Online Speaker Recognition for Teleconferencing Systems

Christoph Dominik Kozielski, Martin Rothbucher, Klaus Diepold

Technical Report

Online Speaker Recognition for Teleconferencing Systems

Christoph Dominik Kozielski, Martin Rothbucher, Klaus Diepold

April 14, 2014

Lehrstuhl für Datenverarbeitung
Technische Universität München

Submitted on April 14, 2014
by Christoph Dominik Kozielski, Martin Rothbucher, Klaus Diepold

Supervised by Prof. Dr.-Ing. K. Diepold and Dipl.-Berufspäd. Martin Rothbucher

Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

Abstract

This thesis describes an online speaker recognition as part of an immersive system for teleconferencing, developed at the *Institute for Data Processing*. With this conference system, it is possible to create an individual audiostream for every speaker in a conference room by using only a device on the conference table. The current active speaker in a teleconference is identified online.

Therefore, a model of each conference participant is constructed by using short time spectral features. In the system approach Gaussian Mixture Models (GMMs) are applied for modeling individual speakers. Features are extracted from the incoming audio stream, constituting a likelihood score for every model and thus identifying the present speaker. Mel-frequency cepstral coefficients (MFCCs) are chosen as features to represent spectral attributes of different sounding voices. Due to the required low computational complexity of an online speaker recognition task, a Universal Background Model (UBM) in combination with maximum aposteriori (MAP) adaptation is used to provide a very fast creation of speaker-dependent GMMs, with only few training data needed.

Experiments are carried out on the AMI meeting corpus the active speaker is correctly determined with a recognition accuracy of 78 %.

Diese Diplomarbeit befasst sich mit der Online Sprechererkennung als Teil eines immersiven Telekonferenzsystems, dass am *Lehrstuhl für Datenverarbeitung* entwickelt wird. Mit diesem System ist es möglich einen separaten Kanal für jeden Sprecher im Konferenzraum nur durch die Nutzung einer Konferenzspinne zu erzeugen. Der momentan aktive Sprecher einer Telekonferenz wird online erkannt.

Hierzu wird für jeden Konferenzteilnehmer ein Modell aus Merkmalen des Kurzzeitspektrums erstellt. Gaussian Mixture Models (GMMs) bilden die Grundlage dieser Modelle. Es werden Merkmale aus dem eintreffenden Sprachsignal extrahiert, die Ähnlichkeit zu den bestehenden Modellen berechnet und damit wird der aktive Sprecher identifiziert. Mel-frequency cepstral coefficients (MFCCs) sind besonders gute Merkmale, um spektrale Besonderheiten verschieden klingender Stimmen abzubilden. Da der Rechenaufwand aufgrund der Onlinefähigkeit des Systems gering zu halten ist, wird ein Universal Background Model (UBM) in Kombination mit der maximum aposteriori (MAP) Adaption verwendet, um eine sehr schnelle Erzeugung sprecherabhängiger GMMs bei geringem Bedarf an Trainingsdaten zu garantieren.

Experimente mit dem AMI meeting corpus zeigen, dass so die aktiven Sprecher mit einer Erkennungsrate von 78 % erkannt werden können.

Contents

1. Introduction

Future telecommunication systems will provide an immersive experience for users. Immersion in the context of teleconferencing systems describes the presence within a virtual surrounding where every user is spatially placed in relation to his communication partners. The perception of audio and video for every user is strongly related to the relative position between him and the other users in a conference.

Four attributes define immersive communication [18]:

- full-duplex exchange;
- freedom of movement without body-worn or tethered microphones (hands free);
- high-quality speech signals captured from a distance;
- spatial realism of sound rendering.

These attributes inherit a lot of challenges for developing an immersive system: sound acquisition and processing, multi-party immersive audio mixing and management, and sound rendering for untethered immersive perception [19].

This chapter gives a short introduction about new generation teleconferencing systems, immersive audio perception and the use of speaker recognition in a teleconferencing system environment. The first section describes the contents and the motivation of this thesis. Section 1.2 defines the thesis objectives. Section 1.3 gives a short outline of this thesis.

1.1. Motivation of this thesis

Systems for operating teleconferences gain importance and popularity especially for enterprises, companies and private users [13]. Teleconferences save travel expenses and time and enable efficient teamwork by bridging long distances.

The fast growing market of conferencing solutions faces a noticeable lack of innovation of the products offered. The disadvantages of today's teleconferencing solutions, like poor quality of speech in unfavourable environment conditions, the absence of visual information or the missing overview of the participants of a teleconference and "who speaks now" are not satisfactory compensated by modern conferencing solutions.

Innovation in conferencing systems is limited to improving already existing techniques instead of developing new methods based on state-of-the-art research. It would be desirable to utilize the advantages of current research experiences for immersive audio communication, as a useful system solution for teleconferencing. This would not only lead to a remarkable improvement of speech intelligibility and compensate today's system disadvantages but also to a more effective, efficient and lifelike meeting communication.

A traditional telephone conversation works fine for two colleagues in a company quickly exchanging information but has a lot of drawbacks when used for multi-party conferencing. The human audiovisual sensory perception gives great examples on how to improve future communication systems. The most prominent idea for improving telephone audio is the so called cocktail party effect [7]. Humans are able to understand one particular speaker even if there are simultaneously active speakers around them. Spatial hearing plays a big role: By concentrating on one spatial direction, the listener is able to fully understand the corresponding speaker in a situation of multiple active speakers. Teleconferencing systems would benefit of such capabilities.

To increase comprehensibility of speech sources, audio signals of different speakers in a conference room have to be assigned to individual audio channels in order to enable immersive 3D sound synthesis at the remote site. Every participant is assigned to his own channel, so all users are transmitted in separate streams. Together with a specific spatial position for every speaker, spatial rendering of the conference situation is possible [32]. In order to assign different speakers in a conference room to individual audio channels it would be helpful to have a speaker recognition analyzing the incoming speech. Then, the speech signal is assigned to a channel linked to a certain speaker and spatially rendered, so that the listener at the remote site is able to distinguish between speakers by the direction they are speaking from. The main constraint of a speaker recognition system to be used for teleconferencing is its capability of working online. In an active conference, all recorded speech signals directly need to be analyzed. This shows the necessity of a system with low computational complexity and a minimum need of memory and computing time.

Besides profiting from the cocktail party effect, speaker recognition also improves the performance of an automatic speech recognition [15]. This speech recognition can be used to create an automatic protocol of a conference.

Altogether, a new generation teleconference system benefits from an automatic online speaker recognition.

1.2. Definition of the thesis objectives

The aim of this thesis is the development of an online speaker recognizer for the usage in an immersive teleconference system. The recognition rate is evaluated in experiments and compared to other existing speaker recognition approaches. In addition, the influence of preprocessing like spatially localizing and separating audio streams of conference participants is evaluated and it is tested whether speaker recognition benefits the result of automatic speech recognition for the creation of written protocols for conferences.

This thesis covers an online speaker recognition approach and its implementation in a teleconference system. Therefore, speakers are represented using statistical models. For this purpose, features will be extracted from a speech signal, representing speaker-specific details in a parametrical form. The statistical models are built upon these features. Pattern matching of features extracted from a recording with unknown speakers to the existing models lead to a speaker decision. The focus lies on an online version of the recognizer, presuming there is no prior knowledge about the incoming speech signal and limited computational power. This also inherits permanent adaptation of the existing models, improving the whole system functionality while running.

The performance of the online speaker recognizer is compared to other approaches in literature, using the AMI corpus of meeting recordings [1] to achieve comparable results. Also the performance as module of the proposed immersive teleconference system is measured. Finally, automatic speech recognition results are produced, showing the influence of prior speaker recognition. Altogether, it is evaluated whether the implemented online recognizer is capable of producing robust results and reliably improving the functionality of a teleconference system.

1.3. Outline of the thesis

This thesis comprises six chapters. Chapter 2 presents an example architecture of a new generation teleconferencing system as researched on at the *Institute for Data Processing*. Chapter 3 gives insight into the topic of speaker recognition. The fundamentals of speaker recognition are defined and the different approaches to the procedure of speaker recognition are explained. Chapter 4 describes the system implemented for this thesis in detail. In chapter 5 the experiment methods are defined and the results are arranged and pictured. The last chapter gives a short résumé and conclusion of the methods chosen and the experiments made. The thesis ends with an outlook and possible future work.

2. Teleconferencing System

This chapter introduces an immersive teleconferencing system, as developed at the *Institute for Data Processing*. The role and position of the speaker recognition within the system architecture, and how it is used to separate speaker channels, is shown.

The first section in this chapter gives a system architecture overview. Section 2.2 explains the method of recording and the hardware device designed for the system. Section 2.3 briefly describes the processing needed for speaker localization, the separation of different speakers and the elimination of noise and disturbances. The following section 2.4 shortly denotes the role of the speaker recognition module in the system environment and the defined interfaces. Section 2.5 gives a brief overview of the speech recognition module and its use for teleconferences. It is followed by section 2.6 which deals with the sound mixing and transmission of the speech signals and the network architecture for teleconferences. Finally, the last section gives a short introduction on how the processed and transmitted signal and all the information is synthesized and reproduced for the receiver.

2.1. System overview

The teleconference system developed at the Institute for Data Processing is capable of assigning different speakers to individual audio channels depending only on audio recordings of a microphone array [31] and spatially render them to create a 3D sound impression for the receiver.

The system considered in this thesis has a modular design. Its architecture can be simplified to a classic communication system, composed of a sender, transmitter and receiver. The sender records a speech signal, the transmitter transfers it to the receiver and the receiver reproduces the signal. In a teleconference scenario, the sender is a conference telephone with the correspondent hardware to record a speech signal. The transmitter is a server structure with the functionality of receiving and sending audio signals. The receiver is a telephone or Voice-over-IP client with the respective hardware to reproduce an audio signal. To evolve this system into a new generation immersive communication system, every block needs to be enhanced and extended. However, the architecture keeps its tripartite and serially ordered design, so it can be easily extended and is compatible to current communication systems.

Every block (sender, transmitter, receiver) contains several moduls. Every modul adds a certain functionality to the system. The whole system is not only able to communicate with

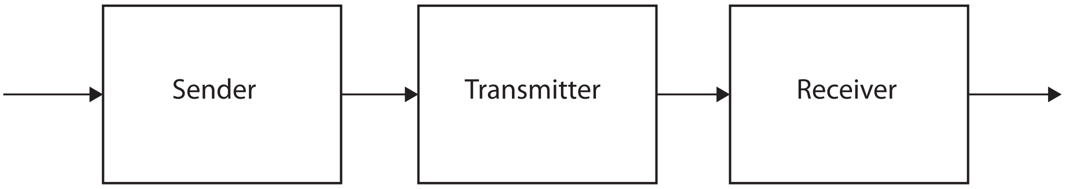

Figure 2.1.: Classic tripartite communication system

other external systems, but it can easily be extended by additional modules. Every modul has defined interfaces and can be tested and developed independently. In the described scenario, the conference telephone will be extended by speaker recognition, localization and separation features to assign different speakers in a conference room to individual channels. The server will have the ability to recognize speech, store it centrally and process all the incoming channels together with localization data for 3-D sound rendering. If compatible, the receiver will be extended by a client software to easily define the spatial position in the virtual environment.

Altogether, the different modules, as described in the following sections, turn a classic teleconferencing system into an immersive audio conferencing system.

2.2. Speaker recording

A big issue for conference telephones placed on a table in a room with multiple conference participants is the distant acquisition of the speech signal [2]. It is very vulnerable to interference from concurrent sound sources and noise distortion through reflection. The audio recording module needs to be able to handle all types of noise occurring. To control noise, reverberation, and competing speech, multiple microphones are generally more powerful than a single microphone [5].

The recording device in the proposed system is a circular microphone array with an omnidirectional directivity pattern. This is important in order to uniformly record speakers from every direction and to increase localization performance [5, 23]. Figure 2.2 shows the technical drawing of the microphone array used for the teleconference system.

2.3. Speaker localization and separation

As described in the previous section, the recording is done by a specifically designed recording device. Then, to ensure high quality speech acquisition even in noisy and echoic rooms, our system approach utilizes a Steered Response Power - Phase Transform (SRP-PHAT) sound localization with particle filtering in combination with Geometric Source Separation (GSS). This way, echoes and noise can be minimized in the recordings and the speech quality of the audio signals that are passed to the speaker recognition system is

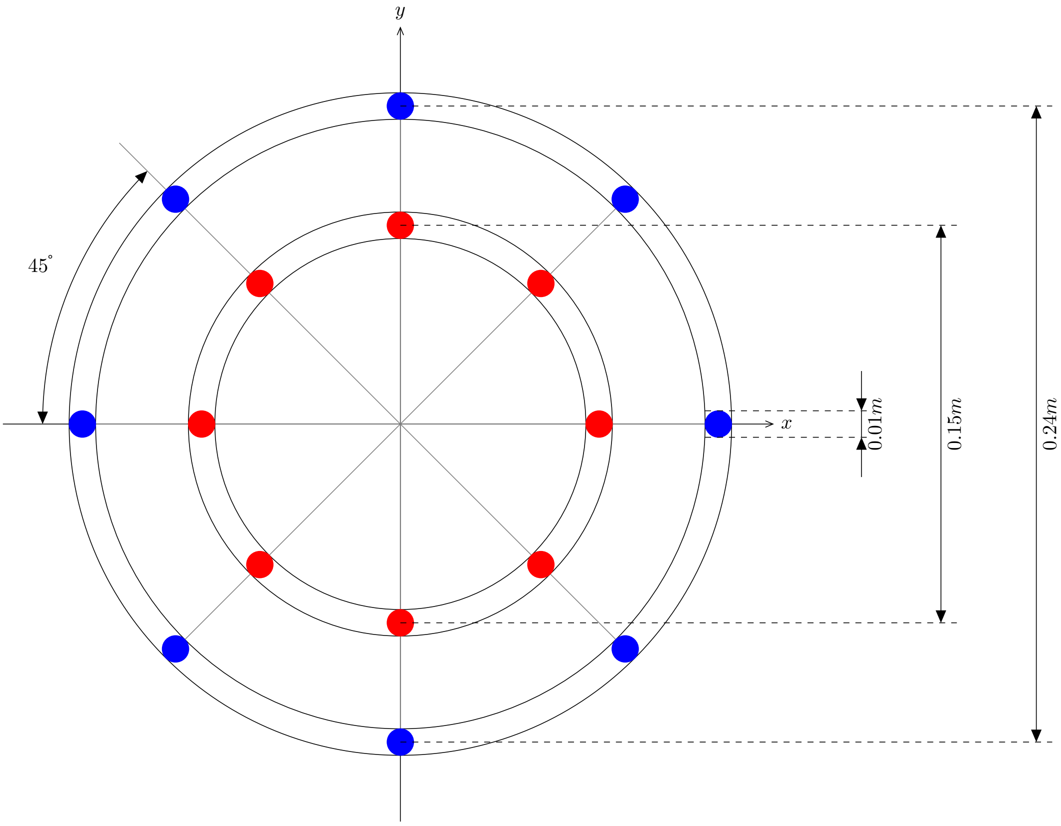

Figure 2.2.: Technical drawing of the circular microphone array used in the teleconference system

increased. Moreover, in case of simultaneously active conference participants, the sound mixture is separated [31].

This module is an important part of the immersive teleconferencing system in order to control noise, reverberation, and competing speech. The output of this module is a number of audiostreams, depending on the number of located sound sources. Regulated by the localization, the streams comprise of separated speaker streams but not in a predefined order. Both tasks, localization and separation, are explained in detail within the Diploma thesis of *Dipl. Ing. (Univ.) Johannes Feldmaier* [11].

2.4. Speaker recognition

The speaker recognition is part of the sender block, to enable a separate transmission of every speaker on its own channel, although only one device in the conference room is used for recording. The input of the speaker recognition module ideally is a single channel speech recording or stream. This audio input, with its preprocessing, separation and filtering, is as much as possible free of noise, interference, echo and reverberation.

The result of the speaker recognition is the classification of a speech signal to a defined speaker name and the assignment to a certain output channel corresponding to the speaker.

This thesis focusses on the speaker recognition module.

2.5. Speech recognition

The speech recognition is part of the transmission block, so mainly a module of the server handling the teleconference.

The speech recognition module automatically creates a written protocol of the conference and stores it centrally to be available for all participants of a meeting. The input are the separated channels, as allocated by the speaker recognition, and the associated speaker names. It creates a document with the words spoken and the speaker name attached to it, similar to a chat log.

The automatic creation of a protocol tremendously benefits of the assignment of each voice channel to an individual speaker by the speaker recognition. By that, each voice stream can be recognized independently and with a minimum of interference or disturbances by other speakers. The quality of this transcript is not only influenced by the preceding separation and speaker recognition but also strongly depends on the used speech recognition module. Nowadays, there are commercial solutions available on the market basing on huge training databases. With these professional speech recognizers it is possible to achieve fair recognition rates.

It takes a high development effort generating such large training sets. Therefore, the speech recognition module in the proposed system is an external system.

2.6. Speech transmission

Signals recorded at each conference site are sent to the transmission module. This includes a mixer that processes all inbound signals and presents a mixed signal to each conference room. The transmissions are handled by a central server. A server-client architecture is the best choice for an immersive teleconference system [19].

Different mixtures for the receiving devices are handled centrally in this module. It includes a mixer that processes all incoming signals and creates appropriate streams according to device capabilities so that the conference is presented in an optimal manner. That means, for single channel devices a mono signal is created and for multichannel room solutions a spatially distributed signal is generated. For this purpose, the central transmission component needs efficient sound rendering and management techniques.

Additionally, after the mixing process, the transmission component picks an appropriate compression codec according to the receiving device, so that the signals are transmitted with a minimum delay and the highest possible quality while acquiring a minimal bandwidth.

A further aspect of this component is to ensure the downward compatibility between different devices. For example, it should be possible to connect mobile phones with a VoIP client presenting 3D sound and the new recording device with its separation feature to a conventional desktop phone. Therefore, every device should be handled in according to its specific capabilities delivering the best possible conferencing experience. This task is explained in detail within the thesis of *Matthias Kaufmann, B.Sc.* [22].

2.7. Speech synthesis

As already mentioned, the system is developed to connect conventional telephones and VoIP clients with a device recording and separating speakers spatially. It creates an appropriate sound signal for each participating device. In order to exploit the full potential of separately recorded speakers on the receiver side, a client software supporting binaural sound is needed. Therefore, in previous work at the *Institute for Data Processing* [32] a Voice over IP (VoIP) software client plugin was developed which presents each teleconference participant spatially distributed over a stereo headset. This client uses a set of Head Related Transfer Functions (HRTFs) to generate the spatial sound signal.

In future, additional sound synthesis techniques will be needed in order to improve the quality of the headphone based sound synthesis. This will also bring lifelike multi-party conferencing to full room solutions using multichannel audio systems and techniques like wave field synthesis (WFS) [19]. WFS uses a large number of loudspeakers (tens to hundreds) to reproduce a sound field in a larger space enclosing possibly multiple listeners.

Altogether, a system with a sound rendering, where the user can distinguish between speakers by their spatial location, is an important part of an immersive teleconferencing system.

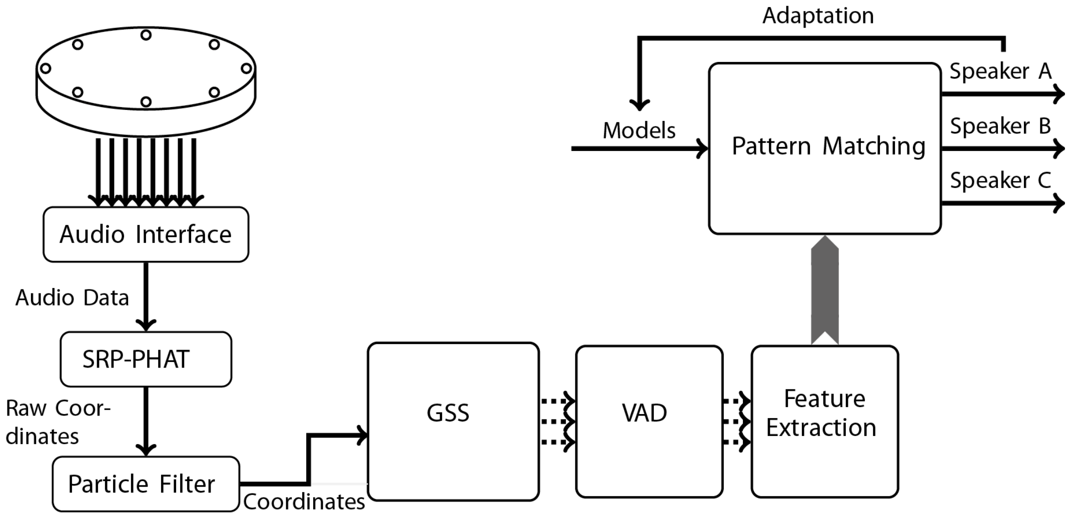

Figure 2.3.: System approach for a conference telephone with separated channels for each speaker as developed at the *Institute for Data Processing*.

3. Speaker Recognition: An Overview

Humans are able to distinguish between people just by the sound of their voice. Even tiny differences in the sounding of voices are noticeable. This makes the human capability of recognizing a speaker very reliable, even when imitators or impostors try to copy the voice of another person.

A lot of different factors play a role in the human ability of recognizing a speaker. Low-level information like the talking frequency mainly define the sound of a voice but high-level information like the words used, the talking speed and how pauses are positioned between words also play a role in distinguishing one speaker from another [3].

A huge challenge in speaker recognition is the robustness for variances. The same words, spoken by the same speaker in two different moments of time, never sound exactly the same. These variances arise from the speaker itself, where mood, health or condition can have effects on the voice, as well as from the surrounding environment and noise level that is active during the recording of the speech.

The aim of speaker recognition is to model speakers as representative as possible and unsusceptible against variances so that a recorded test utterance of this speaker can be labeled with the speaker's name.

Speaker recognition is a task of pattern recognition. A system for pattern recognition typically consists of three blocks:

- **Preprocessing**: Before processing a speech signal, it needs to be recorded and digitalized. This input may comprise errors, caused by the recording techniques and devices used, and the signal itself may be distorted or noisy. The task of preprocessing is to eliminate disturbances and errors and prepare the input for feature extraction.

- **Feature extraction**: This block tries to represent the preprocessed signal by a small number of parameters. These parameters have to be deterministic, should be decorrelated and ideally contain only relevant information.

- **Classification or pattern matching**: The classification block labels extracted features by assigning them to a class. This classification is done by matching the features to all possible classes, using a decision function.

Speaker recognition systems compare a data sample with a database of speaker models to find the most likely originating model for this sample. The data sample is a set of feature vectors extracted from the analysed speech signal. The speaker models are built

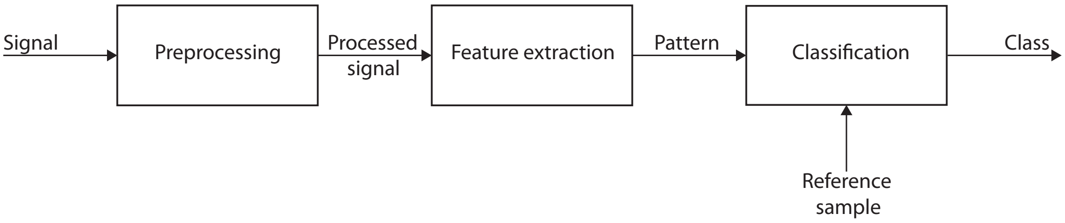

Figure 3.1.: Schematic layout of a pattern recognition system

on a high number of similar measurements, representing speaker-specific feature values. The extracted features are assigned to a speaker model and so labeled with a speaker name.

This chapter gives an introduction to the topic of speaker recognition, its various tasks and the state-of-the-art speaker recognition methods.

The first section gives an overview of different tasks for speaker recognition, its definition and the applications it can be used for. Since speaker recognition on audio signals is a digital signal processing task, a short introduction to digital speech processing is given in section 3.2. Section 3.3 gives insight into the feature extraction of a speech signal. Section 3.4 shows how a speaker model is built up and used to classify a speech signal. These speaker models do not have to be static, section 3.5 shows how to dynamically adapt speaker models. The last section illustrates how the decision to allocate a speech signal to a certain speaker is made. In addition, the different measurement values for evaluating a speaker recognition system are shown.

3.1. Tasks and Applications

Different tasks are defined under the general heading of speaker recognition [3].

- Speaker *verification*
- Speaker *identification*
- Speaker *detection* (with segmentation or diarization)

The following section provides an overview over these tasks and their applications.

3.1.1. Tasks

Speaker verification is the simplest task of speaker recognition. In a speaker verification process a speech sample is given in connection with an identity claim. It is only checked whether the identity claim is correct or not. The unknown voice sample is compared with a speaker model whose label corresponds to the identity claim and is either accepted or rejected. The decision is made by comparing to a threshold: If the comparison value is exceeding a certain threshold, the speaker claim is accepted. This threshold is the most important parameter when designing a speaker verification system and has to be chosen carefully [24]. If it is too low, more false speaker claims are accepted. Is it too high, correct speaker claims often get rejected which could be annoying to users of the system. The best example for the use of speaker verification is a system for access control.

Speaker identification is a more general task: It is given a speech sample only. Now this sample is compared to all available and considered speaker models. The identification is either *open-set* or *closed-set*.

When it is known that the set of speaker models includes all speakers of interest the task is referred to as *closed-set* identification. The label of the best matching speaker is taken to be the identified speaker. Most speaker identification applications are *open-set*, meaning that it is possible that the unknown speaker is not included in the set of speaker models. In this case an additional speaker verification is needed: First, the speech sample is matched to all available models. Then, the comparison value of the best matching speaker is checked, whether it exceeds the threshold. This way, it is also possible to label an incoming speech sample as unknown. So speaker verification can also be seen as a special speaker identification task with only one model to consider.

Speaker detection is a task of speaker recognition used for conversational speech only. The input is a stream of speech containing multiple speakers. The function of speaker detection is to check whether certain speakers appear in the input stream [25]. Speaker detection can be used to find those sound files out of a database of recordings, where a certain speaker is present.

An enhancement of the detection task is the speaker diarization [16]. Speaker diarization (or segmentation) determines the intervals in a multispeaker test sample during which a detected speaker is talking. Therefore, the speech sample is segmented into parts with only one active speaker and then labeled with the speaker's name. With this procedure, speaker diarization answers the question of "who spoke when?" [21].

Speaker diarization can also be performed online. In this case, the input is a continuously incoming multi-speaker audio stream. There is no prior knowledge about start and end points of talking, speaker changes and useful information like level differences. A speech segment has to be labeled with a speaker name as soon as it arrives or in defined time intervals. Online speaker diarization changes the definition of speaker diarization ("Who speaks when?") into "Who speaks now?" [12, 14]. A big challenge of such a system is the demand for very low computational complexity. Also, there is usually only few input data available to work with.

All of these speaker recognition tasks can be realized in two different ways:

- **Text-dependent**: If the speaker is prompted or expected to provide a known text and if speaker models have been trained explicitly for this text, the input mode is said to be *text-dependent*. Text-dependent speaker recognition relies mostly on how specific words or phonemes, that are known in advance, are pronounced [3].

- **Text-independent**: If the speaker cannot be expected to utter specified texts the input mode is *text-independent*. In this case speaker models are not trained on explicit texts. A text-independent system distinguishes between speakers by speaker-specific spectral attributes defining and shaping the sound of the voice [30].

High-end conferencing systems mostly come with a video camera, transmitting not only the recorded audio but also a video stream of the conference. Video can be used to perform or support speaker recognition tasks. Speaker recognition with the help of a video stream is an interesting topic but exceeds the range of this thesis and will not be covered.

3.1.2. Applications

A wide variety of applications exists for speaker recognition systems. This section will name three examples where speaker recognition is already used in real systems and products.

The most widespread applications for speaker recognition are for security. Most of them are typically speaker verification applications intended to control access to privileged transactions or information remotely. Telephone banking or access control systems are good examples for such applications. Mostly, text-dependency is given: The user is prompted to speak a verification phrase or a PIN number, known beforehand to the system. Models are trained by recording and processing speech input in an enrollment session [3].

Another important field of application is forensics. Most of the individuating characteristics, such as fingerprints, iris patterns and DNA structure are hard to perceive and need complex measurements. Voice is easy to acquire and can help to identify a perpetrator. Forensic applications are likely to be open-set speaker identification tasks, where a sample of speech from a suspect is compared to a recording of the perpetrator. They can either be text-dependent or text-independent, depending on the data available [3].

A new field of application, where this thesis will focus on, is teleconferencing or telepresence. To increase comprehensibility of speech sources and audio immersion for teleconferencing systems or for operations with high-fidelity telepresence and teleaction systems, audio signals of different speakers in the conference room or around a robot have to be assigned to individual audio channels in order to enable immersive 3D-Sound synthesis at the remote site. The assignment of the audio signals of different speakers to individual audio channels can be achieved by using a speaker recognition system [31]. Speaker recognition for conferencing situations needs to be text-independent.

Altogether, speaker recognition is an important topic and useful for many different systems and applications.

3.2. Digital speech signal analysis

The human speech apparatus is able to generate sounds with a wide variety of spectral structures. The shape of the vocal tract (tongue, jaw, lips) defines peaks in the spectrum associated to periodic resonances. These resonances are called speech formants. Together with the basic fundamental frequency created in the excitation tract (lung, vocal chord system), a certain spectrum is generated. The locations of fundamental and formant frequencies and, to a lesser degree, the shapes of the resonances distinguish one speech sound from another [20].

In order to process a speech signal digitally, it has to be recorded and digitalized. The A/D conversion consists of two steps: sampling and quantization.

Sampling represents an analog signal as a sequence of equidistant impulses. The sampling frequency f_A needs to follow Nyquist's sampling theorem

$$f_A \geq 2f_{max} \tag{3.1}$$

to avoid spectral aliasing in the analyzed frequencies up to f_{max}. The spectrum of human voice does not exceed a frequency of 8 kHZ, so a sampling frequency of 16 kHz is adequate for speech processing.

Quantization maps all sampled values of the analog signal to values processable by a computer. A precision of 16 bit per sample is sufficient for keeping the dynamic range of a speech signal.

A speech signal has a low-pass characteristic, resulting from the anatomy of the vocal tract [4]. To compensate this effect, a simple recursive filter (preemphasis filter) is used for amplifying high-frequency spectral components.

$$H_{pre}(z) = 1 - \alpha z^{-1} \tag{3.2}$$

This mitigates the low-pass effect of the speech signal. Moreover, it enhances individual voice characteristics of a speaker because high-frequent formants bear a lot of speaker-dependent information [3]. Figure 3.2 shows the frequency response of the filter for different values of α.

An essential parameter of a speech signal is the signal energy E. The energy of the digital signal $s(k)$ is defined as

$$E = \sum_{k=-\infty}^{+\infty} s^2(k). \tag{3.3}$$

3.2.1. Spectral analysis of a speech signal

To analyze the spectral attributes of a speech signal, the signal $s(k)$ is transformed into frequency domain. S_k is called the Fourier transform of $s(k)$.

$$STFT\{s(k)\} \equiv S_k(m,\omega) = \sum_{k=-\infty}^{\infty} s(k)w(k-m)e^{-j\omega k} \tag{3.4}$$

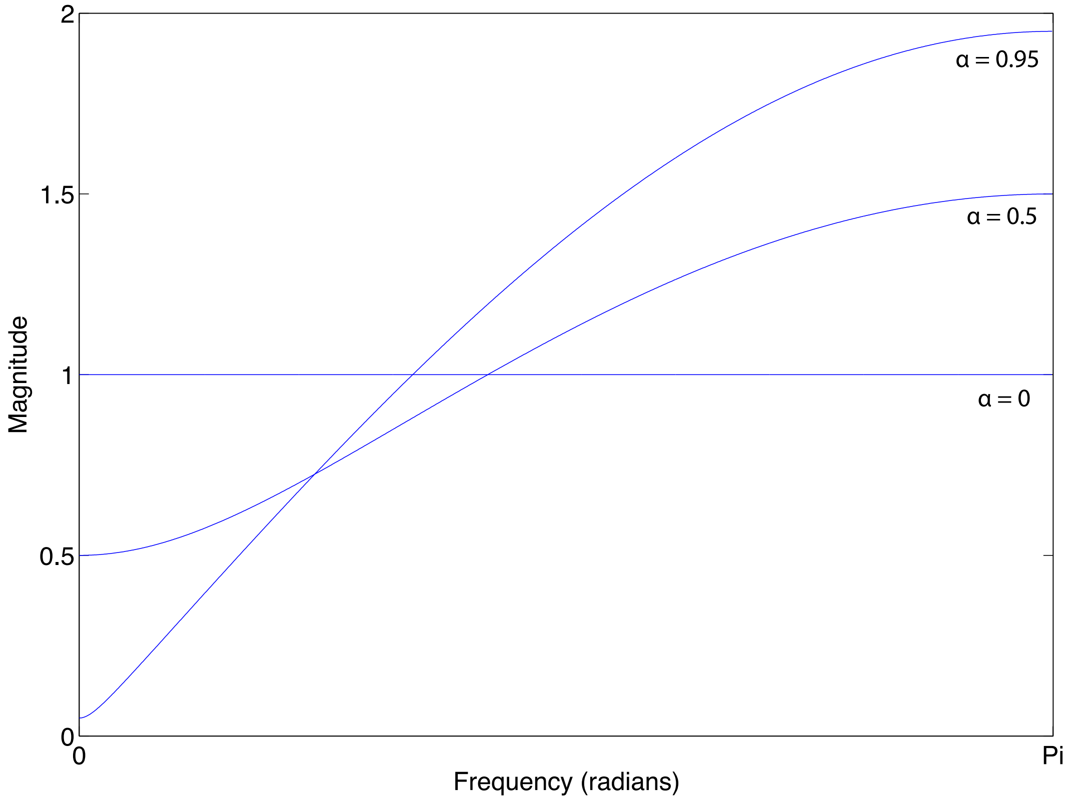

Figure 3.2.: Magnitude of the frequency response of the preemphasis filter for different values of α

This is only possible for short time intervals since a signal is required to be stationary for frequency analysis. For very short time intervals, every interval can be seen as satisfactorily stationary. The signal intervals are called frames.

Before transforming a signal into the frequency domain, it needs to be weighted by a window function w. The window used should be narrow-banded in frequency domain but at the same time fade out quickly in time domain. A good tradeoff is the hamming window, often used in speech processing tasks [3].

$$w(\tau) = 0.54 + 0.46\,\cos(2\pi\frac{\tau}{T}) \quad \tau = -\frac{T}{2} \ldots + \frac{T}{2} \tag{3.5}$$

Figure 3.3 shows a hamming window.

The length of the window is an important parameter and needs to be well chosen. A window that is too long will violate the short time requirement and the signal can no longer be considered stationary. A window that is too short will not contain enough information to reliably work with. A window length of 20-30 ms progressing in 5-25 ms steps has shown to be a good value for feature extraction of speech.

A very fast implementation of the short-time Fourier transformation is the Fast-Fourier-Transformation (FFT) which is used for all implementations for this thesis.

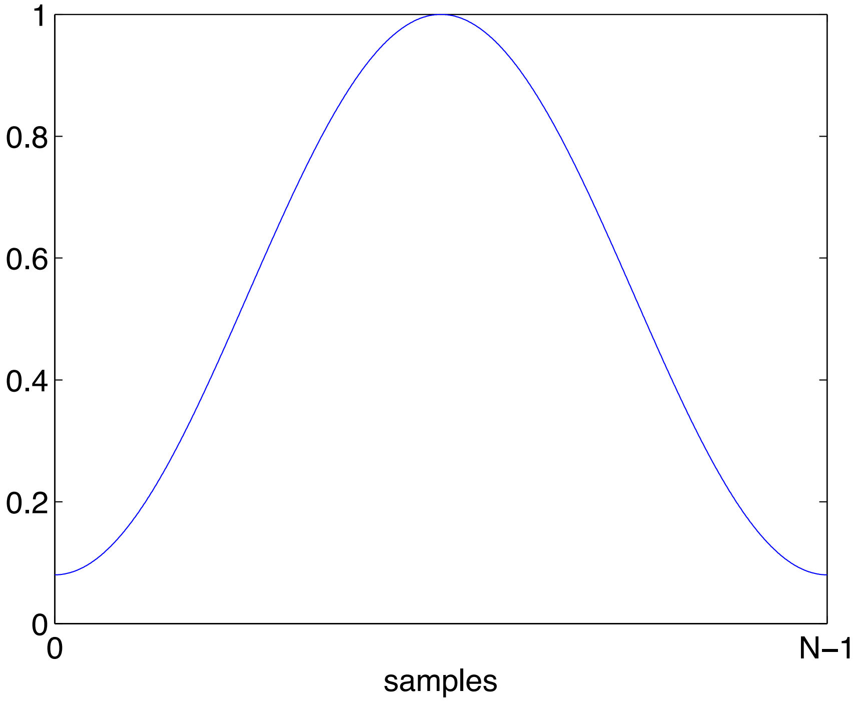

Figure 3.3.: Hamming window in time domain

3.3. Feature extraction

The task of feature extraction is to separate between relevant and irrelevant information in a signal and reduce its dimension. Every frame is parametrized by a vector of features $\vec{x}_i$:

$$\vec{x}_i = \begin{pmatrix} x_{i,1} \\ : \\ x_{i,N} \end{pmatrix} \tag{3.6}$$

For every frame i the feature vectors $\vec{x}_i$ are concatenated to a feature matrix $\vec{X}$:

$$\vec{X} = [\ \vec{x}_1,\, \vec{x}_2,\, \ldots\ ] \tag{3.7}$$

This sequence of feature vectors is given to algorithms that build up speaker models or do pattern matching to label the corresponding segment of the speech signal with a speaker name.

3.3.1. Mel Frequency Cepstral Coefficients (MFCCs)

The same sounds spoken by different speakers vary essentially in formants and spectral attributes based on the physiological differences of the vocal tract. In order to resolve this speaker-dependent information from a speech signal the feature extraction is performed. Commonly used for speaker modeling are low-level acoustic features such as short-time spectra. The most popular spectral features are the so called Mel-Frequency Cepstral Coefficients (MFCCs) [27, 33, 14]. MFCCs have proven to represent individual voice spectra very well [27]. In previous work they were used to identify the gender of the vocalist in recorded popular music [33].

The average signal energy is calculated and filtered on triangular-band filters in W frequency intervals, which are non-linear distributed on the frequency scale. These filters are aligned according to the physiology of the human audio perception. Adjacent filters overlap half of the filter length. This filter is called a mel filter bank, since the filter transforms the signal non-linear, according to the mel scale. Figure 3.4 shows a mel filter bank with 20 triangular filters.

With $F_{mel}^{(w)}(n)$ defined as the frequency response of the w-th filter, the mel-energy of a frequency segment with K samples is

$$E_{mel}^{(w)} = \sum_{n=0}^{K/2-1} F_{mel}^{(w)}(n)\,|\,S(k)\,|^2 \quad 1 \leq w \leq W \tag{3.8}$$

Then, the logarithm is calculated and a discrete cosine transformation (DCT) is applied. The result are M MFCCs.

$$c_{MFCC}^{(i)} = \sum_{w=1}^{W} \log(E_{mel}^{(w)}) \cos[\,i\,(w-0.5)\frac{\pi}{W}\,] \quad 1 \leq i \leq M \tag{3.9}$$

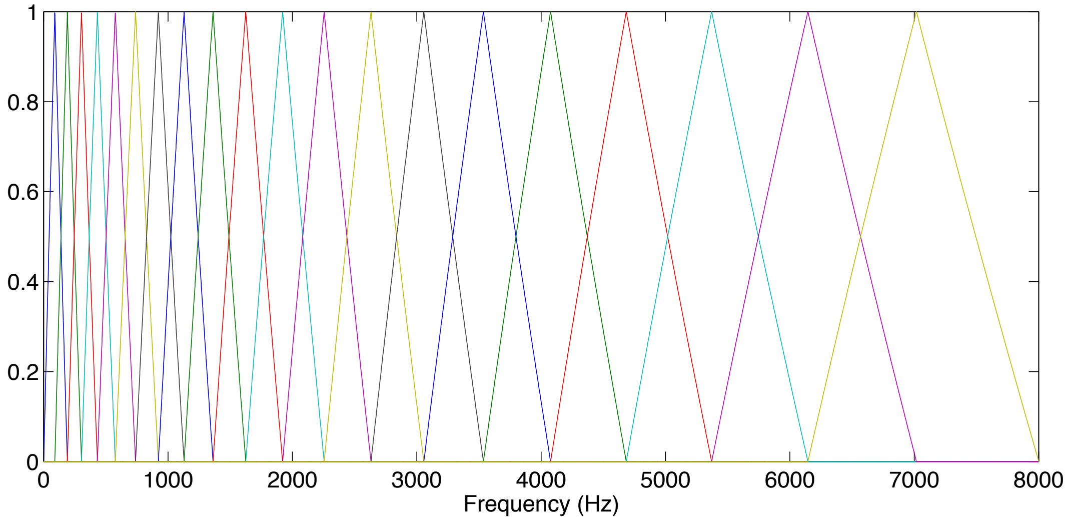

Figure 3.4.: Mel filter bank with triangular bandpass filters

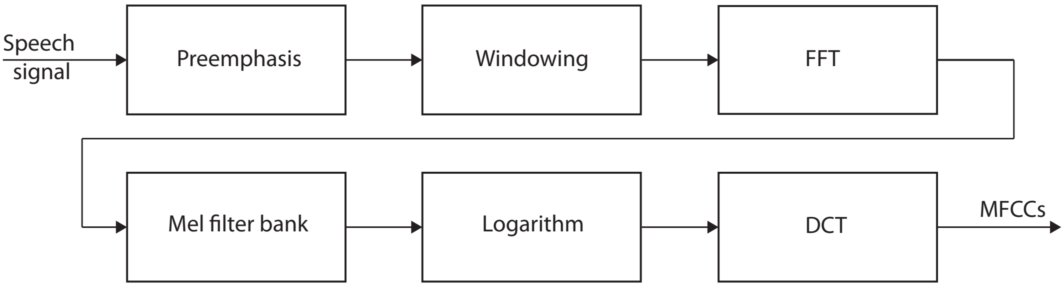

Figure 3.5.: Block diagram of the MFCC extraction procedure

Figure 3.5 shows the procedure of extracting MFCCs. Experiments have shown that these coefficients are highly suitable as features for speaker recognition [27]. The frequency scale is weighted in according to the human auditory system and all features are highly decorrelated. Figure 3.6 shows the MFCCs of a speech signal example.

3.3.2. Other features

The signal energy itself can serve as a feature for speaker recognition tasks. If there is no voice activity detection (VAD) in place, which classifies into speech and non-speech samples, signal energy is a highly meaningful feature. This is often related to as the zeroth MFCC [27, 33].

There are a lot of other features in speech processing for a lot of different tasks. The fundamental frequency of speech is an important feature when distinguishing speaker gen-

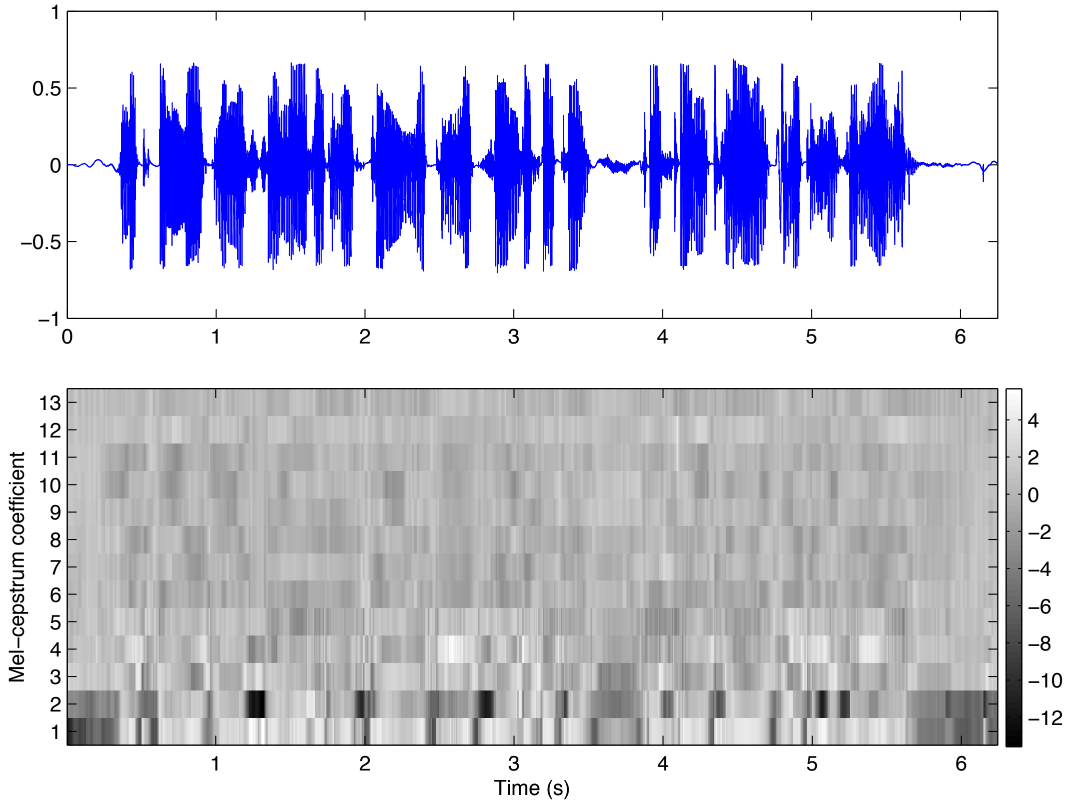

Figure 3.6.: Example of MFCC values for a given speech signal of a male speaker. It can be seen that silent parts with no speech lead to a completely different MFCC pattern.

der and the zero-crossing rate of a speech signal denotes whether a phoneme spoken is voiced (like a vowel) or unvoiced (like a fricative) [33], just to name a few.

3.4. Speaker models

The modeling of speakers is the fundament of every speaker recognition system and the basis of comparison for unknown speech samples. The accuracy of a speaker recognition system strongly depends on the speaker model quality. Therefore, an efficient and well-thought modeling is essential. The model approach to use is tightly coupled with the application. A text-independent recognition needs less detailed models to be more general whereas a text-dependent system benefits of detailed modeling of words, phonemes or phrases.

Speaker models can be divided into parametric and non-parametric approaches.

3.4.1. Non-parametric approaches

Non-parametric approaches are mostly effective with much training data. The approaches are mostly simple and used especially for speaker verification tasks.

Templates

An easy way of speaker modeling uses so-called templates. The model data just consists of a certain number of feature matrices being representative for all appearing sequences of features. These feature matrices are called templates. Test sequences are compared to these templates by simply measuring the distance in feature space. If the distance is below a certain threshold, the identity claim of a person is accepted. This approach is seldom used and only adequate for password applications [3].

Nearest-Neighbor Modeling

Nearest-Neighbor modeling is the most popular non-parametric approach [3]. Training data is used to calculate local densities in feature space for each class. This approach acts on the assumption that different classes differ in the distribution of local densities. A test sequence is assigned to the class with the maximum local density.

3.4.2. Parametric approaches

Vector quantization

Vector quantization (VQ) constructs a set of representative samples of the target speaker's training sequences by clustering the feature vectors. An algorithm is needed to build up these clusters, the most commonly used is k-means clustering [17] by *Hartigan* and *Wong*.

After establishing the model clusters, test sequences are assigned to a class by measuring the distance between the feature vectors and the centroid of the cluster associated with that class. This approach has been shown effective in text-dependent and text-independent applications [3] but has been overtaken by the more general Gaussian mixture models.

Gaussian Mixture Models (GMM)

A Gaussian mixture model (GMM) is a statistically based representation of the speaker identity. GMMs can be seen as a more general VQ, assuming the feature vectors are drawn from a probability density function that is a mixture of Gaussians. The Gaussian mixture speaker model was introduced by *D.A. Reynolds* and *R.C. Rose* in [30].

For a N-dimensional feature vector, $\vec{x}$, the mixture probability density function is defined by

$$p(\vec{x}|\lambda) = \sum_{k=1}^{K} w_k \, \mathcal{N}(\vec{x} \mid \vec{\mu}_k, \Sigma_k) \ , \tag{3.10}$$

where $\mathcal{N}(\vec{x} \mid \vec{\mu}_k, \Sigma_k)$ is a unimodal Gaussian density

$$\mathcal{N}(\vec{x} \mid \vec{\mu}_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\{-\frac{1}{2}(\vec{x}-\vec{\mu}_k)^T \Sigma_k^{-1}(\vec{x}-\vec{\mu}_k)\} \tag{3.11}$$

parametrized by a mean vector, $\vec{\mu}_k$, and a covariance matrix, Σ_k:

$$\vec{\mu} = \{\vec{\mu}_1, \ldots, \vec{\mu}_K\} \ , \tag{3.12}$$

$$\Sigma = \{\Sigma_1, \ldots, \Sigma_K\} \ . \tag{3.13}$$

Part of a unimodal Gaussian density is the so-called Mahalanobis distance Δ^2:

$$\Delta^2 = (\vec{x}-\vec{\mu}_k)^T \Sigma^{-1} (\vec{x}-\vec{\mu}_k) \tag{3.14}$$

$$\mathcal{N}(\vec{x} \mid \vec{\mu}_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\{-\frac{1}{2}\Delta^2\} \tag{3.15}$$

The mixture density (3.9) is a weighted linear combination of K Gaussian densities with mixture weights w_k

$$\vec{w} = \{w_1, \ldots, w_K\} \ . \tag{3.16}$$

These weights furthermore satisfy the constraint

$$\sum_{k=1}^{K} w_k = 1 \ ; \quad 0 \leq w_k \leq 1 \quad . \tag{3.17}$$

Collectively, the parameters of the density model are denoted by

$$\lambda = \{w_k, \vec{\mu}_k, \Sigma_k\} \quad k = 1, ..., K \ . \tag{3.18}$$

Let $\vec{X}$ be a sequence of N feature vectors $\vec{x}$:

$$\vec{X} = \{\vec{x}_1, \ldots, \vec{x}_N\} \tag{3.19}$$

When building up a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising means $\vec{\mu}_k$, covariances Σ_k and mixing coefficients w_k), assuming the observed sequence of feature vectors X is drawn from this density function:

$$\ln p(\vec{X} \,|\, \lambda) = \sum_{n=1}^{N} \ln p(\vec{x}_n \,|\, \lambda) \tag{3.20}$$

$$= \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} w_k \ \mathcal{N}(\vec{x}_n \mid \vec{\mu}_k, \Sigma_k)\} \tag{3.21}$$

Out of the techniques used to build speaker models, Gaussian mixture models have proven to be effective, efficient and easy to implement, especially in the case of text-independent speaker recognition where the system has no prior knowledge of the spoken text [30, 28, 29].

Once the GMMs for every speaker are trained, test utterances can be compared to each model just by calculating the log-likelihood as a score. This is computationally very inexpensive, so GMMs are a good choice for modeling speakers in an application that needs to recognize speakers online. Log-likelihood as score for pattern matchingis explained in more detail in section 4.5.

Hidden Markov Models (HMM)

Another effective method for speaker recognition, especially in text-dependent systems, are Hidden Markov Models (HMMs). In applications where the system has prior knowledge of the text, HMMs are much more effective than GMMs. Because of this, HMMs are also the basis of almost all speech recognition systems [26]. Each word can be characterized by a HMM with a certain number of states, in which each state is represented by a probability density function. In general, a GMM can be seen as a single-state HMM with a Gaussian mixture density.

Since this thesis focusses on the application in a teleconferencing system which is text-independent, HMMs are not considered for this task.

3.4.3. Other models

There is a large number of other interesting speaker modeling approaches that have been shown to be useful. *Support Vector Machines (SVMs)* [37] are often used for tasks including speaker identification and provide a way for training classifiers using discriminative criteria [6]. *Eigenvoice* modeling is an approach in which the speaker models are confined to a low-dimensional linear subspace. The advantage is that this approach can be effectively used, even when training data is too limited for the effective use of other modeling approaches [35]. *Artificial neural networks* have also been shown to be useful [10].

3.5. Adaptation

In systems where training data is limited and proper modeling of speakers during the enrollment period cannot be assured, adaptation is a good possibility to automatically multiply training data. With adaptation, speaker models ideally become more representative over time and recognition accuracy improves with every correct recognized speaker. Mostly, the model training does not adequately characterize the range of test conditions. Adaptation is a very effective way to mitigate these effects [3].

Adaptation can be unsupervised or supervised. Unsupervised adaptation uses data from previous identifications and has no need of additional input. However, there is the possibility that models are adapted on imposter utterances. This can also lead to an impairment of the models, lowering the recognition accuracy. Supervised adaptation needs additional information to not adapt on wrongly identified speakers. This can be implemented using an interaction system with the user or likewise methods.

3.6. Evaluation

To measure the performance of a speaker recognition system quantitatively, an evaluation is carried out. Usually a system is evaluated with data from a large database. This database needs to be divided into a training set and a test set. This way, the system is not evaluated with data it was already trained on. This is necessary to simulate a real application and to not falsify the evaluated measurements.

Important evaluation measurements for any pattern recognition system are *accuracy*, *precision* and *recall*. They can be descriptively explained by calculating them from a confusion matrix.

	Class 1	**Class 2**
Classified 1	A	B
Classified 2	C	D

Table 3.1.: Confusion matrix: Values A, B, C, D are absolute numbers of classified segments

The most obvious measurement is the recognition rate, or accuracy. It is calculated by the ratio of correct classified segments to the number of segments classified.

$$Accuracy = \frac{A+D}{A+B+C+D} \tag{3.22}$$

For a binary class problem, precision and recall can be calculated for every class:

$$Precision = \frac{A}{A+B} \tag{3.23}$$

$$Recall = \frac{A}{A+C} \tag{3.24}$$

Since most of the speaker recognition tasks are non-binary (except speaker verification), a different measurement, especially for conversational speech needs to be found.

In the field of speaker diarization, the main error measurement is the so-called diarization error rate (DER) [36]. The DER is determined by calculating the weighted sum of different error components. These components are the miss-error (speech segment not detected), the false alarm (segment incorrectly declared as speech) and the speaker error (wrong speaker identified):

$$DER = \delta_{miss-error} + \delta_{false-alarm} + \delta_{speaker-error} \tag{3.25}$$

The DER also considers speech segments with overlapping speakers. If one speaker out of multiple overlapping speakers is not detected, the whole segment is declared as error.

4. Online Speaker Recognition for Teleconferencing Systems

This chapter describes the usage of a speaker recognition system in a real teleconferencing application. It mainly focuses on the constraints and requirements that need to be considered and the approaches chosen to meet these instructions. Parameter values and used algorithms are explained in detail and why they were picked for this task. The online speaker recognition system as described in this chapter was implemented at the *Institute for Data Processing*.

4.1. System requirements

Any real-world speaker recognition system has a number of practical constraints [3]:

- **Efficient speaker modeling**: The number of parameters of each model should be kept as small as possible to minimize the required amount of training data. This is not only efficient in terms of storage and retrieval but also important to users since the training material ideally should consist of only a single recording made in a very short time.
- **Short recognition time**: Recognition must need only a limited amount of speech data to ensure a short recognition phase for user comfort and practicability.
- **Relaxed text restrictions**: Restrictions on which text has to be spoken by the user should be as relaxed as possible.

These requirements are applicable to any speaker recognition application. Speaker recognition to be used in teleconferencing systems inherits a lot of additional challenges and requirements:

- **Online processing**: Speakers need to be identified on a continuous input audio stream. There is no prior knowledge about the incoming data and the computational complexity needs to be very low.
- **Text-independence**: There is also no prior knowledge about the text spoken by the users. Features need to represent the overall sound of a voice rather than certain words or phrases.

- **Detection of speech activity**: To extract features only from segments containing speech, it needs to be determined whether a segment can be classified as speech or pause segment. Speaker models built on features containing silence or noise perform worse.
- **Fast and reliable adaptation**: Adaptation of models with classified and labeled speech data needs to be computationally inexpensive to not slow down the system. In addition, adaptation should benefit the system rather than impairing the speaker models.
- **Parallel processing of multiple input streams**: Multiple input streams should be computed parallel and independently. This should not jeopardize temporal synchronization or online processing capability.

For the system implemented in this thesis, the number of speakers needs to be known in advance. This is because of the difficulty of implementing a reliable detection of the number of speakers present in a conference [14]. The use of video for the task of speaker recognition is not part of this thesis since the developed teleconferencing system aims at an audio-only solution.

The online speaker recognition system is developed step by step in the following sections, trying to meet all the above mentioned requirements.

4.2. Preprocessing

In a conference situation the speakers are recorded with a microphone array. Localizing the speakers and separating simultaneously active speech sources is carried out. The input of the speaker recognition module now is one audio stream per localized active source. These audio streams can contain multiple speakers but the separation module made sure that only one speech source is active in one segment. Before speaker recognition can be performed the input signal has to be preprocessed.

4.2.1. Input signal

As previously mentioned, a sampling frequency of 16 kHz is sufficient for speech signals since human speech never exceeds a maximum frequency of 8 kHz. The localization and separation is carried out at 48 kHz, so the input signals need to be downsampled. The whole system works with a quantization resolution of 16 bit and so the input needs no additional quantization.

Then, the input streams are filtered with the preemphasis filter to amplify the high-frequency spectral components. A value of 0.95 is selected for the parameter α of the preemphasis filter.

Before transforming the signal into frequency domain, it is weighted and split up into overlapping short-time frames by a window function. A hamming window, with a length of

20 ms and progressing in 10 ms steps, has been chosen for this purpose. Then the frames are transformed into frequency domain by a Fast Fourier Transform (FFT).

4.2.2. Voice activity detection

A voice activity detection (VAD) locates frames with active speech by comparing the frame energy to a threshold, and discards silence frames, similar to the segmentation method in [14].

First, the energy of every frame is calculated. This energy value is compared to a certain threshold. If the energy exceeds the threshold, the associated frame is labeled as *speech*, otherwise as *silence*. It takes a certain amount of *speech* frames to get a *speech* segment. Once a speech segment is detected an additional frame at the beginning and the end of the segment is added to ensure smooth transitions. The result of VAD is a shortened speech signal where all *silence* frames were discarded.

A threshold value of about 20-40 dB above minimum signal energy has been proven to be useful for the proposed conferencing system. In recordings of conference conversation the voice activity detection typically discards about 25% of the signal. Figure 4.1 shows an example of the VAD applied to a speech signal.

4.3. Feature extraction

The reduced speech signal is now sent to the feature extraction where the speaker-dependent information is parametrized into features. The use in conferencing applications raises the constraint of text-independence. For this reason, low-level acoustic features such as short-time spectra are chosen to represent the overall sound of a speaker's voice.

As previously mentioned, MFCCs represent the characteristics of individual voices very well. With a frame length of 20 ms and a sample rate of 16 kHz, one frame consists of $20ms * 16kHz = 320$ samples. With a maximum frequency of 8 kHz the mel-filterbank consists of $W = 20$ triangular filters.

Then, the mel-energy per filter w of a frame with 320 samples is

$$E_{mel}^{(w)} = \sum_{n=0}^{159} F_{mel}^{(w)}(n)\,|S(k)|^2 \quad 1 \leq w \leq 20 \tag{4.1}$$

Finally, the resulting M MFCCs are calculated:

$$c_{MFCC}^{(i)} = \sum_{w=1}^{20} \log(E_{mel}^{(w)}) \cos[i\,(w-0.5)\frac{\pi}{20}] \quad 1 \leq i \leq M \tag{4.2}$$

$M = 12$ MFCCs are used as features. This value has been chosen to have enough MFCCs containing speaker-dependent information as well as keeping the overall number

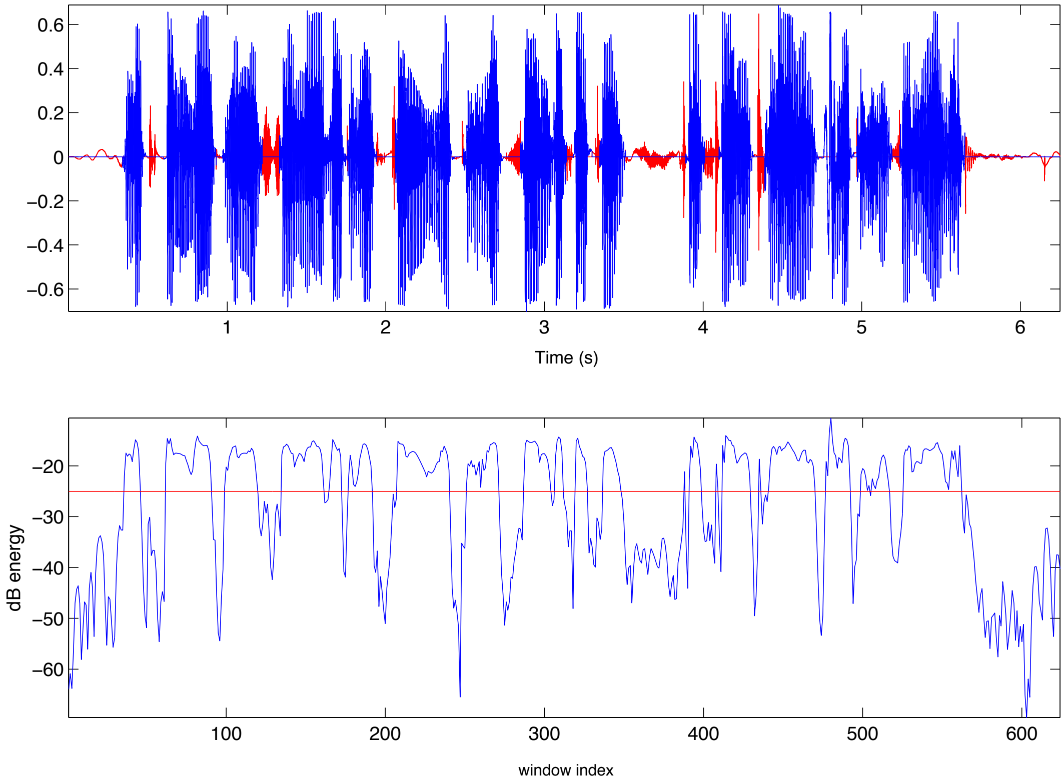

Figure 4.1.: Voice activity detection: The speech signal of a male speaker is framewise analyzed. The energy per frame is calculated and compared to a threshold. If the energy is below this threshold, the frame is discarded (marked red in the figure).

of features low to meet the above mentioned constraints. The spectral frame energy is also added as feature.

The feature vector is extended by the respective first- and second order delta regression coefficients to incorporate dynamic information. The MFCCs just take a single frame into account, not the progression over time. The derivatives add dynamic information to the feature vector and have proven to be very useful as recognition feature [27].

Thus, altogether a set of 39 features is employed.

4.4. Model training and adaptation

Out of the speaker modeling approaches presented in chapter 3, Gaussian mixture models were chosen to represent individual speakers. To build up a GMM with training data, the parameters $\{w_k, \vec{\mu}_k, \Sigma_k\}$ of the model λ need to be estimated. This is done using the expectation-maximization (EM) algorithm [8].

4.4.1. Expectation-maximization (EM) algorithm

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means $\vec{\mu}_k$ and covariances Σ_k of the components and the mixing coefficients w_k). First, we use the K-means algorithm to initialize the parameters of the component densities, which is an efficient way to find initial values, rather than initialize randomly [17].

Then, the expectation-maximization (EM) algorithm iteratively refines the model parameters to increase the likelihood of the estimated model for the training sequence $\vec{X}$. The EM algorithm consists of two steps: The *expectation* step, where the training data is applied to the current model, and the *maximization* step, where the model parameters are refined and adapted to the training data. These two steps are iterated until a convergence criteria is satisfied.

EM algorithm

I Use the K-means algorithm to initialize the parameters of the component densities.

II E step: Compute the posterior probability that $\vec{x}_n$ is drawn from the k-th component $p_k(\vec{x}_n \,|\, \lambda_k)$:

$$P(k \,|\, \vec{x}_n, \lambda) = \frac{w_k \, p_k(\vec{x}_n \,|\, \lambda_k)}{p(\vec{x}_n \,|\, \lambda)} \tag{4.3}$$

III M step: Update the model using these equations:

$$\vec{\mu}_k = \frac{\sum_{k=1}^{K} P(k \,|\, \vec{x}_n, \lambda)\, \vec{x}_n}{\sum_{k=1}^{K} P(k \,|\, \vec{x}_n, \lambda)} \,, \tag{4.4}$$

$$\Sigma_k = \frac{\sum_{k=1}^{K} P(k \,|\, \vec{x}_n, \lambda)\, \vec{x}_n \vec{x}_n^T}{\sum_{k=1}^{K} P(k \,|\, \vec{x}_n, \lambda)} - \vec{\mu}_k \vec{\mu}_k^T \,, \tag{4.5}$$

$$w_k = \frac{1}{K} \sum_{k=1}^{K} P(k \,|\, \vec{x}_n, \lambda) \,. \tag{4.6}$$

IV Evaluate log-likelihood. If it is below a certain threshold → Go back to step **II**.

4.4.2. Universal Background Model (UBM)

The EM algorithm builds up very reliable and representative speaker GMMs. However, it takes a long time especially when building GMMs with many mixture components. Building up individual speaker models with the EM algorithm does not fulfill the requirements for a speaker recognition system in teleconferencing applications as stated in section 4.1.

Therefore, a Universal Background Model (UBM) is used to support the speaker modeling approach with GMMs. A UBM is a single GMM that is trained on speech samples from a large number of representative speakers. So it is not only built on data from a specific speaker, but on data from a lot of example speakers. By building up a model on such diverse data, the model is not capable of recognizing particular speakers but represents the common attributes of all different voices.

To receive speaker-specific models for the recognition task, every speaker model is derived from this UBM by individually adapting it. Only a small amount of training data is needed to adapt an already existing model and models can be adapted very fast, as shown in the following section. The UBM is pre-built and stored within the conferencing system. At the beginning of each conference, a short enrollment session introduces the present speakers to the system and a small amount of training data is gathered from every speaker. This training data is used to adapt a specific model for every speaker from the UBM.

The main advantage is that the UBM has to be trained only once, which can be computed in advance for a wide variety of possible speakers. The speaker-specific models are then adapted from this UBM. This leads to a very fast creation of speaker-dependent GMMs, benefiting the user comfort and practicability, and it has been shown that GMMs adapted from a well-trained UBM also yield better speaker recognition results [29].

4.4.3. MAP adaptation

To adapt models from the UBM or to dynamically adapt models while recognizing in a teleconference, an effective yet computationally inexpensive adaptation algorithm has to be used. The adaptation used in the teleconferencing system is done by a form of Bayesian adaptation, known as Bayesian learning or *maximum a posteriori* (MAP) estimation [9]. MAP adaptation takes the existing model and adapts it using incoming speech information after recognizing the associated model. To adapt the means of an existing GMM, the following calculation needs to be done:

For the mixture component k the posterior probability is

$$P(k\,|\,\vec{x}_n,\lambda) = \frac{w_k\, p_k(\vec{x}_n\,|\,\lambda_k)}{p(\vec{x}_n\,|\,\lambda)} \tag{4.7}$$

This is the same as the *expectation* step in the EM algorithm. Then needed statistic values are determined:

$$n_k = \sum_{n=1}^{N} P(k\,|\,\vec{x}_n,\lambda)\ , \tag{4.8}$$

$$E_k(\vec{X}) = \frac{1}{n_k}\sum_{n=1}^{N} P(k\,|\,\vec{x}_n,\lambda)\vec{x}_n\ . \tag{4.9}$$

Finally, the adapted mean parameter for mixture k is created:

$$\vec{\mu}_{k,new} = \alpha_k E_k(\vec{X}) + (1 - \alpha_k)\vec{\mu}_k \tag{4.10}$$

with

$$\alpha_k = \frac{n_k}{n_k + r} \; . \tag{4.11}$$

Here, r is a fixed relevance factor that controls the amount of adaptation. A higher value of r leads to a weaker adaptation with only small adaptation changes whereas a small value prioritizes the new data and therewith strongly adapts the model. In the performed experiments a parameter value of $r = 16$ has shown to deliver the best results.

Although the weights and covariance matrices of speaker models could also be adapted by MAP, it makes sense just to adapt the means of the model. Adapting only the mean parameters is not only saving computational cost but it has been shown that speaker recognition performance benefits from this approach [29].

Adapting speaker models is not only used for generating unique speaker models from the UBM but also for improving the speaker models while actually running tests. Since the training material for building up models is limited and may not adequately characterize the range of test conditions, speaker models can be improved by adapting them with already recognized test data. The model adaption also mitigates the effect of changes in channel or speaker condition. However, there is the possibility that mixture models are adapted on imposter utterances which leads to a bad adaptation and decreasing the system performance.

4.5. Classification

The classification of a sequence of feature vectors can be done computationally inexpensive once the speaker-dependent GMMs are generated. Since a GMM is a probability density function, it automatically is defined as likelihood function:

$$\ln p(\vec{X}\,|\,\lambda) = \sum_{n=1}^{N} \ln p(\vec{x}\,|\,\lambda) \tag{4.12}$$

This log-likelihood provides a score measuring the match between analyzed speech data and speaker models. The best matching speaker model is picked and the test segment is labeled with the associated speaker name. This method is called *maximum likelihood* (ML) classification

$$\Theta^{ML} = \arg\max_{\Theta}(\ln p(\vec{X}\,|\,\lambda_\Theta)) \; . \tag{4.13}$$

This chapter defined all the tools needed to use a speaker recognition system in a teleconference application. The input signals are preprocessed, features are extracted and models built up by adapting from a UBM using MAP adaptation. A speaker label is assigned to an input speech segment by evaluating the maximum likelihood. The assignment of a speech stream to a certain output channel of the teleconference system is made in according to the assigned label.

5. Experiments & Conclusion

Experiments were carried out to evaluate the performance of the implemented online speaker recognition system. In the first experiment the performance of the speaker recognition is measured on a widely used database of meeting recordings. In the second experiment the performance of the speaker recognition module as part of the immersive teleconference system is evaluated. It shows the performance of the channel assignment of different speakers carried out by the speaker recognition module.

5.1. Speaker recognition experiments

To perform experiments in speaker recognition a database of speech recordings is needed. The first speaker recognition experiment in this thesis is carried out on the AMI meeting corpus.

5.1.1. AMI Corpus of meeting recordings

The AMI corpus is a huge database of meeting recordings established by the AMI Consortium [1]. The whole corpus consists of more than 100 hours of meeting recordings, carried out in different meeting rooms under varying acoustic circumstances. The data available comprises audio and video recordings together with associated protocols and ground truth of spoken words, active speakers, written notes, and even gestures.

For the experiments the meetings of the so-called Edinburgh scenario are used. This data set consists of 15 meeting settings ("ES2002-ES2016") with four participants in each meeting. Every meeting itself is divided into four parts (a,b,c,d) with a duration of approximately 30 minutes each. This results in 30 hours of meeting recordings. The meetings are conducted like real-world meetings, containing background noise, disturbances like slamming doors, laughing and coughing, and regions with overlapping active speakers.

Every participant is equipped with a headset microphone. The audio data used for the experiments is a mixed mono signal summing up all four mono headset streams. The sampling frequency of the audio data is 16 kHz. The ground truth was automatically created by a voice activity detection on every headset signal, resulting in a table of time intervals and the name of the speaker talking in this segment.

For the experiments the database was split up into training data and test data. Meetings ES2002a - ES2010a are used as training data, ES2011d - ES2016d are used as test data. By this, the speakers of the test data are completely unknown to the system beforehand.

They are also not used for building up the UBM, so this challenge is as close to a real-world application as possible. In addition, only very short excerpts of ES2011c - ES2016c were used to adapt the UBM to the speaker-specific models.

5.1.2. Results

The classification is carried out as a five-class problem, meaning that the classifier decides between one of the four speakers and the UBM. This way we generate an open-set speaker identification, if a speech segment is classified as UBM, it should optimally not belong to one of the four speakers and should rather be noise or distortion.

In the experiments on the AMI corpus the focus is laid on two important parameters:

- The number of Gaussian mixture components for speaker modeling
- The length of a speech segment analyzed

In addition, it is examined whether the continuous adaptation of models benefits the speaker recognition.

Table 5.1 shows the confusion matrix for the classification on meeting 2014d using 64 mixture components and a segment length of two seconds. It can be seen that speaker A is very present in this meeting since a lot of segments are labeled with speaker A. This leads to the assumption that speaker A maybe is the presenter or moderator in this meeting. In addition, it is interesting to check the false detections. Segments of speaker D are often classified as speaker A. The reason could be a high similarity in the voice sound of both speakers, since both speakers are male. It is also possible that speaker A often talks at the same time as speaker D, resulting in overlapping speech and a lot of false classifications.

The confusion matrix has a higher number of segments in the upper right triangle, meaning that there are more segments labeled with a speaker name than segments with an active speaker exist. The system over-classifies on the speakers which also can be seen by taking a closer look at the segments of the UBM: There are more UBM segments classified as a speaker than speaker segments classified as UBM. This leads in terms of the Diarization Error Rate (DER) to less miss-errors but therefore more false-alarm errors.

	Speaker A	**Speaker B**	**Speaker C**	**Speaker D**	**UBM**
Classified A	593	20	19	84	18
Classified B	10	200	6	45	7
Classified C	5	3	113	29	13
Classified D	5	10	3	205	11
Classified UBM	2	5	9	12	27

Table 5.1.: Confusion matrix for classification of meeting 2014d using 64 mixture components and a fixed segment length of two seconds

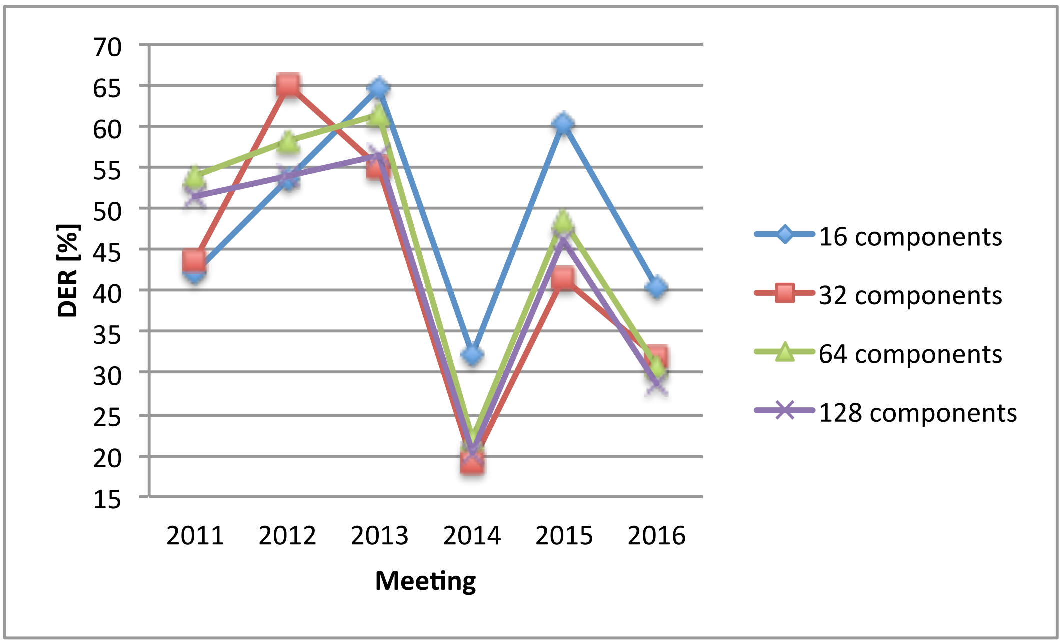

Figure 5.1.: Results of the DER calculation on the test meetings for different numbers of Gaussian mixture components, without continuous adaptation.

For this classification, the accuracy is

$$Accuracy = \frac{593+200+113+205+27}{1454} = 78,27\,\% \tag{5.1}$$

Figure 5.1 shows the results on all test meetings. Continuous adaptation while recognizing has not been performed for these results. As can be seen in the figure, there is a huge difference between the meetings, getting DER as low as 20 % in meeting ES2014 up to 65 % for meetings ES2012 and ES2013. The reason could be differences in quality of speech or bad chosen training data as well as meeting participants that are hard to distinguish by the sound of their voice. The recognition works remarkably well in meeting ES2014.

A difference can be seen between the recognition rate of models with a small number of components compared to the models with a high number of mixture components. The model with only 16 components has a higher error rate in most of the meetings. The more components, the more robust it seems to get. 32 components already achieve great results but seem not to work so well in all meetings. The difference between 64 and 128 components is very small and so it should be considered if 128 mixture components are really worth the higher computational cost.

Figure 5.2 displays the same experiments on the same meetings, but this time with the use of a continuous adaptation of the speaker models while recognizing is performed. As

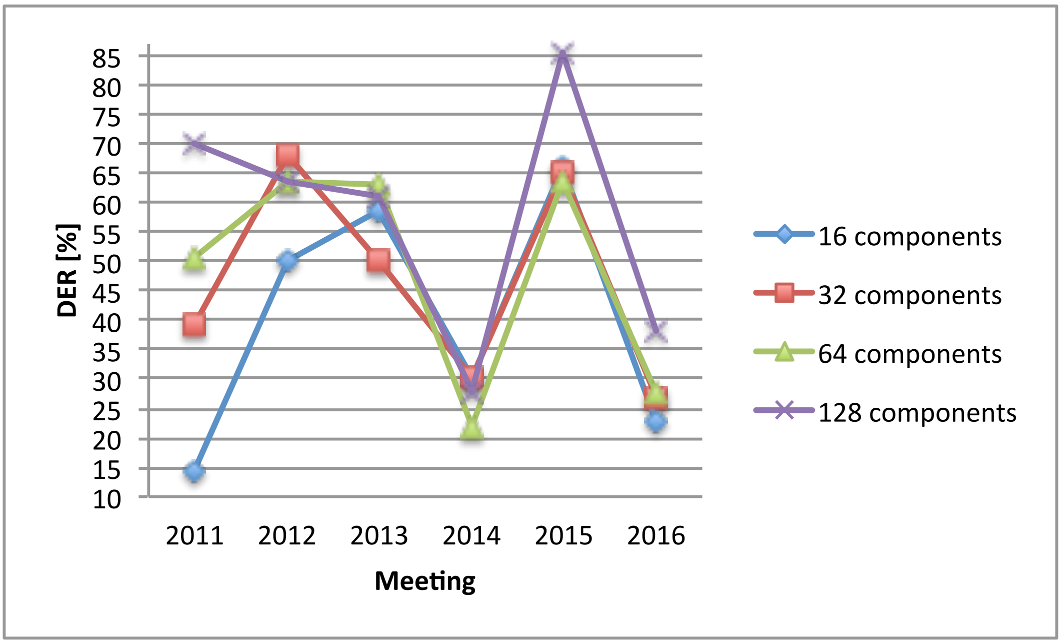

Figure 5.2.: Results of the DER calculation on the test meetings for different numbers of Gaussian mixture components, with continuous adaptation.

soon as a segment is labeled with a speaker name, the associated model is adapted on the speech segment data. Although Figure 5.2 tends to picture it differently, the adaptation slightly improves overall recognition rate. However, as can be seen in the Figure, the results are less consistent. A system performing adaptation all the time seems to tend to one out of two extremes very easily: Whether one speaker is heavily active in a meeting, then he is often recognized, the model is improved and he gets even more recognized. This significantly enhances recognition rate, like in meeting ES2011 or ES2016. Or, as can be seen especially in meeting ES2015, has a lot of false detections which leads to false adaptation which is desastrous for the speaker models.

Figure 5.3 shows the average DER on all test meetings for the different numbers of mixture components. It can be seen that at least 32 mixture components should be chosen to model speakers.

Finally, Figure 5.4 shows the average DER on all test meetings for different speech segment lengths. It can be seen very clearly that a segment length below 1 second is not recommendable for speaker recognition. This is because in this short time interval, it is not enough test data gathered to reliably determine the speaker.

But also for a segment length greater than 2 seconds, recognition performance decreases. This is because a speaker can easily speak a few words in a time interval shorter

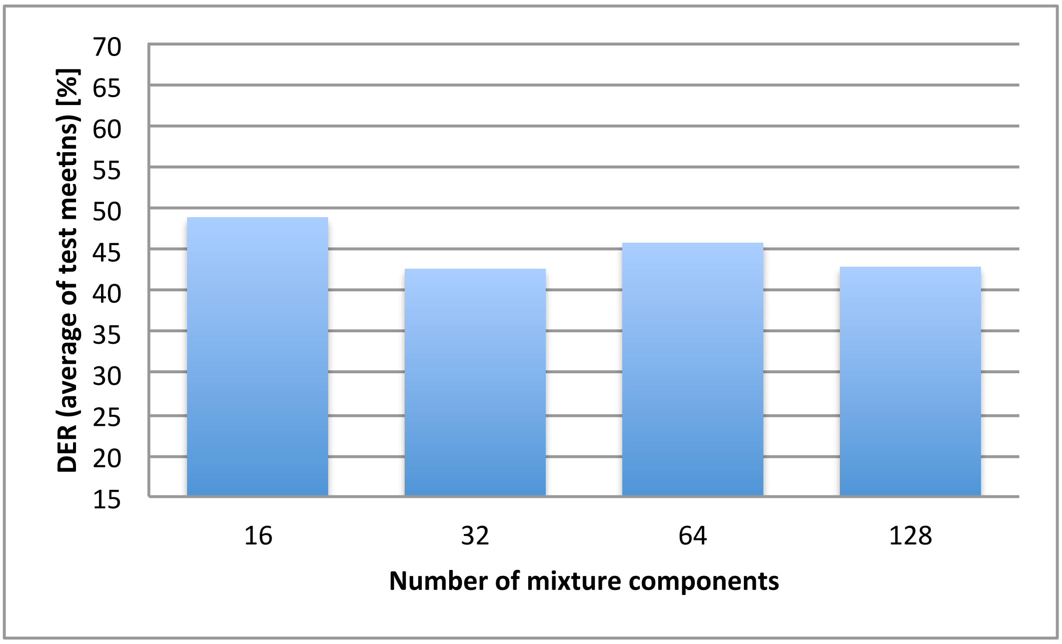

Figure 5.3.: Average DER of all test meetings for different numbers of Gaussian mixture components.

than three seconds. Also, speaker changes are detected very late in the worst case. This leads to a false labeling of the segment.

Altogether, the experiments on the AMI meeting corpus show that speaker recognition is possible for conference situations. The experiments also raise the awareness that parameters like the number of components or the segment length have to be chosen very carefully because they have a high impact on the recognition performance.

5.2. Application in the teleconferencing system

5.2.1. Database

The database for this experiment is a collection of podcasts from the German radio station SWR [34]. The podcast database contains 107 podcast recordings of 21 different speakers with an average length of 3 minutes. The choice was made for this database to have enough speech material of single speakers to split the sound files into training and test data. The radio podcasts are recorded professionally with a minimum of noise or echo. This way, they are appropriate for this experiment.

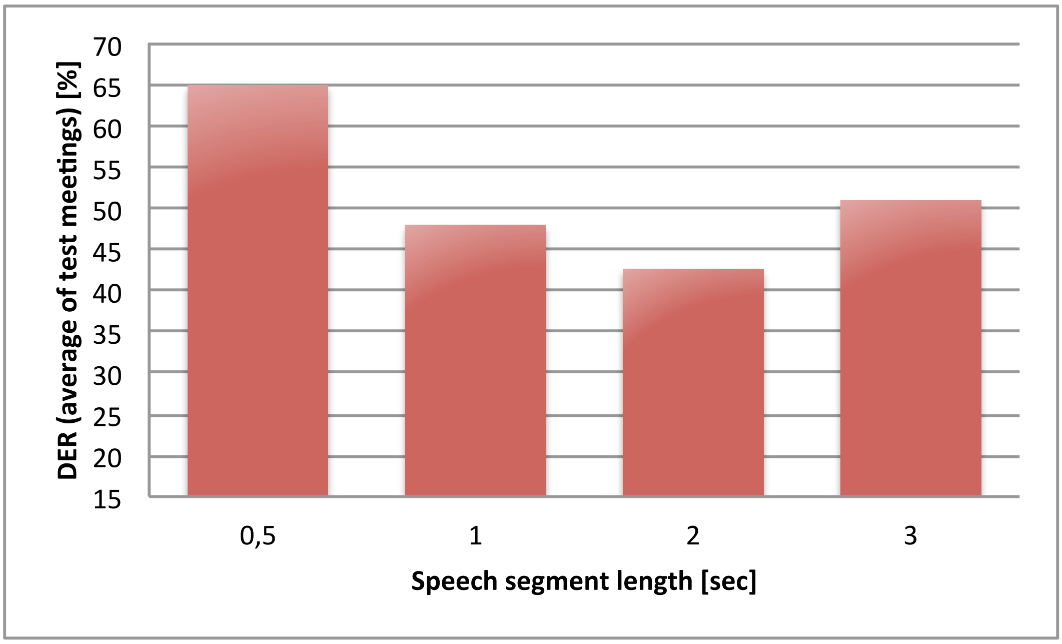

Figure 5.4.: Average DER of all test meetings for different segment lengths.

5.2.2. Experimental setting

In the following, the experimental settings are described in detail. Recordings are performed in a real-world setting inside an office room. The circular microphone array, as shown in chapter 2, consists of eight microphones and is placed on a desk in the office room. Four loudspeakers are placed around the microphone array with a distance of 1.15 m.

A dialogue of four speakers, two female and two male speakers, has been synthetically created out of the radio podcasts. This dialogue is set to simulate different conference situations with multiple speakers active at the same time. This dialogue is played back with each speaker on its assigned loudspeaker. The dialogue is constructed to have parts with overlapping speech sources as well as sequences, where only one speech source is active. The speaker models are adapted in the style of a round of introductions, which is common in teleconference situations: 10 s speech of each individual speaker is played back on the associated loudspeakers. A UBM was created beforehand, modelling all speakers of the podcast database. The data used to train the GMM or adapt the speaker models in this round of introductions is not part of the following diarization task.

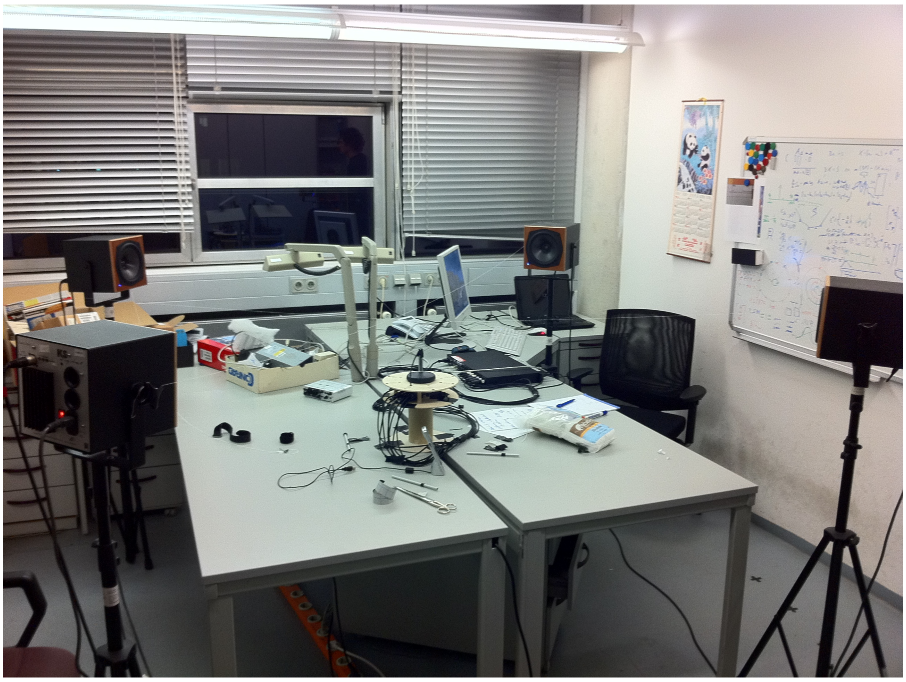

Figure 5.5.: Recordings were performed in a real-world setting inside an office room

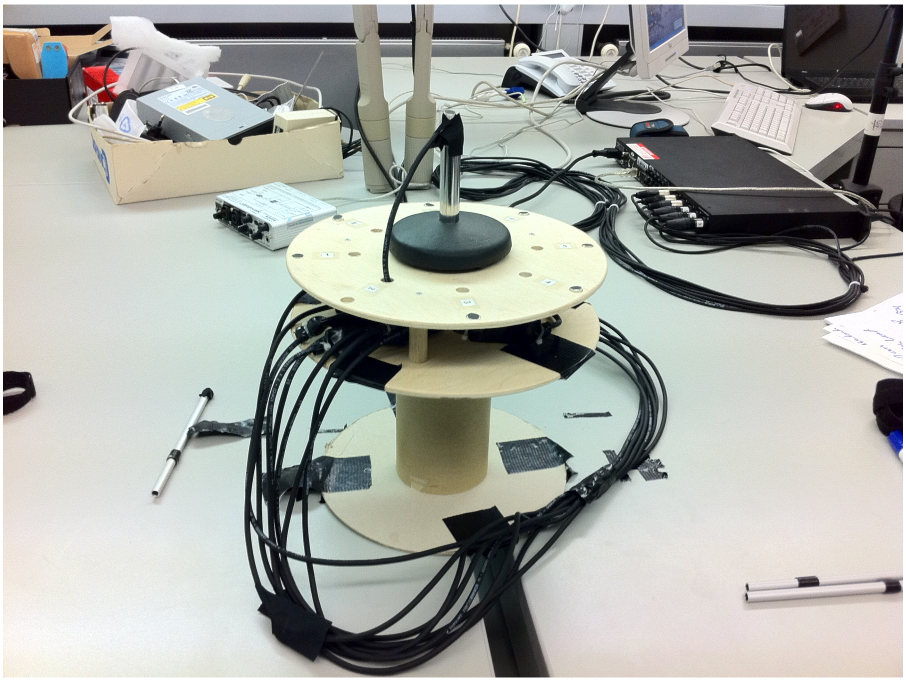

Figure 5.6.: The circular microphone array

5.2.3. Experimental results

Three experiments are conducted. In the first experiment, the sound source localization and separation algorithm of our teleconferencing system is applied to localize the active speech sources. In the second experiment, stand alone implementation of our online speaker recognition system is used to recognize active speakers and assign them to their individual audio channel. Finally, in the third experiment, joint sound localization and separation is utilized in combination with the speaker recognition.

Figure 5.7a) illustrates a 55s sequence of a discussion of two male and two female conference participants. Looking at the thin lines with the waveforms attached to it, the first active speech source is located at 225 degrees azimuth. Then a second speech source, located at 45 degrees, starts talking. It is worth noticing that there is a small overlap between the first two speakers. The solid blue and green markers in Figure 5.7a) depict the estimates of the localization and separation algorithm. It can be seen, that the implementation of the localization and separation algorithm detects the active sound sources, even if sources overlap. Due to the reverberant office environment, the algorithm also localizes echoes, which can be seen at time instances 18 s and 40 s respectively. The output of the localization and separation system in this case is composed of two audio streams. Each stream consists of one of the active speech sources. The overlapping parts are separated by a Geometric Source Separation (GSS) algorithm [11]. However, as seen in Figure 5.7a) the green and blue markers sometimes switch channels which means that the localization and separation algorithm is not able to ensure a fixed channel-speaker-alignment.

To overcome the problem of channel-speaker-alignment, in the second experiment stand alone speaker recognition is applied on the recordings. As shown in Figure 5.7b), the speaker recognition module reliably recognises the respective speech source, denoted by A,B,C,D in the Figure, in case of non-overlapping speech signals. If speakers overlap, the recognition algorithm has to decide for one source, leading to distorted audio information on the channels and a loss of speech information in the aligned audio streams. For example at instance 10 s the speech signal is switched instantaneously from A to C, possibly interrupting speech of speaker A. This experiment shows that speaker recognition is suitable for assigning speakers to individual channel, but in case of overlap errors will always occur.

In the last part of this experiment, performance of a joint system is investigated. According to 5.7c), the system is able to properly assign two overlapping speech sources to the corresponding audio channels. Compared to the second experiment, there is no loss of speech information in case of two simultaneously talking conference participants. Furthermore, the joint system is able to correctly assign echoes due to the fact, that also the active speaker within the echo signal is correctly recognized and assigned to the proper channel. This can be seen in Figure 5.7a) at time instance 18 s. False assignments, illustrated in Figure 5.7c) at time instance 39 s, hardly influence the perceived conference quality due to the low signal energy.

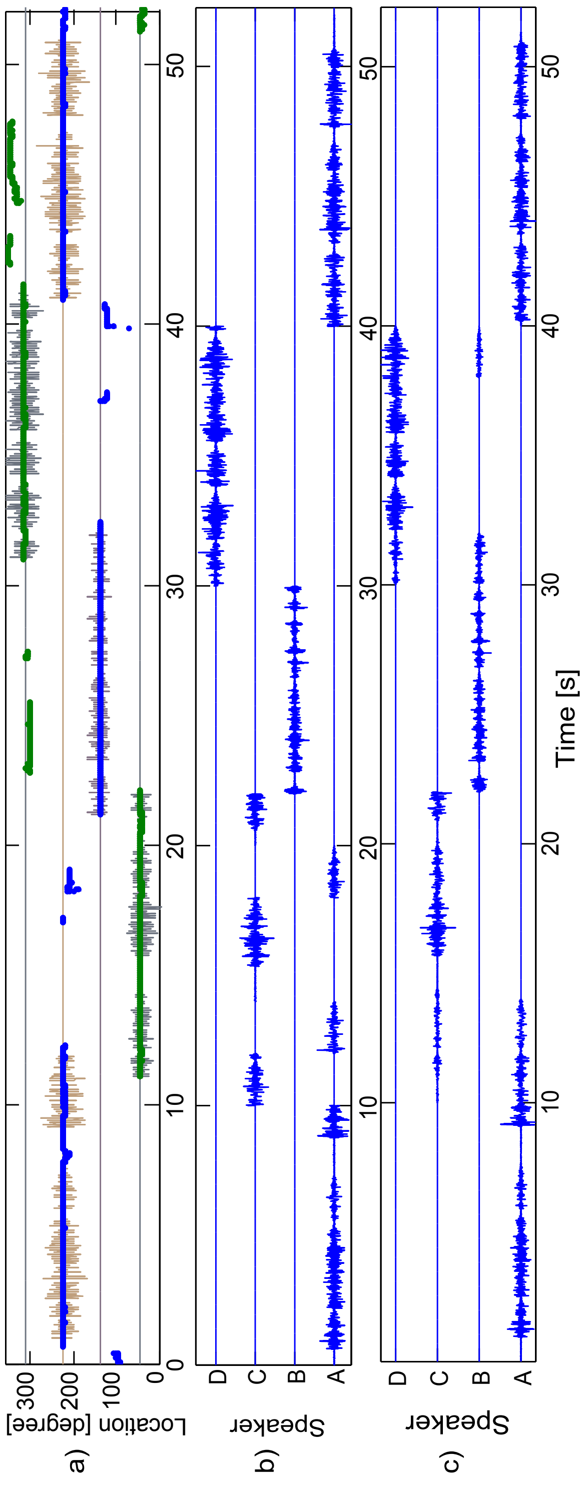

Figure 5.7.: Experimental results of the joint sound localization, separation and speaker recognition

6. Conclusion & Outlook

6.1. Conclusion

The aim of this thesis is to implement and evaluate an online speaker recognition for immersive teleconferencing systems. The experiments and analyses of the previous chapters yield the following conclusions:

The first experiment measured the overall quality of the implemented approach. An accuracy of 78 % was achieved on the AMI meeting corpus. It is shown that certain parameters of the system massively influence the outcome. For example, the number of Gaussian mixtures not only defines how computationally expensive the calculation is but needs to be chosen carefully in order to correctly represent the speakers. The length of the analyzed speech segment needs to be high enough to gather enough information to reliably recognize the speaker but also needs to be low in order to not add even more latency to the system. The results on the AMI corpus show the difficulties of the speaker recognition task, always in danger of over-adaptation, bad speaker modeling and even worse input data.

However, taking the real-world constraints and requirements for an online system into consideration, the implemented speaker recognition system performs very well and fulfils all requirements. Moreover, the system has proven to give reliable results in a live demonstration at the *Institute for Data Processing* and shows it's real strength in the second experiment.

In the second experiment, the performance of assigning different speakers to individual audio channels by using sound source localization, separation and online speaker recognition is evaluated. The experiments demonstrate that the system is able to assign different speakers to individual audio channels, even if the speech sources are overlapping. Furthermore, localization performance is improved by assigning falsely detected echoes to the appropriate audio channels of the respective speech sources. Altogether, this experiment can be seen as success. The online speaker recognition module adds an important feature to the whole system and in addition helps improving the performance of connected modules.

In this particular application, using a test set consisting of multiple speakers that are sometimes simultaneously active and overlapping, the speaker recognition accuracy is high when combined with a sound source localization and separation. This is an improvement of solo speaker recognition, where a loss of 20 % of the actual signals is poduced by assigning to the wrong channel. The aim of improving both localization/separation performance and speaker recognition by creating a joint system is achieved.

Gaussian Mixture Models were chosen to model the individual voices of speakers. In the experiments they prove to be the right choice for an online speaker recognition, reliably modeling speakers and having a computationally inexpensive measurement score to quickly evaluate test utterances. MFCCs are chosen to represent spectral attributes of different sounding voices and the experiments confirm this choice. The adaptation of the UBM helps to get individual speaker models in a fast and efficient way. However, the effect of adaptation on just recognized data is not as high as expected which is founded in the vulnerability to false detections.

The experiments show that this system approach is capable of performing online recognition.

6.2. Outlook

Although the system performed well in the carried out experiments, there is still some space to boost performance, refine parameters and algorithms and of course take the online speaker recognition system one step closer to the deployment in a real teleconferencing system. A few improvement approaches and ideas are shared in this section.

The system introduced in this thesis only uses fixed-length segments. This leads to inaccuracies in detecting speakers, especially when active speakers overlap. This problem could be resolved by implementing a dynamic segment length, automatically checking for the start and end of a segment and immediately starting a new segment as soon as a speaker transition is detected.

There is also the possibility that mixture models are adapted on imposter utterances by unsupervised MAP adaptation. it would be extremely helpful to introduce a way of implementing a supervised adaptation system into the teleconference application. This would make sure that the system really improves reliability and recognition rates over time.

There is plenty of space to improve the UBM. It would be worth thinking of gender-specific UBMs since the difference in voice frequency can make a big difference. It is also worth evaluating UBMs for noise, disturbances and non-speech sounds. This so-called garbage recognition can improve voice activity detection and overall recognition rate since noises are classified as such.

Finally, it would be interesting to incorporate video and the localization data into the speaker recognition. Speakers would not only detected by their voices but also if their mouth is moving in the video. Also, a speech signal getting located at 45 degrees and a correct recognized speaker localized at 45 degrees a few segments beforehand will most likely to the decision that it is still the same speaker on 45 degrees. So spatial location also plays a role and could be used as feature or influence the likelihood decision.

It is not only interesting to see how the implemented system could be improved but also how it could be used in future to improve the performance of other systems. The most prominent example is the speaker-dependent speech recognition where speech recognition rates can be dramatically improved by also training the models on speaker-dependent features. Errors and ambiguities in speech recognition transcripts can be corrected using the knowledge provided by speaker segmentation assigning the segments to the correct speakers.

Altogether, there is no doubt that immersive teleconferencing systems will replace conventional phones and conferencing solutions over time and that speaker recognition is an essential part of it.

A. Appendix

A.1. Audio Processing Parameters

Table A.1.: Parameters of audio processing

General Audio Processing Parameters	
Sampling frequency	16 kHz
Quantization	16 bit
FFT length	20 ms
Window overlap	10 ms
Window type	Hamming
Feature Extraction Parameters	
Number of features used	39
Preemphasis roll-off α	0.95
Number of triangular mel filters	20
Voice Activity Detection Parameters	
Energy threshold	30 dB
Number of frames discarded	about 25 %
Minimum number of speech frames to count a segment as speech	3
Maximum number of silence frames to end a segment	8
Gaussian Mixture Models Parameters	
Number of mixture components	32
EM convergence threshold	10^{-5}
MAP relevance factor r	16

A.2. AMI Corpus

Meeting	s001	s002	s003	s004
ES2002	m0007	m0006	m0008	f0005
ES2003	m0011	m0009	m0012	m0010
ES2004	m3015	f0013	f0016	m0014
ES2005	m0018	f0019	m3020	m0017
ES2006	f0024	m3022	f3023	f0021
ES2007	m0025	f3026	m0027	f0028
ES2008	f0029	f0030	f0032	m0031
ES2009	m0034	m0033	m0035	f0036
ES2010	f0037	f0038	f0039	f0040
ES2011	f0043	f0041	f0044	f0042
ES2012	m0045	f0047	f0046	m0048
ES2013	f0049	f0050	f0051	f0052
ES2014	m0053	m0054	f0055	m0056
ES2015	f0057	f0060	f0058	f0059
ES2016	m0061	f0064	m3062	m0063

Table A.2.: Overview of meeting participants per meeting with annotated gender

A.3. MATLAB Implementation

A digital copy of the Matlab source code is provided alongside the thesis. The attached DVD contains the following MATLAB scripts and functions:

Table A.3.: Matlab scripts

Scripts	
adaptUBM.m	adapt GMMs from a UBM
live.m	Live version of the recognizer
main.m	Main script for simple recognition tests
main2.m	Spare version
main_online.m	main function for online recognition
PROPERTIES.m	Defines all important parameters centrally
start.m	main script to start recognition
startSingle.m	Used for a single GMM
trainEval.m	Spare trainModel
trainModelOnline.m	trainModels for live usage
trainModels.m	Train models with EM
trainModels2.m	Spare version
trainUBM.m	Train a UBM

Table A.4.: Matlab functions and tools

Functions	
adaptModel.m	Adapt a model
EM.m	Implementation of the EM algorithm
enframe.m	window a signal
extractFeatures.m	extract features out of a signal
initEM.m	initialize EM algorithm by k-means
logLikelihood.m	Calculate log-likelihood
map.m	MAP adaptation
melcepst.m	Calculate the MFCCs
trainGMM.m	Train a GMM
vad.m	Voice activity detection
vad_old.m	Old version with a different approach
Tools	
activlev.m	estimates the active speech level
estnoisem.m	estimates the ground noise level
frq2mel.m	Transforms linear frequency into mel scale
gaussmix.m	Another adaptation algorithm
gaussPDF.m	Computes PDF of a Gaussian
lmultigauss.m	Computes multigaussian log-likelihood
logsum.m	log(sum(exp()))
lsum.m	Sum up logarithmically
m2htmlpwd.m	Creates a HTML documentation of the current folder
maxfilt.m	Find max of a filter
mel2frq.m	Transforms mel scale to linear frequency
melbankm.m	Mel bank filter function
nearnonz.m	Create a value close to zero
rdct.m	Calculate DCT of real data
rfft.m	Calculate DFT of real data

A.4. DVD content

The attached DVD contains, besides the MATLAB functions described in the previous section, the complete databases and ground truths that were used for carrying out the experiments of this thesis.

- The Edinburgh recordings of the AMI meeting corpus together with the ground truth for every meeting, provided as Microsoft Excel file in the folder "amicorpus",
- training extracts of some meeting recordings of the AMI corpus where just one speaker is active in the folder "training data",
- The podcast database containing 107 podcast recordings of 21 different speakers in the folder "podcast",
- The test file Kozfeld.wav for evaluating joint localization, separation and speaker recognition performance, together with the ground truth as Microsoft Excel file in the folder "Kozfeld"

List of Figures

Bibliography

[1] AMI Consortium. The AMI Meeting Corpus. URL `http://corpus.amiproject.org`. Accessed at 08.09.2011.

[2] S. Araki, M. Fujimoto, K. Ishizuka, H. Sawada, and S. Makino. A DOA based speaker diarization system for real meetings. In *Hands-Free Speech Communication and Microphone Arrays*, pp. 29–32. 2008.

[3] J. Benesty, M. Sondhi, and Y. Huang. *Springer Handbook of Speech Processing*. Springer, New York, NY, 2008.

[4] C.P. Browman and L. Goldstein. Articulatory phonology: an overview. In *Phonetica*, 49(3-4), pp. 155–180, 1992.

[5] C. Busso, P. Georgiou, and S. Narayanan. Real-time monitoring of participants' interaction in a meeting using audio-visual sensors. In *International Conference on Acoustics, Speech and Signal Processing*, volume 2, pp. II–685. 2007.

[6] W.M. Campbell, J.P. Campbell, D.A. Reynolds, E. Singer, and P.A. Torres-carrasquillo. Support vector machines for speaker and language recognition. In *Computer Speech and Language*, 20, pp. 210–229, 2006.

[7] E. Cherry. Some experiments on the recognition of speech, with one and with two ears. In *Journal of the Acoustical Society of America*, 25, pp. 975–979, 1953.

[8] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. In *Journal of the Royal Statistical Society, Series B*, 39(1), pp. 1–38, 1977.

[9] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, 1973.

[10] K.R. Farrell, R.J. Mammone, and K.T. Assaleh. Speaker recognition using neural networks and conventional classifiers. In *IEEE Transactions On Speech And Audio Processing*, 2(1), p. 194–205, 1994.

[11] J. Feldmaier. Sound Localization and Separation for Teleconferencing Systems. 2011. Diploma thesis at the *Institute for Data Processing, Technical University of Munich*.

[12] G. Friedland and O. Vinyals. Live speaker identification in conversations. In *Proceeding of the 16th ACM international conference on Multimedia*, MM '08, pp. 1017–1018. 2008.

[13] Gartner, Inc. Dataquest insight: Videoconferencing products and services market forecast, worldwide, 2007-2013. 2009.

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

[15] H. Gish and M. Schmidt. Text-independent speaker identification. In *IEEE Signal Processing Magazine*, 11(4), pp. 18–32, 1994.

[16] H. Gish, M.H. Siu, and R. Rohlicek. Segregation of speakers for speech recognition and speaker identification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 873–876. 1991.

[17] J. Hartigan and M. Wong. A K-means clustering algorithm. In *Applied Statistics*, 28, pp. 100–108, 1979.

[18] Y. Huang, J. Chen, and J. Benesty. *Acoustic MIMO signal processing*. Springer, New York, NY, 2006.

[19] Y. Huang, J. Chen, and J. Benesty. Immersive Audio Schemes. In *IEEE Signal Processing Magazine*, 28(1), pp. 20–32, 2011.

[20] L. Jamieson. Course notes for speech processing by computer. 2007. URL `http://cobweb.ecn.purdue.edu/ee649/notes/`.

[21] S. Johnson. Who spoke when? - automatic segmentation and clustering for determining speaker turns. In *PROC. EUROSPEECH*, 5, pp. 2211–2214, 1999.

[22] M. Kaufmann. Abwärtskompatible SIP Telefonkonferenzserver Software mit räumlichem Höreindruck. 2011. Bachelor thesis at the *Institute for Data Processing, Technical University of Munich*.

[23] K. Otsuka, S. Araki, K. Ishizuka, M. Fujimoto, M. Heinrich, and J. Yamato. A realtime multimodal system for analyzing group meetings by combining face pose tracking and speaker diarization. In *Proceedings of the international conference on Multimodal interfaces*, pp. 257–264. 2008.

[24] J. Pelecanos and S. Sridharan. Feature warping for robust speaker verification. In *Proc. ISCA Workshop on Speaker Recognition*. 2001.

[25] M. Przybocki and A. Martin. The 1999 nist speaker recognition evaluation, using summed two-channel telephone data for speaker detection and speaker tracking. In *EUROSPEECH*. 1999.

[26] L. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[27] D. Reynolds. Experimental evaluation of features for robust speaker identification. In *IEEE Transactions on Speech and Audio Processing*, 2(4), pp. 639–643, 1994.

[28] D. Reynolds. Speaker identification and verification using gaussian mixture speaker models. In *Speech Communication*, 17(1-2), pp. 91–108, 1995.

[29] D. Reynolds, T. Quatieri, and R. Dunn. Speaker verification using adapted gaussian mixture models. In *Digital signal processing*, 10(1-3), pp. 19–41, 2000.

[30] D. Reynolds and R. Rose. Text independent speaker identification using automatic acoustic segmentation. In *International Conference on Acoustics, Speech and Signal Processing*, volume 1, pp. 293–296. 1990.

[31] M. Rothbucher, J. Feldmaier, C. Kozielski, T. Habigt, M. Durkovic, and K. Diepold. Joint sound localization, sound separation and online speaker recognition. In *International Conference on Acoustics, Speech and Signal Processing*. 2011. Submitted for review.

[32] M. Rothbucher, T. Habigt, J. Feldmaier, and K. Diepold. Integrating a HRTF-based sound synthesis system into Mumble. In *IEEE International Workshop on Multimedia Signal Processing*, pp. 24–28. 2010.

[33] B. Schuller, C. Kozielski, F. Weninger, F. Eyben, and G. Rigoll. Vocalist gender recognition in recorded popular music. In *Proceedings of ISMIR*, pp. 613–618. 2010.

[34] SWR. SWR contra - Das Informationsradio. URL `http://www.swr.de/contra/-/id=7612/nid=7612/did=2056832/1471juo/index.html`. Accessed at 22.09.2011.

[35] O. Thyes, R. Kuhn, P. Nguyen, and J. Junqua. Speaker identification and verification using eigenvoices. In *INTERSPEECH*, pp. 242–245. 2000.

[36] S. Tranter and D. Reynolds. An overview of automatic speaker diarization systems. In *IEEE Transactions On Audio, Speech, and Language processing*, 14(5), pp. 1557–1565, 2006.

[37] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.