Teleconference channel assignment

Korbinian Steierer, Martin Rothbucher,
Klaus Diepold

Technical Report

Teleconference channel assignment

Korbinian Steierer, Martin Rothbucher, Klaus Diepold

June 3, 2013

Lehrstuhl für Datenverarbeitung
Technische Universität München

Korbinian Steierer, Martin Rothbucher, Klaus Diepold. Teleconference channel assignment. . Technische Universität München, LDV. June 3, 2013.

Abstract

The Institute for Data Processing at the Technical University of Munich develops an online teleconference system. At the moment the system consists of separate devices and algorithms, which were implemented in previous works. This includes the recording hardware such as the microphone array and the software for the speaker localisation, separation and recognition. In this thesis I connected the algorithms at the recording side to make channel assignment possible. Channel assignment means to put every speaker at an own audio channel to support the human hearing system at the playback side via 3D-sound.

The other task of this thesis is to evaluate the teleconference system, in special the speaker identification approach. This is done in two ways. The first compares the developed recognition system to an offline speaker diarization algorithm from the ICSI, to have a minimal reachable threshold for the Diarization Error Rate. This is also used to find the best parameters for the recognition task. In the second evaluation simulated and real conferences are tested to achieve a quality statement about the whole teleconference system in view of the channel assignment.

Der Lehrstuhl für Datenverarbeitung der Technischen Universität München entwirft ein online Telekonferenzsystem, das momentan aus den einzelnen Einheiten und Algorithmen besteht, welche in vorherigen Arbeiten entwickelt wurden. Dies beinhaltet die Aufnahmegeräte wie das Mikrophone Array sowie die Software für die Sprecherlokalisierung, Separierung und Erkennung. In dieser Arbeit werden die einzelnen Komponenten der Aufnahmeseite eines Telekonferenzsystems verknüpft um Channel Assignment möglich zu machen. Die Definition von Channel Assignment ist, jeden Sprecher auf einem eigenen Audiokanal zu übertragen, um auf der Abspielseite das menschliche Gehör durch 3D-Sound zu unterstützen.

Die andere Aufgabe dieser Arbeit ist das Telekonferenzsystem, im besonderen die Sprechererkennung, zu evaluieren. Dies wird auf zwei unterschiedlichen Wegen umgesetzt. Der Erste vergleicht die umgesetzte Erkennung mit einem, von der ICSI erhaltenen, offline speaker diarization Algorithmus um eine minimale Schwelle für die Diarization Error Rate zu erhalten. Diese Evaluation wird umgesetzt um die besten Parameter für Sprechererkennung zu finden. Die zweite Testreihe evaluiert das ganze Telekonferenzsystem durch eigene Aufnahmen in Bezug auf das Channel Assignment evaluiert.

Contents

1. Introduction

Teleconference Systems will play a more important role in meetings than they did in the past. This is due to increasing travel expenses. Higher prices for fuel and, of course, the unusable travel time are two points in this calculation. On the other hand there is an environmental aspect. CO_2 is a big problem nowadays and teleconference systems can take a part in reducing the CO_2 emission.

This is accompanied by the lack of innovations in teleconference systems in the past. The last few years show that there is an interest in a bigger innovation width, but most of the new technologies and algorithm have not been fully developed yet. I will show some examples of these approaches in a later chapter of this thesis. Another point is the definition of an evaluation standard, which helps to compare different approaches and shows what an important role teleconference meetings will have in the future. An example for such a standard is [19]. The main problem at the moment is to simulate a real conversation between humans in meetings where the participants are not sitting in the same room. Problems like helpful support in case of multiple people speaking simultaneously has to be solved. At this point I would like to refer to the Cocktail party effect [12]. In this work the author shows, that people can concentrate on one speaker even with noise and other conversations around them. Now the idea is to give participants of meetings the same or similar ways to define the actual speaking person. To make this happen you need algorithms which locate, separate and recognise the actual speaker in order to transport each participant on a separate channel and make 3D-sound possible. The recognition algorithm is needed, too, to give a visual help about the actual speaker. If these three technologies are perfected a conversation similar to real meetings is possible.

In the next chapter I will motivate this thesis and in the following I will define how we will try to solve the problems mentioned above. In the last part of this chapter I will give an outline on the rest of this work.

1.1. Motivation

"Over 80% of knowledge workers communicate on a regular basis with co-workers in other offices and nearly 90% interact with people outside the company in other offices." is said in the article of Debra Chin [13]. In her article it is also said, that a big part of these conversations are made by using the "old ways", like telephone, email and face to face meetings. In my opinion not only the last point can be replaced by a modern, cheap and effective teleconference approach.

Telephone calls are a pretty good way to exchange information between two humans. But very often the information has to be shared with a couple of people and there the classical telephone only has the possibility to call everybody separately or to make a conference call in which it is hard to tell "who is speaking".

If information is exchanged with emails and the mailing list contains many entries discussions can end in an email chaos and the chronology is often no longer reproducible. In a teleconference you do not have this problem, because speech recognition approaches are already showing very good results today [19].

Face to face meetings are only possible if the participants work at the same place. If this is not the case, the above mentioned points of costs, time effort and environmental causes has to be count to the negative effects of face to face meetings.

Nowadays meetings often use headsets, which seems to be difficult, because of the cable there are always limitations in mobility and on the other hand the wireless headset needs a loaded battery which is a growing problem as the number of participants become bigger. The more participants, the bigger the chance of someone having to change the batteries. Without additional technology it is in this case not possible to make 3D-sound. Again, this makes it hard to differ between the speakers. Of course there would be the possibility to show a visual speaker name, but a 3D-sound is faster and easier to interpret for the human hearing apparatus.

A possible solution for the named problems is a video conference. Here all the points are solved, but new issues occur because of the expensive equipment and time effort to make the hardware ready for a meeting. So I think a better solution should be found.

In my opinion new approaches in teleconference systems can be used to find answers for the named issues. The audio signal must be recorded by a microphone array. Now every speaker can be separated, located and recognized. Through this it is possible to transmit every speaker in its own audio channel and generate 3D-sound. So participants on the remote side can distinguish between the speakers because of the sound source location. A speaker recognition system is here helpful, too. Another point is that a speaker recognition system can label the spoken words with a name through which a diarization is possible. In my opinion such a teleconference system can help much in the future of meetings.

1.2. Definition of this thesis

The first point of this thesis is to combine the separate devices of the teleconference system. This should happen in a manner, that a stable audio output with high quality is generated. The delay has to be kept as short as possible without a loss of performance.

The second part is to evaluate the speaker recognition and the whole teleconference system from the Institute for Data Processing at the Technical University of Munich. For the first evaluation a speaker diarization approach should be found and implemented if it is necessary, what of course needs a big time investment in reading about related works. After this an evaluation criterion has to be found, so that the speaker recognition and diarization approach can be significantly compared. For that matching audio files are needed. We decided to compare our speaker recognition approach with an offline one, because we think that a state of the art speaker diarization is the line we nowadays can maximally reach with an online implementation. I will point here out again, that this test procedure does not include the whole teleconference system. It only evaluates the speaker recognition system.

The second evaluation should bring results about the whole system. We have three different speaker localisation and separation approaches implemented. Now we want to test how good the evaluation winner of [21] works with the speaker recognition system. For that conference recordings have to be simulated to evaluate the combined teleconference system in sight to the channel assignment. A point of this thesis is to define an evaluation criterion, too. The evaluation should contain the possibility to interpret our results in the competition to other state of the art approaches.

So the last task of this work is to find the state of the art in speaker recognition systems and compare our work with them.

1.3. Outline of this thesis

In chapter 2 an overview of other speaker recognition systems, and approaches to improve such systems will be given. Also complete state of the art teleconference systems and channel assignment approaches will be explained. In chapter 3 of this thesis I will introduce the teleconference system from the Institute for Data Processing. The microphone array, the preprocessing steps, the speaker localisation, separation and recognition approaches will be introduced here. The channel assignment algorithms developed in this thesis are discussed, too. The next section 3.7 will introduce the speaker diarization implementation we chose and the evaluation ideas we had, too. In chapter 4 the results of the evaluation will be shown. The last part will give a conclusion and a look into the future of this teleconference systems.

2. Related Works

Speaker recognition, separation and the localisation will play an important role in a lot of future technologies. For example a Smart Home without a good speaker recognition and localisation is impossible. A Smart Home has to know who gave the command to react in a learned manner. The other point is the localisation, which is needed to define where something, like opening a door, should happen. The robotic field is another example where through recognition and localisation a good simulated human to human conversation is possible, because the head of the robot adjusts to the speaker and reacts again individually specific. The last example which is mentioned here is the psychology. In the future much more evaluation of speaker behaviour can be made if the whole process works completely automatic. Simple things like "who spoke how often" and "which words were said" till more difficult statements, like "what emotions does the person feel", can be made. But again, a reliable speaker recognition is needed.

In this chapter an overview about different speaker recognition systems and some interesting parts of them is given. Here the relevant steps of speaker recognition will be defined and in every step some similar algorithms will be shown. Additionally a brief introduction in speaker localisation and separation algorithms will be given. Then whole systems, that are related to the one introduced in chapter 3, should be explained. After that some relevant algorithms or systems to *channel assignment* and *speaker localisation in acquisition to speaker recognition* will be presented. The last point in this chapter will introduce important evaluation standards and their adaptation to our case.

2.1. Overview

Speaker identification systems can be divided into different approaches. The most meaningful is the distinction between an online and an offline approach. In the first case the whole computation of the meeting should happen in nearly real time conditions. That is called speaker recognition. In the second case the audio file from a meeting is used to identify "‘who spoke when"’ in an offline scenario. So no real time conditions have to be considered. The name of this is speaker diarization. In this chapter the main attention lies in the online approaches.

A second differentiator are the text-dependent and the text-independent speaker identification [5]. For example a text-dependent recognition is often used in an authentication process with a keyword, also a known word or text. In the thesis of Homayoon Beigi [4] a third model is named, the text prompted speaker identification. This means that the text is known beforehand by the system, but not by the speaker. In this thesis only the text-independent case is important. This means that it is beforehand not known what the speaker will say and therefore it is necessary to use speaker dependent features to identify the person who speakes. The text-independent approaches will only be observed in this chapter.

Another difference are techniques which are used for meetings. It is possible to identify the speaker with the help of video streams or with a combination of video and audio, or like in this work, identification is only computed through audio. Because of that this chapter will concentrate on audio only approaches.

At this point the more general components, then in section 3.5.3, of a speaker recognition system will be introduced. The principle of a speaker recognition system can be divided into two steps:

1. Training step
2. Recognition step

The training phase is required to get features that are typical for a speaker. This is done beforehand the meeting in an, for example, introduction round. After transforming that training features into a speaker model the recognition of a speaker is possible through comparing the speaker attributes of a meeting against the model created in the training phase. By using an Universal Background Model (UBM) the speaker models get more robust. An UBM is a collection of a lot of speaker data from a big amount of meetings.

In the next section the localisation and separation approaches, developed by the Institute for Data Processing are introduced. Afterwards an overview about the different speaker dependent features will be given. In the next part some different speaker models will be introduced and in the following Voice Activity Detection approaches and compensation as normalization methods will be shown. Channel assignment is the topic of the next section and the last one is about other teleconference approaches.

2.2. Speaker localisation and separation

After the recording the first processing steps include the localisation and separation of the actual speaker. To have a choice between different algorithms and their quality more implementations are made [16], [55] and [15]. A good conclusion about the quality of these approaches is given in [21]. For the localisation of the speaker, two ways are possible. The first is called Steered Response Power - Phase Transformation (SRP-PHAT) in the combination with a particle filter and the second one uses a binaural algorithm. Both algorithms deliver a position for the actual speaker. After that it is tried to separate the actual speaker from noise and other simultaneously speaking meeting members. The eight channels, one from every microphone, are reduced to the same amount of channels as the number of active speakers. If there are more than one speakers active, every channel contains speech but in every channel only one speaker is the dominant. At the Institute for Data Processing following algorithms were implemented:

- Blind Source Separation (BSS) [16]
- Geometric Source Separation (GSS) [55]
- Binary Masking [15]

In this section a introduction of ever localisation and separation algorithm will be given.

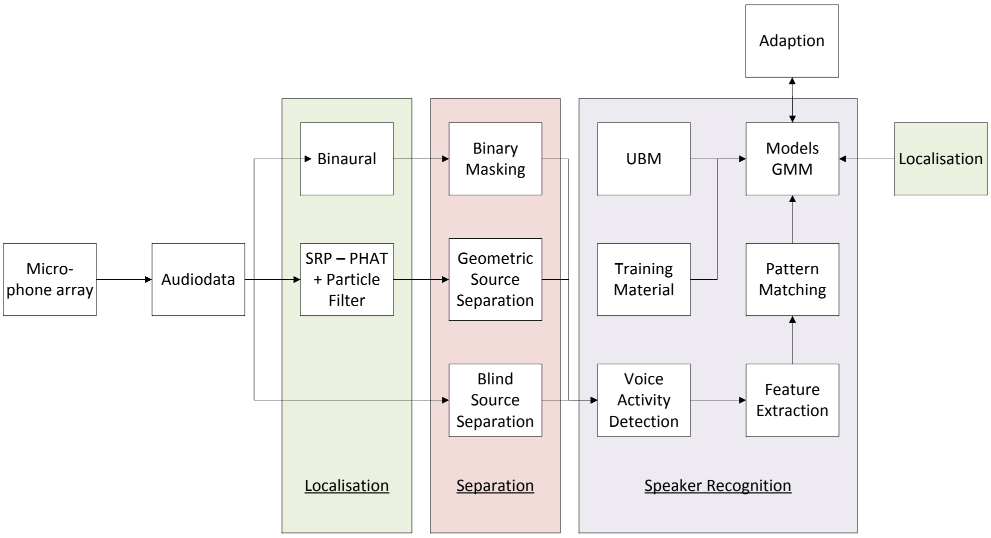

Figure 2.1.: The teleconference system from the Institute for Data Processing of the Technical University at Munich. In the localisation and separation column only the best algorithm is used.

2.2.1. Localisation

In figure 2.1 it can be seen that at our institute two localisation approaches are implemented. The first is the so called binaural localisation [15] which has been adapted in [44] from the robot field to the teleconference scenario with eight microphones. The idea behind this algorithm is to compare a speech utterance with the in advance measured impulse responses to locate the source. The second localisation algorithm is the so called Steered Response Power - Phase Transform (SRP-PHAT) [16]. This approach uses the geometric information from a beamformer and the phase difference from the microphone pairs to locate a source. A particle filter makes the algorithm more time stable trough source tracking.

2.2.2. Separation

In this section the separation algorithms are explained. The separation is first of all needed to separate the speakers out of a mixture of noise and other talking persons. For humans this is not a problem but the mathematical definition of this issue is a challenge.

In this section a short introduction to every algorithm will be given.

Blind Source Separation (BSS) is the task to calculate out of mixture the separated components and that without the knowledge about the source or the mixing process [36]. The developed algorithm is gathered from the work [55]. The idea is to use the statistical independence between different sources and try to maximize it, because it can be assumed that

- The components of different sources within one frequency bin are independent
- The components of one source over all frequency bins are dependent.

This problem is defined in equation 3.2 and has to be solved. One approach is the Independent Component Analysis. This algorithm tries to calculate

$$\hat{\mathbf{s}}(t) = \mathbf{W} \cdot \mathbf{x}(t) = \mathbf{A}^{-1} \cdot \mathbf{x}(t). \tag{2.1}$$

A flow diagram of the approach shows figure 2.2. For more details I want to refer to [55].

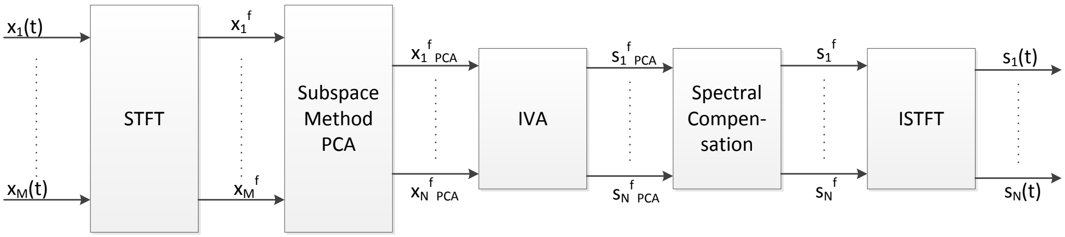

Figure 2.2.: All algorithms used in the Blind Source Separation [55]

.

The second approach is the Geometric Source separation which was implemented in [16]. This algorithm uses a combination of beamforming and BSS to solve the separation issue. A more detailed definition can be found in chapter 3, because it can be anticipated that this approach was the evaluation winner in [21].

The last implemented algorithm is the so called binary masking [15]. This separation segments the signal in the frequency domain into small bins. Every sound source is dominant in some certain frequency bins and so every segment can be labelled with a source.

2.3. Speaker dependent features

The biggest effort in speaker recognition is to calculate speaker dependent features that are as robust as possible against session variability, which means that a speaker never has the same emotional state or health constellation in two meetings [30]. Channel differences have to be considered, under the point of session variability, too. So the elected speaker features are very meaningful for the system quality and they should have following characteristics [31]:

- A large between-speaker variability and a small within-speaker variability
- A robust noise and distortion behaviour
- A frequent and natural occurrence in speech
- Easy to measure from a speech signal
- Difficult to impersonate
- Not be affected by the speaker's health or long-term variations in his voice.
- The number of features should be small [46]

The last point in the list above is because of the exponentially increasing effort of training material with growing feature number. In chapter 4 some experiments, showing this, can be found.

In the following sections the different speaker features will be introduced. First I will write about short-term spectral features that are 10-30 ms long and were extracted out of the vocal tract informations. The second section will introduce the voice source features which are typical for the glottal flow [31]. The next section will show an overview about the spectro-temporal characteristics and the following about prosodic features which contains information about the speech rhythm and intonation. So it is easy to imagine that these features need a longer time period of speech to be extracted and out of that fact their qualification for an online speaker recognition system is limited. The last section shows only a short overview about the high-level features.

2.3.1. Short-term spectral features

Short-term means that the signal is long enough to contain speaker dependent information and on the other hand short enough to be assumed as stationary. In practice the frames are in the most cases 20 to 30 milliseconds long and following out of that these features can be used under real-time conditions.

For all short-term spectral features a pre-emphasis and a windowing has to be done. Pre-emphasis filter is necessary to give high frequencies a bigger intensity and balance out the human vocal characteristics. The windowing is normally made by a Hamming window and is needed because of the necessary finite length of the signal for the next processing step, the discrete Fourier Transformation (DFT). In practice of course the faster Fast Fourier Transformation is used to transform a signal into its frequency components.

Some additional pre-processing improvements can be applied, too. A pitch detector is one possibility to get a higher time-frequency resolution after the DFT and this can achieve better speaker features [41], but only with some extra algorithms.

The next common step is to divide the frame in frequency bands due to the psycho acoustic Bark scale [64]. This means that the bandpass filters become closer and in smaller gaps to each other in the lower frequency area. Usually the filters are overlapping. The last step is to collect the features and this is summarized in the following sections.

Mel Frequency Cepstrum Coefficients

Mel Frequency Cepstrum Coefficients are the most common features which were used in this thesis. An exact description can be found in chapter 3 and because of that only a short one is given here. To receive MFCCs first a psychoacoustic filter bank (in this thesis a Mel filter bank) is applied. Then the logarithm of the bandpass filtered signals, followed by a cosine transformation is computed. Additionally it should be mentioned, that Mel Frequency Cepstrum Coefficients were difficult to beat in practice [31].

Mel Frequency Discrete Wavelet Coefficients

In [4] the Mel Frequency Discrete Wavelet Coefficients are described as MFCCs with one difference, which takes part in the last processing step. Instead of discrete cosine transformation of the logarithm, a discrete wavelet transform is used which includes the so called basis wavelet. In [4] it is claimed, that these features are more stable in a noisy environment.

Spectral Subband Centroids

The principle of this technique is to use the centroid frequency of each subband. Every frequency band is divided into parts that are as long as half the sample frequency. So the results are M subbands [54]. Now the m_{th} centroid feature C_m is calculated through

$$C_m = \frac{\int_{l_m}^{h_m} f \cdot w_m(f) \cdot P^f(f) df}{\int_{l_m}^{h_m} w_m(f) \cdot P^f(f) df}. \tag{2.2}$$

where l_m and h_m are the lower and higher constraint of the m_{th} centroid, $w_m(f)$ stands for the filter shape and $P^f(f)$ is the power spectrum at the location f. Out of the single features C_m a speaker model can be calculated. In [54] it is said that Spectral Subband Centroid features work better than MFCCs if the conditions are very noisy.

Line Spectral Frequencies

Linear Spectral Frequencies are gained out of the linear prediction which will be introduced here first. Linear prediction tries to predict the next signal value $x[n]$ out of the past p signal values $x[n-p]$. The filter

$$H(z) = \frac{1}{1-\sum_{k=1}^{p} a_k \cdot z^{-k}} \tag{2.3}$$

determines the equation for the signal which can be calculated by

$$x[n] = \sum_{k=1}^{p} a_k \cdot x[n-k] + e[n]. \tag{2.4}$$

The error $e[n]$ is a value which results out of the prediction. The prediction coefficients a_k can be used as speaker features because every speaker has an own prediction. This coefficients will usually be calculated out of the $Levinson-Durbin$ algorithm [28]. But they are not often used as features because of their stronger correlation and instability [31] compared to the following two algorithms.

The Line Spectral Frequencies are calculated through the roots of

$$P(z) = A(z) + z^{-(p+1)} \cdot A(z^{-1}), \tag{2.5}$$

$$Q(z) = A(z) - z^{-(p+1)} \cdot A(z^{-1}) \tag{2.6}$$

where $A(z)$ is dependent from the considered p signal values. For example

$$A(z) = 1 - a_1 \cdot z^{-1} - a_2 \cdot z^{-2} = 1 - 2 \cdot \rho_0 \cdot \cos(2) \cdot z^{-1} + \rho_0^2 \cdot z^{-2} \tag{2.7}$$

$$\text{with } 0 < \rho_0 < 1 \text{ and } 0 < f_0 < 0.5.$$

is the equation for a second order prediction. To stay at this example it can be shown [28] that $P(z)$ and $Q(z)$ can be depicted as

$$P(z) = 1 - (a_1 + a_2) \cdot z^{-1} - (a_1 + a_2) \cdot z^{-2} + z^{-3}, \tag{2.8}$$

$$Q(z) = 1 - (a_1 + a_2) \cdot z^{-1} + (a_1 + a_2) \cdot z^{-2} - z^{-3} \tag{2.9}$$

and with some transformations we get our Linear Spectral Frequency coefficients for the second order case

$$\cos(2\pi f_1) = \rho_0 \cos(2\pi f_0) + \frac{1-\rho_0^2}{2} \tag{2.10}$$

$$\cos(2\pi f_2) = \rho_0 \cos(2\pi f_0) - \frac{1-\rho_0^2}{2} \tag{2.11}$$

$$\text{with } f_1 < f_0 < f_2 \text{ and } f_1 \rightarrow f_0 \text{ and } f_2 \rightarrow f_0$$

In [28] it is proven that this theory is effective for higher order coefficients. The so obtained features are very sensitive, because the quantization of one coefficient results in changes only around that frequency. These features are rarely used, because of their correlation and their lack in robustness [31].

Linear Predictive Cepstral Coefficients

The Linear Predictive Cepstral Coefficients (LPCCs) are derived from the Linear Prediction Coefficients and the introduction of section 2.3.1 is here available too. The next step is to calculate with the Linear Prediction Cepstrum Filter

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_n \cdot z^{-k}} \tag{2.12}$$

the LPCCs. G is the gain parameter given by the minimum mean squared error. The theory here will not go into further detail. It only should be mentioned that from equation 2.12 the logarithm is taken and then the derivatives are calculated. To the so received cerpstral equation some more calculations and transformations are conducted. The result

$$\hat{h}[n] = \begin{cases} 0 & n > 0 \\ ln(G) & n = 0 \\ a_n + \sum_{k=1}^{n-1} \bigl(\frac{k}{n}\bigl(\hat{h}(k)a_{n-k} & 0 < n < p \\ \sum_{k=n-p}^{n-1} \bigl(\frac{k}{n}\bigl(\hat{h}(k)a_{n-k} & n > p \end{cases} \tag{2.13}$$

contains now the LPCCs which can be used to produce a speaker model. This calculation is not as complex as the computation of the Linear Spectral Frequency coefficients [28]. In contrast to the Linear Prediction Coefficients there is only a finite number of coefficients. In [57] it is written, that LPCCs has a lower computational price than MFCCs.

Perceptual Linear Prediction

These features are introduced in [26] for speech analysis and can easily be used for speaker recognition. After applying the pre-processing steps, mentioned in section 2.3.1 we got

$$\Theta(\Omega_i) = \sum_{\Omega=-1.3}^{2.5} X(\Omega - \Omega_i) \cdot \Psi(\Omega) \tag{2.14}$$

where Ω is the bark-frequency, Ψ is the critical band curve for the dividing filter and X is the recorded speech spectrum. This result is the pre-emphasised signal by an equal loudness algorithm, which is done to create a sensitivity to human hearing. As last step, before receiving the features, an intensity loudness power algorithm is applied to give a consideration to the nonlinearity between loudness and sound. After an inverse discrete Fourier Transformation the $M+1$ autocorrelation values were taken to solve the equation for the M_{th} order autoregressive coefficients. In [26] the authors describe these features as computationally efficient. For more details I want to refer to the mentioned work.

Modified Group Delay Feature

Usually the features will be computed through the magnitude spectrum but the possibility to calculate them out of the phase spectrum [25] is given too. Of course a new feature type is needed for this case. The first step is to calculate the group delay function

$$\tau(\omega) = -\frac{d\theta(\omega)}{d\omega} \tag{2.15}$$

out of the unwrapped phase function $\theta(\omega)$. A more signal based equation is

$$\tau(\omega) = \left(\frac{X_R(\omega)Y_R(\omega) + X_I(\omega)Y_I(\omega)}{S(\omega)^{2\gamma}}\right) \text{ with } 0 < \gamma < 1 \tag{2.16}$$

where $X(\omega)$ and $Y(\omega)$ are the Fourier transformed from $x(n)$ and $n \cdot x(n)$. The indices R and I suits to the real and imaginary part of $X(\omega)$ respectively $Y(\omega)$. $S(\omega)$ is a smoothed version of $|X(\omega)|$. To reduce spikes in the formant location a modified group delay function

$$\tau_m(\omega) = \left(\frac{\tau(\omega)}{|\tau(\omega)|}\right) \cdot (|\tau(\omega)|)^{\alpha} \text{ with } 0 < \alpha < 1 \tag{2.17}$$

is calculated. After that the features were computed through

$$c(n) = \sum_{k=0}^{N_f} \tau(k) \cdot \cos\left(\frac{n(2k+1)\pi}{N_f}\right). \tag{2.18}$$

The use of the second form of a discrete cosine transformation decorrelates the features. The authors of [25] are claiming that these features achieve similar or even better results than the common MFCCs.

2.3.2. Voice Source Features

Voice Source Features are speaker dependent and include glottal excitations. Given the assumption that the vocal tract and the glottal source are independent the Voice Source Features can be measured through calculating out of the received signal and the inverse vocal tract filter the source signal. The obtained features are no strong speaker recognition attributes but they can be used to improve the Short-term spectral features [31]. In the following some of these features will be introduced.

Autoassociative Neural Network Model

This approach calculates out of 19 LPCCs features a new and additional feature model. This works through putting the features into this Autoassociative Neural Network Model and compare the output to the input [61]. So an error E_i for frame i can be calculated. Now the confidence value

$$c_i = \exp^{-E_i} \tag{2.19}$$

can be computed. This value can be used as additional speaker feature. In [61] it is mentioned that the there developed features may provide robustness against channel and handset effects.

Glottal Flow Derivative Waveform

The method described in [43] uses features which can be divided in coarse and fine features. Coarse features try to modulate the glottal flow mathematically and divides itself again in an open, a close and a return phase. The duration of one phase to the next is used as characteristics. The fine features have the same durations as the coarse ones additionally it concludes the closed and opened phase for formant modulation. The other difference of these features is that the fine ones calculate the energy of a phase.

The problem with the fine features is that they are hard to estimate. To calculate them the glottal flow derivative is used. This method estimates the inverse vocal tract filtering of the waveform through which the glottal pulse can be located. Formants are the lowest stage with which sounds can be depicted. The location makes it possible to define the region where the formant modulation takes place. This can be calculated through the covariance method and the linear prediction. The result $F(i)$ is used to minimize

$$D(n_0) = \sum_{i=n_0}^{n_0+4} |F(i) - F(i-1)| \text{ with } 1 \leq n_0 < N - N_\omega - 5 \tag{2.20}$$

over the region n_0 to $n_0 + 4$ in one frame [43]. In this case $F(n)$ stands for the formant values and $N_\omega = N/4$ is the length of the covariance analysis window. Now the mean and deviation in the stationary region for the first formant is estimated.

To get the fine features from the glottal flow derivative waveform the estimated course features have to be subtracted. The resulting features can then be used for speaker identification. For more information I refer to [43], where additionally it is said that this extension can provide a system improvement of around 5% against their original speaker identification approach.

Wavelet Octave Coefficients of Residues

In the work [63] another algorithm for voice source feature extraction is introduced. After some preprocessing steps linear predictive inverse filtering is applied for a frame length of 30 milliseconds. In the equation

$$e(n) = x(n) \cdot \sum_{k=1}^{12} a_k \cdot x(n-k) \tag{2.21}$$

the filter coefficients a_k are calculated through an autocorrelation algorithm. The neighbouring frames $e(n)$ are joined to receive the vocal tract signal. After locating the pitches in the residual signal a Hamming windowing with a length of three pitches is executed and as result the windowed residual signal $e_h(n)$ is received. The next step is the wavelet transformation

$$w(a,b) = \frac{1}{\sqrt{|a|}} \sum_n e_h(n) \cdot \Psi^* \left(\frac{n-a}{b}\right) \tag{2.22}$$
$$\text{with } a = \{2^k | 1, 2 \cdots, K\} \text{ and } 1 \leq b < N$$

which includes the scaling parameter a, b and the wavelet basis function $\Psi^\star$. For more information on that I want to refer to [63]. The signal now has to be divided in frequency sub-bands of K octave levels. Each sub-band is divided again in M equal parts. The coefficients

$$W_k^M(m) = \left[w(2^k, b) | b \in \left(\frac{(m-1)N}{M}, \frac{mN}{M}\right)\right] \tag{2.23}$$

are calculated with $1 \leq m < M$ and $1 \leq k < K$. The last step, before getting the features, is to take the two-norm from W_k^M. In [63] it is said, that these features give an improvement of two percent in relation to MFCCs used alone.

Voice source cepstrum coefficients

The method in [23] describes other features, which are calculated out of the Linear Predictive Cepstrum (LPC). To receive these features the signal $x(n)$ is split into a voiced and an unvoiced part with a frame length of 32 milliseconds and the next frame starts 10 milliseconds after the beginning of the first frame. The Prediction order K is 16. For the voiced part the closed phases of the glottal in the frame are detected, so the closed phase LPC coefficients for a frame can be estimated. Out of the unvoiced part the autoregressive spectral envelope covariance LPC coefficients for each frame are calculated. To the spectral envelope

$$X(n) = \frac{\sigma_u}{\sum_{k=0}^{K} a_k \cdot \exp^{-j2\pi nk/N_s}} \tag{2.24}$$

a mel-filter bank, with $r = 26$ filters, is applied. In equation 2.24 σ_u is the magnitude of the closed-phase LPC and N_s is the frequency resolution. Out of the resulting signal $Y(n)$ the voice tract cepstrum coefficients, a pre step to the source features, can be calculated through implementing a cosine transformation to $Y(n)$ which looks like

$$c_{vt} = \sum_{r=1}^{N_r - 1} \log(Y(r)) \cdot \cos\left(\frac{(2r+1)m\pi}{2N_r}\right). \tag{2.25}$$

For the whole original signal x(t) normal MFCCs $c(n)$ are calculated too. The subtraction $c_{vt}(n) - c(n)$ yield the final voice source feature c_{vs}. For more information I want to refer to [23]. Here it is claimed too, that these features yields a 3% better misclassification rate then MFCCs features.

2.3.3. Spectro-Temporal Features

Spectro-Temporal features are additionally used and in the most cases they are computed out of short-spectral features. They usually show the transition between the features itself and thus can improve the original ones. Here an overview about the most established algorithms will be given. The first is the derivative from short-spectral features, for example in the case of MFCCs these new features are called delta and double-delta coefficients and represent the 1st and 2nd order time derivatives. An advantage of this method is that this coefficients can be appended to the original features and so the system can easily be improved [31]. Another way for LPC coefficients is described in [17]. Here is suggested to use orthogonal polynomial coefficients as feature expansion. These coefficients represent the mean value, the slope and the curvature of the time function. Some similar feature extensions will only be named and you can obtain more information on that in [31]. There are regression line-, simple differentiation-, time-frequency principal- and data-driven temporal filter-components. The modulation frequency is a speaker dependent feature, too.

2.3.4. Prosodic Features

Prosodic features are non-segmental which means that they need a certain amount of time to be calculated. That of course is a problem in a real-time scenario, how it occurs in a teleconference meeting. So they can not be used as recognition feature but as an offline verifying feature that improves, for example, the speaker model adaptation. Typically intonation patterns, speaking rate and rhythm can be measured and used as prosodic features. How the features exactly can be computed goes beyond the scope of this thesis but can be read in [31] or in [4].

2.3.5. High-Level Features

In [14] these kind of features were initiated. High-level means that these features are put in the speaker lexicon. For example special speaker dependent words or utterances like "hmm" or "uhh" are used as features. Another idea is to use the sequence of words or the position of a word in a sentence like "yeah <end>". Of course this features can not be used as features for an online speaker recognition system, because of their time requirement being to big and so they will not be discussed any further here. For more informations I want to refer again to [31].

2.4. Speaker Models

Now for every speaker features are extracted out of the own speech signals. The next step in a speaker recognition system is to create with the features, gained from an introduction round, a model for every speaker. This makes it possible to compare a speech utterance gathered from a meeting to every speaker model and then calculate the speaker probability for every conference participant. After that further processing steps, like channel assignment and a graphical identification help, are possible.

In this section first the simple Vector Quantisation will be introduced followed by the common Gaussian Mixture Models. Then the newer Support Vector Machine will be defined. The last part of this section is reserved for other, not common speaker models.

2.4.1. Vector Quantisation

Vector Quantisation is a model that promises a fast and easy computation of the probabilities. Under the assumption that $X = \{x_1, \cdots, x_T\}$ is the vector of the test features and $R = \{r_1, \cdots, r_K\}$ are the reference feature vectors received out of the training material. Then the average quantisation distortion $D_Q(X,R)$ is calculated through

$$D_Q(X,R) = \frac{1}{T} \sum_{t=1}^{T} \min_{1 \leq k < K} ||x_t - r_k|| \tag{2.26}$$

where $||x_t - r_k||$ is the euclidean distance. The smaller $D_Q(X,R)$ becomes, the bigger the probability that the speaker was the source of the speech utterance. Often k-means or another clustering method is used to reduce the vector set. In spite of the convenience of this model in [24] it achieves, in an adapted version, comparable results in speaker recognition as the more complex Gaussian Mixture Model. But in the most common literature the Gaussian Mixture Models are preferred and much additional developments were published here.

2.4.2. Gaussian Mixture Models

The Gaussian Mixture Model (GMM) is used in the teleconference system from the Institute for Data Processing at the Technical University of Munich. Because of that I want to refer to section 3.5.3 for the technical details. Here it should be said that GMMs are the state of the art model and it is an extension to the Vector Quantisation through overlapping clusters.

Further I want to mention some possible differences to the system introduced in this thesis. In [31] it is said that a female and a male Universal Background Model can be an advantage but in [52] this can not be proven for our system.

Another approach is to use a Monogaussian Model, for example in [7], which only uses a single gaussian component with a full covariance matrix. This technique seems to achieve not as good results in speaker recognition as normal GMMs but they are faster because of the reduction.

The maximum a posteriori adaptation is not the only way to train GMMs out of an Universal Background Model. The maximum likelihood linear regression (MLLR) [34] is an alternative. The idea is to calculate an adapted mean vector which is multiplied with a matrix that maximizes the likelihood of the adaptation. Originally it is developed for a speech recognition, but in the meantime it is used in speaker recognition systems, too.

Some approaches exist for speeding up the slow GMMs. Two of them will be substitutional introduced here. The first is hash GMM that reduces the components of the original GMM [3]. The idea in this approach is to train two GMMs. The first includes all needed components. The second one is reduced and contains only a fraction of the complete one. After that a shortlist is trained which contains index vectors from the small GMM to every, best matching entry in the big GMM.

The second speeding up technology is described in [39]. In this work it is proposed to reduce the input vectors with fixed-rate decimation, variable-rate decimation and adaptive-rate decimation algorithms.

2.4.3. Support Vector Machines

Support Vector Machines are a newer approach to classify the speaker features. A two-class classifier is used to calculate [9]

$$f(x) = \sum_{i=1}^{N} a_i t_i K(x, x_i) + d \tag{2.27}$$

where $K(\cdot,\cdot)$ is the so called kernel function. t_i is either 1 or -1 and is named as the ideal output. d is a bias constant. a_i is a weighting value that has to fulfil the constraint that the sum over $a_i t_i$ results in zero. Through the kernel function a transformation from the feature space to the kernel space is calculated. For that there are some different kernel functions which can not all be named here, but as an overview four will be mentioned:

- Generalized linear discriminant sequence Kernel
- Gaussian Supervector linear Kernel
- GMM L^2 Inner Product Kernel
- High-Level Supervector Kernel

In the thesis [9] it is claimed that the Support Vector Machine achieves similar results in the NIST evaluation from 2003 as GMMs. The system gets better if a combination of them both is used. For more information I want to refer to [31] and [10].

2.4.4. Other Models

The above mentioned feature models are the most common and auspicious models in literature. But there are of course much more. Some of them will be named and shortly described here. The first are Aural neural network models which consist of interconnected processing units where every unit stands for a feature and there is a weighted connection between the units [60].

In [40] the so called speaker mapping is introduced. The idea is to calculate information about the speaker out of the linguistics. This is done with a linear prediction method. The next step is to map the linguistic informations to the speaker informations via, in this case, a neural network.

The last here presented model shapes every speaker in dependency to the other speakers. Every speaker is modelled by an anchor model that contains features from reference speakers [53, 31]. If the speaker of a new speech utterance has to be calculated, the utterance is modelled by the anchor model and then the vector distance has to be compared to get the best suited speaker.

2.5. Voice Activity Detection, compensation and normalization methods

In this section I want to introduce first another Voice Activity Detection and name some similar approaches. In the next part compensation and normalization methods will be discussed.

The Voice Activity Detection (VAD) which is used in this thesis is usually an offline approach but through some adaptations, like a hard energy threshold or the dynamic threshold which is adapted from frame to frame, it is valuable in online scenarios too. For more informations I refer to 3.5.6.

There are some more approaches for an online VAD, which is introduced here. The Long-term spectral divergence seems to be a reliable algorithm. The idea behind it is to measure first the long-term spectrum envelope (LTSE) [45] of the noisy speech signal spectrum $X(k,l)$ where l is the frame number and k is the band issue. In the equation

$$LTSE_N(k,l) = \max\{X(k,l+j)\}_{j=-N}^{j=+N} \tag{2.28}$$

N defines the order. The next step is to calculate the deviation between the LTSE and the average noise spectrum magnitude $N(k)$. This results in the long-term spectral deviation (LTSD) which is defined as

$$LTSD_N(l) = 10\log_{10}\left(\frac{1}{NFFT}\sum_{k=0}^{NFFT-1}\frac{LTSE^2(k,l)}{N^2(k)}\right). \tag{2.29}$$

$k \in 1, 2\cdots, NFFT-1$ is a condition in this equation. NFFT is a threshold which is set to 256. The LTSD can be used as feature for the decision of voice or silence. For more information it is obtained to [45].

Of course there are some more approaches, but the main aspect in this thesis is not the voice activity detection and so I want to advise to [31] for more implementations.

Usually it is tried to compensate or normalize a signal to reduce the influence of noise to the speech and following consequently out of that the influence to the speaker features and models. To achieve this the features can be normalized, a speaker model can be compensated and a score normalization of the speaker model is applicable too [31]. For example in the feature normalisation domain the idea is to subtract the noise from the received signal, for instance a mean value of each feature.

Compensating the speaker model means to use an universal channel model to compensate the influence of the channel to the speaker model.

The score normalization tries to normalize a new speaker model in matters of a cohort model that consists an amount of other speaker models. Usually a normalization looks like

$$s' = \frac{s-\mu_I}{\sigma_I}. \tag{2.30}$$

where s is the score. μ_I and σ_I is the mean and standard deviation of the cohort models.

2.6. Channel assignment

Channel assignment is a main part of this thesis and therefore some other relevant approaches to this topic will be introduced. Channel assignment means that every speakers speech is put on his own channel and this makes it possible to produce, for example, 3D-sound. The implementation chosen in this thesis can be read in section 3.6. The problem in this topic is to have a robust algorithm for every frame, because an error can produce a 3D-sound for one speaker that jumps between different localisations in the output.

2.6.1. Channel assignment using audio data

In [27] the problem is solved with only using an array of eight microphones and because of that it is possible to locate the speakers with a direction of arrival algorithm (here general cross correlation with a phase transformation is used). To make sure that only speech data is processed, a VAD preprocessing is necessary. At last a separation is used to reduce the influence of an overlapping speaker to the original speaker's channel. This is a relatively easy, but very robust implementation with the disadvantage that speakers can not move or change seats between a meeting. The speaker diarization approach labels the appropriate speaker channel.

Another way to assign multi-modal data streams for every speaker is described in [2]. Here only the audio-streams, received from the close-talking microphones, are taken to get a speaker independent channel assignment. There are as much cameras as participants in the room and so every participant has his own video stream. The assignment of video and audio is quite easy, because every speaker has his own audio and video stream that only has to be synchronized. This approach is mentioned under audio only channel assignment, because the close-talking microphones fulfil the requirements through a activity detection.

In the book [50] the word channel assignment is not used but the algorithm introduced here can be adapted for this topic. The idea is to locate a speaker with a Steered Response Power - Phase Transform (SRP-PHAT) algorithm and then track him with a Kalman Filter. This way a moving participant can always be allocated with his produced speech and so channel assignment is possible. A further advantage of this algorithm is that a speaker is not active all the time and so the tracking works with clustering of the last known position of all participants. The clustering approach is adapted in a way that a new speaker can be recognised. Additionally a participant who left the meeting is not any longer tracked through computing a time threshold during which a participant has to be active.

Another Teleconference system is introduced in [29]. Here a Direction of Arrival (DoA) algorithm is implemented, which assumes for every frame the location of the speaking meeting participants through energy measures. This can be seen as a Bag of Words problem as it is used in the comparison, for example, of pictures or an internet search. In the audio case the words can be interpreted as locations and a document is here a histogram of locations over some frames. After that the result has to be modelled in order to be clustered. For this a dynamic Latent Dirichlet Allocation is used, which improves the normal Latent Dirichlet Allocation with a distribution of two following frames. For more information I want to refer to [29]. Now only a variational Bayes algorithm has to be applied, that assumes the variational posterior of the models. Clustering can be used for channel assignment too. The whole system has an advantage that I want to mention in special. This system does not need to know in advance how many speakers are going to attend the meeting.

A further approach to assign speakers to their channel is defined in [59]. Here reinforcement learning is used to identify the actual speaking person. The speaker localisation decides if the algorithm gets a reward. This means if the recognized speaker has the same position as it is saved in the model a reward is received and the speaker model is adapted. At the original work this approach was not used for channel assignment but it is proven that an easy adaptation is possible. This procedure was developed at the Institute for Data Processing at the Technical University of Munich and therefore is used for more evaluations in this thesis to confirm the channel assignment suitability. So an exact definition can be seen in chapter 3 and the results in chapter 4.

2.6.2. Channel assignment using video and audio data

In the work [38] channel assignment is achieved through person tracking with cameras. The so received speaker can now be located in the video stream. A second audio localisation combines the two streams and assigns them to one channel. The channel can afterwards be labelled with a name through a speaker recognition system and a face identification in the video stream. The combined likelihood of these two identification systems labels the stream with a speaker name, if the likelihood is bigger than a threshold. For a closer system introduction I want to refer to section [38].

In the work of Himanshu Vajaria [56] an offline approach is introduced. Here first the video and audio stream is split into homogeneous segments. For every audio segment MFCCs are extracted and modelled with a GMM. In the video segment a motion between three frames is calculated, because a speaker has a bigger movement than a listener and so the camera with the biggest motion percentage belongs to the speech frame. Now a face detector is used in the video stream. The result is an audio segment that is labelled with the speakers face. Now a back segmentation is applied to assign all speech samples belonging to one speaker/face.

Often meeting analysis needs a good channel assignment because a separated stream for every speaker is necessary for a better evaluation. In [42] the speakers were tracked through their faces and in the audio stream the active speaker is located. Supposed that the participants have constant positions in the room the channel can be assigned for one speaker through the located speech source. So it is possible to make an analysis about the attention of the participants, which is measured through the direction where the person is looking.

2.7. Speaker localisation and recognition systems

In a speaker identification and localisation scenario different devices can be used. In the system introduced in this thesis only audio information is handled. But often video information is used too. Here face detection for speaker identification and for localisation, for example through lip motion, are the main approaches invented in such systems.

Another important differentiation attribute is the field of application and with that the involved limitations in the used hardware. The first ambit is the robot domain. Here the robot has to react user specific or look to the speaker to simulate a real human-human conversation. Another field are Smart Homes or rooms which have to know who is in the room and where is the person to react in a desired manner to human orders. The last aspect is the same as introduced in this work, the teleconference scenario. Here persons sit or move around a microphone array and in another room a speech output should be generated that opens the listener possibilities to use his psycho-acoustical abilities. The requirements are again a stable speaker recognition, separation and localisation.

In the further chapter a summarisation about related approaches to our one will be introduced. It is divided into procedures that use only audio and others that combine audio- and visual information.

2.7.1. Audio only processing

The system described in [22] divides the speaker recognition approach into a training- and an identification phase and is implemented for a robot. The training phase is used to create the speaker models out of a single close microphone where MFCCs are extracted and converted into a Vector Quantisation Model. In the identification phase the ManyEars implementation of a Beamformer and a Geometric Source Separation is applied to get an audio stream for every speaker. After that a mask is used to calculate the noiseless features (again MFCCs) of the audio stream. The calculation of an euclidean distance from the trained and extracted models result in a „who speaks when “. The mask is always updated to get the time variance of noise and room conditions. A big difference to the system introduced in this thesis is, that a robot doesn't need to calculate the speech origin in real-time, because the further processing has to be done after the complete order and so the results can be hardly compared.

In the work [49] were four probabilities calculated. The first is the speaker position $p(x^{pos}|c)$ received from a filter and sum beamformer. The binary value c stands for an existing or not existing speech signal. The second probability is $p(x^{bic}|c)$, which is calculated out of MFCCs which were modelled by a GMM and that stands for a speaker change probability. The third is the probability that a speaker was the source of a speech utterance $p(x^{sid}|c)$. This is calculated out of the GMM too. The last likelihood $p(speech|x^{vad})$ is computed through a Voice Activity Detection and declares if a frame contains speech or not. This information is summed in a Hidden Markov Model. The probability that the state j is the current state is calculated through:

$$b_j(x(k)) = p(x^{sid}|c) \cdot p(speech|x^{vad}). \tag{2.31}$$

The state transition probability between i to j is calculated out of the position and speaker change probability. A Viterbi decoder computes the advantageous state sequence.

An approach that uses the speaker recognition for an improvement of the localisation is described in [35]. The identification works again with MFCCs and GMMs and the localisation algorithm uses a beamformer. In an introduction round the speaker models are created and saved. An initial position estimation is calculated too. During the meeting every second a source location is computed. If the localisation gives a position back that does not match to one model location, the speaker recognition is used to cancel this cleavage. A disadvantage of this algorithm is the big amount of time and that no further speaker movement calculation is done.

2.7.2. Audio-visual processing

An audio-visual computing for speaker recognition and localisation has the advantage that more sensor data can be used and so the accuracy of the system can be improved. But such systems have one important disadvantage, they are much more complex in the needed hard- and software that has to be coded and so it is an expense, installation and configuration factor to build and work with that.

In the work of Jörg Schmalenstroer [48] such a system is introduced. An audio signal is captured by a microphone array with a frequency of 16 kHz and a frame size of 128 samples. The further processing includes a 13 MFCCs feature capture and a calculation of their first and second derivatives. The audio features x^{sid} are on the one hand modelled in a Gaussian Mixture Model, which is trained in an introduction round, and on the other hand they are used for speaker change detection through a ΔBIC (Bayesian Information Criterion) x^{BIC}, which is a common way to calculate the difference of following feature vectors. Some speaker localisation techniques are discussed in this work and one of the best suited seems to be the Filter and Sum Beamformer. The result of the localisation is a new feature x^{pos}. At last a face detection and identification algorithm is applied which results in a fourth feature vector x^{vid}. Now a Hidden Markov Model is implemented where every participant is a state and silence is an additional one. There is a transition between every speaker and a transition between speaker and silence. The probability that the model is in a speaker state is calculated out of the identification system x^{vid} and x^{sid} and the probability for the silence state is calculated through a Voice Activity Detection. The transition probabilities are given through the speaker change detection x^{BIC} and the position x^{pos}. The last step is a Viterbi decoder which calculates the best state sequence.

In the article [37] a second way to recognize the speaker is described. Here a mobile robot has to track and identify persons in the room. It does this through video and audio information. A face is localised followed by a skin color histogram. That way an initialisation of the approach is possible through creating a new track with the information of localization, skin color histogram and depth of the person. If a person is speaking additionally MFCCs in a GMM are separated and an audio localisation is executed for this track. An interesting idea is introduced here for the speaker recognition. There is not only a GMM modelled for every speaker but rather for an impostor speaker, consisting training data from all other speakers. So the online recognition task includes the measurement if a frame belongs to the speaker or the impostor.

The tracks are further created if a face is found and deleted if there is no new information of the participants track for longer than five seconds. The identity information is saved in case that the person appears again.

Another approach can be found in [11]. Here not a complete conference system is introduced but an interesting way to combine two speaker identification streams. After extracting MFCCs as audio features and discrete cosine transformed features for the visual identification a Gaussian Mixture Model (GMM) is calculated for each feature stream. The speaker models are trained out of an Universal Background Model and an initialisation utterance. Now a model for the identification is computed through transforming the GMM into a feature optimized space. Here Maximum Likelihood Linear Transformation is used. A second model is computed for the verification of the identification. This is done via Maximum A posteriori Probability adaptation of the background model. For collected frames the distance from participant model to speech and visual features are calculated and the closest model is taken as causer. A verification is applied afterwards through the distance of the speech and visual features from the background model, which has to be bigger than a threshold. For a more detailed statement I want to refer to [11].

In [8] is the position of the speaker calculated through a Time Difference of Arrival algorithm and a 360 degree camera localise the participants too. The speaker identification uses MFCCs modelled by a GMM. This audio information are combined by a Monte Carlo Fusion algorithm, which uses a Particle Filter to track the persons in a room. This results in a correlation measure r_{ij} which contains the probability that speaker j is speaking and if this person stays in area i. Every tracked participant gets an ID.

A further approach can be found in [6]. Here the speaker position is estimated again by a Time Difference of Arrival algorithm followed by a speaker identification with MFCCs modelled with a GMM. The initial speaker model is calculated out of an offline introduction round. The visual recognition contains a face detection and identification as a visual localisation. The whole system works temporarily with the audio technical localisation which sends a speaker position to the tracking algorithm where the speaker is identified by his face. The speaker recognition is only used as verification, because of the lag generated by the amount of audio data that is needed.

In [47] the recognition of a speaker uses GMMs for the audio part and a face identification for the visual one. Additionally is every speaker localised by a SRP-PHAT algorithm combined with Time Difference of Arrival. The visual localisation of participants is calculated by a tracking algorithm which uses a Particle filter. The track starts in the moment an user enters the Smart Room. For the Particle filter every point i in the room gets a weight over the equation

$$w_t^i = w_1 \cdot (POM \cdot FBI) + w_2 \cdot (AcLoc \cdot SpkId). \tag{2.32}$$

The visual probability for localisation POM and identification FBI as the audio probability for localisation $AcLoc$ and identification $SpkId$ are weighted with w_1 and w_2. These two values are received out of experience. Through continuously adapting the weights for the room points a permanent tracking of every participant is possible.

Additionally a state transition with Hidden Markow Models is calculated offline. A three second speech sample is clustered, in as many clusters as there are speakers plus one Universal Background state, through a Baum-Welch algorithm.

2.8. Concluding remarks

The algorithms which are used and combined in our system are SRP-PHAT and GSS for speaker localisation or rather separation and MFCCs as speaker features as well as GMM as model. SRP-PHAT and GSS were the evaluation winner in the thesis [21]. We decided to take MFCCs because they are the state of the art and they are still hard to beat in practice [31]. GMMs are used, because they are fast to compute, efficient and effective [33]. The speaker recognition approach of [59] was further utilised, because the algorithm has achieved good results and was easy to implement in the existing system. The whole system is described in chapter 3.

There are many approaches that combine speaker recognition and localisation, but in the most cases there are no audio only approaches. The visual identification part is often needed to get the system more stable. The systems which use only audio data for the processing have often a big frame length, no speaker position adaptation or do not update the speaker models for a better recognition. Additionally can be said, that the speaker recognition has often not such an important role at all. The teleconference system of the Institute for Data Processing at the Technical University of Munich has not been existing yet in its entirety, but many of the single approaches and algorithms are used elsewhere.

Channel assignment is not a big research part at the moment because there are only a few developed algorithms. The localisation stand alone or linked to visual information is used in the most channel assignment systems. The approach, described in this thesis, uses the speaker recognition as help or to assign the channels itself. This is, under the knowledge of the author, in no other work the case.

2.9. Evaluation

First of all there was a decision to be made, which evaluation standardisation should be chosen. We decided that the NIST Spring 2007 (RT-07) Rich Transcription Meeting Recognition Evaluation Plan [19] is suited best for our case. This gives a structure how to evaluate

- Speech to Text algorithms
- Speaker diarization systems
- Speaker Attributed Speech-To-Text approaches

The speaker diarization analysis is used in this thesis. There is a newer standard but we decided against it, because in this version video streams must be considered too and our system works with audio only. This version is called NIST Spring 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan [18].

Some customizations have to be done, because the used evaluation is designed for a speaker diarization approach, which works offline on a whole meeting file and our system works online. So some specifications are not necessary and others are not feasible. The first adaptation is the handling of speaker pauses. In the RT-07 evaluation plan silence counts as speaker pause if it is longer than 0,3 seconds. This is not usable in our case, because we take a constant frame length and if somebody is talking in a frame it has to be considered as speech and has to be put to the channel assigned to the speaker. Pause in our system only is the case if no speech is included in the frame. The second point we adapted to our case is the forgiveness collar of 0,25 seconds, which is necessary in a speaker diarization approach. The forgiveness collar means, that the diarization approach is allowed to find a speech utterance 0,25 seconds earlier or respectively later than it was the case in reality. In the developed system it is not needed, because the diarization algorithms classify the audio file not in hard time steps, but this is done in our work. The last point we can not consider are the original audio files used in the RT-07 evaluation standard, because there was no possibility to receive them. So we decided to use the international accredited and commonly used AMI meeting corpus [1]. This corpus is employed to have a comparison to other approaches. A second corpus recorded by ourself is used to evaluate the whole system.

The evaluation score in the NIST plan is the so called diarization error score. This error is calculated through

$$Error_{SpkrSeg} = \frac{\sum_{allseg}\{dur(seg) \cdot (max(N_{Ref}(seg), N_{sys}(seg)) - N_{correct}(seg))\}}{\sum_{allseg}\{dur(seg) \cdot (N_{Ref}(seg)\}}, \qquad (2.33)$$

where N_{Ref} is the number of reference speakers in a segment and against that N_{sys} is the number of system speakers in this segment. The difference between the two speaker types is that the reference speaker is similar to the real speaker and the system speakers are the speakers that are found by the algorithm. In a perfect case the number of both should be identical. But the system can find more speakers and then two or more correspond to one reference speaker. $N_{correct}$ are the number of correct found speakers in this segment. A easier expression for equation 2.33 is the following Diarization Error Rate (DER)

$$DER = \delta_{miss-error} + \delta_{false-alarm} + \delta_{false-detection} + \delta_{speaker-error}. \tag{2.34}$$

The $\delta_{miss-error}$ includes the calculated weighted sum over all speech segments that are not detected. The $\delta_{false-alarm}$ stands for all segments that contain no speech but were labelled to a speaker. The last weighted sum $\delta_{speaker-error}$ contains every frame where one or more speakers are not found, assessed with the number of not found speakers in this segment.

A further adaptation was necessary to evaluate the localisation approach for channel assignment. The speaker position has to be labelled by a name and therefore the location is saved in the speaker model, too. This way speech utterances can be classified by a name, if the utterance position deviates not more than a threshold from the location of the model. So a DER calculation as mentioned for the speaker recognition is available too.

There is a second NIST evaluation which is called The NIST Year 2012 Speaker Recognition Evaluation Plan [20]. This plan sounds better suited to our case, but the evaluation expects here only to differ if a model speaker is talking or not. But we need an evaluation which allows us to differ between some speakers like it appears in conference scenarios, so we decided to use the approach from the speaker diarization, which gives us evaluation standards for our case.

3. Developed teleconference system

This chapter will introduce the whole teleconference system from the Institute for Data Processing at the Technical University of Munich. After this chapter it will be clear how this teleconference system is working and a comparison to the in chapter 2 introduced related works should be possible.

In the next part I will give an overview about the components used in the system. In the following section the microphone array will be illustrated. The speaker localisation is introduced in the successive part of this diploma thesis. Here it is pointed out what the idea behind the algorithm is and why it is needed. The next section shows how the speaker separation algorithm is working and why it is needed. The main part of this chapter is the speaker recognition. Here it will be shown how the algorithm is operating, why it is necessary and finally an improvement to the last versions will be declared. The implemented channel assignment algorithms are defined here too. A second speaker identification, based on Reinforcement learning, will be briefly analysed. The ICSI speaker diarization system is introduced at last.

3.1. System overview

In figure 2.1 the whole system can be seen. First of all in the middle of the conference table a microphone array with eight microphones is put. The participants sit around this technical apparatus and start with an introduction round. This information is needed for the training material. So for every meeting member a speaker model out of the Universal Background Model can be created. In this model a speaker position is saved, too. Now the participants can start with the meeting and after a defined frame length the audio files for further processing will be created.

After the recording of a speech utterance in the meeting, the SRP-PHAT algorithm is used to locate the source position to afterwards separate the source from other simultaneously speaking persons and noise through the Geometric Source Separation. The decision felt to their benefit, because they were the evaluation winner in [21].

A further processing step is to recognise the active speakers for every frame [33] [52]. Some separation algorithms, like the GSS, can label a channel as active or inactive by themselves. For the other separation approaches the Voice Activity Detection has to do be adapted to do that. The VAD is needed anyway, because it reduces the amount of data, due to cutting the silence away, up to 25 percent. So the following algorithms will be faster.

The remaining audio data is used to calculate the features of the speakers in the active channels. In this work Mel Frequency Cepstrum Coefficients are used as speaker specific features. Through the existing speaker model and the computed features it is possible to say who the active speaker is. As a second feature the speaker position is used. At last the speaker models will be updated during the meeting and so they will become more accurate over time.

In the following sections an exact definition of every named algorithm will be given. To have a standardisation I want to define some general valid formulations. First of all one microphone input j is defined as

$$x_j(t) = \sum_{i=1}^{N} a_{ji} \cdot s_i(t) + n_j(t). \tag{3.1}$$

and contains the sum over the number of sources s_i which are weighted with a microphone and source dependent factor a_{ji}. At least a noise coefficient n_j is added. The result is the received microphone signal x_j. As a further definition the number of sources is given with N and the number of microphones is M. This can be merged to a Matrix Vector notation which shows

$$\mathbf{x}(t) = \mathbf{A} \cdot \mathbf{s}(t) + \mathbf{n}(t) \tag{3.2}$$

and contains a matrix $\mathbf{A}$ where every entry is given trough a_{ji} with $j \in M$ and $i \in N$. So $\mathbf{A}$ is a $M \times N$ matrix. The vectors $\mathbf{x}(t)$ and $\mathbf{n}(t)$ contain M entries.

3.2. Microphone Array

In a teleconference scenario it can be mentioned that the participants sit around a table and in the mid of the table a microphone array with eight microphones is positioned. The principle structure is always the same and is shown in picture 3.1. In the work of [21] three different microphone arrays were built through a 3D - Plotter. There are eight microphones, which are arranged circular with 45° degree to have the possibility for a signal direction estimation and to separate the speakers. The radius is 12 cm. In the diploma thesis of Thomas Grasser [21] the winning microphone array for the Geometric Source Separation in combination with Steered Response Power - Phase Transform is the simple microphone array without any attachments. Figure 3.2 shows the microphone array used for further processing.

3.3. Steered Response Power - Phase Transform (SRP-PHAT)

The speaker localisation is needed for the later declared speaker separation algorithm and for the improvement of the speaker recognition implementation, as it is introduced in section 3.5.3. In the following section the SRP-PHAT approach will be discussed as it is implemented in the teleconference system, because it was the evaluation winner in [21].

The SRP-PHAT algorithm works with the data received from a beamformer. Beamformers try to separate a signal through geometric information. In the case of a Delay and Sum Beamformer, like in figure 3.3, it is tried that the phase is in accordance through overlaying the signals received from every microphone. After weighting every channel, not only a separated signal is received, but also geometric information which can be used further. In the work of [16] the similar Filter and Sum Beamformer is applied. The obtained geometric information is used to calculate the origin of the maximum signal energy, which should accord to the sound source. This is the SRP part. To include the PHAT part the phase difference between the single microphone pairs has be considered too.

The received localisations are pretty unstable, because no time information is used and spontaneous energy peaks attributed to noise can not be ignored. To solve the problem a particle filter is used, which tracks a sound source over time through a probability density function. The idea is to define particles for a sphere in the room. Every particle is assigned to a cluster which again is assigned to a sound source, if the energy for the particle is high enough. More information can be found in [16].

3.4. Geometric Source Separation

As mentioned above is the separation needed to separate simultaneously speaking persons and to reduce the influence of noise. In [21] the evaluation winner was the Geometric Source Separation and thus this one was implemented in the teleconference system from the Institute for Data Processing at the Technical University of Munich. In this section this approach is defined more exactly.

Geometric Source Separation (GSS) is a mixture of beamforming and Blind Source Separation. The idea is to assume a separation matrix which cancels the influence of the channel, noise and the other simultaneous talking persons. However, the equation

$$\mathbf{y}(\omega) = \mathbf{W}(\omega) \cdot \mathbf{A}(\omega) \cdot \mathbf{s}(\omega) \tag{3.3}$$

is no longer solved through independent vector analysis but rather with the assumption of a beamformer. The transfer matrix $\mathbf{A}$ is calculated through

$$a_{ij}(k) = \exp^{-2\pi k \delta_{ij}}, \tag{3.4}$$

where δ_{ij} is the delay between a microphone pair i and j. k stands for a frequency group.

Now only the separation Matrix $\mathbf{W}(\omega)$ has to be estimated, which is done by calculating iteratively cost functions and their derivatives. One iterative step n is done through

$$\mathbf{W}^{n+1}(\omega) = \mathbf{W}^{n}(\omega) - \mu \left([||x(t,\tau)||^2]^{-2} \cdot \frac{\delta J_1(\mathbf{W}(\omega))}{\delta \mathbf{W}^*(\omega)} + \frac{\delta J_2(\mathbf{W}(\omega))}{\delta \mathbf{W}^*(\omega)} \right) \tag{3.5}$$

where $J_1(\mathbf{W}(\omega)) = ||R_{yy}(t,\tau) - diag[R_{yy}(t,\tau)]||^2$ is the minimization equation with the signal correlation $R_{yy} = y(t,\tau) \cdot y(t,\tau)^H$ and the geometrical influence is given through $J_2(\mathbf{W}(\omega)) = ||\mathbf{W}(\omega) \cdot A(\omega) - I||^2$. μ is an adaptation rate which is set to 0,01. $\mathbf{W}(\omega)$ is initialised by the filter coefficients of the beamformer. That is the way the source signal $\mathbf{s}(\omega)$ can be assumed. More information can be found in [16].

3.5. Speaker recognition

In this part the speaker identification approach of the Technical University of Munich at the Institute for Data Processing will be introduced. In the first step the pre-processing will be discussed to show the exact calculation of the speaker specific features and how to model them in a comparable manner afterwards. In the last part of this section it will be shown how a new speech utterance can be assigned to a trained speaker model. In this thesis the speaker recognition system out of [33, 52] is used and extended, but for further informations I can refer to these two works.

3.5.1. Pre-processing for speaker recognition

The sampling of the speech data has to be done first to receive a digital signal out of the analogue one. The so called sampling theorem has to be considered through proving the guilt of

$$f_{sample} \geq 2 \cdot f_{speech}. \tag{3.6}$$

In the developed system f_{sample} is set to 16 kHz and so the 48 kHz recorded speech has to be down sampled. The human vocal tract has a low-pass characteristic on the human produced sounds which have to be compensated through using a pre-emphasis filter with the shape of

$$H_{pre}(z) = 1 - \alpha z^{-1} \text{ with } \alpha = 0.95. \tag{3.7}$$

Now a hamming window has to be applied as bandpass filter to make the signal static. The window

$$w(\tau) = 0.54 + 0.46 \cos(2\pi \frac{\tau}{T}) \text{ with } \tau = -\frac{T}{2} \cdots + \frac{T}{2} \tag{3.8}$$

has different overlaps and lengths which can be looked up in chapter 4. These two durations have to be chosen in a way, that the signal consists of enough speaker specific information and on the other hand to fulfil a time static signal. Now a Fast Fourier Transformation (FFT) can be used to receive the spectrum.

The last pre-processing step is the Voice Activity Detection (VAD), which reduces the speech signal by subtracting the speechless, thus silence, frames. This is done by calculating an energy threshold over the whole file which distinguishes between silence and speech. In that way this is an offline approach which has to be adapted to our requirement. We decided to test two different VADs. In the first approach a hard threshold is given by hand to the system, gained through experience. The second approach uses a dynamic threshold, which is updated from frame to frame. The results can be seen in chapter 4.

3.5.2. Mel Frequency Cepstrum Coefficients

Every human has his own specific body form and his voice is differentiable by his vocal tract, the shape of his oral cavity and his physique. In this thesis features, through that given knowledge, are calculated with so called Mel Frequency Cepstrum Coefficients (MFCCs). The decision felt to these features, because they are the most common and therefore seems to be the best [31]. In the first step a frame received from the preprocessing has to be divided by using a Mel-filter bank like in figure 3.4. This filter bank consists of triangular bandpass filters which become bigger the higher the frequency is. They overlap exactly at the half of neighbouring filters. For every filtered part the signal energy is calculated by

$$E_{mel}^{(w)} = \sum_{n=0}^{K/2-1} F_{mel}^{(w)} |S(k)|^2 \text{ with } 1 \leq w \leq W \tag{3.9}$$

and then the logarithm is applied, followed by a discrete cosine transformation to decorrelate the signal. This results in

$$c_{MFCC}^{(i)} = \sum_{w=1}^{W} \log(E_{mel}^{(w)} \cos\left(i(w-0.5)\frac{\pi}{W}\right) \text{ with } 1 \leq i \leq M. \tag{3.10}$$

The extracted features are very static and to receive a bigger dynamic the first and second derivation of equation 3.10 is calculated. There is the possibility to use the energy and the zero coefficient (offset) as features too. In this work a variable number of features is compared which can be seen in chapter 4.

3.5.3. Gaussian Mixture Model

The extracted features are hard to compare against characteristics of a speech utterance. This can be solved by transforming the MFCCs into models. In the preparatory work

[33] the decision fell to a Gaussian Mixture Model which is a weighted probability density function with the form

$$p(\mathbf{x}|\lambda) = \sum_{k=1}^{K} w_k P(\mathbf{x}|\lambda_k). \tag{3.11}$$

w is the weighting factor that fulfils $\sum_{k=1}^{K} w_k = 1;\ 0 \leq w_k \leq 1$. The density P is given through the Gaussian Distribution

$$P(\mathbf{x}|\lambda_k) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_\mathbf{k})^T \Sigma_k^{-1} (\mathbf{x}-\mu_\mathbf{k})\right). \tag{3.12}$$

λ is a distribution of w_k, Σ_k, μ_k and is optimized in the further processing in a way that $p(\vec{x}|\lambda)$ is maximum. The initial value for the weighting w_k, the mean μ_k and the covariance matrix Σ_k is computed out of the MFCCs with a k-means clustering algorithm. This first values are further optimized with a Expectation Maximization (EM) algorithm that switches between two cases (Compute and Update). This is done till $p(\vec{x}|\lambda)$ exceeds a threshold, which is normally reached after five durations.

3.5.4. Speaker identification

The speaker GMMs make it possible to calculate how similar a new speech utterance is to one speaker. This is done by a Maximum Likelihood (ML) computation, which maximizes a probability function. A new speech utterance has to be separated and MFCCs must be extracted. Now the ML classification can be applied which classifies the speech samples with a speaker name.

3.5.5. Speaker model adaptation

In a normal teleconference scenario it is typical that every participant is introducing himself and in the system discussed here, this data is used to train initial models. The problem is that this short introduction round, which consists of roundabout 30 seconds to one minute of every speaker, has not all relevant voice information for every person. To solve this issue the speaker model is updated during the meeting and a Universal Background Model, consisting of a big collection of speech data, is taken in the beginning.

Maximum a posteriori (MAP) adaptation

The missing information about every meeting participant can cause an imprecise speaker recognition system and has therefore be solved. One way is to adapt the speaker models continuously with the model appropriate speech utterances. The longer a meeting, the better the models should be. The algorithm that is needed is called Maximum a posteriori. The first step is the calculation of a new GMM out of the received speech, which is used to adapt and improve the existing model. The idea is to determine, out of the probability of a GMM k, the static values followed by updating the means μ_k. The probability looks like:

$$P(k\,|\,\vec{x}_n,\,\lambda) = \frac{w_k\, p_k(\vec{x}_n\,|\,\lambda_k)}{p(\vec{x}_n\,|\,\lambda)} \tag{3.13}$$

$$n_k = \sum_{n=1}^{N} P(k\,|\,\vec{x}_n,\,\lambda) \tag{3.14}$$

$$E_k(\vec{X}) = \frac{1}{n_k}\sum_{n=1}^{N} P(k\,|\,\vec{x}_n,\,\lambda)\,\vec{x}_n \tag{3.15}$$

$$\vec{\mu}_{k,new} = \alpha_k E_k(\vec{X}) + (1-\alpha_k)\vec{\mu}_k \tag{3.16}$$

with

$$\alpha_k = \frac{n_k}{n_k + r}\ . \tag{3.17}$$

r is a relevance factor which says how strong the existing model should be adapted. For more information see [33].

One issue with the developed idea is, that in the beginning the recognition often combines the false speaker model with a speech utterance which therefore conducts in a false model update. This makes the whole system worse. In the work [52] an answer has be found in using the localisation of each speech sample to verify the origin. In each speaker model an azimuth position, calculated out of the training material, is saved. So the likelihood of a false model adaptation is extremely reduced.

Universal Background Model

The lack of speech in the beginning of the meeting results in incomplete speaker models. In the early stage of a meeting it is very important to have a robust speaker model, that makes a reliable speaker identification possible. In [33] an Universal Background Model (UBM) is used to solve this issue. An UBM is a single GMM which is computed out of a big amount of speech data, received from many different people. The participant models, collected out of the trainings material, is adapted from the UBM via the MAP algorithm. So a much more robust speaker model is calculated.

3.5.6. Voice Activity Detection

The problem with the VAD is, that it is written for an offline approach. This means, that the VAD is used on complete meetings and thus the dynamic speech/silence threshold is set nearly perfect. At the Institute for Data Processing we want to use an online speaker recognition algorithm and this means, that the file parts have a defined short length, for example 0,1 seconds. The dynamic threshold is now set with only the knowledge of an audio file with a short length and if this file contains only silence the silence will not be reduced, because of the wrong set threshold. So the existing approach has to be adapted in a way, that the dynamic threshold is saved and adapted for every meeting. This means that at the beginning of a meeting the threshold is to low and silence will not be removed, but in the theory at the end of the meeting the threshold should be very good. A second VAD is implemented too, which has a hard coded threshold, that is gained out of experience. Some experiments to this issue can be seen in chapter 4

3.6. Channel Assignment

One important work of this thesis was to find a stable combination of the single components to have a complete teleconference system at the recording side. The output should result in a separate channel for every speaker.

The winner of the work [21] is used. This means that for localisation the SRP-PHAT algorithm is used and for the speaker separation the Geometric Source Separation is applied. The speaker recognition is used as defined in section 3.5.3 with the winning values out of the tests as shown in chapter 4. This combination is implemented for a stable and certain channel assignment. This is necessary to fulfil the requirement of a 3D-sound at the playback side.

The localisation and separation algorithm needs at least a frame length of 1024 samples. The speaker recognition of course (see chapter 4) is better as longer the frame length is chosen. The compromise between real-time requirements and a long frame length was to take a length of 4 times 1024 samples.

The channel assignment is implemented with three different algorithms. The first simple one uses the speaker recognition and puts every $4 \cdot 1024$ an audio frame at the channel of the calculated speaker. The speaker models were adapted if the computed position accord with the location of the speaker model.

The second idea was found out during the testing of the first implementation, because the localisation algorithm shows pretty good and stable results and thus can be taken as approach for channel assignment. The position of the speaker model is compared to the calculated location of the speech utterance and then the channel assignment is done to the nearest speaker.

In the third approach the localisation is used too and the channel assignment is done if the two azimuth angles do not deviate more then 15 degree. If no speaker model is found which fulfils this demand the speaker recognition is used to assign the speech frame to a channel. Figure 3.6 shows the circumstance. This developed algorithms do not support a free movement of the participants why an adaptation has to be done. The received speech is collected for one second. The source of the speech frame is computed through the recognition. Afterwards the mean azimuth angle of the collected speech is calculated to compare it against the azimuth of the received speaker model. If they accord a $positionChange$ counter is set to zero and if they do not the counter will be increased. If a threshold is exceeded the speaker model is set to a new position. This way the participants can move free in the conference room. The localisation algorithm can be seen in figure 3.5.

After the decision who was the origin of this speech frame the single frames have to be put together again. To have no artefacts the speech is divided by overlapping hamming filters. The overlap accounts 50% and the frame length is $4 \cdot 1024$ milliseconds as mentioned above. The overlapping part is additively merged for every frame. This way the output sounds similar to the source and single mismatches in the channel assignment make no odds because of the overlap. A single false classified speech produces of course noisy sounding artefacts in the other channels. More following false assigned speech frames can corrupt the hear impression in a stronger way. The channels where nobody is assigned during one time step are set to zero. At the playback side $2 \cdot 1024$ samples can be put out with a delay of $4 \cdot 1024$ samples. That correlates, under the assumption of a sample frequency of 48 kHz, to a delay of around about 80 milliseconds. The results can be seen in chapter 4 including a plot of a conference sample after the channel assignment.

3.6.1. Reinforcement Learning for speaker recognition

A second, completely new speaker recognition system has been developed at the Institute for Data Processing. It works with Reinforcement Learning and is described in [59]. The pre-processing and the feature extraction is the same as mentioned above. But then the extracted MFCCs were made binary through calculating ten binary places and give each position a defined MFCC-range. The mean and the standard deviation from the MFCCs has to be calculated to set the range thresholds and the binary position which contains the range of a MFCC is set to one. Now it is possible to compute out of a new speech sample the speaker origin by multiplying the binary MFCCs with a speaker specific weighting function and take the maximum value. This value leads to a speaker name who should be the origin of the speech utterance. The weighting function is received from the introduction round and is updated if the localisation, saved for every speaker, and the extracted positions from the speech utterances, are identically. This is done by a reinforcement learning approach that tries to maximize the reward, which the algorithm gets if the identified person has the same position as the speech frames. Now the problem occurs that, if one time a speaker is found that gives a good reward, the algorithm will never change the identified speaker. This is solved by a probability that another speaker is tried for a sample and if the reward is better the adaptation of the weights will lead to a better speaker recognition. To achieve a more reliable speaker recognition the average over some samples is taken. For a more exact definition of this algorithm I refer to [59] and in chapter 4 the results will be discussed.

3.7. The ICSI speaker diarization system

We choose the speaker diarization system from ICSI [58] to have a maximal reachable threshold for our system, because their approach is an offline one and should work better than our speaker recognition system. At this part this approach will be introduced.

Offline means in a teleconference scenario, that the whole meeting is recorded and it is handled as a complete file afterwards. The first pre-processing steps include wiener filtering to remove corrupting sounds. The next step is a delay and sum beamformer, that reduces the recorded audio channels to one summarized channel. After that the two speaker specific features are extracted. The first are MFCCs and the second are delay features received from the beamformer. The features get different weightings, 0,65 for the MFCCs and 0,35 for the delay characteristics. This is followed by a pre-segmentation of the meeting which results in a speech, silence or non-speech label for every frame. The labelling is done by a GMM, which was calculated from broadcast news data beforehand, through which the meeting can be divided into silence and speech. The silence region is divided again in regions with high energy and one with low energy. This leads to the three different blocks that are now build with a GMM, because this makes a re-segmentation possible through which the segments are optimized. The last step is to look if the non-speech and speech GMMs are similar with a Bayesian Information Criterion, which can be seen as a distance measure but only for Gaussian Mixture Models. If the similarity is given, these two models are connected. All silence and non-speech frames are deleted. The remaining file is divided into K clusters and a model is created for everyone. Now the clusters are segmented with the ΔBIC algorithms, which is an adaptation to the normal BIC algorithm, as mentioned above, through which a decision can be made if two clusters contain the same or different speakers. Clusters that contain the same speaker were merged as long as no further junction is possible, because the ΔBIC value is to big. The result is a clustered meeting file where every cluster is labelled by a speaker number. At the end every speaker number has to be consolidated with a real speaker name.

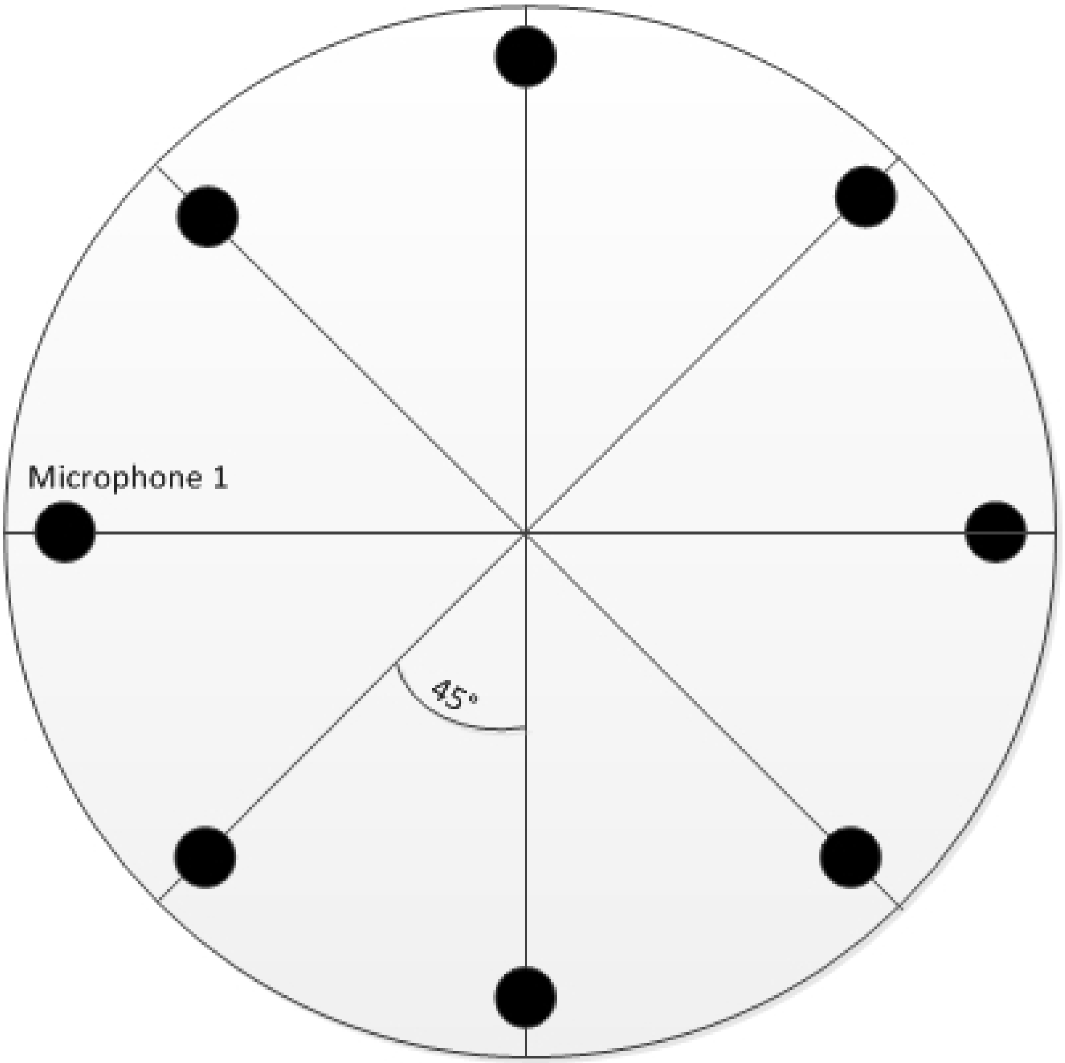

Figure 3.1.: The principle structure of the microphone array

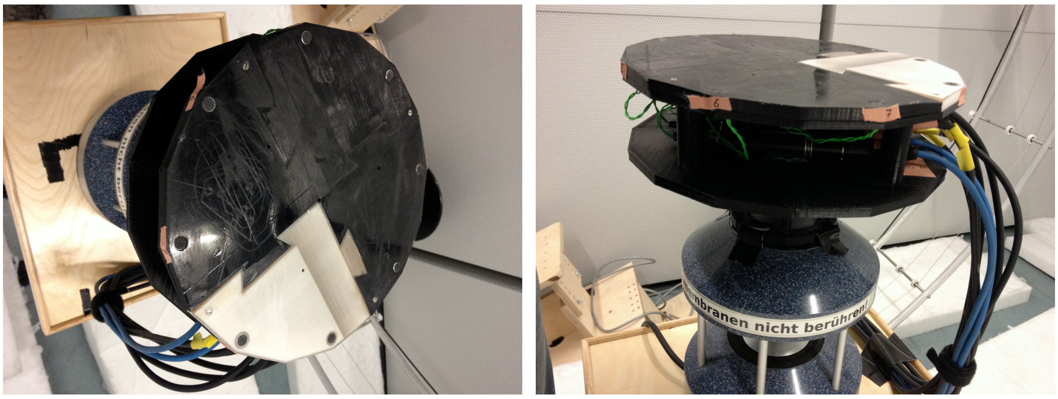

Figure 3.2.: The microphone array out of [21]

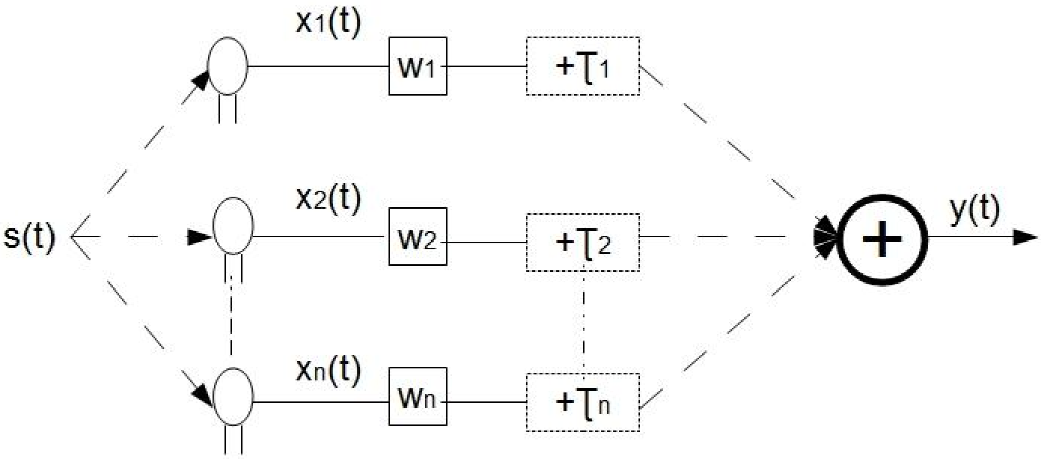

Figure 3.3.: A Delay and Sum Beamformer

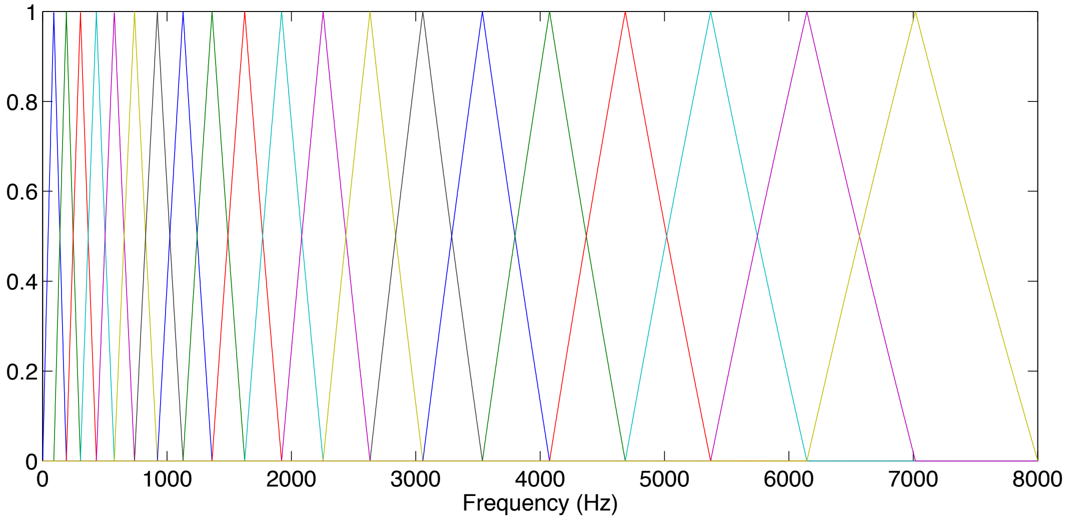

Figure 3.4.: A Mel-filter bank with triangular bandpass filters out of [33]

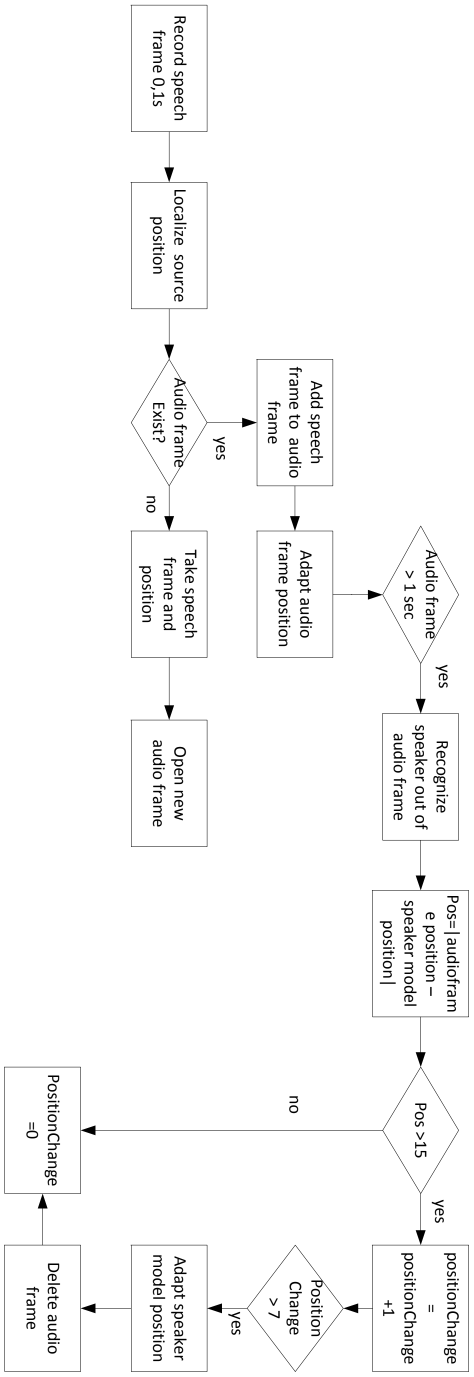
Record speech frame 0,1s
Localize source position
Audio frame Exist?
yes
no
Add speech frame to audio frame
Adapt audio frame position
Audio frame > 1 sec
yes
Recognize speaker out of audio frame
Pos=|audioframe position – speaker model position|
Pos >15
no
yes
PositionChange =0
positionChange = positionChange +1
Position Change > 7
yes
Adapt speaker model position
Delete audio frame
Take speech frame and position
Open new audio frame

Figure 3.5.: Speaker position change algorithm

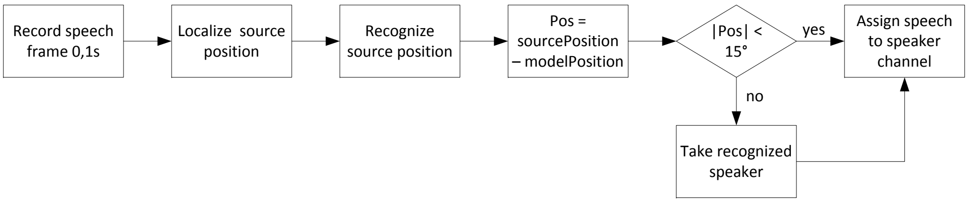
Record speech frame 0,1s
Localize source position
Recognize source position
Pos = sourcePosition – modelPosition
|Pos| < 15°
yes
no
Assign speech to speaker channel
Take recognized speaker

Figure 3.6.: Channel assignment algorithm

4. Evaluation

The evaluation is divided in three parts. In the first part the speaker recognition has to be tested to find an optimum for the parameters. A comparison to the reinforcement approach and the ICSI speaker diarization system will be discussed, too. In the second part the quality of the two main channel assignment algorithms will be evaluated with simulated meetings. In the third evaluation a real meeting is tested and here the differences between some channel assigning parameters based on the DER will be given. Additionally the diverse channel assignment approaches are evaluated on the third conference, too. In this chapter first all approaches will be repeated and will be newly labelled with a better appreciable name. The next section will show all parameters that are used for the first evaluation followed by a definition of the different audio files and recordings. Then every evaluation gets a separate section.

4.1. Evaluation & channel assignment names

In table 4.1 all algorithms are named. $Algorithm1$ consists of the speaker recognition with MFCCs and GMM. This is defined in section 3.5.3. $Algorithm2$ uses only the localisation for channel assignment. In contrast $Algorithm3$ uses additionally the speaker recognition as mentioned in $Algorithmus1$ for position change detection. If the collected utterance position deviates more then a threshold from the location of the speaker model and if this happens oftener then a threshold, the position is adapted as shown in figure 3.5. $Algorithm4$ expands $Algorithm3$ with a second application of the speaker recognition. If the position of the speech utterance deviates more then a threshold from the location of the speaker model, the speaker recognition is used to assign the utterance to a channel. Figure 3.6 show this circumstance. $Algorithm5$ uses the speaker recognition approach through reinforcement learning as developed in [59]. $Algorithm6$ is the implementation of the speaker diarization approach from the ICSI.

4.2. Parameters to evaluate

In a speaker recognition system different parameters can be tested. In our case we examined for the in table 4.2 defined parameters.

Table 4.1.: Channel assignment algorithms

Algorithm1	Speaker recognition only
Algorithm2	Localisation only
Algorithm3	Localisation combined with position verification
Algorithm4	Localisation combined with recognition and position verification
Algorithm5	Recognition through Reinforcement Learning
Algorithm6	ICSI speaker diarization approach

Table 4.2.: Parameters of speaker recognition

Frame length	0,1 s; 0,2 s; 0,25 s; 0,5 s; 1 s
Gaussian Mixture Components	16, 32, 64, 128, 256
Mel-Frequency Cepstrum Coefficients	8, 12, 16, 20
Zero-coefficient	with or without
Silence Model	trained out of UBM or without UBM
VAD	dynamical VAD or hard threshold for VAD
Length of Hamming window	10 ms, 20 ms, 30 ms
Overlap of Hamming window	30%, 40%, 50%

The Frame length was tested to see „How much real-time is possible?“. This means looking for the best compromise between a good DER and keeping the frame length as short as possible. For the GMMs Christoph Kotzielski [33] proved in his thesis that the DER pays off if the Mixtures get bigger than 64 [46]. Opposing to this the literature [31] writes that more mixtures achieve better results. So we decided to make a second evaluation. To the MFCCs in literature [31] is said, that a big amount of training data is needed to use a bigger number of coefficients. The Zero-coefficient, which can be seen as an energy offset, was not used in the original speaker recognition system and it is often mentioned that this coefficient is unreliable, but in [62] it is said that it has its rights to be compared. That is why it will be evaluated here too. The last two points have to be tested because they were newly programmed in this thesis. The window length and the overlapping duration of each hamming window can be evaluated. The Hamming window has a range from 10 milliseconds to 30 millisecond with a overlap of 30% to 50% to be short enough to contain the spectral information and on the other hand long enough to contain a good frequent resolution [32]. Another value which is not compared in this thesis is the roll off value of the pre-emphasis because it is often demonstrated that the optimum is $\alpha = 0.95$ [62, 32]. The evaluation results can be seen in section 4.4.

4.3. Selected audio files

For our evaluation we used three different corpora. The first is the AMI meeting corpus [1] which is a very often used database for evaluations in speaker diarization and recognition topics. The second is a simulated meeting which we recorded ourselves. The last is a real meeting where participants talk freely about an given topic. All together will be introduced in the next two sections.

4.3.1. AMI Meeting Corpus

The AMI meeting corpus is divided into different corpora. In this thesis the Edinburgh meeting compilation has been taken. The meeting names ES2009 to ES2016 were chosen for evaluation. Every meeting is divided into four parts. The "c" part of the ES2010 to ES2016 corpora is taken to train the UBM. "a", "b" and "d" are used for evaluation. Only a small part is separated from every meeting to train the speaker models.

In the AMI Meeting Corpus the main problem is that the audio files contain only one mono stream and with only one channel it is impossible for our algorithm to detect, in a part with overlapping speakers, more than one. For the Diarization Error Rate this is postulated. Another point was the not existing need of a correct non speech classification, as it is named in the Diarization Error Rate, because if silence is labelled as a speaker and the silence is then given out at the speakers channel it is not an issue. Of course for the later processing steps, like automatic speech detection, it is a problem, but this is not a part of this thesis. This are the causes why we decided to differ in our evaluation four cases:

- DER evaluation.
- Evaluation like DER but overlapping speech samples are omitted
- Evaluation like DER but silence samples are omitted
- Evaluation like DER but silence and overlapping speech samples are left out

The first point of the itemization is used to make our algorithms comparable and to use an international recommended standardisation. The other three are checked for our analysis, because these are the issues which are important for the evaluation of our complete system. To get the maximum achievable limitations we used an offline speaker diarization from ICSI [58] which was the best evaluated participant of NIST evaluation from 2007 [19]. We implemented this evaluation and speaker diarization algorithm because of the in section 2.9 mentioned reasons. Through the evaluation an adaptation of the system relevant parameters is possible and thus the optimal one can be found.

To use some important features of our system we decided to compute our localisation algorithm, too. The AMI meeting corpus does not locate there participants during the meeting, so we had to simulate an entrant position. For that we simulate the localisation accuracy from the thesis of Johannes Feldmaier [16], which shows a localisation accuracy of 96,9%, with a deviance of five degrees, which is a compromise of the participants behaviour in real meetings and a good theoretical value for the localisation accuracy. We gave every participant a fixed position for the whole meeting, so we could use the possibility of our algorithm to employ the speaker position for a saver update of the speaker models as mentioned in section 3.5.3.

4.3.2. Audio recordings

The second audio files we evaluated were recorded by ourself. We recorded twelve speakers, eight men and four women, with a single microphone in our anechoic room and a sampling frequency of 48 kHz. Every speaker reads five minutes a text for the meeting simulation and one minute as training material. The meeting audio files are cut and put together again to simulate a real meeting with four speakers. The audio files were only cut in silence parts. The meeting has eleven percent overlap of two simultaneously speaking participants and one percent overlap of three speaking persons at once. This is a measurement which was calculated out of diverse meeting corpora [51]. The length of each speech frame is varied from 2 seconds to 12 seconds. Every speaker is put on an own channel which makes it possible to write scripts that play every speaker on an own loudspeaker.

We put the loudspeakers in two combinations around the microphone array, which is shown in picture 4.2. In configuration one every loudspeaker has a distance to the center of the microphone array of 1,30 m and they were put in 90° degree from each other (from now on called **session1**). In the second combination two participants sit parallel to each other and adverse to the other two. The distance is again 1,30 m and the angle between the parallel sitting users is 40°, this meeting is further called **session2**. In all conferences the elevation angle from loudspeaker to the array is 20° degree. Figure 4.2 displays the circumstances. The recordings are made in an anechoic room and an echoic office room. The room characteristics can be seen in table 4.3 and 4.4 and a picture of the recording in figure 4.1.

Audiolab dimensions	4.7m x 3.7m x 2.84m				
frequency in Hz	250	500	1000	1995	3981
Audiolab reverberation time t_{60} in s	0.1008	0.0554	0.0465	0.0415	0.0416

Table 4.3.: Room characteristics of the audiolab

So we got 24 simulated conference combinations out of the three arranged meetings, each with four different speakers. The sample frequency was again 48 kHz. Addition-

Videolab dimensions	6.3m x 4m x 2.8m				
frequency in Hz	250	500	1000	1995	3981
Videolab reverberation time t_{60} in s	0.2545	0.2169	0.2230	0.2466	0.2149

Table 4.4.: Room characteristics of the videolab

ally the temperature has to be measured during every part, because it is needed for the localisation algorithm.

The last recorded conference is a real one, which is used to evaluate the influence of the speaker recognition to the channel assignment via localisation. This means it is measured how much better the DER gets if the speaker identification fills the gaps of the localisation assignment and proves if a speaker has moved. In the recordings four participants stands around a microphone array like in **session2**. During the meeting a person changes their position to an fictive white board. At this new position the participant is talking round about 20 seconds and moves then back. This happens twice. The real meeting consists of round about 10 seconds introduction of every speaker and the meeting takes 5 minutes and is divided into two parts. The first one consists hard speaker positions and the second part is the one where the speakers are moving. I decided to part the conference this way, because I want to show the differences of $Algorithm2,\ 3$ *and* 4. In the theory $Algorithm4$ should treat best with the moving speakers and a bit better then $Algorithm3$ with the standing participants, because the fourth uses the speaker recognition to fill gaps between model position and utterance location. $Algorithm2$ should work worse as the other two. $Algorithm1$ is also applied to prove its quality in a real scenario.

The last table 4.5 gives an overview which algorithm is used in which evaluation. The tests with the AMI meeting corpus were made to find a parameter optimum and to decide between $Algorithm1$ and $Algorithm5$ for the later channel assignment. $Algorithm6$ was also evaluated to get an idea about a minimal reachable DER threshold. The simulated meetings should only show how good the developed channel assignment algorithms work and in contrast should the evaluation of the real conference point out how important a parameter optimization is. Therefore the four different channel assignment algorithms are here compared.

Table 4.5.: Relationship between evaluation and algorithms

AMI meeting evaluation	Algorithm1, 5, 6
Simulated meeting evaluation	Algorithm1, 3, 6
Real conference	Algorithm1, 2, 3, 4

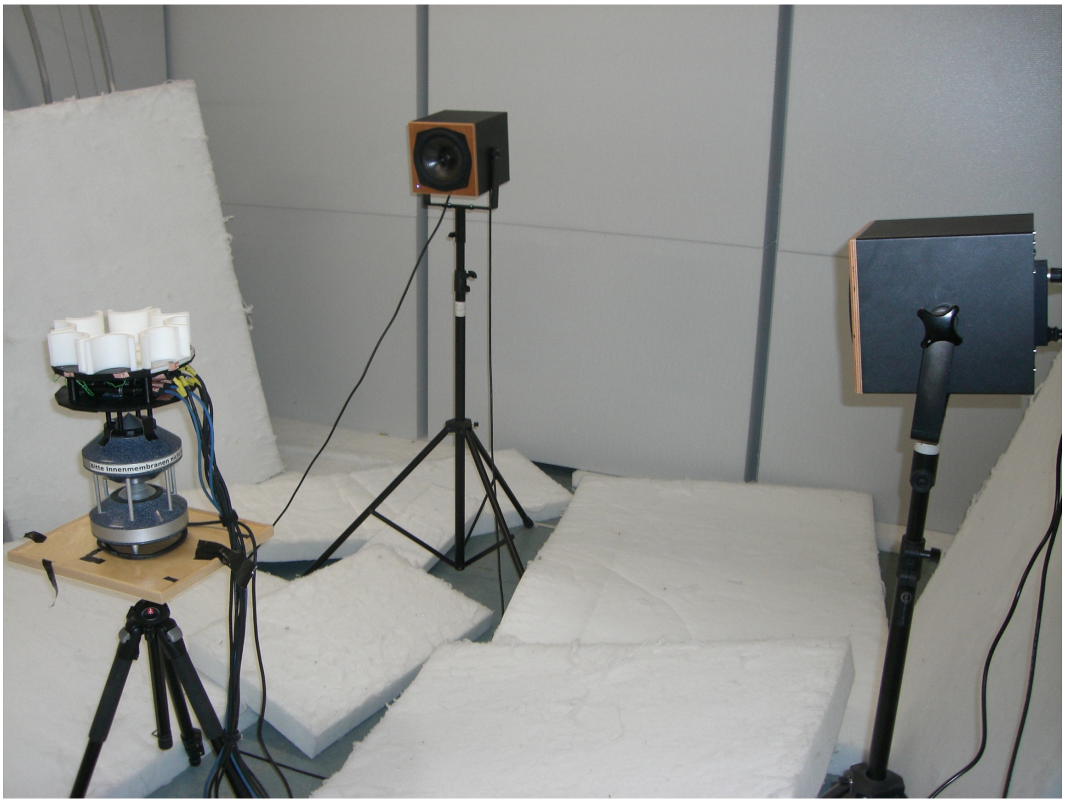

Figure 4.1.: Recordings in the audiolab

4.3.3. Ground truth of recorded meetings

The ground truth for every simulated meeting is needed to calculate the DER. So the ground truth can be made manually, which is very time consuming, or automatically, like it is done in this case. A program was written, that looks into the files, created for the recording, for ten samples in a row which are bigger than a threshold. If this happens the part is labelled as speech till ten samples that are lower than the threshold. There silence begins again. This is done for every speaker channel and so a ground truth is calculated and saved in an Excel file. The ground truth for the real meeting has to be done via hand.

4.4. Results of the AMI meeting corpus evaluation

In the evaluation of the AMI meeting corpus the above mentioned parameters has to be compared. The following statistics show a separate bar for every meeting. In table 4.6 are standard values for $Algorithm1$ mentioned. They are valid for every following evaluation, if

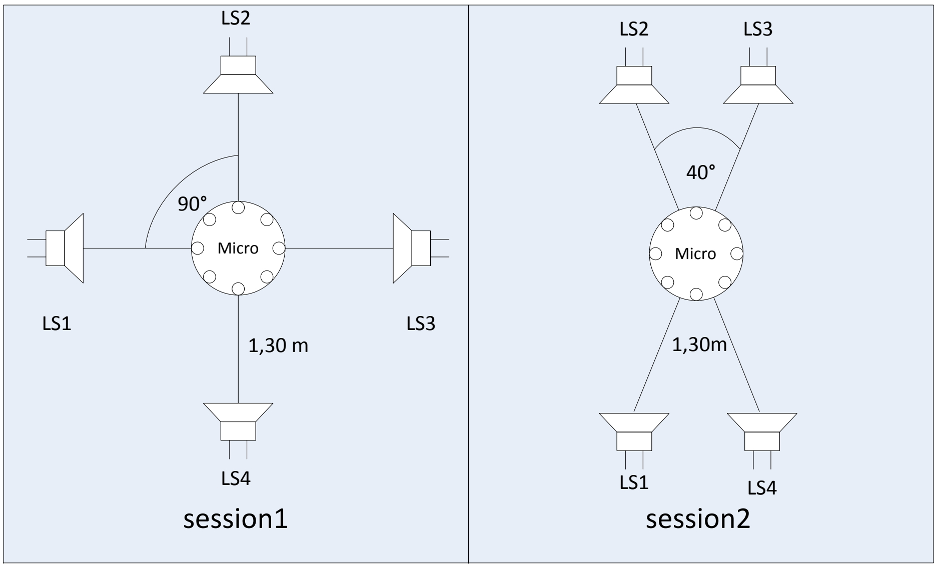

Figure 4.2.: The two meeting combinations for the different participant positions.

no other values are given. This choice is picked, because it is the standard in the literature or is necessary for our system.

First of all it is shown how good the developed $Algorithm1$ is. Differences in frame lengths are compared in figure 4.3 to show what the possibilities are if there is no real-time condition. With a frame length of one second DER up to 28% are feasible but a smaller one produces a much higher DER, for example 0,1 seconds increase the error to a maximum of 68%. This test is also needed to get a good frame length for the, in the channel assignment of $Algorithm3$ *and* 4 used, position verification.

Figure 4.4 pointed the distinction between the number of GMM components out. Additionally a comparison to the $Algorithm6$ from ICSI is possible which shows good DERs up to 20%. The developed speaker recognition displays with 256 GMM components a DER of 47%, which is its best result. The larger the number of the GMMs as smaller becomes the DER, but this is bought with a bigger computing effort.

The MFCC number seems to achieve similar results as the GMM evaluation. As higher the number as better gets the DER. 46% is the top mark with 20 MFCCs. So 0,1 seconds consists of enough speaker information for this many MFCCs. Again is the gain won by a

Table 4.6.: Parameters of speaker recognition

Frame length	0,1 s
Gaussian Mixture Components	128
Mel-Frequency Cepstrum Coefficients	12
Zero-coefficient	without
Silence Model	trained out of UBM
VAD	hard threshold for VAD
Length of Hamming window	30 ms
Overlap of Hamming window	50%

bigger calculation effort. The Zero coefficient has no big influence to the DER, but it seems to make it worse. This can be seen in figure 4.5.

In the next diagram 4.6 the different silence treatments are compared. The best result achieves the static VAD, which uses a hard coded threshold in the combination with a silence model that is trained without using an UBM. This is caused by the big speech parts in the UBM and maybe an UBM, consisting of silence only, can put the things right.

Figure 4.7 shows how the influence of the window sizes and overlaps referred to the DER is. In the average a window length of 10 ms and 30% overlap seems to be advantageous, but it makes no significant difference.

The last evaluation shows how much overlapping speech and silence the AMI meeting corpus is consisting of. This is important for the rating of the developed algorithm, because in overlapping speech only one speaker is detectable and silence is not countable for the DER if it is put at a channel. Figure 4.8 shows that the influence lies in a round about 10% better DER.

At last the speaker recognition based on reinforcement learning is evaluated, what is depicted in figure 4.9. First of all reinforcement learning seems to accomplish really good results if the speaker model is adapted the whole time. But if there is no adaptation the results become worse. This is because of the reward that is necessary to achieve a robust speaker recognition and make the algorithm time memorial. To get a reward the speaker position was simulated beforehand. So the question is now, why to use the speaker position as reward and not for speaker recognition as itself? In the thesis [21] it is shown that the localisation accuracy achieves much better results as the speaker identification via reinforcement learning. An exactly description of this approach can be found in section

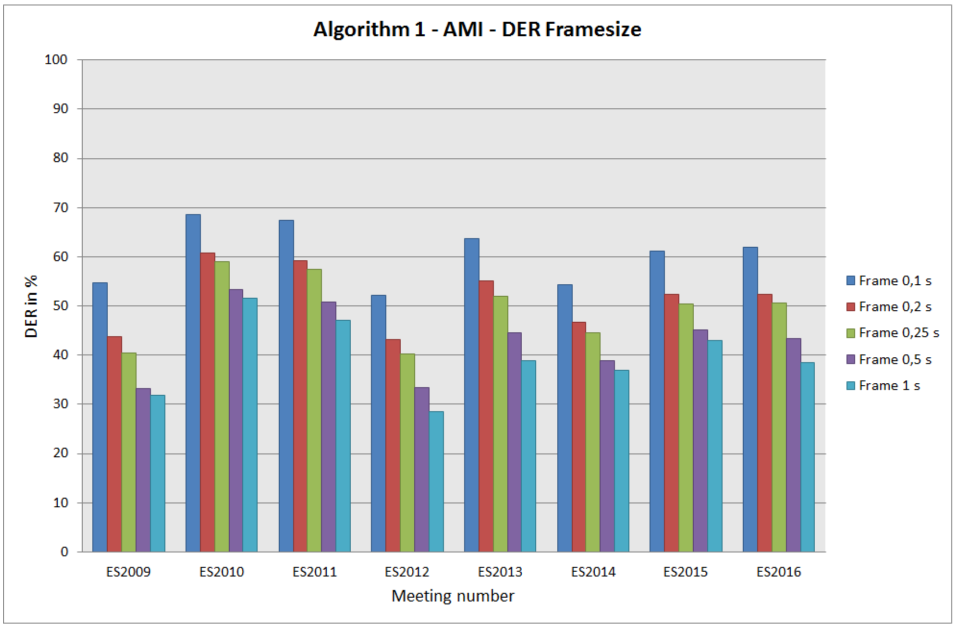

Figure 4.3.: DER for $Algorithm1$ evaluated with the AMI meeting corpus referred to different frame lengths

3.6. Furthermore there is no right localisation if the reinforcement learning should carry the task of $Algorithm1$ in $Algorthm4$.

The best parameters are recapitulated in table 5.1 and used for the further evaluation of the two recorded conferences.

4.5. Evaluation of the simulated conferences

In this section the results of the recorded conference evaluation will be discussed. The main attention lies at a statement about the goodness of the whole developed system. Furthermore a comparison between $Algorithm1$ and $Algorithm4$ is made. The first figure 4.10 pointed out that the assignment with $Algorithm4$ is twice as good as the assignment through $Algorithm1$. This result can be traced over all simulated conference scenarios and the different room characteristics. In the best case a DER of 17% is possible. If the silence is left out of the DER a better value of 9% is achievable. The DER of the videolab is basically worse than in the audiolab and reach 21% in the best case. If silence is omitted 8% can be won to a DER of 13%. The two varying conference setups seems to have no big influence. At the audiolab the **session2** achieves a bit better results. In the videolab it is inverted. In the audiolab the DER is in the mean 4% better than in the other room and if the silence is omitted in the DER a value of 8% is possible. Furthermore $Algorithm6$ from the ICSI was applied to the simulated recordings. It achieves the best results with a

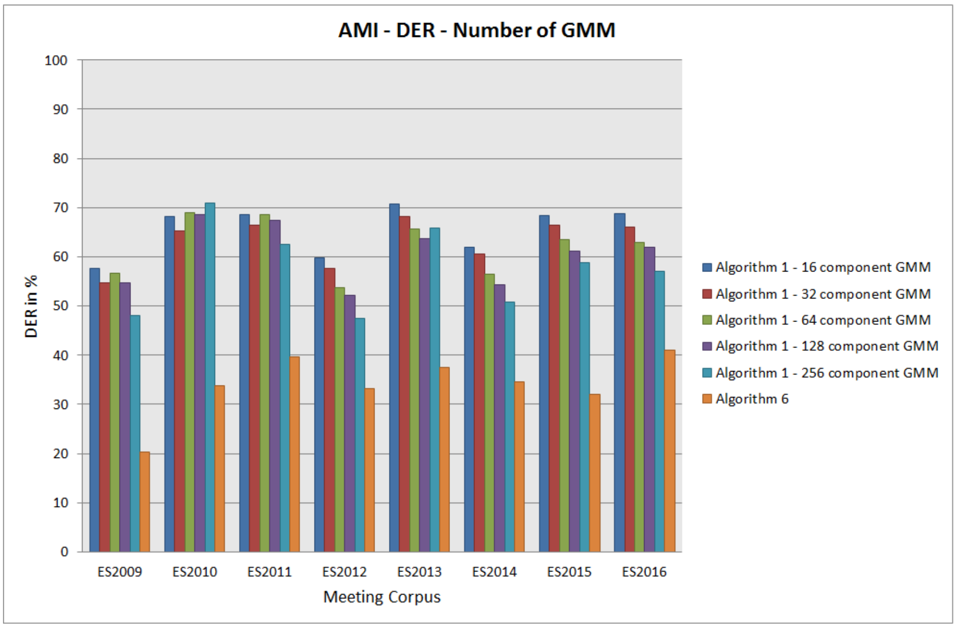

Figure 4.4.: DER for $Algorithm 1$ *and* 2 evaluated with the AMI meeting corpus referred to a different number of GMM components.

DER of 4% in the optimum. Only in conference 2 it achieves a worse DER, because one speaker is not detected by the algorithm.

In figure 4.11 two things are pictured. The first is a difference in the frame length and its effect to the DER of $Algorithm1$. The second shows what happens if the tolerance of the angle difference between the speech utterance and the speaker model is set to 10° or 15° in $Algorithm4$. The frame length was tested to get a knowledge about the quality that is achievable to verify the speaker position. A length of 1 second reaches the best DER with 19% in the videolab. Of course this result must be verified in a real meeting. The change of the localisation tolerance shows that there is only a difference of 0,5% in the DER of $Algorithm4$.

The last figure in this section shows the outputs after the channel assignment. In the left figure $Algorithm4$ is used and the wrong assigned speech utterances are pointed out with a black circle to highlight them. The right one shows the same utterance after an assignment by $Algorithm1$. Here it can be seen that in every channel false assignments occur. Accordingly is the hear impression very bad.

So a clear decision to $Algorithm4$ is fallen in this thesis, because it has the best DER and the channel assignments shows nearly zero false assigned speech utterances. Normally errors occur if a new speaker starts and if more participants talk simultaneously.

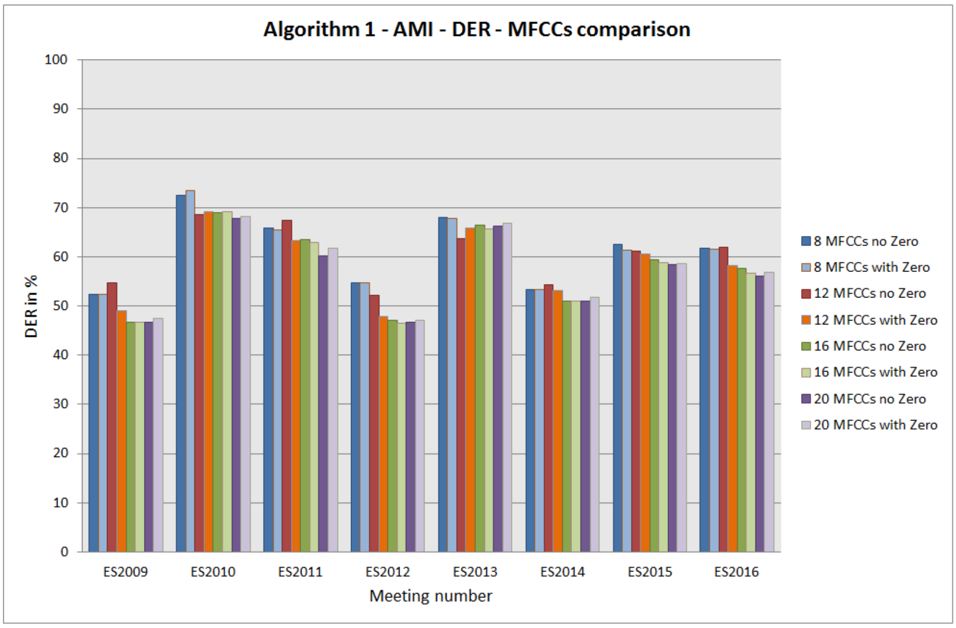

Figure 4.5.: DER for $Algorithm1$ evaluated with the AMI meeting corpus referred to a different number of MFCCs and with or without the Zero coefficient.

4.6. Evaluation of the recorded real meeting

According to the evaluations of the simulated meeting we also want to test a real meeting. Additionally we want to know how big the influence of the speaker recognition is, in its two tasks, to the channel assignment with $Algorithm4$. Figure 4.13 pointed again out that $Algorithm1$ achieves very bad DER results. The other two beams show how big the influence of moving speakers is. If there is no movement $Algorithm4$, with its best parameters, is a bit better than $Algorithm2$. Thus it can be said that the speaker recognition improves the channel assignment. In the case that there is a position change during the meeting, the results are inverted. In my opinion the cause can be found in the not optimal set parameters and in the short position change time. The parameters for position change detection via speaker recognition could not be optimized during this thesis, because of the lack of experience and the enormous time effort for testing. In the introduced case this means that the position change is detected late and then the speaker moves back soon. Thus a bit higher DER is achieved. I think in a real meeting a position change will last longer as in the tested case, because the person will usually change his position to explain something at, for example, the white board. However with optimal parameters $Algorithm4$ should maybe beat $Algorithm2$. Anyhow I have to pointed out that a DER of 23% for the position change section and 21% for the hard positions is possible if the silence is omitted.

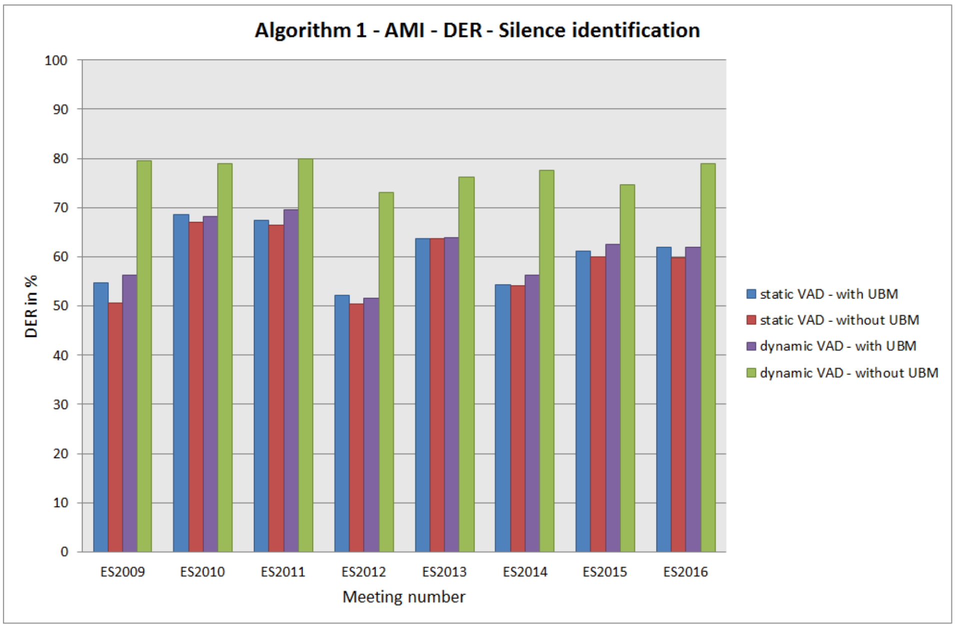

Figure 4.6.: DER for $Algorithm1$ evaluated with the AMI meeting corpus referred to a difference in silence treatment.

These values also show, that the meeting with the position changing participants has a much bigger silence part.

The next diagram 4.14 shows why it was so hard to find the optimal parameters. $Algorithm4$ is compared with $Algorithm3$ and the parameters for position change detection were the same. Table 4.7 displays the exact values. In the figure can be seen that $Algorithm4$ is always better if no movement in a conference occurs. Again here can be said that the speaker recognition improves the channel assignment via $Algorithm4$. But if there a participant changes his position bad chosen parameters made $Algorithm4$ fall into an abyss. On the other side good chosen parameters improve the system against $Algorithm3$ even in the case of a moving person. But again I have to mention here, that the recorded real meeting was only a first test with a short duration, to get fast as many evaluation results as possible. But because of that the speakers are not long enough at the new position. I think to get the most suitable results a longer meeting has to be recorded and more parameter evaluations have to be done. However for the first short evaluation of the developed $Algorithm4$ the results are auspicious.

As a comparison of all tested algorithms in the real conference, I want to refer to table 4.8.

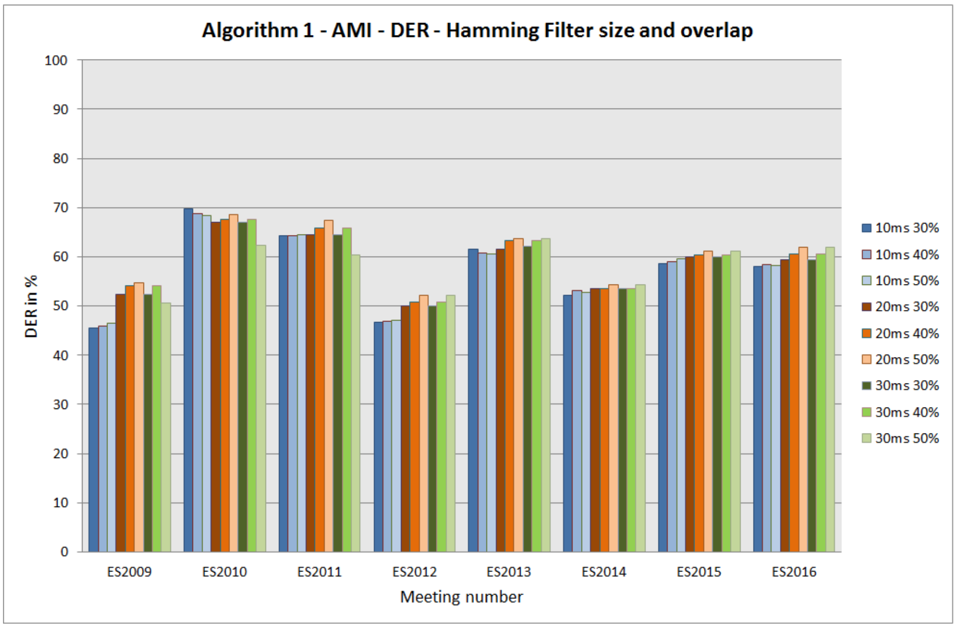

Figure 4.7.: DER for $Algorithm1$ evaluated with the AMI meeting corpus referred to a difference in the window length and the overlap of the Hamming filters.

Table 4.7.: Good and bad parameters for $Algorithm4$

	good parameters	bad parameters
Position Change detection	7	7
Frame length	1s	0,3s
Angle tolerance	10°	5°

Table 4.8.: Best DER of the different algorithms in the real meeting

	no movement	with position change
$Algorithm1$	51%	65%
$Algorithm2$	24%	33%
$Algorithm3$	24%	38%
$Algorithm4$	23%	34%

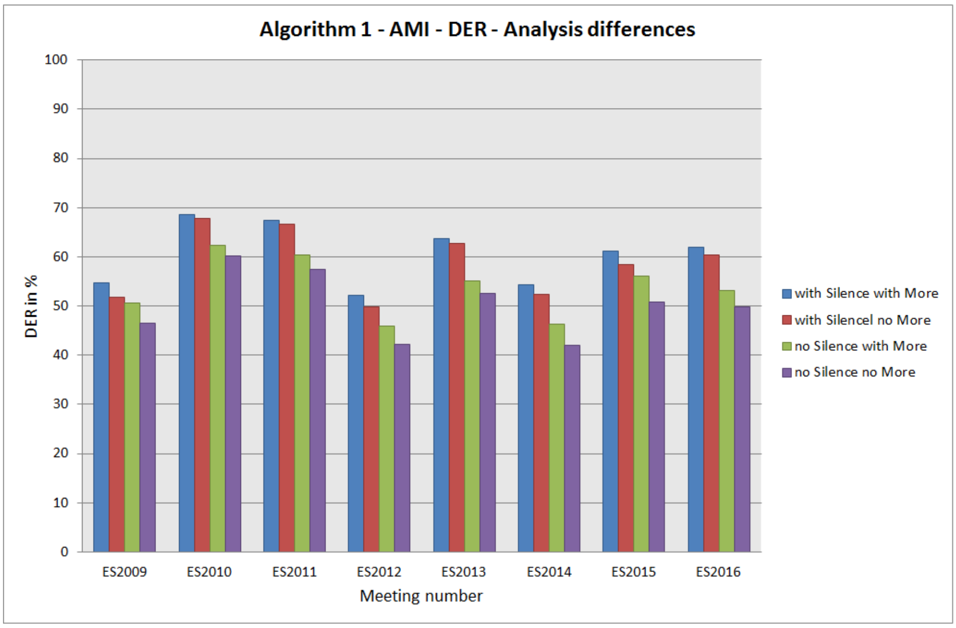

Figure 4.8.: Differnet DER for $Algorithmus1$ referred to silence and overlapping speech in the AMI meeting corpus.

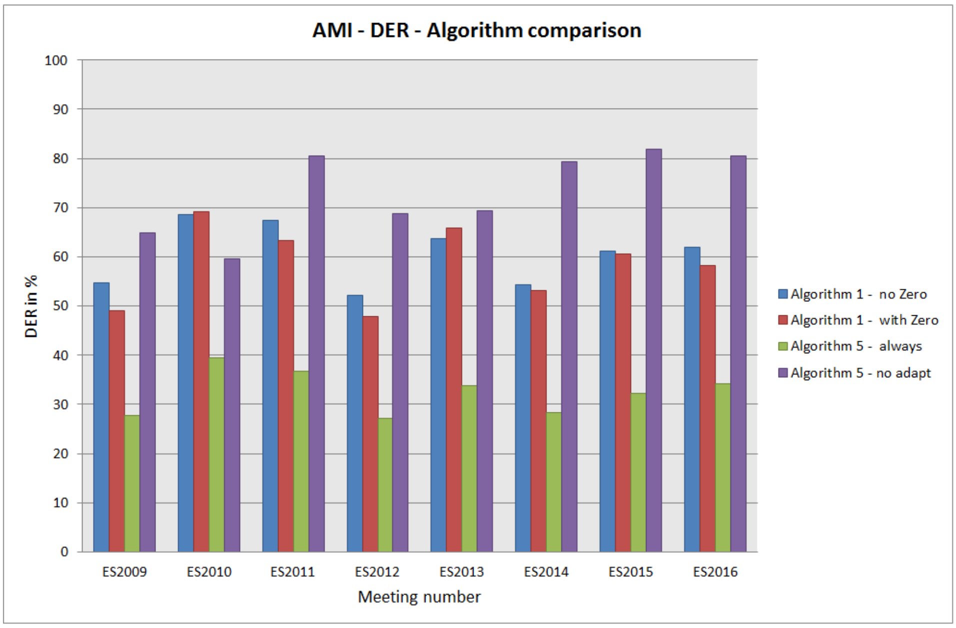

Figure 4.9.: DER for a comparison between $Algorithmus1$ (with and without zero coefficient) and $Algorithmus5$ (with adaptation and without).

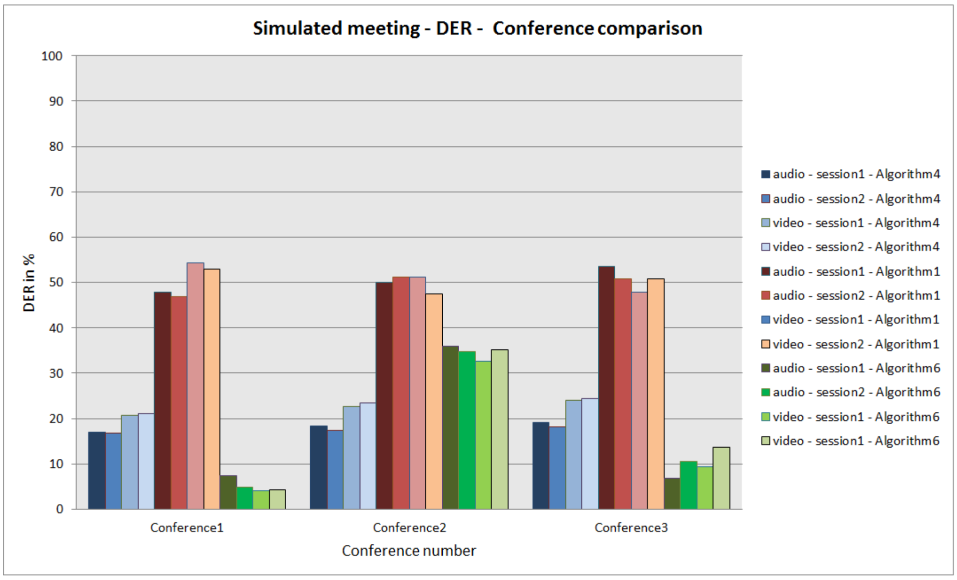

Figure 4.10.: DER for the conference simulations at the audio- and videolab for the two meeting variations. $Algorithm1$ and $Algorithm4$ are tested.

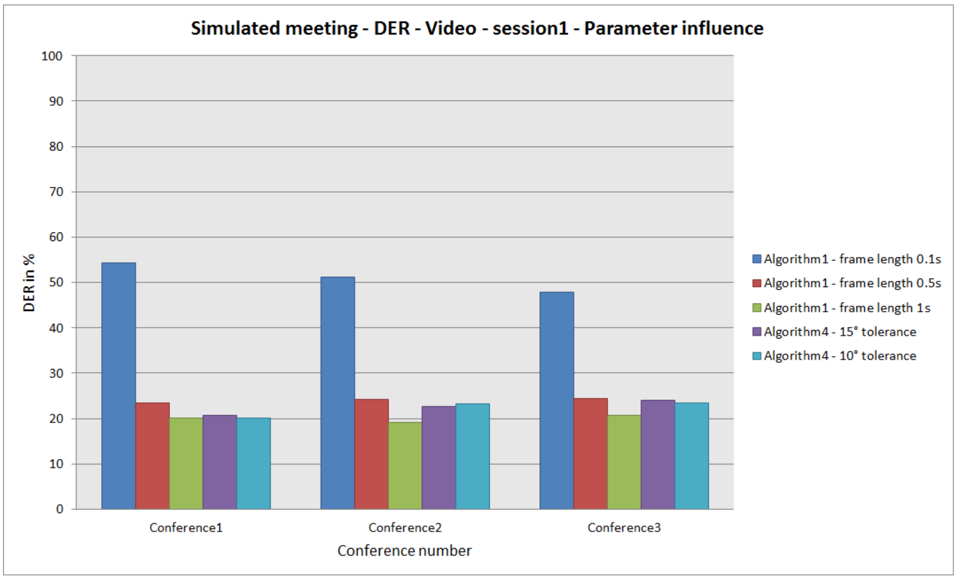

Figure 4.11.: DER for the conference simulations in the videolab with **session2** under different parameter values for $Algorithm1$ and $Algorithm4$.

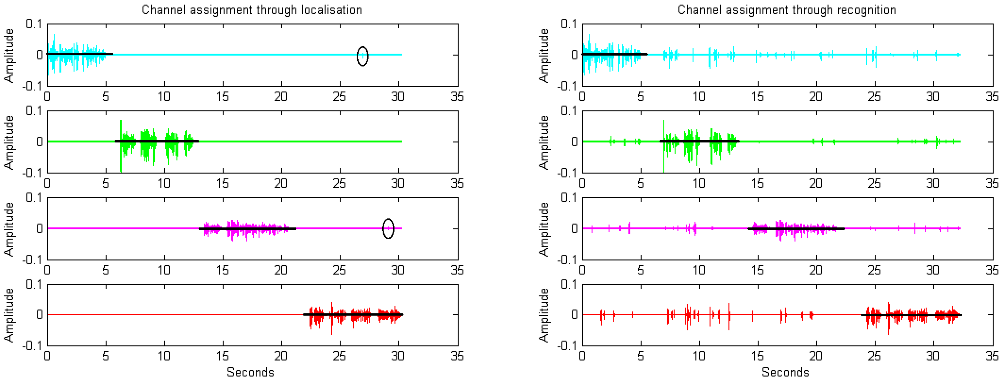

Figure 4.12.: The single speaker channels after the processing. The black line displays the active speaker. The black circle pictures a wrong assigned speech utterance with $Algorithm4$.

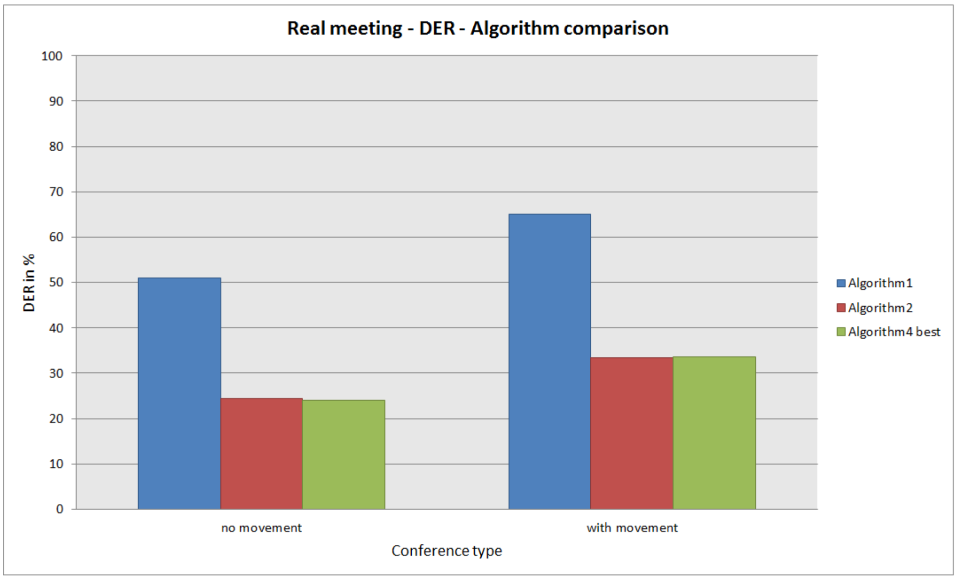

Figure 4.13.: DER for the real conference in the videolab referred to $Algorithm1$, $Algorithm2$ and $Algorithm4$.

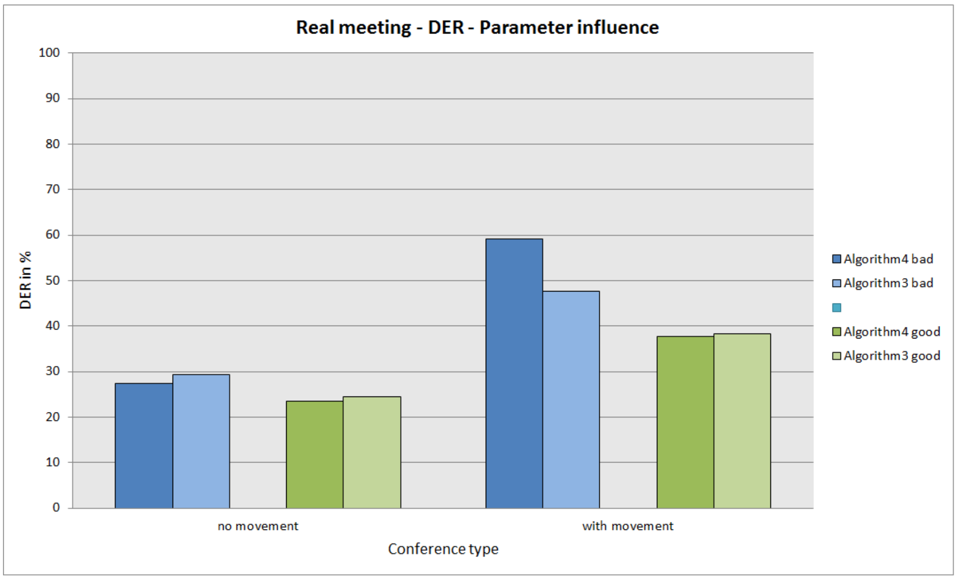

Figure 4.14.: DER for the real conference in the videolab under different parameter values for $Algorithm3$ and $Algorithm4$.

5. Conclusion & Outlook

In this chapter I will give a short conclusion to the received results and developed algorithms. Furthermore I will write about some additionally ideas to improve the introduced system and at this point some additionally evaluations will be discussed, too.

5.1. Conclusion

The main point of this thesis was to combine the existing single algorithms and devices in an adequate manner and evaluate then the whole system. Additionally a channel assignment algorithm is developed, which has the task to put the speech of every speaker at his own channel to make 3D-sound possible. This is strongly dependent from speaker recognition and thus the parameters of this algorithm are evaluated to find the best. Afterwards the whole system is evaluated with simulated meetings that are recorded and handled by the developed system. At least a small real conference is recorded and processed through the algorithms.

The recordings were done by a microphone array with eight microphones. The SRP-PHAT algorithm locates for every speech utterance the source position. The so achieved results are used to separate overlapping speakers and noise with the Geometric Source Separation. Now the speaker recognition, which consists of MFCCs modelled in a GMM, is applied. The speaker recognition is improved through adapting every speaker model from an UBM and verifying the computed speaker with the saved localisation. Another approach to identify the speaker is the introduced reinforcement algorithm, which uses the localisation as reward. The channel assignment is implemented in four algorithms. The first uses the speaker recognition and the second the sound source localisation stand alone. The third is implemented with the localisation combined with the speaker identification as position change detection and adaptation. The fourth uses the same as the third but extended with a localisation misclassification correction through speaker recognition.

The parameter evaluation for $Algorithm1$ achieves the in table 5.1 mentioned results as its best for a frame size of 0,1 seconds. Of course the recognition scores a better DER if the frame length becomes larger. How expected the speaker diarization algorithm from the ICSI has a better DER than the developed online speaker recognition. The reinforcement learning approach for speaker identification achieves pretty good results (a minimal DER of 27% is achievable), but fails on the absence of a time memory from speaker characteristics. This can be seen if the adaptation of the speaker model through the reward is turned off, then a minimal DER of 59 % can be received. Anyhow this algorithm does not reach the DER of a localisation approach as it is used in this thesis.

Table 5.1.: The best parameters for $Algorithm1$

Gaussian Mixture Components	256
Mel-Frequency Cepstrum Coefficients	20
Zero-coefficient	without
Silence Model	trained without UBM
VAD	hard threshold for VAD
Length of Hamming window	not significant
Overlap of Hamming window	not significant

In the simulated meetings it can be seen how strong the whole system works. If the silence is left out of the DER the speech utterances which are assigned to the false channel can be located at 13% with a frame size of 0,1 seconds. This small value combined with the frame size lead to an output where, in opinion of the author, no error is audible if only one speaker is active. If there are more active speakers there occurs the problem, that the localisation not only assigns the, through the separation dominant speaker to the right channel but rather the not dominant. So here is an error audible.

At the end the Institute for Data Processing has developed a good teleconference system at the recording side which achieves a well DER and the audio stream after the computing sounds in the most cases as good as the input. The system altogether is not implemented elsewhere and can easily keep up with the state of the art.

5.2. Outlook and improvement ideas

The developed channel assignment algorithm $Algorithm4$ has very much parameters which has to be adapted to optimize the whole system in a real conference scenario. That is not realizable with such a short evaluation time as it was the case in this work. Longer and more conferences has to be recorded and tested. It is proven that the discussed system works for simulated meetings and it shows first good results in a real meeting. But to demonstrate this and to optimize the parameters more evaluations must be done. To mention some parameters the location tolerance, the position change counter threshold and the frame size for position checking will be named here.

The reinforcement learning approach is promising a good DER but can not be used in the actual state, because of the lack of time memory. Some additions can be done like a speaker model for every letter or as mentioned in [59] a lip movement camera can be used as reward. Of course this is no desirable approach for this thesis, because it needs visual information.

A further idea, to improve the system is to save the received audio stream as long as the position does not change. This way a larger frame can be used if the localisation fails and the speaker recognition has to do the channel assignment. I think this concept has a great potential. Furthermore can the idea be used to find faster a new position for moving speakers.

A compensation or normalisation method is not implemented yet, too. So an algorithm of section 2.5 can be programmed in the future.

A. Appendix

A.1. Audio Processing Parameters

Table A.1.: Parameters of audio processing

General Audio Processing Parameters	
Recording frequency	48 kHz
Sampling frequency	16 kHz
Quantization	16 bit
FFT length	10 ms, 20 ms, 30 ms
Window overlap	30%, 40% 50%
Window type	Hamming
Feature Extraction Parameters	
Number of features	8, 12, 16, 20 with 1_{st}, 2_{nd} order derivatives
Preemphasis roll-off α	0.95
Number of triangular mel filters	20
Voice Activity Detection Parameters	
Energy threshold	30 dB
Minimum number of speech frames to count it as speech	3
Maximum number of silence frames to end speech	8
Gaussian Mixture Models Parameters	
Number of mixture components	16, 32, 64, 128, 256
EM convergence threshold	10^{-5}
MAP relevance factor r	16

A.2. List of acronyms

Table A.2.: Used acronyms

Acronym	Advertised
BIC	Bayesian Information Criterion
BSS	Blind Source Separation
EM	Expectation Maximization
FFT	Fast Fourier Transformation
GMM	Gaussian Mixture Model
GSS	Geometric Source Separation
HMM	Hidden Markov Model
MAP	Maximum a Posteriori
MFCCs	Mel Frequency Cepstral Coefficients
ML	Maximum Likelihood
MLLR	Maximum likelihood linear regression
LPCCs	Linear Prediction Cepstrum Coefficients
PCA	Principle Component Analysis
SEND	Spherically symmetric exponential norm distribution
SRP-PHAT	Steered Response Power - Phase Transform
SSL	Spherically symmetric Laplacian distribution
STFT	Short Time Fourier Transformation
TDoA	Time Difference of Arrival
UBM	Universal Background Model
VAD	Voice Activity Detection

A.3. AMI Corpus

Meeting	**s001**	**s002**	**s003**	**s004**
ES2009	m0034 ev	m0033 ev	m0035 ev	f0036 ev
ES2010	f0037 ev	f0038 ev	f0039 UBM	f0040 ev
ES2011	f0043 ev	f0041 ev	f0044 UBM	f0042 ev
ES2012	m0045 ev	f0047 ev	f0046 UBM	m0048 ev
ES2013	f0049 ev	f0050 ev	f0051 UBM	f0052 ev
ES2014	m0053 ev	m0054 ev	f0055 UBM	m0056 ev
ES2015	f0057 ev	f0060 ev	f0058 UBM	f0059 ev
ES2016	m0061 ev	f0064 ev	m3062 UBM	m0063 ev

Table A.3.: Overview of meeting participants per meeting with annotated gender and the useage in the evaluation

A.4. 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.4.: In this work developed algorithms

File name	Function
AMICorpusTestenWithMore.m	AMI evaluation no silence
AMICorpusTestenWithOut.m	AMI evaluation no silence and overlaps
AMICorpusTestenWithSil.m	AMI evaluation no overlaps
AMICorpusTestenWithSilMore.m	AMI evaluation after DER
conferenceAngle.m	Assigns the speaker channels with localisation
conferenceRec.m	Assigns the speaker channels with recognition
direction.m	Extracts the speaker position out of a file name
evaluationICSI.m	Calculated the DER out of the ICSI diarization results
groundTruth.m	Calculated the ground truth out of a recording file
improveModel.m	adapts a speaker model with the help of the position
konferenz.m	Channel assignment via localisation
konferenzAngle.m	DER conference evaluation + chan. assign. via loc
konferenzRec.m	DER conference evaluation + chan. assign. via rec
positionChange.m	Proves if the speaker position has changed
PROPERTIES.m	Defines all important parameters centrally
SkriptAMI.m	Calls the single AMI evaluations
SkriptKonferenz.m	Calls the single conference evaluations
trainUBM.m	Train an UBM

Table A.5.: Matlab functions and tools out of [33] and [21]

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
folder gss	Consists of all matlab algorithms to localize and separate a speaker
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.5. DVD content

- The Edinburgh recordings of the AMI meeting corpus together with the ground truth for every meeting
- The recorded conferences and there ground truth
- The results of the evaluation
- The ICSI speaker diarization system with the evaluation results
- The above mentioned matlab files
- The diploma thesis as pdf and latex file

List of Figures

Bibliography

[1] AMI Consortium. The AMI Meeting Corpus. URL `http://corpus.amiproject.org`. Accessed at 10.03.2013.

[2] A.M. Arthur, R. Lunsford, M. Wesson, and S. Oviatt. Prototyping novel collaborative multimodal systems: simulation, data collection and analysis tools for the next decade. In *Proceedings of the 8th international conference on Multimodal interfaces*, pp. 209–216. 2006.

[3] R. Auckenthaler and J.S. Mason. *Gaussian selection applied to text-independent speaker verification*. 2001.

[4] H. Beigi. *Fundamentals of speaker recognition*. Springer, New York, 2011.

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

[6] K. Bernardin and R. Stiefelhagen. Audio-visual multi-person tracking and identification for smart environments. In *Proceedings of the 15th international conference on Multimedia*, pp. 661–670. 2007.

[7] L. Besacier and J.F. Bonastre. Subband architecture for automatic speaker recognition. In *Signal Processing*, 80(7), pp. 1245–1259, 2000.

[8] C. Busso, S. Hernanz, C.W. Chu, S.i. Kwon, S. Lee, P.G. Georgiou, I. Cohen, and S. Narayanan. Smart room: participant and speaker localization and identification. In *International Conference on Acoustics, Speech, and Signal Processing.*, volume 2, pp. ii–1117. 2005.

[9] 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 & Language*, 20(2), pp. 210–229, 2006.

[10] W.M. Campbell, D.E. Sturim, and D.A. Reynolds. Support vector machines using gmm supervectors for speaker verification. In *Signal Processing Letters*, 13(5), pp. 308–311, 2006.

[11] U.V. Chaudhari, G.N. Ramaswamy, G. Potamianos, and C. Neti. Information fusion and decision cascading for audio-visual speaker recognition based on time-varying

stream reliability prediction. In *International Conference on Multimedia and Expo.*, volume 3, pp. III–9. 2003.

[12] 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.

[13] Chin, D. Next Generation Video Conferencing. In *Arkadian Global Conferencing*, pp. 3–4, 2011.

[14] G. Doddington et al.. Speaker recognition based on idiolectal differences between speakers. In *Proceedings Eurospeech*, volume 1, pp. 2521–2524. 2001.

[15] M. Durkovic. Localization, Tracking, and Separation of Sound Sources for Cognitive Robots. 2012. Dissertation at the *Institute for Data Processing, Technical University of Munich*.

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

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

[18] J. Garofolo. The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan. In *RT-06S Transcription Evaluation Plan*, 2009.

[19] J. Garofolo, J. Fiscus, and J. Ajot. Spring 2007 (RT-07) Rich Transcription Meeting Recognition Evaluation Plan. In *RT-06S Transcription Evaluation Plan*, 2007.

[20] J. Garofolo, J. Fiscus, and J. Ajot. The NIST Year 2012 Speaker Recognition Evaluation Plan. In *2012 NIST Speaker Recognition Evaluation*, 2012.

[21] T. Grasser. Speaker Localization and Separation in Teleconferences . 2013. Diploma thesis at the *Institute for Data Processing, Technical University of Munich*.

[22] F. Grondin and F. Michaud. Wiss, a speaker identification system for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA).*, pp. 1817–1822. 2012.

[23] J. Gudnason and M. Brookes. Voice source cepstrum coefficients for speaker identification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4821–4824. 2008.

[24] V. Hautamaki, T. Kinnunen, I. Karkkainen, J. Saastamoinen, M. Tuononen, and P. Franti. Maximum a posteriori adaptation of the centroid model for speaker verification. In *Signal Processing Letters*, 15, pp. 162–165, 2008.

[25] R.M. Hegde, H.A. Murthy, and G.V.R. Rao. Application of the modified group delay function to speaker identification and discrimination. In *IEEE International Conference on Acoustics, Speech, and Signal Processing.*, volume 1, pp. 1–517. 2004.

[26] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. In *The Journal of the Acoustical Society of America*, 87, p. 1738, 1990.

[27] T. Hori, S. Araki, T. Yoshioka, M. Fujimoto, S. Watanabe, T. Oba, A. Ogawa, K. Otsuka, D. Mikami, K. Kinoshita et al.. Real-time meeting recognition and understanding using distant microphones and omni-directional camera. In *Spoken Language Technology Workshop*, pp. 424–429. 2010.

[28] X. Huang, A. Acero, and H.W. Hon. *Spoken language processing*, volume 15. Prentice Hall PTR New Jersey, 2001.

[29] K. Ishiguro, T. Yamada, S. Araki, T. Nakatani, and H. Sawada. Probabilistic speaker diarization with bag-of-words representations of speaker angle information. In *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2), pp. 447–460, 2012.

[30] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel. Speaker and session variability in gmm-based speaker verification. In *IEEE Transactions on Audio, Speech, and Language Processing.*, 15(4), pp. 1448–1460, 2007.

[31] T. Kinnunen and L. Haizhou. An overview of text-independent speaker recognition: From features to supervectors. In *Speech communication*, 52.1, pp. 12–40, 2010.

[32] T. Kinnunen. Spectral features for automatic text-independent speaker recognition. In *Licentiate's Thesis*, 2003.

[33] C. Kozielski. Online Speaker Recognition for Teleconferencing Systems. 2011. Diploma thesis at the *Institute for Data Processing, Technical University of Munich*.

[34] C. Leggetter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. In *Computer speech and language*, 9(2), p. 171, 1995.

[35] E. Lleida, J. Fernandez, and E. Masgrau. Robust continuous speech recognition system based on a microphone array. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 1998.*, volume 1, pp. 241–244. 1998.

[36] S. Makino, T.W. Lee, and H. Sawada. *Blind Speech Separation*. Signals and Communication Technology. Springer Dordrecht, 2007.

[37] E. Martinson and W. Lawson. Learning speaker recognition models through human-robot interaction. In *IEEE International Conference on Robotics and Automation.*, pp. 3915–3920. 2011.

[38] I. McCowan, D. Gatica-Perez, S. Bengio, D. Moore, and H. Bourlard. Towards computer understanding of human interactions. In *Machine Learning for Multimodal Interaction*, pp. 56–75. Springer, New York, 2005.

[39] J. McLaughlin, D.A. Reynolds, and T. Gleason. A study of computation speed-ups of the gmm-ubm speaker recognition system. In *Proceedings Eurospeech*, volume 99, pp. 1215–1218. 1999.

[40] H. Misra, S. Ikbal, and B. Yegnanarayana. Speaker-specific mapping for text-independent speaker recognition. In *Speech Communication*, 39(3), pp. 301–310, 2003.

[41] H. Nakasone, M. Mimikopoulos, S.D. Beck, and S. Mathur. Pitch synchronized speech processing (pssp) for speaker recognition. In *ODYSSEY04-The Speaker and Language Recognition Workshop*. 2004.

[42] 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 10th international conference on Multimodal interfaces*, pp. 257–264. 2008.

[43] M.D. Plumpe, T.F. Quatieri, and D.A. Reynolds. Modeling of the glottal flow derivative waveform with application to speaker identification. In *IEEE Transactions on Speech and Audio Processing.*, 7(5), pp. 569–586, 1999.

[44] T. Plutka. Extension of a binaural localization and tracking algorithm. 2012. Bachelor thesis at the *Institute for Data Processing, Technische Universität München*.

[45] J. Ramirez, J.C. Segura, C. Benitez, A. De La Torre, and A. Rubio. Efficient voice activity detection algorithms using long-term speech information. In *Speech communication*, 42(3), pp. 271–287, 2004.

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

[47] A.A. Salah, R. Morros, J. Luque, C. Segura, J. Hernando, O. Ambekar, B. Schouten, and E. Pauwels. Multimodal identification and localization of users in a smart environment. In *Journal on Multimodal User Interfaces*, 2(2), pp. 75–91, 2008.

[48] J. Schmalenstroeer and R. Haeb-Umbach. Online diarization of streaming audio-visual data for smart environments. In *IEEE Journal of Selected Topics in Signal Processing.*, 4(5), pp. 845–856, 2010.

[49] J. Schmalenstroeer, M. Kelling, V. Leutnant, and R. Haeb-Umbach. Fusing audio and video information for online speaker diarization. In *Proceedings ASRU*, pp. 1163–1166. 2007.

[50] C. Segura, A. Abad, J. Hernando, and C. Nadeu. Multispeaker localization and tracking in intelligent environments. In *Multimodal Technologies for Perception of Humans*, pp. 82–90. Springer, New York, 2008.

[51] O. Setin and E. Schriberg. Speaker overlaps and ASR errors in meetings: Effects before, during and after the overlap. In *Acoustics, Speech and Signal Processing*, 1, 2006.

[52] K. Steierer. Ausnutzen der Richtungsinformationen bei der Sprechererkennung. 2012. Studienarbeit at the *Institute for Data Processing, Technical University of Munich*.

[53] D.E. Sturim, D.A. Reynolds, E. Singer, and J.P. Campbell. Speaker indexing in large audio databases using anchor models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001.*, volume 1, pp. 429–432. 2001.

[54] N.P.H. Thian, C. Sanderson, and S. Bengio. Spectral subband centroids as complementary features for speaker authentication. In *Biometric Authentication*, pp. 631–639. Springer, New York, 2004.

[55] M. Unverdorben. Blind Source Separation for Speaker Recognition Systems. 2012. Diploma thesis at the *Institute for Data Processing, Technical University of Munich*.

[56] H. Vajaria. *Diarization, localization and indexing of meeting archives*. ProQuest, 2008.

[57] J.F. Wang, T.W. Kuan, J.c. Wang, and G.H. Gu. Ubiquitous and robust text-independent speaker recognition for home automation digital life. In *Ubiquitous Intelligence and Computing*, pp. 297–310. Springer, Berlin Heidelberg, 2008.

[58] C. Wooters and M. Huijbregts. The ICSI RT07s speaker diarization system. In *Lecture Notes in Computer Science*, 4625, pp. 509–519, 2008.

[59] S. Wulff. Reinforcement for Online Speaker Recognition in Teleconferencing Systems. 2013. Master thesis at the *Institute for Data Processing, Technical University of Munich*.

[60] B. Yegnanarayana and S. Kishore. Aann: an alternative to gmm for pattern recognition. In *Neural Networks*, 15(3), pp. 459–469, 2002.

[61] B. Yegnanarayana, K. Sharat Reddy, and S. Kishore. Source and system features for speaker recognition using aann models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing.*, volume 1, pp. 409–412. 2001.

[62] F. Zheng, G. Zhang, and Z. Song. Comparison of different implementations of mfcc. In *Journal of Computer Science and Technology*, 16(6), pp. 582–589, 2001.

[63] N. Zheng, T. Lee, and P. Ching. Integration of complementary acoustic features for speaker recognition. In *Signal Processing Letters, IEEE*, 14(3), pp. 181–184, 2007.

[64] E. Zwicker. Subdivision of the audible frequency range into critical bands (frequenzgruppen). In *The Journal of the Acoustical Society of America*, 33, p. 248, 1961.