3D Audio Conference System with Backward Compatible Conference Server using HRTF Synthesis

Martin Rothbucher, Matthias Kaufmann, Johannes Feldmaier, Tim Habigt, Marko Durkovic
Christoph Kozielski, Klaus Diepold
Institute for Data Processing
Technische Universität München Arcisstr. 21
80333 Munchen, Germany
{martin.rothbucher, tim, johannes.feldmaier, durkovic, kldi}@tum.de, matthias.kaufmann@mytum.de,
christoph.kozielski@gmail.com

***ABSTRACT*: *This article describes a system that is able to perform online sound localization, separation and speaker recognition for the purpose of channel assignment in teleconferencing scenarios. The assigned channels are then virtually synthesized to different positions around the listener by a conference server using Head-Related Transfer Functions (HRTFs), which enables the user to identify active speakers in an audio conference. Intelligibility of communication is also increased by making use of the cocktail party effect which describes how a user can concentrate on one specific participant, even if there are more simultaneously active speakers. To support regular telephone systems, we focus in particular on compatibility to existing Voice over IP (VoIP) infrastructure by using standardized protocols and central signal processing. Both traditional telephone devices and software clients are put into one heterogeneous conference. While the conference server transmits the down-mixed, single-channel conference signal to regular phones, the software clients can receive a stereo-signal enabling full 3D audio.*

Keywords: 3D Telephone Conferencing System, HRTF Synthesis, Sound Localization, Speaker Recognition, Speaker Assignment

Received: 11 August 2011, Revised 17 October 2011, Accepted 24 October 2011

1. Introduction

Nowadays, multi-party conferencing is more and more important to manage projects and work progress. There has been great progress concerning the quality of speech and visualization of conference rooms to reach a higher degree of immersion [1]. High quality but very expensive available solutions for professional teleconferencing consist of systems that record conference attendants with multiple cameras and microphones and display them on their own screen at the remote conference room. Such a system is usually too expensive for most users and moreover, all conference participants are restricted to a well equipped conference room to benefit from the system's features. Therefore, remote participants using e.g. mobile phones or regular VoIP systems do not benefit in such a scenario.

As a simple solution, most users utilize freeware to conduct teleconferences. One popular teleconferencing freeware is *Skype*[1]. Audio codecs used by *Skype* seek to improve speech quality but do not use binaural techniques to generate 3D sound.

[1] http://www.skype.com

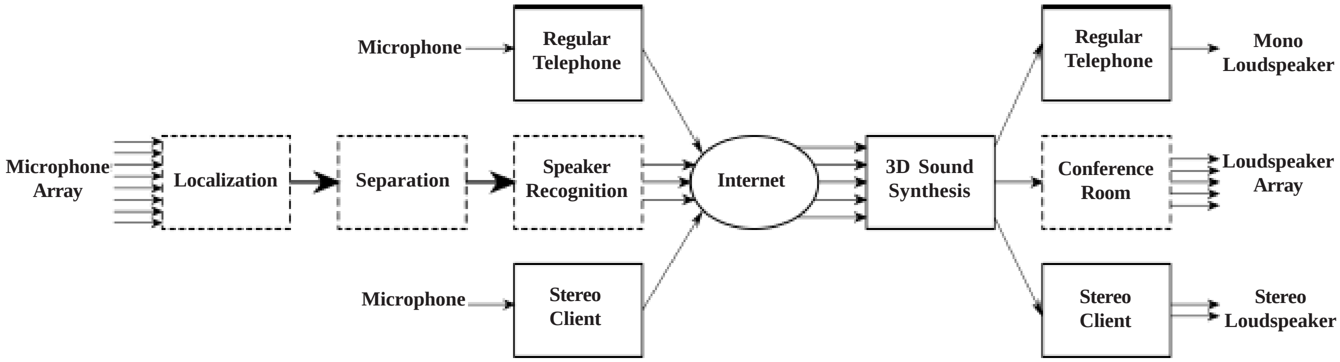


Figure 1. Schematic view of the proposed system. Solid line boxed components show signal processing of single conference participants, dashed line boxed components denote the required speaker assignment if participants are located in a conference room with a microphone array

Recently *Ekiga*, an open source VoIP software, was equipped with binaural sound rendering to make use of the well-known cocktail party effect [2, 3], enabling the user to virtually differentiate and understand simultaneously talking speakers [4]. However, all participants of this conference application have to be equipped with a computer and therefore it is not possible to remotely access the conference through traditional telephone infrastructure or by using mobile phones.

In [5], HRTF sound synthesis was integrated into the VoIP conferencing system *Mumble*, where also a computer is needed for each conference participant.

In this work, our approach to extend a telephone conference server software by the functionality of virtually placing conversational partners at different locations around the respective participant is described.

To support existing telephone systems, our system focuses in particular to compatibility on VoIP infrastructures and the Session Initiation Protocol (SIP). The central conference server extends all participants' voice signals by binaural effects using HRTFs. Therefore, users who are able to receive stereo signals via their headphones, can hear various speakers at different positions. Traditional telephone devices participating the conference, which still use the current mono standard, can nevertheless take part at the same conference so that a heterogeneous conference situation ensues in which stereo participants are enabled to virtually locate traditional devices. As illustrated in Figure 1 (solid line boxed), it is straigtforward to virtually place single conference participants that use telephones or VoIP systems by 3D sound synthesis. However, to virtually place different conference participants that are located in one conference room around a remote listener, the conference contributions of each speaker have to be assigned to individual transmission channels.

The assignment of the audio signals of different speakers to individual audio channels can be achieved by combining face tracking with a camera and microphone data.

In [6] a microphone array and two cameras with fish eye lenses are utilized to acquire an omnidirectional view to analyze group dynamics. In [7], the data of an omnidirectional camera and a microphone array is used for speaker identification and localization in a meeting scenario. Often it would be advantageous using only microphone signals, especially in low complexity stand-alone systems. Furthermore, there are often situations where the conference participants are not in the field of view of a camera system.

In [8] a Direction of Arrival (DOA) based speaker diarization system is introduced that is working purely on audio data, without speaker recognition. But this system suffers on high false detection rates.

Our approach illustrated in Figure 1 (dashed line boxed) to assign different speakers to individual audio channels is based on joint sound localization, sound separation and online speaker recognition and only depends on audio recordings of a microphone array. To ensure high quality speech acquisition even in noisy and echoic rooms, we utilize a Steered Response Power - Phase Transform (SRP-PHAT) sound localization with particle filtering in combination with Geometric Source Separation (GSS) [9, 10].

This way, echoes and noise can be minimized in the recordings and the speech quality of the audio signals that are passed to the speaker recognition system is increased. Moreover, in case of simultaneously active conference participants, the sound mixture is separated. Then, an online speaker recognition system is utilized which identifies the active speaker. Therefore, models are constructed by short-time spectral features of each conference participant. A likelihood score between the model and an active speaker decides which particular person is speaking. Since a system for conference applications has no prior knowledge about the text spoken by the users, the speaker recognition needs to be text-independent. Mel-frequency cepstral coefficients (MFCCs) are chosen to represent spectral attributes of different sounding voices. Due to the required low computational complexity of an online speaker recognition task, the system is based on Gaussian Mixture Models (GMMs). An Universal Background Model (UBM) in combination with maximum aposteriori (MAP) adaptation, as proposed in [11], leads to a very fast creation of speaker-dependent GMMs, with only few training data needed. In [12] it is shown that this system approach is capable of performing online recognition.

In this work, we recall the aforementioned backward compatible 3D audio conference approach presented in [13] and substantiate it with a microphone array that is able to assign different conference participants in a conference room to individual audio channels that can then be processed by the central 3D conference server. The paper is organized as follows. Sections 2 and 3 describe the speaker assignment system using sound source localization in combination with speaker recognition. Sections 4, 5, 6 and 7 provide knowledge about HRTF data, a fast sound convolution technique, called real time partitioned convolution and summarize communication models for generating 3D sound and introduce our approach of integrating 3D sound rendering in the conference software, respectively. The experimental results are presented in Section 8 and finally, a conclusion is given in Section 9.

2. Sound Source Localization and Separation

This section describes our adaption of localizing and separating sound sources based on [9, 10], within a conference situation with a circular microphone array. The output of the localization and separation module features a sound stream, in which echoes and disturbing noise are reduced compared to the raw microphone signals. A speaker recognition module then assignes an individual channel to each speaker. The speaker assignment system is illustrated in Figure 2. The details of the illustrated sound localization system is explained in the following subsections and the speaker recognition system of Figure 2 is described in section 3.

2.1 Steered Response Power (SRP) Localization

One promising approach to localize multiple simultaneously active sound sources in echoic and noisy environments is the SRP-PHAT algorithm [14]. SRP-PHAT estimates the DOA by determining and aligning the cross-correlation functions of the array's microphone signals according to pre-computed delays and the PHAT weighting function. The SRP $P(\mathbf{q})$ of a filter-and-sum beamformer of a sound source at position **q** is given by

$$P(\mathbf{q}) = \sum_{l=1}^{N} \sum_{k=1}^{N} \int_{-\infty}^{+\infty} \Psi_{lk}(\omega)\, X_l(\omega)\, X_k^*(\omega)\, e^{j\omega(\tau_k - \tau_l)}\, d\omega \qquad (1)$$

where $\Psi_{lk}(\omega) = \frac{1}{\| X_l(\omega) X_k^*(\omega) \|}$ is the PHAT weighting and τ_k and τ_l are the delays between a pair of microphones signals and a sampling point **q** of the search region [14]. The search region of the localizer can be described as a grid of points that covers preselected possible directions of arrival. $P(\mathbf{q})$ has to be computed between each microphone pair and sampling point. The summation of all $P(\mathbf{q})$ results in an energy map that has peaks at positions where sound sources are expected. In accordance with [9], we utilize an array of eight microphones for our teleconferencing system. Depending on the software configuration, the search region can be restricted to areas where conference participants could be present. Since we focus on teleconferencing, where the microphone array is placed on a conference table, we restrict the search region to a upper hemisphere.

2.2 Particle Filtering

The SRP-PHAT localizer on its own produces instable sound localization results, including localization of noise sources and echoes. To overcome this problem, a particle filter is integrated [15] for temporal smoothing of noisy measurements. Every possible sound source is therefore considered to be a set of particles, where each particle is assigned a distinct position in space, velocity and weighting. The sound localization estimations of the SRP-PHAT algorithm are then used to update the particle positions, directions and weightings, resulting in a permanently updated probability density function (PDF) of the estimated positions of sound sources. By computing the mean value of the PDFs, a stable sound source localization can be achieved.

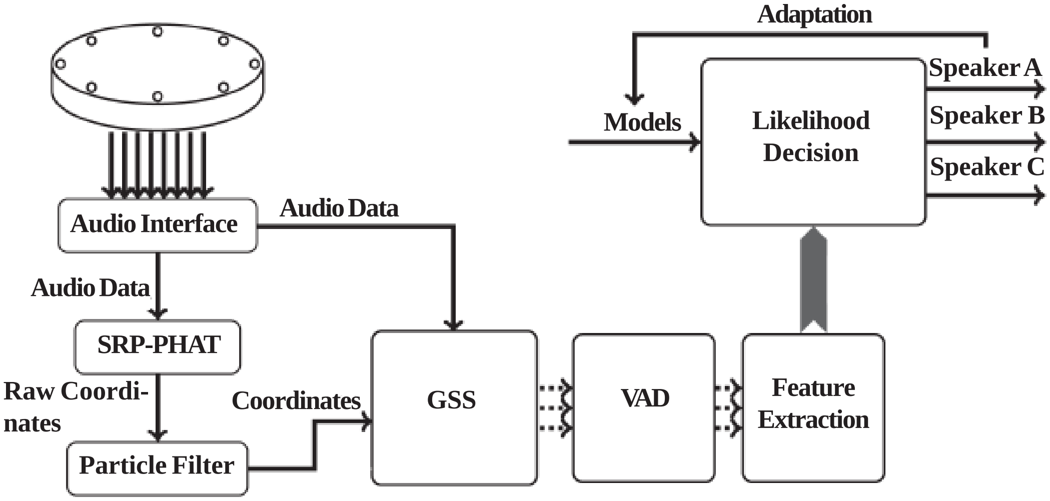


Figure 2. Schematic view of the speaker assignment system

2.3 Geometric Source Separation (GSS)

GSS [16] seeks to combine benefits of Blind Source Separation (BSS) and beamforming by fusing cross-power minimization of convolutive mixtures with geometric information provided by sound localization. In accordance with [16, 9], the cross-talk is minimized by cost functions, given by

$$J_1(\mathbf{W}(\omega)) = \| \mathbf{R}_{\mathbf{y}\,\mathbf{y}}(t, \omega) - diag[\mathbf{R}_{\mathbf{y}\,\mathbf{y}}(t, \omega)] \|^2 \qquad (2)$$

and

$$J_2(\mathbf{W}(\omega)) = \| \mathbf{W}(\omega)\mathbf{A}(\omega) - \mathbf{I} \|^2 \qquad (3)$$

where $J_1(\mathbf{W}(\omega))$ expresses the cross-talk minimization of the output signals $\mathbf{y}(t)$ and $J_2(\mathbf{W}(\omega))$ is the geometric constrain containing the estimated linear transfer functions $\mathbf{A}(\omega)$ between the sources and the microphones. The cross-power spectra $\mathbf{R}_{\mathbf{y}\,\mathbf{y}}$ has to be estimated in an on-line algorithm as a running estimate of the outputs $\mathbf{y}(t)$ for time instances t. The entries of $\mathbf{A}(\omega)$ are determined using the sound localization information of the localizer. With $J_1(\mathbf{W}(\omega))$ and $J_2(\mathbf{W}(\omega))$ the separation matrix $\mathbf{W}^n(\omega)$ is then updated using a gradient decent algorithm:

$$\mathbf{W}^{n+1}(\omega) = \mathbf{W}^n(\omega) - \mu \left[\alpha(\omega) \frac{\delta J_1(\mathbf{W}(\omega))}{\delta\, \mathbf{W}^*(\omega)} + \frac{\delta J_2(\mathbf{W}(\omega))}{\delta\, \mathbf{W}^*(\omega)} \right] \qquad (4)$$

where μ is the adaptation rate and $\alpha(\omega) = \| \mathbf{R}_{\mathbf{xx}}(t, \omega) \|^{-2}$ is an energy normalization factor. Finally, the separated output $\mathbf{y}(\omega)$ can be computed by $\mathbf{y}(\omega) = \mathbf{W}(\omega)\,\mathbf{x}(\omega)$, where $\mathbf{x}(\omega)$ describes the input signals.

3. Online Speaker Recognition

For our system approach we propose an online speaker recognizer, applying Gaussian mixture models for modeling individual speakers. Features are extracted from the incoming audio stream, constituting a likelihood score for every model and thus identifying the present speaker. The recognizer is not only capable of online speaker recognizing but also continuously improving recognition performance by online adaptation of the speaker models.

3.1 Preprocessing and MFCC Extraction

To extract spectral features of a speech signal, we first divide the incoming audio stream into frames by 20 ms hamming windows with an overlap of 10 ms. Using the FFT, it is required that the signal can be assumed to be stationary. However, to achive adequate resolution for robust feature extraction, a sufficient number of samples is required. 20 ms block length is a good compromise to overcome the aforementioned problems. A voice activity detection (VAD) locates frames with active speech by comparing the frame energy to a threshold, similar to the segmentation in [12].

Then, a sequence of feature vectors is generated that represents speaker-dependent information in every speech frame of the preprocessed signal. To represent the characteristics of individual voices, the mel-frequency cepstral coefficients (MFCCs) have proven meaningful as spectral features for speaker recognition tasks [17]. In our system approach we calculate MFCCs 1-12 for each speech frame. In addition, we use the spectral frame energy as feature. The feature vector is extended by the respective firstand second-order delta regression coefficients to incorporate dynamic information. Thus, altogether we employ a set of 39 features.

3.2 Gaussian Mixture Models

In our system approach, Gaussian mixture models (GMMs) are considered to model different speakers. For a *N*-dimensional feature vector $\vec{x}$, the Gaussian mixture density $p(\vec{x} \mid \lambda)$ is defined by

$$p(\vec{x} \mid \lambda) = \sum_{k=1}^{K} w_k N(\vec{x} \mid \vec{\mu}_k, \Sigma_k) \tag{5}$$

where $\mathcal{N}(\vec{x} \mid \vec{\mu}_k, \Sigma_k)$ is a unimodal Gaussian density, parametrized by a mean vector, $\vec{\mu}_k$, and a covariance matrix Σ_k. Therefore, the mixture density is a weighted linear combination of *K* Gaussian densities with mixture weights w_k, that satisfy the constraint

$$\sum_{k=1}^{K} w_k = 1; 0 \le w_k \le 1 \tag{6}$$

Collectively, the parameters of the density model are denoted by $\lambda = \{w_k, \vec{\mu}_k, \Sigma_k\}$, where $k = 1, ..., K$. The GMM for speaker modeling was introduced in [18] and has proven to be efficient and effective for text-independent speaker recognition tasks.

Given a collection of training vectors, the model parameters λ are estimated using the iterative expectation-maximization (EM) algorithm [19]. The EM algorithm iteratively refines the model parameters to increase the likelihood of the model for the training data. The EM equations for training a GMM can be found in [18].

The log-likelihood provides a score, measuring the match between a collection of feature vectors $\vec{X}$ of analyzed speech and speaker GMMs:

$$ln\, p(\vec{X} \mid \vec{w}, \vec{\mu}, \Sigma) = \sum_{n=1}^{N} ln\{\sum_{k=1}^{K} w_k N(\vec{x} \mid \vec{\mu}_k, \Sigma_k)\} \tag{7}$$

A speaker is assigned to the analyzed data by picking the GMM with the highest score.

Our system approach uses $N = 39$ dimensional feature vectors and $K = 32$ Gaussian density components.

3.3 Universal Background Model and MAP adaptation

We use an Universal Background Model (UBM) in our system approach. An UBM is a single GMM that is trained on speech samples from a large number of representative speakers. The main advantage is that the UBM has to be trained only once, which can be computed in advance for a wide variety of possible speakers. Then, the specific speaker models for a conference are derived from this UBM by individually adapting it. This leads to a very fast creation of speaker-dependent GMMs, benefiting the user comfort and practicability. Furthermore, it has been shown that GMMs adapted from a well-trained UBM also yield better speaker recognition results [11].

The adaptation in our system is done by maximum a posteriori (MAP) estimation (also known as Bayesian adaptation [20]). In order to create a new model for a certain speaker, the UBM is taken and adapted with the training data of this speaker. We use MAP adaptation to adapt only the means of the speaker GMM, which is saving computational cost and increases speaker recognition performance [11].

Besides generating speaker models, we also use MAP adaption for online improvement of already generated models during speaker recognition. Since the training material for building models is limited and may not adequately characterize the range of conference conditions, speaker models can be improved by adapting them with already processed test data as shown in [12]. This online adaptation leads to more comprehensive models, mitigating the effects of changes in conference situation or speaker condition.

4. Head Related Transfer Functions

After the assignment of the audio signal in the conference room via speaker localization and speaker recognition, the conference room audio streams and audio streams of other conference participants at remote places can be transformed into 3D audio signals by utilizing HRTFs. Head Related Transfer Functions (HRTFs) describe spectral changes of sound waves when they enter the ear canal, due to the diffraction and reflection properties of the human body, i.e. the head, shoulders, torso and ears [21]. Since the diffractions and reflections differ from direction to direction, HRTFs can be considered as direction dependent filters. Furthermore, a set of HRTFs is unique for each individual due to unique geometric features of each person.

Usually, a set of HRTFs is generated by measuring the Head Related Impulse Responses (HRIRs), the time domain representation of the HRTFs, in a time-consuming procedure in an anechoic chamber [22]. Small microphones are placed in the entrance of sealed ear canals and record sound presented by a loudspeaker that is moved to the positions on the spatial sampling grid. Depending on the density of the sampling grid, the HRTF measuring process can last several hours. Figure 3 schematically illustrates a setup to measure HRIRs. The mono signal *x* is unequally delayed before arriving at the ears due to path differences, which lead to the so-called interaural time difference (ITD). Additionally, the loudness of the signals is influenced by shadowing effects of the head, resulting in interaural level differences (ILD). Finally, diffraction and reflection from the listener's body are changing the spectrum of the signal. The recorded left and right signals x_L and x_R can be described by

$$x_L(t, d) = h_L(t, d) \ast x(t) \qquad (8)$$

and

$$x_R(t, d) = h_R(t, d) \ast x(t) \qquad (9)$$

where *d* is the direction of the sound source relative to the ears and describes the convolution operation in time domain.

With a set of measured HRTFs, like the publicly available MIT HRTF database [23], it is possible to convert a mono signal to a 3D presentation via stereo headphones. Computing (8) and (9), one can generate the spatial sound signals $x_R(t, d)$ and $x_L(t, d)$ by convolving a mono signal $x(t)$ with $h_L(d)$ and $h_R(d)$, respectively.

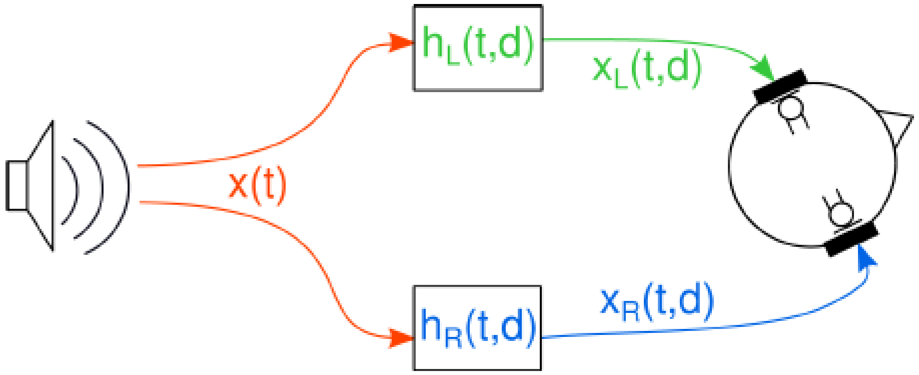


Figure 3. Schematic illustration of the measurement of HRIRs

5. Real-Time Partitioned Convolution

A key requirement of telecommunication systems is a low delay. Therefore, the signal processing should, in the best case, not add any additional delay. The conferencing server in our application has to calculate the convolution of the incoming signals with a pair of HRIRs. This convolution can be calculated straightforwardly in the time-domain, which has the advantage that it can be calculated for every sample without any algorithmic delay. On the other hand, this approach has a very high computational cost, especially for long impulse responses.

The convolution of two signals can be calculated much more efficiently in the frequency domain where the convolution operation is a simple multiplication. Using the overlapadd or overlap-save algorithm, the same result as in the time-domain can be achieved. The major disadvantage here is the fact that a discrete Fourier transform of the input signal has to be calculated. This leads to a delay of at least the length of the impulse response.

Real-time partitioned convolution [24] offers a good compromise between delay and computational cost. Here, a long impulse response of length N is split into P smaller blocks h_i of block size K

$$h_i(\tau) = h(t) \text{ for } t \in (i-1).K + \tau, i \in [1..P] \tag{10}$$

on which the frequency-domain convolution can be applied in parallel. The block size K is the new latency for the convolution and is usually chosen to be $K << N$. We zero pad each block to a size of $L = 2 \cdot K$ and obtain its frequency domain counterpart $H_i(\omega)$ using Fourier transform. We split the input stream $s(t)$ also into K sized blocks $s_i(t)$ and $S_i(\omega)$, respectively. The convolution for the input block m is computed by

$$Y_m(\omega) = \sum_{i=1}^{P} H_i(\omega) \cdot S_{m+1-i} \tag{11}$$

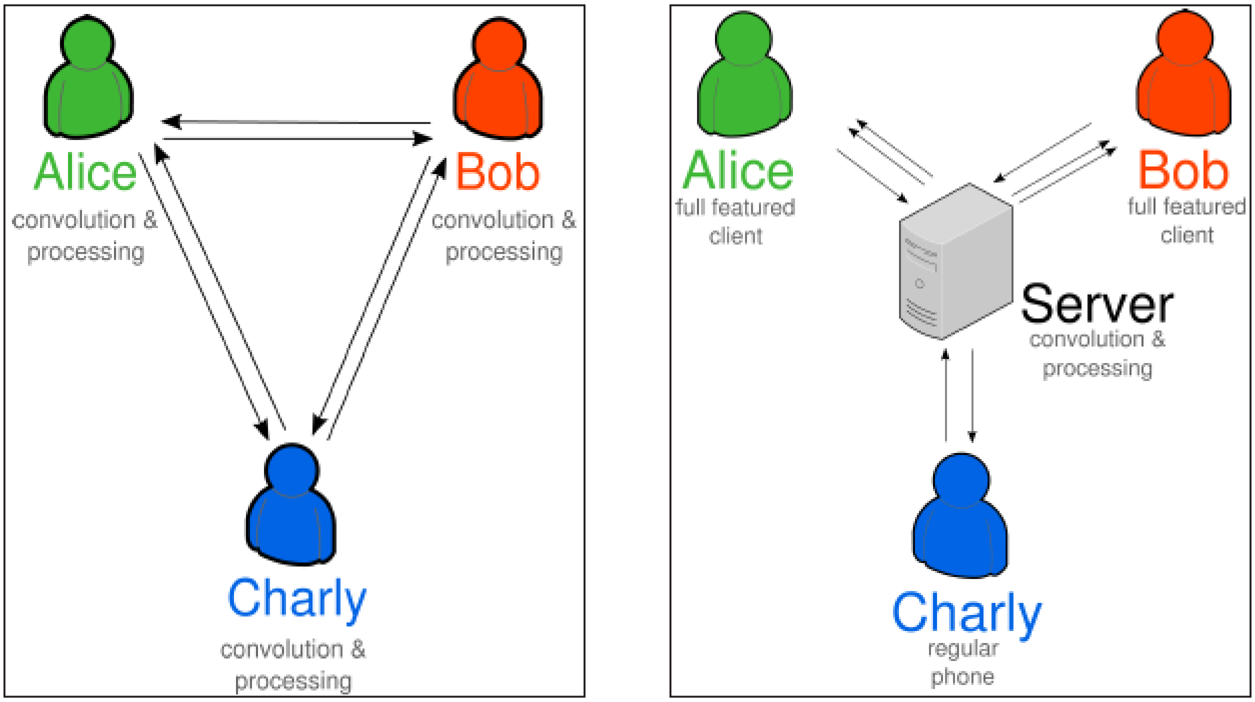
The synthesized signal is obtained by transforming each $Y_m(\omega)$ back to time domain and discarding the first K samples from the result.

6. Communication models for generating 3D audio effects

In this section we discuss which network component is capable of generating the desired 3D audio effects and focus on the approaches of decentral and central processing.

6.1 Decentral processing approach

As illustrated in Figure 4(a), binaural stereo-signals are processed by the clients and the system does not use any central network component.



(a) Decentral processing

(b) Central processing

Figure 4. Different approaches in signal processing

Therefore each participant has to send its own single-channel voice signal to the other clients. In this example, we assume that the clients can receive stereo signals which have double the datarate of regular single-channel signals. Concerning the datarate, one client has to send it's own signal (uplink) $N_u = n - 1$ times and receives (downlink) also $N_d = n - 1$ different signals from the other $n - 1$ participants, where n is the number of participants in the conference. To generate the individual binaural conference signal, every client has to compute $2(n - 1)$ HRTF convolution operations for the left and right ear and mix both channel of the stereo signal, which results in 2 mixing computations. In total for the whole scenario, the number of convolutions is $N_c = 2n (n - 1)$ and for mixing computations $N_m = 2n$.

6.2 Central processing approach

Instead of interchanging the voice signals decentrally, the central processing approach, illustrated in Figure 4(b), uses a central server for broadcasting and signal processing. Every client uploads its single-channel voice signal to the central server once ($N_u = 1$) which generates binaural stereo signals for the participants. As a result, the clients download one stereo signal from the server requesting $N_d = 2$ datarate units. Using global, fixed conference positions for each participant in the conference as we will discuss in section 7.3, every participant's voice signal is copied to the left and right stereo channel and filtered by the corresponding HRIR pair. Hence the sum of HRIR convolution operations will be $N_c = 2n$. The server mixes the stereo voice signals to one conference. In the central processing approach, the participants' own speech signals have to be removed from the respective mixtures. The amount of signal mixing computations (summations and differentiations) is therefore $N_m = 2(n + 1)$.

6.3 Comparison of required resources

Comparing the decentral with the central approach one can see major differences in required resources e.g. datarate, mixing and convolution computations. Here, the datarate is specified without units since the datarate depend on the codec.

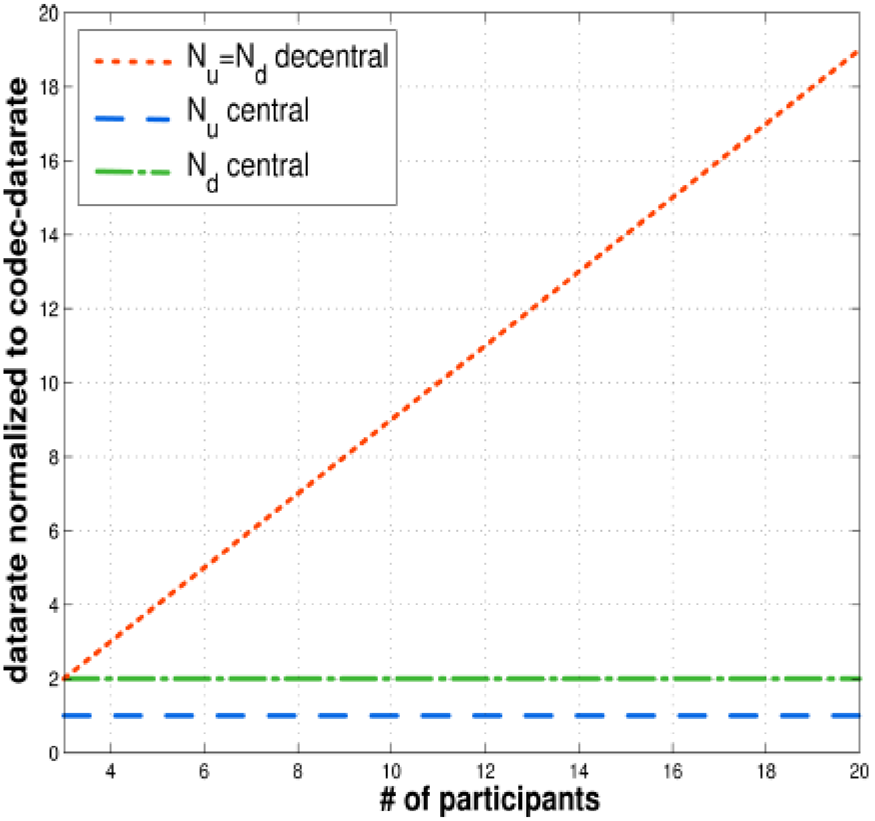


Figure 5. Datarate resources over the number of participants

Figure 5 shows the required datarate from the clients' point of view. The datarate of the decentral approach scales with the number of participants while the central approachkeeps these resources constantly low. This fact is very important for clients like mobile phones, where a broadband connection is mostly not available.

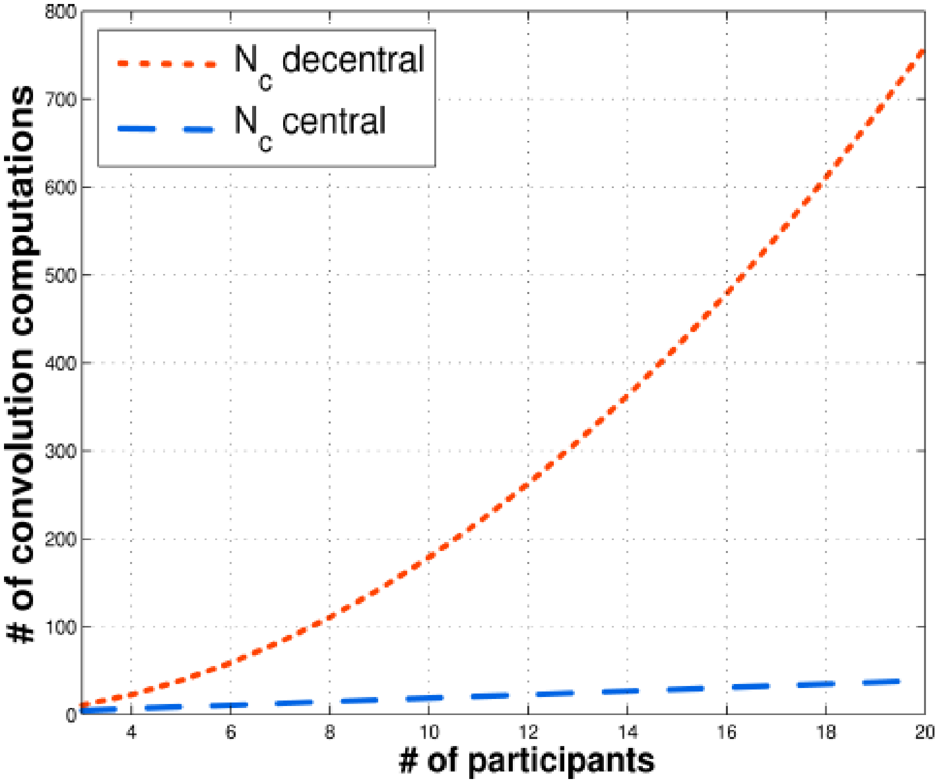


Figure 6. Total number of convolution computations over the number of participants

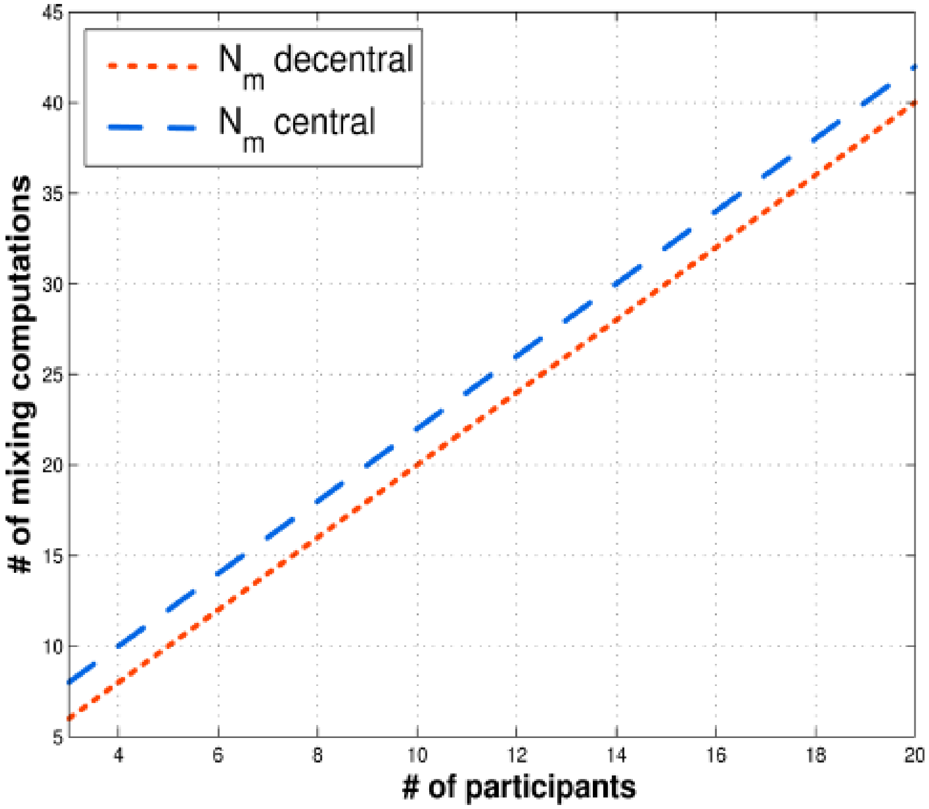


Figure 7. Total number of mixing computations over the number of participants

While the central server computes the stereo signals for all participants, every client has to process the binaural conference signal itself. Therefore to compare the computations fairly the whole scenario including all participants has to be considered. The total number of convolutions required for 3D spatialization using HRTFs is shown in Figure 6. It illustrates clearly the efficiency of fixed, global conference positions, which is discussed in Section 7.3.

The total number of mixing computations is illustrated in Figure 7 where again the whole conference scenario with all participants is considered. The difference between the two approaches is marginal and can be neglected.

Having an example with ten conference participants, the central processing approach saves 88.9% in uplink and 77.8% in downlink datarate usage compared to the decentral approach. In addition 88.9% less convolutions and only 10.0% more mixing operations have to be computed. This example highlights the efficient usage of resources.

6.4 Central vs. Decentral

In this section, the aforementioned approaches are discussed. The decentral processing allows the user to easily change the used HRTF database and influence the own position within the virtual conference room as the database is stored at the client's machine and the mixing is done locally. Utilizing the central approach, the user needs an interface to choose or upload the HRTF database and to influence his conference position, which is initially given by the central server. The major advantage of the central approach is the possibility of heterogeneous conferences with full featured stereo clients in parallel with regular fixed network phones, mobile phones or existing, regular VoIP infrastructure. One more positive aspect of the central solution is a very efficient usage of resources. Neither are redundant audio signals needlessly transported over the network nor is the convolution processing with HRIR filters done redundantly. In addition, the server can easily support many different audio codecs and not every single client needs to support all of them but only its own codec. Also the performance of the client and the bandwidth of the connection to the server can be limited. The client does not need to execute expensive signal processing and the required datarates are minimized. Those effects – compatibility to existing infrastructure by using heterogeneous conference rooms, efficient usage of resources, an easy support of many different audio codecs, low performance requirements of the client's hardware and low datarates – are the main benefits of the central approach.

7. Server Implementation

In this section the main aspects of our conferencing server are discussed. We focus on a maximum of compatibility and a minimum of datarate using an approach with central signal processing and global, fixed conference positions. The core network technologies are the Session Initiation Protocol (SIP) and the Real-time Transport Protocol (RTP) based on the modification of *PJSIP's* media library (*PJMEDIA*). We also show the general communication procedure for both, full featured stereo clients and regular phones.

7.1 Compatibility

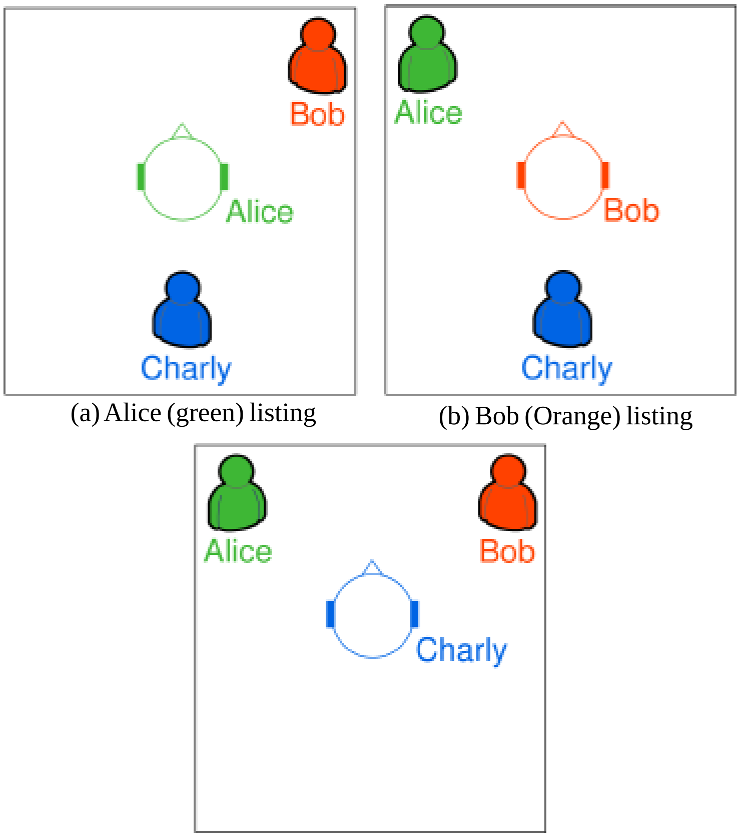
The main goal of our solution is maximum compatibility to existing telephone and VoIP infrastructure like the plain old telephone service, mobile phones, SIP infrastructure or Skype network, while adding HRTF sound synthesis to enhance the perceived quality and comprehensibility of telephone conferences. In comparison to existing solutions using local signal processing [5] we focus on a network infrastructure in which a central server extends all participants' signals by binaural effects using HRTFs. All speakers are put into one heterogeneous conference and only the client's supported features decide whether the participant enjoys the enhanced 3D-quality or gets a regular mono audio signal.

7.2 Minimizing datarate

In our approach every participant only needs to send its own single-channel voice signal once to the central server. Depending on the supported features of every participant, the server sends a stereo voice signal to full featured clients or a regular mono signal to traditional phones. Figure 4(b) illustrates the central signal processing with two full featured clients (Alice & Bob) and one regular phone (Charly). Obviously every client needs to send only its own voice signal once to the central server and, depending on its supported features, receives the already processed conference signal with one or two channels. Therefore, the required datarate is restricted to a minimum, which is particularly very important for mobile participants.

7.3 Global, fixed conference positions

As the voice signal for every conference position needs to be processed using HRTF technique, the server avoids redundant signal processing by using global, fixed conference positions. Same speakers are placed at the same position in the conference room and are located at this global, fixed position by all other remote participants. The result is a high efficiency by convolving every voice signals only once using HRTF technique. In Figure 8 one can see a conference situation with Alice, Bob and Charly and their individual perception of the scenario. Figure 8(a) and 8(b) show Charly being located in the back by both Alice as well as Bob. This 3D-sound synthesis for Charly's voice signal is done once and used for both participants (Alice and Bob). Figure 8(c) completes the scenario with Charly's point of view and the usage of Alice's and Bob's global, fixed conference position.



(a) Alice (green) listing

(b) Bob (Orange) listing

(b) Charly (blue) listing

Figure 8. Avoidance of redundant signal processing by using global, fixed conference positions

7.4 Network technology

To control the communication sessions we use the widespread Session Initiation Protocol(SIP) based on the Transmission Control Protocol (TCP) and for transporting the audio signals the Real-time Transport Protocol (RTP), which is based on the User Datagram Protocol (UDP). Besides the effect of using well-known and supported protocols one more important benefit is the compatibility to existing VoIP infrastructure which is mainly based on SIP while other protocols can be connected using gateways.

7.5 PJSIP

We base our idea on existing open source SIP software as they offer most of the required basic features and therefore need to select the most suited project. After a pre-selection we decide between *Asterisk*[2], *FreeSWITCH*[3], *PJSIP*[4] and *Yate*[5]. In order to choose one project, we specify three major groups of requirements: *Functionality*, *program code* and *additional features*.

We decide to utilize *PJSIP*, an open source SIP stack, as it supports in particular full stereo processing and a number of different audio codecs. It also has a reference implementation called *PJSUA* and can be used as a central conference server. We modify

[2] http://www.asterisk.org

[3] http://www.freeswitch.org

[4] http://www.pjsip.org

[5] http://yate.null.ro

the code which is responsible for the conference in the *PJMEDIA* library. As a result, the server software of our system only accepts stereo connections. Mono connections have to be mixed up externally, which is discussed in section 8.2.

7.6 Communication procedure

After the initialisation process at the server's startup, all new connections are treated the same way: if a client connects to the conference, the server automatically assigns a free virtual conference position to the new participant and saves this mapping. During the conversation, every new audio packet needs to be processed regularly as it arrives at the conference bridge. The left and right audio channels are saved interleaved inside one data fragment as illustrated in Figure 9. The channels are split and filtered by the corresponding HRIR pair using the position-mapping. After HRIR convolution, the modified stereo signal is passed to the stereo conference bridge. Full featured clients (stereo clients) receiving the stereo conference signal without their own voice signal and will enjoy virtual 3D placement of the conference participants. Regular phones receive the down-mixed single-channel version of the conference signal. They can hear all speakers but can not benefit from virtual 3D placement of the other conference participants. As soon as a participant leaves the conference, the server frees the corresponding virtual conference position for new participants. The mentioned functionality is realized by adapting the conference functions of PJSIP and integrating the additional 3D feature. Both assignment of a new 3D conference position to new participants and the clean up when a participant leaves the conference is done within the regular PJSIP functions.

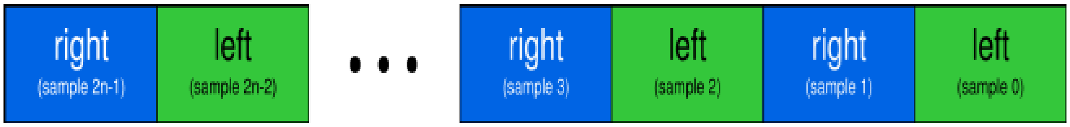


Figure 9. n Interleaved stereo samples inside one audio packet

8. Testing and Results

In this section, we individually investigate the performance of the speaker assignment and the backward 3D conference server system.

8.1 Localization and Speaker Recognition Performance

Here, we apply our joint sound localization, sound separation and online speaker recognition approach on real recordings, that we do in an echoic office environment.

8.1.1 Experimental Setting

In the following, the experimental settings are described in detail. Real-world recordings were performed in an office room ($5.10\text{m} \times 3.49\text{m} \times 3.09\text{m}$) with a reverberation time $RT_{60} = 0.64$ s. Our circular microphone array, that consists of eight low-budget microphones (seven microphones are attached on a circle, one is mounted in the middle of the circle, 10 cm above the other microphones), is placed on a desk in the office. Four loudspeakers with a distance of 1.15m around the microphone array with an elevation angle of 15° played back a dialogue (about seven minutes) of two male and two female speech signals taken from professionally recorded podcasts. Each speech source is connected to its individual loudspeaker. The dialogue is constructed to have parts with overlapping speech sources as well as sequences, where only one speech source is active. The speaker models are trained in the style of a round of introductions, which are common in teleconference situations: 10 s speech of each speaker is played back by the corresponding loudspeaker. The data used to train the speaker models are not part of the following diarization task.

Finally, our system records the communication of the four conference participants, represented by loudspeakers. Based on the recordings, the algorithm localizes and separates the speech sources and passes the separated streams to the speaker recognition module, which segments and assorts the communication to an individual audio channel for each participant.

8.1.2 Experimental Results

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

8.1.2.1 SRP-PHAT

Figure 10(a) illustrates a 55 s sequence of discussion of two male and two female conference participants. It can be seen at the thin lines with the waveforms, that the speech source, which starts talking first, is located at 225°. Then a second speech source, located at 45° starts talking. It is worth noticing, that there is a small overlap between the first two speakers. The solid blue and green markers in Figure 10(a) depict the estimates of the localization and separation algorithm. It clearly can be seen, that our implementation of the localization and separation algorithm robustly detects the active sound sources, even if sources overlap. Due to the reverberant office environment, the algorithm also localizes echoes, which can be seen at time instances 18 s and 40 s, where also sound sources are localized at the opposite site of the array. The output of the localization and separation system in this case is composed of two audio streams. Each stream consists of one of the active speech sources. The overlapping parts are separated by GSS. Furthermore, in situations with only one active speech source, the output signal includes less audible echoes and noises. However, as seen in Figure 10(a) the green and blue markers switch channels, meaning the localization and separation algorithm is not able to ensure a fixed channel-speaker-alignment.

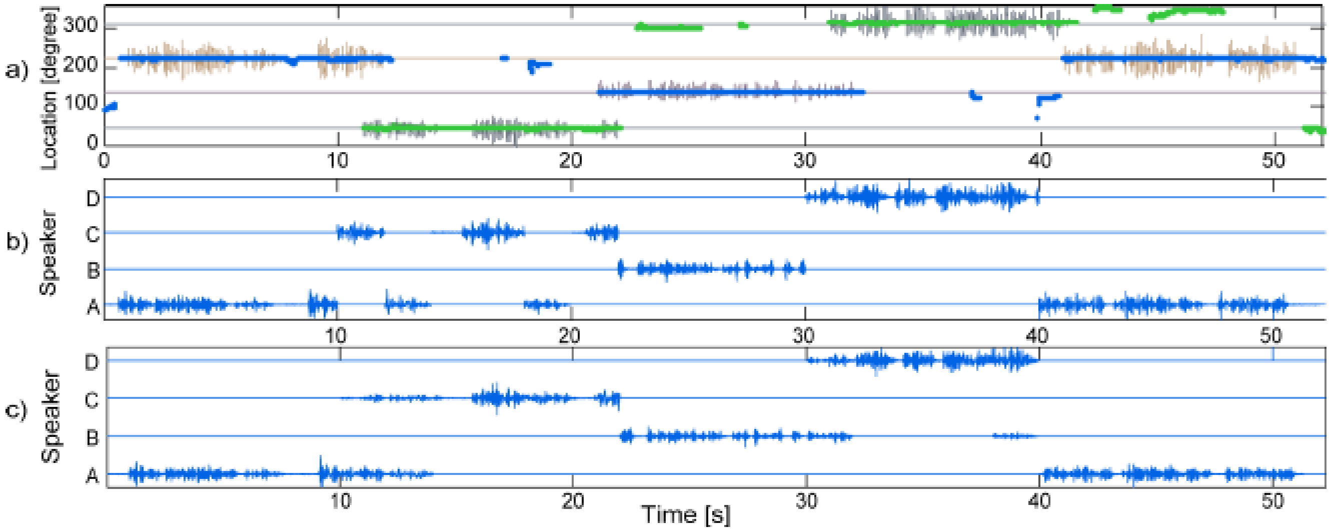


Figure 10. Speaker localization and assignment results. (a) Results of the active speaker localization. The thin lines at 45, 135 , 225 and 305 with the waveforms indicate the actual position and activity of the respective speech sources. The blue and green markers depict the localization algorithms' estimates of the sound sources. Figure (b) illustrates the result of the speaker recognition system without previous localization and separation. The audio channels A, B, C and D are assigned to a speech source, overlapping speech sources are not be assigned properly. In (c) channel assignment with previous speaker localization and separation is shown. One can see the improvements in case of overlapping sound sources. Furthermore, there is no loss of speech information compared to (b)

8.1.2.2 Speaker Recognition

To overcome the problem of channel-speaker-alignment, in the second experiment, standalone speaker recognition is applied on the recordings. As shown in Figure 10(b), the speaker recognition module reliably recognises the respective speech source, denoted by $A, \ldots, D$ in the Figure, in case of non-overlapping speech signals. If speakers overlap, the recognition algorithm has to decide for one source, leading to distorted audio information on the channels and a loss of speech information in the aligned audio streams.

8.1.2.3 Joint System: SRP-PHAT and Speaker Recognition

In the last experiment, performance of our joint system is investigated. According to 10(c), the system is able to properly assign two overlapping speech sources to the corresponding audio channels. Compared to the second experiment, there is no loss of speech information in case of two simultaneously talking conference participants. Furthermore, the joint system is able to correctly assign echoes due to the fact, that also the active speaker within the echo signal is correctly recognized and assigned to the proper channel.

This can be seen in Figure 10(a) at time instance 18 s. Preliminary hearing tests have shown, that false assignments, illustrated in Figure10(c) at time instance 39 s, hardly influence the perceived conference quality due to the signals low power level.

8.2 Backward Compatible 3D Conference Server Performance

We use *PJSUA*, the reference implementation of *PJSIP*, for both, the server and the client with different configurations. Figure 11 shows the complete test installation where the blue circle highlights the stereo connections.

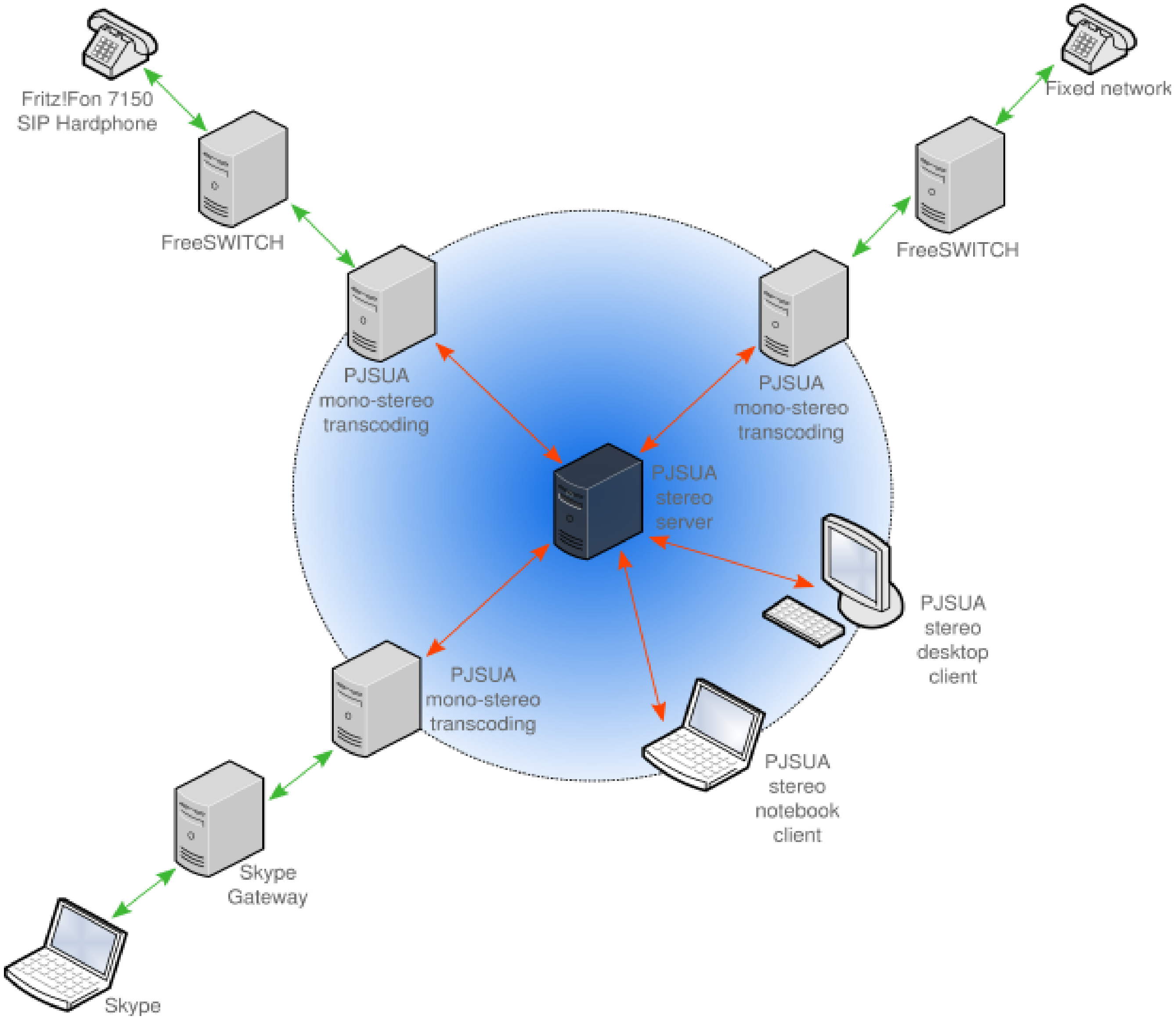


Figure 11. Schematic view of the test installation

The core of our setup consists of a central stereo conference server using the modified *PJMEDIA* library running on an Ubuntu 10.04 32 bit 800 Mhz system. As mentioned in section 7.5 the central server only accepts stereo connections. The reason is to keep modifications as small as possible. Therefore single-channel connections have to be transcoded before connecting to the server. As the central signal processing server tries to focus only on conferencing and does not offer any telephone management, in real scenarios additional authentication instances are necessary. Those machines could easily transcode the single-channel signals to stereo-channel signals by acting as a back-toback- user-agent. An additional advantage of this limited access to the central server is security: if external clients need to authenticate externally, the central server only needs to accept connections from certain, well-known communication servers. In our approach, we transcode the single-channel signals externally by one more instance of *PJSUA* which acts as back-to-back-user-agent and bridges the external connections to the internal 3D conference server.

Two full featured clients – one desktop and one laptop computer – are connected directly to the server. To demonstrate maximum compatibility, we integrate the fixed network (plain old telephone service) and a wireless SIP hardphone (AVM FRITZ!Fon 7150). For external networks, like the assigned audio streams of the teleconference room, we use *FreeSWITCH* as a gateway service and an additional instance of *PJSUA* for transcoding to stereo. New participants are automatically connected to the server.

8.2.1 Delay measurement

In order to guarantee high quality communication, we want to measure the effective mouth-to-ear delay between two full featured stereo clients. The stereo signals are coded with the L16 PCM algorithm at 16 kHz and 16 bit each channel (512 kbit/s in total for each stereo-connection).

To measure the mouth-to-ear delay, we sent an analog white Gaussian noise signal from a separate measurement computer to the headphone jack of the stereo desktop client. In addition we captured the signal from the stereo laptop client by the same measurement computer. The playback and capture were done simultaneously using Audacity. The delay was extracted by finding the maximum of the cross-correlation between the played and captured signal. Knowing the sampling rate, the time delay can be easily calculated.

We do four series of measurements with 20 test runs each. The first measurement extracts the self delay of the measurement equipment (mean 61.7 ms). The three measurements are corrected by this value. The most interesting measurement is the one between two *PJSUA* stereo clients connected to the server using the L16 PCM stereo codec. With a corrected mean delay of 178.0 ms this value is satisfying. In the following we want to extract the influence of the used audio codec and datarate. Therefore, we change the codec to G.722 mono. The result is a corrected mean delay of 223.7 ms. This shows us that the influence of datarate in the local network is smaller than the additional encoding and decoding delay. In addition, the weak hardware platform of the laptop client has to be considered. The last test isolates the influence of the server's signal processing and the additional server's network hop. Hence we connect the two stereo clients directly without the central conference server using the L16 PCM stereo codec. The result is a corrected mean delay of 127.9 ms. The difference to the second measurement including the server is 50.1 ms which is an acceptable delay for the additional network hop and the complete signal processing including the 3D sound synthesis.

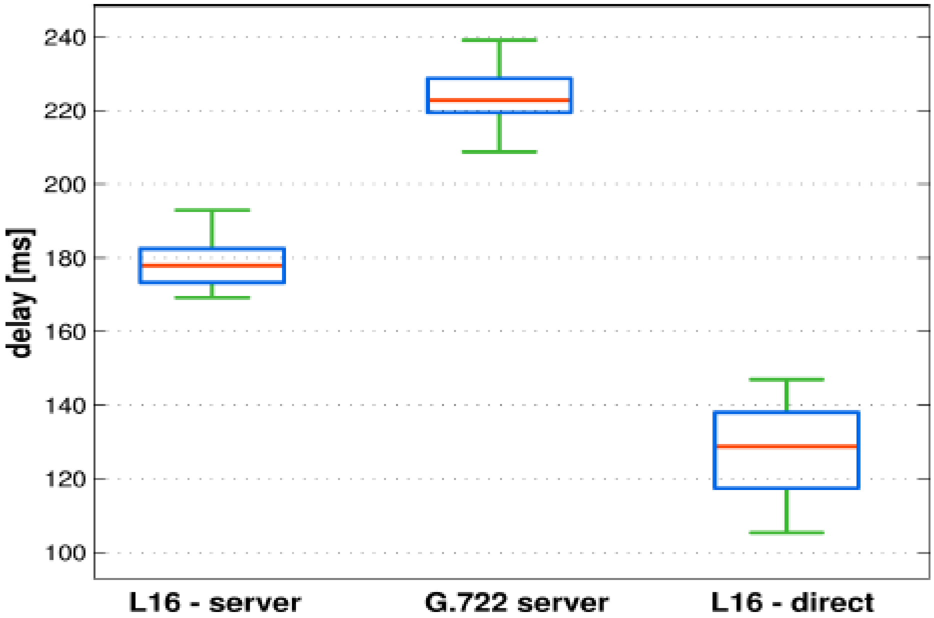


Figure 12. Box-and-whisker diagrams of the four measurement series

Figure 12 shows box-and-whisker diagrams of all four measurements. The last three measurements are corrected by the mean value of the self-delay. The L16 PCM stereo connection measurement including the implemented central 3D conference server is labeled as *L16 - server*, the one using G.722 mono connections over the server as *G.722 - server* and the direct connection without the central server again using L16 PCM stereo codecs as *L16 - direct*.

The results presented in this section demonstrate that the 3D-sound synthesis processed by the central conference server needs only 50.1 ms extra processing time and the mouth-to-ear delay of our system is 178.0 ms. Delays up to 200 ms are considered as *very user satisfying* according to the ITU G.114 recommendation [25].

9. Conclusion

In this article we enhance a teleconference server software solution by integrating 3D sound synthesis using HRTFs, which provide an improved identification of conference participants and increase intelligibility in situations with more than one active speaker (cocktail party effect). Different approaches of integrating HRTF sound synthesis are analysed and compared. Our approach, using a central processing with global, fixed conference positions, is resource-efficient and furthermore, compatibility to existing telephone networks, mobile phones and VoIP infrastructure is ensured. Low computational complexity of our system is demonstrated by evaluating the effective mouth-to-ear delay. Moreover, we address the problem of assigning different speakers placed in one conference room to individual audio channels by using sound source localization, separation and online speaker recognition for the HRTF-based 3D-teleconferencing system. Our experiments demonstrate that our system is able to assign different speakers to individual audio channels, even if the speech sources are overlapping.

10. Acknowledgment

This work was supported by the German Research Foundation (DFG) within the collaborative research center SFB-453 “High-Fidelity Telepresence and Teleaction”, by the DFG excellence initiative research cluster Cognition for Technical Systems CoTeSys and by the TUM Graduate School’s Faculty Graduate Center FGC-EI.

References

[1] Mortlock, A. N., Machin, D., McConnell, S., Sheppard, P. (1997). Virtual conferencing, *BT Technology Journal*, 15 (4) 120–129.

[2] Arons, B. (1992). A review of the cocktail party effect, *Journal of the American Voice I/O Society*, 12 (7) 35–50.

[3] Drullman, R., Bronkhorst, A. (2000). Multichannel speech intelligibility and talker recognition using monaural, binaural, and three-dimensional auditory presentation, *Journal of the Acoustical Society of America*, 107 (4) 2224–2235, Apr.

[4] Hyder, M., Haun, M., Hoene, C. (2010). Placing the participants of a spatial audio conference call, *In*: *IEEE Consumer Communications and Networking Conference- Multimedia Communication and Services*.

[5] Rothbucher, M., Habigt, T., Feldmaier, J., Diepold, K. (2010). Integrating hrtf sound synthesis into mumble, *In*: *International Workshop on Mulitmedia Signal Processing*, p. 24–28.

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

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

[8] Araki, S., Fujimoto, M., Ishizuka, K., Sawada, H., Makino, S. (2008). A doa based speaker diarization system for real meetings, *In*: *Hands-Free Speech Communication and Microphone Arrays*, p. 29–32.

[9] Valin, Michaud, F., Rouat, J. (2007). Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering, *Robotics and Autonomous Systems*, 55 (3) 216–228, 2007.

[10] Valin, J., Rouat, J., Michaud, F. (2004). Enhanced robot audition based on microphone array source separation with post-filter, *In*: *Proceedings of the international conference on Intelligent Robots and Systems*, 3, 2123–2128.

[11] Reynolds, D., Quatieri, T., Dunn, R. (2000). Speaker verification using adapted gaussian mixture models, *Digital Signal Processing*, 10 (1-3) 19–41.

[12] Geiger, J., Wallhoff, F., Rigoll, G. (2010). GMM-UBM based open-set online speaker diarization, *In*: *Proceedings of Interspeech*, p. 2330–2333.

[13] Rothbucher, M. Kaufmann, M., Habigt, T., Feldmaier, J., Diepold, K. (2011). Backwards compatible 3d audio conference server using hrtf synthesis and sip, *International IEEE Conference on Signal-Image Technologies and Internet-Based System*, p. 111–117.

[14] DiBiase, J., Silverman, H., Brandstein, M. (2001). Robust localization in reverberant rooms, *Microphone arrays: signal processing techniques and applications*, p. 157– 180.

[15] Isard, M., Blake, A. (1998). Condensationâ AT conditional density propagation for visual tracking, *International journal of computer vision*, 29 (1) 5–28.

[16] Parra, L., Alvino, C. (2002). Geometric source separation: Merging convolutive source separation with geometric beamforming, *IEEE Transactions on Speech and Audio Processing*, 10 (6) 352–362.

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

[18] Reynolds, D., Rose, R. (1990). Text independent speaker identification using automatic acoustic segmentation, 1, 293–296.

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

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

[21] Blauert, J. (19970. *Spatial Hearing*. Cambridge, MA: MIT Press.

[22] Møller, H., Sørensen, M., Hammershøi, D., Jensen, C. (1995). Head-related transfer functions of human subjects, *Journal of the Audio Engineering Society*, 43 (5) 300–321.

[23] Gardner, W., Martin, K. (1995). HRTF measurements of a KEMAR, *Journal of Acoustical Society of America*, 97, 3907–3908.