective companies.

## History

| Version | Date | Reason | RevisedBy |
|---|---|---|---|
| 0.1 | 2010-12-02 | created [MMK] | Juergen Geiger |
| 0.2 | 2011-01-27 | first draft [EURECOM] | Ravichander Vipperla |
| 1.0 | 2011-01-31 | final version | Juergen Geiger |

## Authors


| Partner | Name | Phone / Fax / Email |
|---|---|---|
| MMK | Juergen Geiger | Tel: ++49 89 289 25734<br>Fax:<br>Email: geiger@tum.de |
| MMK | Tobias Rehrl | Tel: ++49 89 289 25734<br>Fax:<br>Email: tobias.rehrl@tum.de |
| EURECOM | Ravichander Vipperla | Tel: ++33 4 93 00 82 62<br>Fax:<br>Email: vipperla@eurecom.fr |
| EURECOM | Nick Evans | Tel: ++33 4 93 00 81 14<br>Fax:<br>Email: evans@eurecom.fr |

**Table of Contents**

# 1 Introduction

The primary objective of the ALIAS project is to develop a mobile robot platform that is designed to assist elderly users and people in need of care to continue independent living with minimal support from carers. The functionalities of the robot platform will include among other important services, the ability to interact with users, monitor their well being and provide cognitive assistance to them in day-to-day life.

The system is intended to be used for care at homes or in facilities such as nursing homes or elderly care homes. In such scenarios the robot is typically expected to interact with more than one user. Hence it is imperative for the robot to identify the current user correctly in order to deliver appropriate services and to personalise such services. Rather than using passwords or other more cumbersome, intrusive means of identification, the ALIAS robot will be equipped with face and speaker recognition capabilities so that identification may be performed automatically from distance and with ease and convenience.

Both modes of identification have their particular merits in this context. While generally giving acceptable levels of performance speaker recognition can be troublesome in the presence of background, ambient noise coming from the radio, television of other, competing speakers for example, and naturally requires the user to speak. Given the need for proactive care, face recognition is the most appealing means of identification in this case. In varying or poor lighting conditions, however, or when the user is out of the field of view, then identification by speaker recognition might be the only option and thus a combined identification approach has been adopted for the ALIAS project.

This document outlines the general identification algorithms and related enrollment procedures that will be employed within the ALIAS project to provide for user identification. The document also contains a brief overview of speaker diarization which is employed in combination with speaker recognition in order to handle multiple sound sources. The face recognition module will be developed by TUM whereas the speaker recognition module will be developed by EURECOM.

# 2 Biometrics

How best to confirm or verify someone's identity is an age-old problem. There are three different approaches in the form of something that we know, something that we have and something that we are: a biometric. Billed as a more efficient, universal, reliable and low-risk means of identification, biometric technologies have received a great deal of attention in the last decade.

Each biometric has its advantages and disadvantages, in terms of performance, cost, acceptability, etc. Among the so-called physical biometrics there are our fingerprints, hand geometries, retina, iris and faces. Among the more behavioral biometrics there are our signatures, gait and voices. The choice of biometric depends very much on the application, some diverse examples of which include access and border control, electronic commerce, telephone banking and user profiling for system personalization.

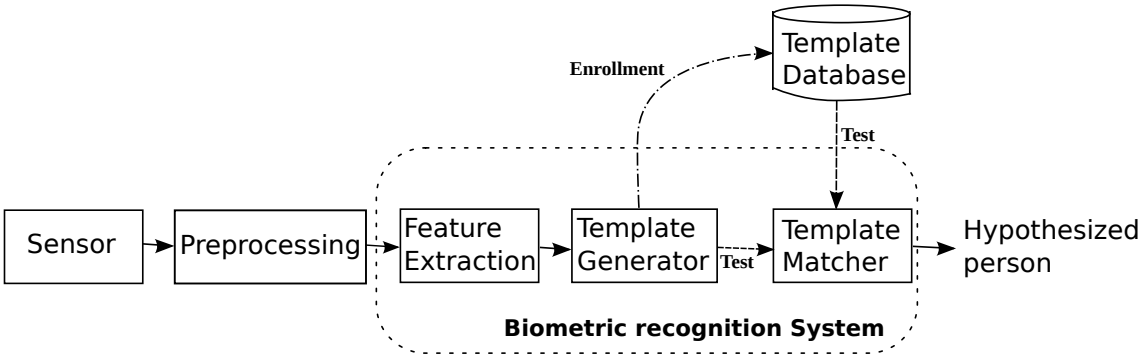



Figure 2.1: Biometric recognition system

Figure 2.1 illustrates the block diagram of a typical biometric recognition system. As seen in the figure, there are in general, at least two identifiable modules in a typical biometric system: that of learning or enrollment and that of comparison or recognition: identification (one-to-many) or verification (one-to-one). Optionally there is a third adaptation module which aims, for example, to track changes in a person's biometric over time.

During learning the biometric characteristics or traits are captured through an appropriate sensor. Captured signals are rarely stored in raw form and are, instead, usually transformed or parametrized to extract features, a compressed representation better suited to statistical pattern recognition. In doing so, we aim to remove redundant and/or noisy information and keep only that which is useful for biometric recognition. Features are first used to learn a statistical model, otherwise known as a template, which is a compact representation of the signals which facilitates recognition and reduces the quantity of data or features that must be stored by the system.

During recognition the biometric characteristics are captured again and are compared to the stored template(s). If they are sufficiently similar then they are deemed to come from the same person and a match is declared. If they are sufficiently dissimilar then a false match is declared. It is important that the sensor used during recognition should be as close a match as possible to that used for learning. Any differences between the sensors used for learning and recognition will result in differences in the statistical properties of the extracted features. Some form of compensation or normalization is then usually required to limit the subsequent degradation in recognition accuracies. What follows the generic 'recognition' stage differs according to the whether the biometric system performs identification or verification.

In identification mode the system aims to discover a person's identity, i.e. the system is required to answer the question, 'who am I?' In this mode the system compares the biometric signal with different models contained in its database (a 1-to-n problem). In general, when we refer to identification, we suppose that the problem is closed-set, i.e. that every potential user has a corresponding template in the database. Open set refers to the situation which includes previously unseen persons and is a potentially more difficult problem.

In verification mode there is a notion of a claimed identity and the system is required to answer the question, 'Am I who I say I am?' The user claims an identity and the system has to verify whether or not the identity of the individual is the same as that claimed. For verification, it is only necessary to compare the biometric signal with a single template in the database (a 1-to-1 problem). Here we generally suppose that the problem is open-set, in that not all potential users have a corresponding template in the database.

Identification and verification are two different problems. Identification can be a daunting task when the database contains thousands or even millions of identities, especially when the system is subject to real-time constraints. When a system functions in verification mode there are two types of error. It can either (i) reject a legitimate user, which we refer to as a false rejection, or (ii) it can accept an impostor, which we refer to as a false acceptation. The first cause of error stems from the inevitable variations in the environment or context in which they are used. In face recognition, for example, there is high potential for variability (e.g. lighting, pose and expression) and as a result it is possible to confuse two different people as the same person and equally two biometric samples from one person as belonging to two different people. This latter case is an example of a false rejection (FR). If the FR rate becomes too high then the system becomes unusable. To alleviate these problems the authentication protocol may be modified to make the matching process less stringent. This invariably leads to an increase in the false acceptance (FA) rate, the second cause of error. Together the FR and FA rates reflect the inevitable trade off between usability and security.

In the following chapters, we review the two biometric modes that are relevant to the ALIAS project. They are face and speaker recognition.

# 3 User Identification by Face

In recent years the detection as well as the recognition of persons have gained more and more attention in the image-processing domain. Several reasons are accountable for this fact: the processing power of computers has increased, cameras have become cheaper, sophisticated algorithms have been developed, large data sets have become available, etc. This chapter describes some well-known approaches as well as the algorithms used for the ALIAS system.

## *3.1 Basic Concepts*

In general up to now, two different approaches in the image-processing domain for recognizing persons are available: via face or via gait. Face identification and recognition can be used for admission control, as the storage of digital image data on passports is a first indicator of heading into that direction. Besides, automatic face identification and recognition can be used for the surveillance of public spaces as well as search for criminals. For the human-machine interaction point of view, the knowledge of the identity of the interaction partner can be used for the adaptation towards specified personal profiles as well as authentication for access to the computer. Especially the user identification and recognition basing on image-processing can be easily integrated in common media devices, due to the fact that most of the systems are equipped with a camera (e.g. notebooks, mobile phones).

As for the human-machine interaction point of view, in a robotic environment, the image-based person identification can also be used for the adaptation of the robotic platform as well as for authentication of the user. By knowing the identity of the user, the robot can for example present user-specific data (e.g. images) or the robot can adapt its behavior to the specific needs and abilities of the user. Furthermore, it is possible to provide certain applications only for specific users. For example, it would be possible that the robot can only be controlled by a user from a whitelist.

### 3.1.1 Task

In this chapter, we concentrate on the image-based identification of persons using face recognition. The ALIAS face recognition module performs two tasks, face detection and face identification. Before we delve into details of these separate tasks, we want to address some major parameters that can cause problems. There are several determining factors for the performance of a face recognition system. The main parameters are the illumination conditions, the head pose and the styling, occlusions and mimics. Illumination plays a

major role in face identification. The best condition would be constant lighting. In this case, no special algorithms to reduce the influence of lighting are required. In other cases, performance of a face identification system can degrade heavily due to changing lighting conditions. The biggest problems occur due to shadows and under- or overexposure. Especially in a robotic environment, constant illumination conditions cannot be taken for granted. A robot in a home scenario will cause many different conditions. For example, the robot will move into different rooms with different conditions. Furthermore, the robot may be directed towards a window or a source of light, or, in the other case, may look in the opposite direction. In order to cope with these changing conditions, the influence of other factors should be kept at a minimum. The second major factor is the head pose. The geometric orientation of the face that is to be identified influences the performance of the identification system. When the face is always recorded from a frontal point of view, the task is kept simple. For ALIAS, we can solve this problem by providing a module that is only activated when a person is directly in front of the robot and looking at the camera. Head styling is another important factor for face identification. Cosmetics, hair style, facial hair and glasses all influence face identification process. Occlusions can also be ruled out for the ALIAS system, as it is assumed that the ALIAS face identification module is only activated when a person is directly in front of the robot.

### 3.1.2 Face Recognition

The area of face recognition covers several different tasks, which we want to describe in more detail in this chapter. In order to be able to identify a face, it has first to be detected in the current image. Afterwards, the area in the image containing the face can be passed to the face identification or face verification module. For the task of face identification, the system has several hypotheses for the identity of a face and has to decide for one of them. A different task is face verification. In face verification, for a given face, the system has only two hypotheses. It must decide if a face is of a given identity or not.

#### Face Detection

Face detection is the task of determining if and where in an image faces are located. In the face detection step, sensor data are the input. The output are coordinates of regions containing faces. A common description of such a region is a bounding box, with its coordinates $x \in [x_l, x_r]$ and $y \in [y_b, y_t]$, where $x_l$ and $x_r$ are the left and right borders, respectively, and $y_b$ and $y_t$ are the bottom and top borders of the bounding box, respectively. Typically, a face detection system does not provide scaling and rotation of the detected face.

**Face Identification**

The task of face identification can be described as follows. An image segment with a detected face is provided to the system. The system has a database of known faces and must decide for one of these, whichever is the most similar to the face in the given image segment. This can be done by calculating a similarity (or distance) measure between the unknown face and all of the faces in the database. The system then decides for the face in the database with the largest similarity (or smallest distance). Hence, face identification is a $1 : n$ comparison.

**Face Verification**

Face verification (also known as face authentication) covers the scenario when a person claims to have a certain identity and the system has to decide if this is true. The system decides if the person has the claimed identity or if not. Thus, this is a 1:1 comparison. Face verification does also make use of a database for all known faces. This database is constructed in the training phase. Furthermore, an appropriate threshold for a similarity or distance measure has to be found.

## *3.2 General Approaches*

For the task of identifying a person by its face seen from a frontal direction under good illumination conditions, several different approaches have been developed over the years, all of which show good results. In the next sections, we describe different approaches to the tasks of face detection and face identification.

### 3.2.1 Face Detection

Face detection is the task of finding regions in an image that contain faces. The most common approaches are the approach of Viola and Jones [58] using adaptive boosting [26] and Haar like features and the approach of Rowley [51] which uses neural networks [8] to find faces. These two detection algorithms will be described in the following two sections. An overview over other approaches can be found in [63]. Apart from the specific approaches for face detection, more general approaches like foreground and background segmentation by background subtraction, detection of specific colors, detection of movement also play a role. These approaches are very often combined with the approaches described here. A description of these approaches can be found in standard references like [34, 3, 52].

### Detection with Neural Networks

The approach of detecting faces with neural networks [8] can find frontal faces in gray scale images. The algorithm applies a sliding window technique to classify the selected region with regard to a two class problem: presence of a face or absence of a face. A sliding window is a region with a defined size, which is stepwise placed on the input image to crop certain regions. These selected regions are afterwards presented to a classifier. However, to accomplished the detection of faces varying in size, the input image is downscaled by applying a low-pass filter operation followed by an undersampling step to decrease the size of the input image stepwise.

The classification of the selected regions is performed via a multi-layer perceptron [8]. The approach of [51] bases on a input layer of 20x20 pixel region, a hidden layer, and an output layer, which represents the presence or absence of a face. Rowley [51] applied the following structure for the neural network, which proved itself as suitable for detecting faces. Three different forms of receptive fields are used, each receptive field is connected with one neuron of the hidden layer. The three receptive field forms were designed according to the task to represent structures, which can be used to characterize a face. The structure of the receptive fields and their three different forms can be seen in Figure 3.1. One form of the hidden unit consists of 4 10x10 pixel subregions, the second receptive field form is composed of 16 5x5 pixel subregions, and the third form comprises 6 overlapping 20x5 pixel horizontal stripes of pixels. The three different receptive field forms were designed to represent specific face features. The idea behind the horizontal stripes is to emulate the horizontal eyes region as well as the mouth region. The hidden layers connected to the receptive fields having a square form should detect the following face features: nose, corners of the mouth, and eyes.

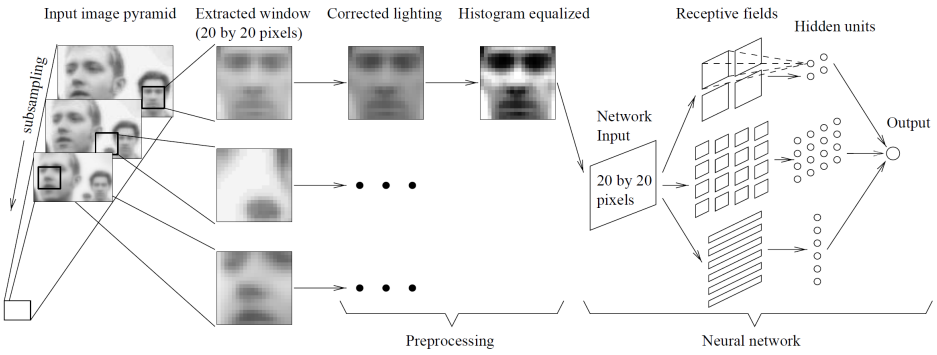



Figure 3.1: Neural Network approach presented in [51].

In general, to train a neural network classifier to the task of detecting a face in an image, a lot of training data is required to accomplish a good classification performance. For the training, the position of the eyes, the tip of the nose, the corners and the center of the mouth are labelled by hand as input for the normalization of the image data with regard to

scale, orientation, and position. From each original face of the training set additional 15 training images are generated by additional rotation, translation, scaling, and mirroring. A further preprocessing step of the training phase is the lighting correction and histogram equalization of the 20x20 pixel subregion, which are handed over to the neural network for training purposes. The instances for the negative training examples are composed of random extracts of images comprising no faces.

### Detection with Viola-Jones Algorithm

The reason why the approach of P. Viola and M. Jones [58, 59] became so popular in the image-processing community to detect faces is its effective and fast processing and the achievement of high detection rates. The fast processing speed of this approach is based on three facts: first, the applied Haar-like features computed on a so-called *integral image*, second, adaptive boosting (adaBoost) [26], and third, the cascade structure of classifiers (within increasing complexity).

The features used in the cascade classification structure are so-called Haar-like features, because they have resemblance with Haar-Wavelets basing on the theory of orthogonal functions by A. Haar [30]. The computation of the Haar-like features is conducted by relying on the *integral image* described below. The reason for applying features and not directly pixels in that case is due to the following considerations: first, a higher processing speed can be achieved by applying features, second, features represent semantic information, which can be easier extracted from a finite training set.

The *integral image ii* can be used to compute the sum of an arbitrary rectangular image subregion fast and efficiently. The *ii* contains the sum of all pixels from the left upper corner to the current pixel position $(x, y)$ of the input image $I$, thus the $ii$ is given by:

$$ii(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j). \tag{3.1}$$

For enhancing the efficiency of the computations of the $ii$, the neighboring relation can be exploited. The summation process is split up into two steps: summation column-by-column and summation line-by-line, thus an efficient way of computing the $ii$ can be found. The cumulative line sum is given by:

$$s(x, y) = \sum_{i=0}^{x} I(i, y), \tag{3.2}$$

which is followed by cumulative column sum:

$$ii(x, y) = \sum_{j=0}^{y} s(x, j). \tag{3.3}$$

The summation of the cumulative sums can be expressed in an iterative way, thus resulting in an efficient implementation:

$$s(x+1,y) = s(x,y) + I(x+1,y), \tag{3.4}$$
$$ii(x,y+1) = ii(x,y) + s(x,y+1). \tag{3.5}$$

Having the efficient computation for an $ii$ at hand, the sum of pixels within an arbitrary image rectangular can be determined in the following way:

$$S = ii(x_4,y_4) + ii(x_1,y_1) - ii(x_2,y_2) - ii(x_3,y_3), \tag{3.6}$$

where $p(x_1,y_1)$ is the upper left corner of the image rectangular, $p(x_2,y_2)$ is the upper right corner, $p(x_3,y_3)$ the lower left corner, and $p(x_4,y_4)$ is the lower right corner. The computation of the pixel sum of the rectangular $S$ is shown in the Figure 3.2 depicted below.

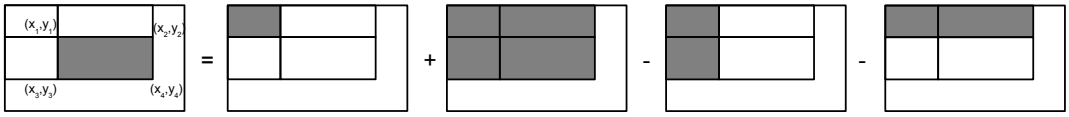



Figure 3.2: Computation of the pixel sum in the rectangular $S$.

The reason for setting up this efficient method is due to the computation of the Haar-like features. As mentioned above these so-called Haar-like features have resemblance with the Haar Wavelets, and they are designed the following way: Within a defined rectangular region of an input image $I$, the sum between different subregions is computed. Subregions indicated by a white color are weighted in a positive way, whereas the subregions indicated with a black color are weight in a negative way. In general, the Haar-like features can vary in size and position and are composed of two, three, or four rectangular subregions (see Figure 3.3 for some examples). The base resolution for the applied face detector is 24x24, thus over 180.000 rectangular features are thinkable.

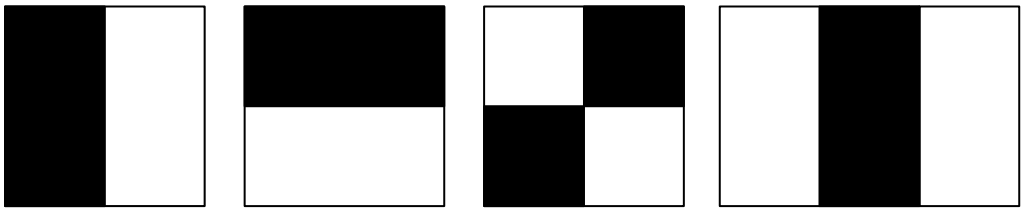


Figure 3.3: Examples of Haar-like features.

The result of the computation of the Haar-like features represent certain characteristic of the input image: edges, texture changes, borders between light and dark image regions.

The value of the applied Haar-like feature $f$ is given by the sum of the positive and negative weighted image subregions.

As mentioned above, adaBoost is applied to set up a cascade of weak classifiers for detecting the human face, therefore, the negative and positive training examples are used to determine an optimal classification threshold ensuring a minimal number of misclassifications.

For a Haar-like feature $i$ and a corresponding threshold value $\theta_i$ a weak classifier $h_i$ can determine in a defined image region $x$ if either a face is contained or not. The classifier can be interpreted in a geometric way, building a hyperplane separating the two classes. The parity $p_i$ is applied to determine the location of the two classes with regard to the hyperplane. Thus, the weak classifier is given by:

$$h_i(x) \begin{cases} 1 & \text{if } p_i f_i(x) < p_i \theta_i \\ 0 & \text{else.} \end{cases} \tag{3.7}$$

As mentioned above adaBoost is applied for detecting a face. The idea behind adaBoost is to combine several $T$ weak classifiers $h_t(x)$ in a cascade resulting in one strong classifier $H(x)$.

$$H(x) \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{else.} \end{cases} \tag{3.8}$$

In the following the adaBoost algorithm for classifier learning is given (see Algorithm 1), in each round one feature is selected from 180.000 possible candidates.

A detector, trained with Algorithm 1, already provides a suitable detection rate, however, the computational speed is related in a linear way to the selected features. For increasing the processing speed, a cascade of classifiers can be set up combining several classifiers $H_i(x)$.

The approach of constructing the cascade of classifiers is the following: In the beginning boosted classifiers are used, which are smaller and thus having a more effective processing speed. These classifiers are designed to have a high detection rate and reject many subregions comprising no face. These simple classifiers are followed by complex classifiers achieving low false positive rates.

The general idea behind the cascade approach is the following: The simple classifiers in the beginning should select all subregion having a face and discard many subregions having no face. Thus, many subregions having no faces should be discarded in the early stages of the processing. In the following cascade steps, the classifiers are more complex to reduce the false positive rate. The increase in the processing speed is due to the fact, that many subregions comprising no faces are discarded by the simple classifiers, which have good processing speed, thus the complex classifiers have to process much less subregions.

**Algorithm 1** AdaBoost algorithm for face detection from [58].

The following steps have to be performed:

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ describe negative or positive examples, respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where $m$ and $l$ are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
     $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
     such that $w_t$ is a probability distribution.
  2. For each feature $j$, train a classifier $h_j$ which is restricted to using a single feature. The error is evaluated with respect to $w_t$, $\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$.
  3. Choose the classifier $h_t$, with the lowest error $\epsilon_t$.
  4. Update the weights:
     $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$
     where, $e_i = 0$ if example $x_i$ is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:

$$H(x) \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{else.} \end{cases}$$

  where $\alpha_t = \log \frac{1}{\beta_t}$.

The classification cascade composed of the classifiers $H_1(x), \ldots, H_T(x)$ is depicted in Figure 3.4 shown below.

### 3.2.2 Face Recognition

In the image-processing community several approaches were proposed to fulfill the task of face recognition, however, the differences between recognition approaches and feature extraction approaches are not sharply defined. Often a specific feature extraction method is coupled with a classical classification method (e.g. distance classification) forming face recognition approaches. Popular approaches for face recognition are eigenfaces, pseudo two-dimensional Hidden Markov Models (p2dhmm), Active Appearance Models or Elas-

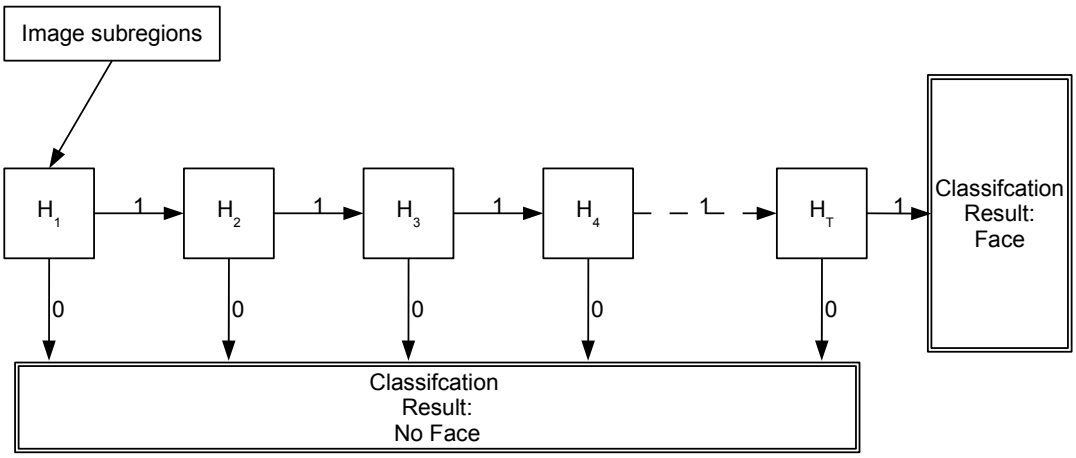



Figure 3.4: Cascade of classifiers.

tic Bunch Graph Matching, for example. A description of these and other approaches can be found in [65]. In the following two subsections, only the eigenfaces approach and the p2dhmm approach will be delineated.

### Eigenfaces

The so-called eigenfaces [54, 56, 57] is a well-known approach in the image-processing community. A crucial processing step in the eigenfaces approach relies on the principal component analysis (PCA) [8].

Starting point for the face recognition via eigenfaces are face images $I(x, y)$, which are gray scale and have the same size $N$x$N$ (it is assumed that the images have a quadratic form, however, it is not a neccessity). The idea behind is that face images can be represented in a small dimensional subspace of the original image space, which is given by $N$x$N$, because the face images have similarities in their entire configuration and thus are not randomly distributed in the entire image space.

The face images $I(x, y)$ are represented via column vectors having the length of $N^2$. From these column vectors $\vec{x}_1, \ldots, \vec{x}_M$ an average vector is computed the following way:

$$\overline{\vec{x}} = \frac{1}{M} \sum_{i=1}^{M} \vec{x}_i. \tag{3.9}$$

Having the average vector, a matrix $\mathcal{A}$ is computed consisting of the input data $\vec{x}_1, \ldots, \vec{x}_M$

subtracted by the average vector $\overline{\vec{x}}$.

$$\mathcal{A} = [\vec{x}_1 - \overline{\vec{x}}, \ldots, \vec{x}_M - \overline{\vec{x}}]. \tag{3.10}$$

Depending on the matrix $\mathcal{A}$ the covariance matrix $\mathcal{C}$ is determined:

$$\mathcal{C} = \frac{1}{M-1}\mathcal{A}\mathcal{A}^T. \tag{3.11}$$

The eigenvectors $\vec{e}$ of the covariance matrix $C$ hold the following equation:

$$\mathcal{C}\vec{e} = \lambda\vec{e}. \tag{3.12}$$

The matrix $\mathcal{C}$ is real-valued and symmetric, thus the eigenvectors of $\mathcal{C}$ are orthogonal. If the eigenvectors are converted back into two-dimensional images, it can be seen that the eigenvectors represent characteristic face structure. This is the reason for naming these features eigenfaces.

A subset $k$ of the eigenvectors $\mathcal{E}_k(\vec{e}_1, \ldots, \vec{e}_k)$ and the average face image $\overline{\vec{x}}$ is selected to represent faces by projecting the face image $\vec{x}$ to a subspace of dimensionality $k$ by selecting $k$ eigenvectors resulting in a vector $\vec{w}$. The representation of a face $\vec{x}$ is conducted the following way:

$$\vec{w} = \mathcal{E}_k^T(\vec{x} - \overline{\vec{x}}). \tag{3.13}$$

As can be seen in Equation (3.13), the amount of selected eigenvectors is variable. Therefore, the quality of the representation $\vec{w}$ depends on the selected amount of eigenvectors. The eigenvectors $e_i$ are selected according to their eigenvalue $\lambda_i$, which are sorted with regard to their value. The lower bound for the quality $q$ is related to the $k$ first selected eigenvalues the following way:

$$q(k) = \frac{\sum_{i=1}^{k}\lambda_i}{\sum_{j=1}^{N}\lambda_j}, \tag{3.14}$$

where $q$ is in the interval $[0, \ldots, 1]$, whereas the values close to $1$ represent higher quality. The backprojection $\vec{x}_b$ is given by:

$$\vec{x}_b = \overline{\vec{x}} + \mathcal{E}_k\vec{w}. \tag{3.15}$$

The classification in the eigenfaces approach is based on distance measure determining the difference of the projection $\vec{w}_y$ of an unknown input image $\vec{y}$ to the projections

$\vec{w}_{db_1}, \ldots, \vec{w}_{db_N}$ of all images $\vec{x}_{db_1}, \ldots, \vec{x}_{db_N}$ stored in the testing data base. The distance can be measured for instance via the Euclidean distance

$$d_{euc}(\vec{w}_y, \vec{w}_{DB_i}) = \sqrt{(\vec{w}_y - \vec{w}_{DB_i})^T(\vec{w}_y - \vec{w}_{DB_i})}, \tag{3.16}$$

or the Mahalanobis distance

$$d_{mah}(\vec{w}_y, \vec{w}_{DB_i}) = \sqrt{(\vec{w}_y - \vec{w}_{DB_i})^T\mathcal{C}^{-1}(\vec{w}_y - \vec{w}_{DB_i})}. \tag{3.17}$$

The assignment of the unknown image $\vec{y}$ to one of the faces $i$ of the data base is conducted by finding the database representative $\vec{w}_{DB_i}$ minimizing the $d_{measure}(\vec{y}, \vec{w}_{DB_i})$ given by:

$$\widehat{i} = \underset{i}{\operatorname{argmin}}\, d_{measure}(\vec{y}, \vec{w}_{DB_i}). \tag{3.18}$$

**Pseudo Two-Dimensional Hidden Markov Models**

Hidden Markov Models are stochastic finite automatons, where a first order Markov chain $X_t$ generates observations $Y_t$. These two processes are random, where the underlying generating process of the Markov chain is hidden, thus the name Hidden Markov Model. However, the name hidden refers only to the actual states of the Markov chain and not to the parameter set $\theta$ describing the Hidden Markov Model itself.

The area of application of Hidden Markov Models covers a wide range, most commonly they are used for speech recognition. Other applications are hand writing as well as gesture recognition and face recognition. In general, Hidden Markov Models are a special case of Dynamic Bayesian Networks, where the topology of the underlying graph connecting the different nodes of the random processes can vary. Hidden Markov Models for face recognition have itself a special structure known as pseudo two-dimensional Hidden Markov Models (p2dhmm). This structure results in a topological order improving the representation of two dimensional data structures like images.

Before the p2dhmm is introduced, some important characteristics about Hidden Markov Models are given, easing the understanding of p2dhmms.

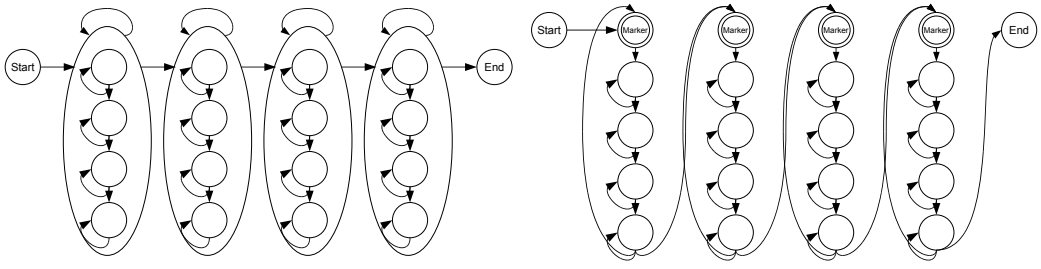
The underlying hidden Markov chain $X_t$ is in general of discrete nature comprising $K$ possible states, whereas the observation sequence can be either discrete comprising $L$ states or continuous $Y_t \in \mathbb{R}^L$. The parameters $\theta$ of the Hidden Markov Model can be subdivided into three different kinds:

- The initial state distribution $\pi(i) = P(X_1 = i)$ given the probability that the hidden Markov chain starts with state $i$.

- The transition model $A(i,j) = P(X_{t+1} = j|X_t = i)$, where $A$ is a matrix, where each row sums to one. Often, the matrix $A$ is sparse, or the structure gives a left-right transition matrix, meaning that states can only transit into themselves or higher numbered states.
- The observation model characterizes the relationship between the hidden state of the Markov chain $X_t$ and the actual observation $Y_t$. Depending on the nature of the observation (discrete or continuous), the observation model is defined via a matrix (discrete case) or via a Gaussian or Mixture of Gaussians (continuous case).
  - Discrete case:
    $$B(i,k) = P(Y_t = k|X_t = i). \tag{3.19}$$
  - Gaussian:
    $$P(Y_t = y|X_t = i) = \mathcal{N}(y; \mu_i, \Sigma_i), \tag{3.20}$$
    $$\text{where } \mathcal{N}(y; \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{L/2}|\Sigma|^{\frac{1}{2}}} \exp(-\tfrac{1}{2}(y-\mu)'\Sigma^{-1}(y-\mu))$$
  - Mixture of Gaussians:
    $$P(Y_t = y|X_t = i) = \sum_{m=1}^{M} P(M_t = m|X_t = i)\mathcal{N}(y; \mu_{m,i}, \Sigma_{m,i}). \tag{3.21}$$

The classification applying Hidden Markov Models tries to find the the parameter set $\theta^*$ which maximizes the probability $p(Y_T|\theta^*)$ for a given observation sequence $Y_T = (y_1, \ldots, y_T)$.

As mentioned above p2dhmms have a certain topological order, which is more suited to emulate two dimensional signals (e.g. images). This topological order can be seen in Figure 3.5(a).




(a) p2dhmm: Compact representation. (b) p2dhmm: Detailed representation.

Figure 3.5: Overview over p2dhmm representations.

The general idea behind this structure is the following: The columns of the two-dimensional input signal (here: the gray scale image) are mapped to left-right transition Hidden Markov Models. These Hidden Markov Models of the image columns are afterwards connected

building themselves a right-left transition Hidden Markov Model. The connection of the columns is accomplished by connecting the last hidden state of column $c_{n-1}$ to a so-called marker state of column $c_n$. The integration of the marker states into the p2dhmm can be seen in Figure 3.5(b). The purpose of the marker state is to ensure that the signal is represented in the two-dimensional form, because only when the marker state of column $c_{n-1}$ has a certain value a transition to the first hidden state of column $c_n$ takes place. This measure ensures, that only entire image columns are modeled by the column Hidden Markov Models.

For p2dhmms the general tools the *Expectation Maximization Algorithm* [6] for learning as well as the *Viterbi Algorithm* [61] for finding the most probable sequence of hidden states can be applied. One important fact that should be mentioned here briefly is that the observations for the Hidden Markov Models are not directly the pixel values of the gray scale images, but features computed on them or on blocks of pixels.

## *3.3 ALIAS Face Identification System*

In this section, we present the approaches we use for the ALIAS face identification module. For face detection, the Viola-Jones Algorithm as described in Section 3.2.1 is used. For face identification, we use an eigenface approach, see Section 3.2.2.

### 3.3.1 Algorithm

Our approach uses the Viola-Jones Algorithm for face detection and eigenfaces for face identification. The system fetches an image from the camera and first of all transforms the image to greyscale, since the Eigenface approach is done on greyscale images. Afterwards, the face detection is started. This is done with the Haar cascade classifier (Viola-Jones Algorithm). Only the hypothesis with the highest probability is returned, in the form of a rectangle as a bounding box of the face. The image segment containing the face is then normalized in contrast and brightness, to minimize the influence of the lighting conditions. Then, the face identification is started. For this purpose, the image segment is projected onto the PCA subspace. The identity is then determined by classifying the unknown face with a nearest neighbor classifier. In addition to the identity, a so-called confidence $c_y$ is calculated. This is done in the following way, using the Euclidean distance from Equation (3.16):

$$c_y = 1 - \frac{\sqrt{\frac{d_{euc}(\vec{w}_y, \vec{w}_{DB_i})}{n_{train} * K}}}{255}, \tag{3.22}$$

whereby $n_{train}$ is the number of training images and $K$ is the number of eigenfaces. The confidence $c$ lies always in the interval between 0 and 1. In order to make the recognition

results more robust, $N$ images are taken and classified. The final decision is then made on the average confidence $\bar{c}$,

$$\bar{c} = \frac{1}{N} \sum_{n=1}^{N} c_y \tag{3.23}$$

over all $N$ images. The average confidence $\bar{c}$ is calculated for every identity in the database and the system then decides for the identity with the maximum $\bar{c}$ . To be able to classify an unknown face, a data base of known faces is needed. Therefore, in a training phase, training images are recorded, whereby the identity of the user is known.

### 3.3.2 Implementation

The ALIAS prototype makes use of an openCV (version 2.1) implementation.

#### Training phase

The module can be started in a training mode. In this mode, the system needs to know the identity of the user it wants to learn. The system also needs to be served with the length of the training phase. The user should be in front of the camera. Then, for the length of the desired training phase, the camera takes pictures and performs face detection. The image segments of the detected faces are preprocessed, stored and added to the face data base. If the person was previously unknown, the name of the new person (as provided to the module) is also added to the list of known persons. The training mode can be started several times to add pictures of different persons or of the same person in order to reduce the influence of the recording conditions.

#### Recognition Phase

In the recognition phase, the described module can be used to identify the person that is in front of the robot. When the module is started it needs the length of a time interval as a parameter, similar to the training phase. For the length of the time interval, it will then take pictures, detect the faces, and assign one of the identities of its database to the unknown faces. When the time interval ends, a decision for the identity of the unknown person is made by deciding for the identity with the largest average confidence over all images recorded during this interval. The result of the identification process can then be given to the dialog manager, for example. Currently, the results are written in a text file.

There are also several other functions, for example to show all users that are in the database or to delete a specific user.

### 3.3.3 Restrictions and open issues

For the ALIAS face identification system, we make several recistrictions and assumptions to simplify the recognition procedure and to improve the recognition results. The ALIAS face identification module will be triggered on demand. This means, that it is assumed that the module is only started when a person is in front of the camera and that only one person is there. This simplifies the task as the system can assume to have a frontal image to work with. Furthermore, the system assumes that there are no occlusions. Additionally, the system has to work with a limited set of persons. The recognition accuracy increases with decreasing number of possible persons, therefore a small number of possible persons is beneficial for the system. There are several specific problems for the ALIAS face identification module. For example, as the robot can move around, it will always encounter different illumination and background conditions. This can lead to difficulties in the identification process, as the recognition results are highly dependant on the conditions during the training phase. In order to minimize the effect of changing illumination and background conditions, there is always the possibility to retrain the system by adding new images of an already known person to the database. The system can then, for example, be trained several times, each time with different conditions.

With the current version of the module, the results and feedback of the module are written in a text file. In order to be able to communicate with the dialog manager, an interface needs to be defined. Furthermore, the current version is just a prototype that runs with a webcam. The integration on the robotic platform with the connection to the omnicam needs to be done.

# 4 User Identification by Speech

Speech signals not only carry the message to be communicated but also paralinguistic information about a speaker's identity. Each speaker has a distinct voice signature characterized by pitch and timbre, speaking style, accent, prosody and speaking rate. There are also subtle differences in the language usage and frequently used words which can also help in distinguishing between speakers to a certain extent.

Automatic speaker recognition (ASR) systems attempt to identify a person based on his/her input speech. Such systems have widespread appeal in several real life scenarios. In applications such as secure telephone banking, speech is the only biometric mode available to authenticate a user. Speaker recognition is a useful tool in forensic analysis to establish the identity of speakers in general surveillance or suspicious conversations. In tasks such as multimedia indexing, for example, automatic annotation of meeting room discussions or broadcast news, associating segments of speech to a particular speaker is an important sub-task. Speaker recognition is also an essential step towards personalizing spoken dialogue systems in order to make man-machine interactions as natural as possible. This latter example is the most relevant to the use of ASR within the ALIAS project.

One particular advantage in using speech for identifying users of the ALIAS Robot is the ease with which the signal can be obtained. Speech signals are readily captured in almost any environment using standard microphones and recording equipment and do not depend of the orientation of a camera or relative position of the subject. It is further independent of occlusion and inter-session variations in illumination, pose or expression which often degrade the performance of face recognition systems in similarly uncontrolled contexts. Speaker recognition, however, is not without its own specific issues related to inter-session variation. Ambient noise, differences in the linguistic context, a persons state of health or emotional state all influence performance. The quantity of data is also an important factor. Whereas face recognition may only require a single image, speech signals are dynamic, i.e. information is contained within its variation over time. Sufficient data is thus required for acceptable performance and place certain constraints on viable applications and contexts relevant to the ALIAS project.

In the following we outline the basic concepts and general approaches to ASR. Since it is inextricably linked to speaker recognition we further describe related speaker diarization technology which allows speaker recognition to be deployed in multi-speaker contexts. Finally we describe the specific ASR system that will be exploited in the ALIAS project.

### 4.1 Basic concepts

Depending on the application speaker recognition systems can be either a) text-dependent or b) text-independent. In text-dependent systems, the utterances to be spoken by the user for identification are predefined. Such systems are typically used in biometric authentication applications where the user is cooperative. For instance in secure telephone banking, the user speaks his password and the system determines whether or not the utterance corresponds or matches the template. Since the recognition task is constrained, recognition performance is typically higher than it is for the text-independent case but places strict constraints on the dialogue.

In text-independent systems there is no constraint on the spoken text as well as the duration of the speech and hence the task can be relatively more challenging. Such systems are typically used in non-pervasive scenarios such as home care systems and identify the speaker without his/her explicit effort, co-operation or dependence on pre-defined utterances. Text-independent speaker recognition systems are also the norm in media indexing and forensics. From the ALIAS project perspective, since the Robot is required to recognize the speaker from any utterance directed towards it, the ALIAS speaker recognition system will be text-independent. This will not, however, prevent text-dependent operation if this is later deemed necessary.

Depending on the task, speaker recognition systems can be classified as either a) speaker verification or b) speaker identification systems. In the same way as described for general biometric systems in Chapter 2, the question asked of a speaker verification system is simply ‘is the person who they say they are?’, i.e. there is a notion of a claimed identity, which the system should verify or otherwise. Such systems are essentially binary classifiers which provide a ‘Yes/No’ response and are widely used in biometric authentication applications.

Speaker identification systems, on the other hand, have to address a slightly more challenging problem of determining the identity of the speaker from an open or closed set of speakers. Such systems are multiclass classifiers which, in an open set scenario, also have to determine whether the input speech is from a non-enrolled person. In the ALIAS project, the system will function in an identification context where the Robot may be required to distinguish between more than a single enrolled user. The speaker set is, however, nonetheless open but with a closed subset including enrolled users, family members and carers. Occasional visitors and speech from other background sources correspond to the open subset.

### 4.2 General Approaches

In the following we outline the various components that are common to almost all speaker recognition systems and show how they are combined together, potentially with speaker

diarization, in order to perform identification.

### 4.2.1 Speech Activity Detection

Speech activity detection (SAD) is an important component in any real-world speech based application. It aims to segment the input audio stream into speech and non-speech segments. This is an important functionality from both a computational and performance point of view since processing the non-speech input is unnecessary and contains no information about the speaker identity. By concentrating only on intervals of the audio signal which contain active speech the resulting speaker models are more discriminative and lead to improved recognition performance.

SAD is primarily based on the energy content in the signal, zero crossing rates and line spectral frequencies [46]. These approaches are usually based on thresholding by analyzing the whole utterance before SAD. On the other hand, approaches such as long term spectral divergence (LTSD) [47] have been successfully used in online detection in real world situations.

### 4.2.2 Feature Extraction

As in all statistical speech pattern recognition tasks one theme of active research relates to the choice of features that give the best discrimination between classes, here speakers. Speech signals are quasi-stationary in nature and hence features are usually extracted from short frames of 20-40 msec in duration. The signal is assumed to be stationary in such short windows (frames) and most state-of-the-art feature extraction techniques are based on short-term spectral estimates over these frames which are preemphasized to boost the higher frequencies.

For each frame, the spectrum is further processed through a filter bank analysis followed by a decorrelation process using cepstral analysis. First and second order derivatives are typically appended to the cepstral parametrization to capture the correlation between adjacent frames. Cepstral based parameters that have found widespread usage are Mel Frequency Cepstral Coefficients (MFCCs) [41] where the frequency scale is warped to Mel-scale to replicate the logarithmic frequency resolution of human ears and Perceptual linear prediction prediction coefficients (PLPs) [31] which are inspired by the principles of human auditory perception. The block diagram representation for these cepstral based feature extraction methods is shown in Figure 4.1.

Unfortunately features always carry certain channel characteristics which manifests as convolutional noise. In order to attenuate these effects, which can otherwise lead to non-negligible degradations in performance, various feature-level normalization approaches have been investigated. These approaches include cepstral mean subtraction (CMS) [27], RASTRA filtering [33], feature warping [44] and feature mapping [48].

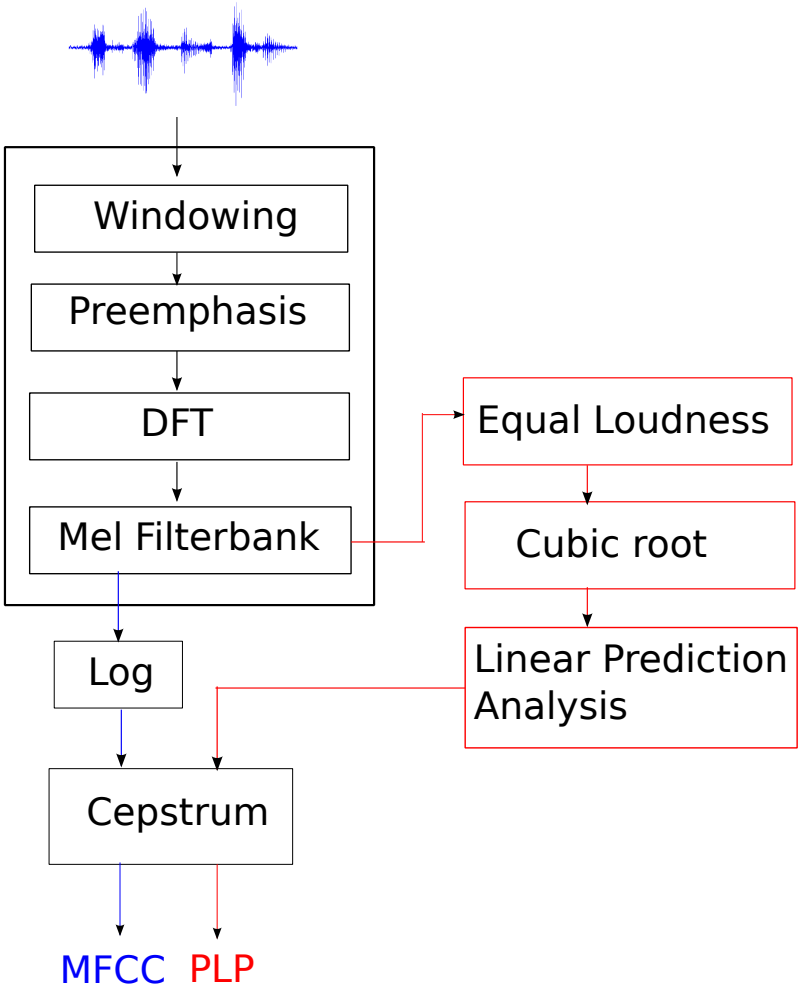



Figure 4.1: Feature extraction

Apart from short term features, features extracted over longer windows or multiple frames have also been widely investigated. Spectro temporal features such as modulation cepstrum [32], voice source and prosodic features [4] such as fundamental frequency, jitter, shimmer, intonation and rhythm and higher level characteristics [22] that capture so-called supra-segmental stylistic qualities fall under this category. The combined use or fusion of both shorter term and longer term features has been investigated for a number of years and has been shown to give better recognition performance under certain conditions [49]. Most of these features have been used successfully with the now-standard, baseline classification approach which is based on Gaussian mixture models (GMMs).

### 4.2.3 Speaker Modeling

Speaker modeling is the core task in speaker recognition. Speech has so much variety that a few samples of speech cannot be used as templates for the speaker and this is especially true in text-independent speaker recognition. Stochastic models are hence widely used to represent speaker characteristics. In this section, the standard Gaussian mixture based models and other new emerging approaches to stochastic speaker modeling are discussed.

#### GMM-UBM models

Many current state-of-the-art approaches to classification have their roots in the standard Gaussian mixture model(GMM) with a universal background model - the so-called GMM-UBM approach [50]. In this framework, each speaker is modeled by a mixture of multivariate Gaussians denoted by $\lambda$:

$$P(x|\lambda) = \sum_{k=1}^{K} w_k \mathcal{N}(x; \mu_k, \Sigma_k), \tag{4.1}$$

where $K$ is the number of Gaussian components, $w_k$ is the apriori weight of the component $k$ and

$$\mathcal{N}(x; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} exp\left\{-1/2(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)\right\} \tag{4.2}$$

is a D-dimensional multivariate Gaussian.

The most common implementation utilizes a UBM which is trained using expectation maximization (EM) [21] and large amounts of data from a pool of background speakers. The Gaussian components are typically initialized used k-means clustering, or any similar approach. Since cepstral features are mostly decorrelated and in order to reduce the computational requirements, diagonal covariance matrices are preferred over full covariance matrices.

Due to the common lack of speaker-specific data, target speaker models are generally adapted from the UBM during enrollment as shown in Figure 4.2. The adapted models are stored as templates and are retrieved during the recognition phase to determine the speaker identity.

Speaker specific models are commonly derived using Maximum a posteriori (MAP) adaptation [28]. Although all the parameters of the UBM can be adapted, adapting only the means of the Gaussian components has been found to work well in practice [50]. Given the enrollment data $X = (x_1, x_2 \ldots, x_T)$ and the UBM, $\lambda_{UBM} = \{w_k, \mu_k, \Sigma_k\}_{k=1}^{K}$, the adapted mean vectors ($\mu_k^{'}$) in the MAP sense are a weighted average of the maximum

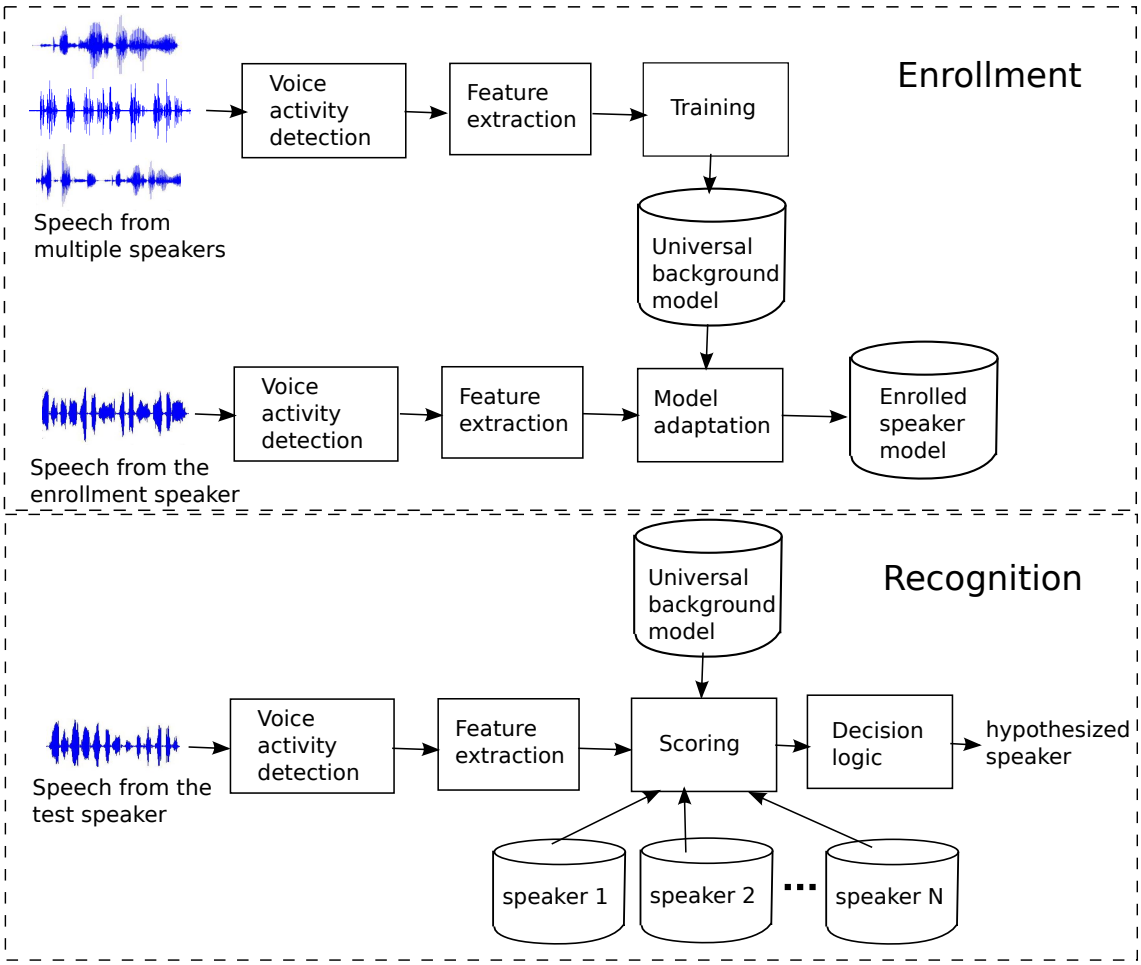



Figure 4.2: Speaker identification

likelihood (ML) estimate from the training data and the parameters of the UBM according to:

$$\mu_k^{'} = \alpha_k \tilde{x}_k + (1 - \alpha_k)\mu_k \tag{4.3}$$

where,

$$\alpha_k = \frac{n_k}{n_k + r} \tag{4.4}$$

$$\tilde{x}_k = \frac{1}{n_k} \sum_{t=1}^{T} P(k|x_t) x_t \tag{4.5}$$

$$n_k = \sum_{t=1}^{T} P(k|x_t) \tag{4.6}$$

$$P(k|x_t) = \frac{w_k \mathcal{N}(x_t; \mu_k, \Sigma_k)}{\sum_{m=1}^{K} w_m \mathcal{N}(x_t; \mu_m, \Sigma_m)} \tag{4.7}$$

Here $r$ is the relevance parameter and $n_k$ is the occupation count of component $k$ for the training data. The parameter $\alpha_k$ controls the contribution of the original model parameters and the adaptation data in the adapted model parameters and can be set heuristically by modifying the parameter $r$.

When the amount of adaptation data is limited [40], MAP estimation suffers from the disadvantage of the stronger contribution from non-discriminative background models compared to the target speaker data. To overcome this problem, more recently Maximum likelihood linear regression (MLLR) [39] adaptation has been investigated to model the speakers [55, 35, 60] for speaker recognition task. It is shown to give better results with small enrollment data but with the availability of larger training set, MAP adaptation outperforms the maximum likelihood estimates.

During recognition, scores correspond to log-likelihood ratio of the target model on the test set normalized with respect to the background models.

$$LLR(s|L) = log\left(\frac{P(s|L)}{P(s|W)}\right) \tag{4.8}$$

where, $s$ is the test segment, $L$ and $W$ are the target model and the universal model respectively.

Other normalization approaches such as test normalization (Tnorm) [5] normalize the score with respect to the scores from the acoustically close cohort speakers

$$L_{TNORM}(s|L) = \frac{log\left(P(s|L)\right) - \mu_I}{\sigma_I} \tag{4.9}$$

where, $\mu_I$ and $\sigma_I$ are the mean and variance of the scores evaluated on impostor speaker models w.r.t the test segment $s$.

Additional normalization strategies that are commonly used operating at the score level include zero normalization (Z-norm) and handset normalization (H-norm) [23].

Decision logic is usually based on a threshold which is determined using a large development set during the training phase. Recognition accuracies in terms of False Accepts and False Rejects can be traded-off by varying this threshold.

**Other advanced models**

The state-of-the-art has advanced significantly since the early days of GMM-based approaches. Support vector machines (SVMs) [20] have become a popular approach to pattern classification and speaker verification is no exception. Early attempts to use SVMs for speaker verification appeared in the mid-to-late 90's e.g. [53, 15]. These early approaches used cepstral based parametrization and led to results that were inferior to a standard GMM. More recent SVM-based approaches such as the generalized linear discriminant sequence kernel (GLDS) [16] and the GMM supervector linear kernel (GSL) [17] approaches are capable of outperforming the standard generative GMM-based approach [50]. The GSL approach is one example where the input to the SVM classifier comes from a conventional GMM and is here the concatenation of the GMM mean vectors [35] better known as the GMM supervector.

Despite harnessing the discriminative power of the SVM the above approaches do not explicitly model inter-session variability which the next generation of speaker verification system sought to achieve. There have been two main approaches, namely nuisance attribute projection (NAP) [18] and joint factor analysis (JFA) [36]. The NAP approach aims to attenuate session effects in a discriminative SVM framework. JFA has received a huge amount of attention and there are numerous implementations reported in the literature, e.g. [25, 14]. In contrast to feature mapping [166] the JFA approach assumes that the channel variability space is continuous instead of discrete and combines a model of both speaker and session variability. Joint factor analysis approaches have proved to be among the best performing approaches to date.

Finally, any state-of-the-art review would not be complete without referring to the many attempts to bring additional improvements in performance through the fusion of different systems and scoring approaches, some notable examples including [14, 37] but the above text necessarily focuses only on the underlying, core modeling and classification technologies which have led to the greatest contributions to the field of text-independent speaker recognition over the last decade. An excellent comparison of these approaches using common parametrizations and datasets is presented in [38, 7].

### 4.2.4 Speaker Diarization

Speaker diarization is inextricably linked to speaker diarization which aims to facilitate application in multi speaker contexts by identifying intervals during which different speakers are active. This entails segmenting an audio stream into homogeneous segments

based on speaker turns (segmentation) and in grouping together all the segments from the same speaker (clustering). Speaker diarization systems are often used in conjunction with both speech and speaker recognition systems when there are multiple competing input sources for eg., in home care scenarios, meeting rooms and public spaces. The modeling procedure is almost exactly the GMM-UBM approach described in Section 4.2.3 and thus models produced through speaker diarization can be used directly for speaker recognition.

The problem is usually unsupervised, i.e. no a priori knowledge is available. This leads to a trial-and-error search for an optimal speaker inventory and the two dominant approaches to speaker diarization: bottom-up and top-down hierarchical clustering [24]. The bottom-up approach based on hierarchical agglomerative clustering is by far the most popular and systems based on this approach have consistently achieved the good levels of performance in the NIST RT evaluations [42], e.g. [62]. On the other hand top down, divisive hierarchical clustering approaches are initialized with a single speaker model which is repeatedly divided into new models until the desired number of speakers is achieved [12, 13, 11].

While some recent online or near-realtime approaches have been proposed [29] but most speaker diarization algorithms are normally offline in their operation. Both online algorithms, similar to that described in [29] and inspired from the top-down approach in [12] will be used in the ALIAS project.

## *4.3 ALIAS Speaker Recognition System*

In this section, we first give a brief overview of the speech related components to be deployed on the ALIAS Robot, followed by specific description of the ALIAS speaker recognition system.

### 4.3.1 Speech components on the ALIAS Robot

Figure 4.3 shows various speech related components that will be deployed on the ALIAS Robot. All the components will be interfaced to and will be controlled by the dialogue manager

- **Speech Enhancement**: This module will take as input the noisy speech signal and removes the ambient background noise. Input from the two microphones on the ALIAS Robot could possibly be used to produce a high fidelity signal.
- **Speech Activity Detection**: The role of this module is to partition the input audio stream into speech and non-speech segments. On detection of speech, this module will trigger the dialogue manager.
- **Speaker Diarization**: When there are multiple competing sources of speech, this module will partition the speech segment based on speaker turns, and cluster the

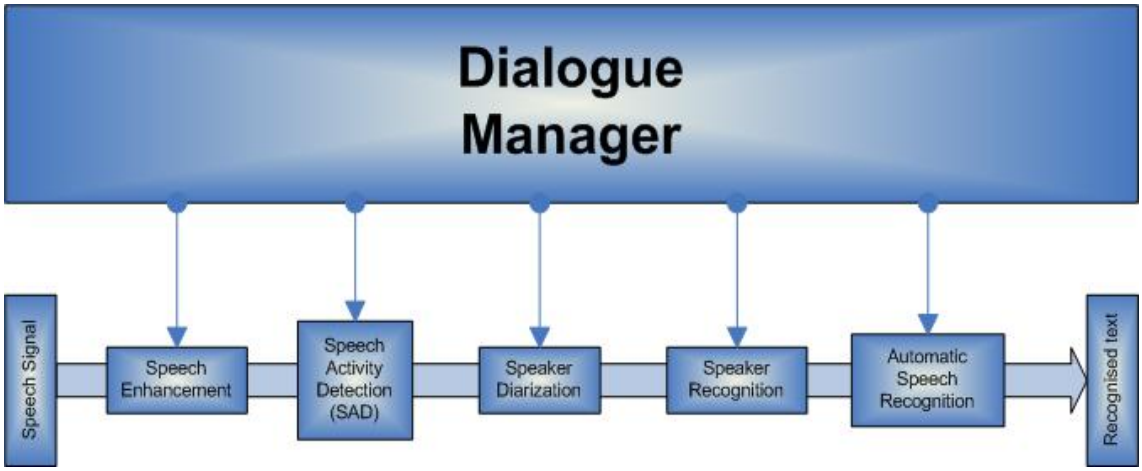



Figure 4.3: Speech based modules in ALIAS system

speech chunks associated with each user separately. Current publicly available implementations of speaker diarization systems process an audio segment offline. For the ALIAS project an online realtime system is being implemented in collaboration between Eurecom and TUM.

- **Speaker Recognition**: This module when triggered by the dialogue manager identifies the speaker of the requested speech segment and provides the dialogue manager with the speaker identity.
- **Automatic Speech Recognition**: It plays the role of transcribing the input speech. This module will have speaker independent acoustic models, language models and dictionary resources required to perform the task. The speaker identity information available from the speaker recognition module could be used to improve the recognition accuracy by using switching to speaker adapted/dependent acoustic models.

All these modules will be independent and will be called upon by the dialogue manager based on the requirements and context. Certain components in the signal path can be bypassed and the speech signal redirected to the appropriate module by the dialogue manager, i.e. where speech recognition is required, but speaker recognition is not.

### 4.3.2 Speaker Recognition System

The speaker recognition system to be used in the ALIAS project will be primarily based on the LIA speaker detection system [9] which was developed using the open source speaker recognition toolkit - ALIZE [1, 10]. ALIZE is now part of the MISTRAL project [19] which is an open source tool for biometric recognition in general. ALIZE-MISTRAL tool-kits provide robust C++ implementations of the state-of-the-art techniques in speaker

recognition and have been used to build several systems used in recent internationally competitive NIST speaker recognition evaluations. The ALIAS system may also use some libraries and tools from the Hidden Markov Model Toolkit (HTK) [64] and its application program interface namely ATK [2] that are extensively used in speech recognition research.

The ALIAS speaker recognition system will be completely modular with clearly defined interfaces. It will be based on the standard GMM-UBM principles as described in Section 4.2.3. The components of the system are as follows:

- **Front end:** This module will include pre-processing, feature extraction and feature normalization steps. In the pre-processing step, the speech signal will be normalized to remove any DC offset and will be pre-emphasized to boost the energy in higher frequencies. Feature extraction will typically involve short term cepstral features such as MFCCs and PLPs with their derivative coefficients appended. Other features based on spectro-temporal characteristics of the speech signal might be investigated and integrated into this module. The final submodule will involve a feature normalization component that will include cepstral mean subtraction and Gaussianization of the features.

- **Universal background model:** The UBMs will be trained as GMMs with diagonal covariance matrices with speech data from several speakers. In the initialization step, the seed models will be set to 16 components which will be initialized by k-means clustering from the available data. These models will be retrained using an expectation-maximization algorithm and, after every few iterations, the number of mixture components will be increased. The number of mixture components will be empirically optimized.

- **Speaker modeling:** The UBMs will be adapted to each target speaker using maximum aposteriori estimation. Only the Gaussian means will be adapted. Maximum likelihood based approaches such as MLLR may also be investigated to compare the accuracies in limited data conditions.

- **Recognition module:** Speaker recognition will be based on log-likelihood ratio tests which is a ratio of the likelihood score of the target model and the UBM model for a test speech segment. These scores will be further normalized using ZNorm and TNorm before arriving at the final decision about the identity of the speaker.

Other recently proposed advances in speaker recognition such as Joint Factor Analysis and Nuisance Attribute Projection can be computationally expensive and may have implications with respect to memory and CPU usage when deployed on the ALIAS Robot in a real time application scenario. Hence the speaker recognition system will use the standard approach as described above and will use more advanced features where necessary and depending upon initial field tests.

### Development phase

During the development of the ALIAS speaker recognition system, it will be extensively tested using NIST SRE (National Institute of Standards and Technology Speaker Recognition Evaluation) corpora in an offline mode. The NIST SRE datasets and experimental protocols are the defacto platform to evaluate the state-of-the-art algorithms and systems in speaker recognition research. It is a biennial worldwide evaluation [43] and a corpus is released in each evaluation. For developing the ALIAS system, appropriate NIST datasets will be used. These corpora have speech from a large number of speakers and are partitioned into training and test sets to suit speaker recognition experiments.

The system will be tested with varying amounts of enrollment and test data and under matched and mismatched conditions in enrollment and test data to baseline the results against the state-of-the-art systems. Tests under such varying conditions would indicate the accuracy tradeoffs with the amount of available enrollment and test data and thus lead to the development of data guidelines for user enrollment with the ALIAS Robot.

### Deployment phase

For the deployment on the ALIAS Robot, additional C++ wrappers will be written for the speaker recognition system to interface with the dialogue manager in consultation with the ALIAS consortium partners. The universal background models generated from NIST corpus will be used in the deployed system as the training of UBMs needs speech from several speakers which cannot be realistically collected using the Robot microphones during the project lifespan. Based on the detection error tradeoff curves obtained during NIST-style development evaluations an optimal amount of required enrollment data will be determined beforehand and each user will be asked to record a set of appropriate utterances. Recordings will be made from different directions and distances relative to the ALIAS Robot. Typical background noises originating in home care setting such as ambient noise, sound from microwave oven, fans etc. will also be recorded to build a noise model. An online speaker diarization system will also be developed in collaboration with TUM and will be field tested in conjunction with the speaker recognition system.

### Open issues

The mobile ALIAS Robot platform has only two inbuilt microphones towards the front. As a result the fidelity of the speech signal captured from behind the ALIAS Robot as well as from various directions is not well understood. There would be some research effort to understand the signal quality and the impact on speaker recognition accuracies arising from these issues.

An interface needs to be defined and developed for integration with the dialogue manager.

A speech activity detector module will run continuously in the background to detect and alert the dialogue manager to the presence of any speech activity. The speaker recognition module, possibly in conjunction with the speaker diarization module, will then be triggered by the dialogue system to identify the user on an as needed basis. The communication interfaces between the dialogue manager and the various speech based modules will need discussion and agreement between project partners.

More speech data will become available with the use of the system by the target users. This data could potentially be used to further adapt the speaker models. A strategy for online adaptation may be considered if it is deemed necessary. Speaker model adaptation should ideally happen when the ALIAS Robot is in a dormant mode. There are some practical open issues such as the memory constraints on the ALIAS Robot, unsupervised selection of the speech based on the signal quality and the frequency of model adaptation. All of these issues will be addressed once the first demonstrator is complete.

## Bibliography

[1] Alize: Open source tool for speaker recognition `http://mistral.univ-avignon.fr/en/index.html`.

[2] The ATK Real-Time API for HTK `http://mi.eng.cam.ac.uk/research/dialogue/atk_home.html`.

[3] T. Acharya and A. K. Ray. *Image Processing - Principles and Applications*. Wiley-Interscience, 2005.

[4] A. Adami, R. Mihaescu, D. Reynolds, and J. Godfrey. Modeling prosodic dynamics for speaker recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 788–791, 2003.

[5] R. Auckenthaler and J. S. Mason. Gaussian selection applied to text-independent speaker verification. In *In Proc. Speaker Odyssey 2001*, pages 83–88, 2001.

[6] J. A. Bilmes. A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical report, 1997.

[7] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. A. Reynolds. A tutorial on text-independent speaker verification. *EURASIP J. Appl. Signal Process.*, 2004:430–451, January 2004.

[8] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, October 2007.

[9] J.-F. Bonastre, N. Scheffer, C. Fredouille, and D. Matrouf. NIST 04 speaker recognition evaluation campaign: New LIA speaker detection platform based on ALIZE toolkit. In *NIST SRE 04 Workshop: Speaker Detection Evaluation Campaign*, 2004.

[10] J.-F. Bonastre, N. Scheffer, D. Matrouf, C. Fredouille, A. Larcher, A. Preti, G. Pouchoulin, N. Evans, B. Fauve, and J. S. Mason. ALIZE/SpkDet: a state-of-the-art open source software for speaker recognition. In *Speaker and Language Recognition Workshop (IEEE Odyssey)*, 2008.

[11] S. Bozonnet, N. Evans, X. Anguera, O. Vinyals, G. Friedland, and C. Fredouille. System output combination for improved speaker diarization. In *Interspeech*, 2010.

[12] S. Bozonnet, N. Evans, and C. Fredouille. The LIA-Eurecom RT09 speaker diarization system: Enhancements in speaker modelling and cluster purification. In *ICASSP*, pages 4958–4961, 2010.

[13] S. Bozonnet, N. Evans, C. Fredouille, D. Wang, and R. Troncy. An integrated top-down/bottom-up approach to speaker diarization. In *Interspeech*, 2010.

[14] N. Brümmer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim. Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2072–2084, sept 2007.

[15] W. Campbell and K. Assaleh. Polynomial classifier techniques for speaker verification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 321–324, mar 1999.

[16] W. Campbell, J. Campbell, D. Reynolds, E. Singer, and P. Torres-Carrasquillo. Support vector machines for speaker and language recognition. *Computer Speech & Language*, 20(2-3):210–229, 2006.

[17] W. Campbell, D. Sturim, and D. Reynolds. Support vector machines using GMM supervectors for speaker verification. *Signal Processing Letters, IEEE*, 13(5):308–311, May 2006.

[18] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff. SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, page I, may 2006.

[19] E. Charton, A. Larcher, C. Levy, and J.-F. Bonastre. Mistral: open source biometric platform. In *ACM Symposium On Applied Computing*, 2010.

[20] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995. 10.1007/BF00994018.

[21] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical society*, 39:1–38, 1977.

[22] G. Doddington. Speaker recognition based on idiolectal differences between speakers. In *Seventh European conference on Speech Communication and Technology (Eurospeech)*, pages 2521–2524, 2001.

[23] R. Dunn, T. Quatieri, D. Reynolds, and J. Campbell. Speaker recognition from coded speech and the effects of score normalization. In *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on*, volume 2, pages 1562 –1567 vol.2, 2001.

[24] N. Evans, S. Bozonnet, D. Wang, C. Fredouille, and R. Troncy. A comparative study of bottom-up and top-down approaches to speaker diarization. (Research report RR-10-243). Technical report, EURECOM, 2010.

[25] B. Fauve, D. Matrouf, N. Scheffer, J.-F. Bonastre, and J. Mason. State-of-the-art performance in text-independent speaker verification through open-source software. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(7):1960 –1968, sept 2007.

[26] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, London, UK, 1995. Springer-Verlag.

[27] S. Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(2):254–272, Apr. 1981.

[28] J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, 1994.

[29] J. Geiger, F. Wallhoff, and G. Rigoll. GMM-UBM based open-set online speaker diarization. In *Interspeech*, pages 2330–2333, 2010.

[30] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69:331–371, 1910. 10.1007/BF01456326.

[31] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

[32] H. Hermansky. Should recognizers have ears? *Speech Communication*, 25(1-3):3 – 27, 1998.

[33] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, Oct. 1994.

[34] B. Jähne. *Digitale Bildverarbeitung*. Springer, 5., überarb. u. erw. aufl. edition, June 2001.

[35] Z. Karam and W. Campbell. A new kernel for SVM MLLR based speaker recognition. In *Interspeech*, pages 290–293, 2007.

[36] P. Kenny. Joint factor analysis of speaker and session variability : Theory and algorithms crim-06/08-13. Technical report, CRIM, 2006.

[37] T. Kinnunen and P. Alku. On separating glottal source and vocal tract information in telephony speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4545–4548, 2009.

[38] T. Kinnunen and H. Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52(1):12–40, 2010.

[39] C. Leggetter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9(2):171–185, 1995.

[40] J. S. Mason, N. W. D. Evans, R. Stapert, and R. Auckenthaler. Data-model relationship in text-independent speaker recognition. *EURASIP Journal on Applied Signal Processing*, 2005(4):471–481, 2005.

[41] P. Mermelstein. Distance measures for speech recognition, psychological and instrumental. In C. H. Chen, editor, *Pattern recognition and artificial intelligence*, pages 374–388. Academic Press, New York, 1976.

[42] NIST. The NIST rich transcription 2009 evaluation `http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-eval-plan-v2.pdf`, 2009.

[43] NIST. 2010 NIST speaker recognition evaluation `http://www.itl.nist.gov/iad/mig/tests/sre/2010/NIST_SRE10_evalplan.r6.pdf`, 2010.

[44] J. Pelecanos and S. Sridharan. Feature warping for robust speaker verification. In *Speaker Odyssey, The Speaker Recognition Workshop*, pages 213?–218, 2001.

[45] N. Poh and S. Bengio. Why do multi-stream, multi-band and multi-modal approaches work on biometric user authentication tasks? In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 893–896, May 2004.

[46] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall Signal Processing Series, 1993.

[47] J. Ramírez, J. C. Segura, C. Benítez, A. de la Torre, and A. Rubio. Efficient voice activity detection algorithms using long-term speech information. *Speech Communication*, 42(3-4):271 – 287, 2004.

[48] D. Reynolds. Channel robust speaker verification via feature mapping. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 53–56, april 2003.

[49] D. Reynolds, W. Andrews, J. Campbell, J. Navrátil, B. Peskin, A. Adami, Q. Jin, D. Klusáček, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang. SuperSID project final report. Technical report, Johns Hopkins University, 2002.

[50] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

[51] H. A. Rowley, S. Baluja, and T. Kanade. Neural Network-Based Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[52] J. C. Russ. *Image Processing Handbook, Fourth Edition*. CRC Press, Inc., Boca Raton, FL, USA, 4th edition, 2002.

[53] M. Schmidt and H. Gish. Speaker identification via support vector classifiers. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 105–108, 1996.

[54] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am. A*, 4(3):519–524, March 1987.

[55] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A. Venkataraman. MLLR transforms as features in speaker recognition. In *Interspeech*, pages 2425–2428, 2005.

[56] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3:71–86, January 1991.

[57] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591. IEEE Comput. Sco. Press, 1991.

[58] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–511–I–518, Los Alamitos, CA, USA, April 2001. IEEE Comput. Soc.

[59] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57:137–154, May 2004.

[60] R. C. Vipperla, S. Renals, and J. Frankel. Augmentation of adaptation data. In *Interspeech*, pages 530–533, 2010.

[61] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260 – 269, apr. 1967.

[62] C. Wooters and M. Huijbregts. Multimodal technologies for perception of humans. chapter The ICSI RT07s Speaker Diarization System, pages 509–519. Springer-Verlag, Berlin, Heidelberg, 2008.

[63] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, Jan 2002.

[64] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book (for Hidden Markov Model Toolkit Version 3.4)*, 2006.

[65] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35:399–458, December 2003.