FP7-ICT-2013-C TWO!EARS Project 618075

Deliverable 3.5

Report on evaluation of the Two!Ears expert system

WP3 *

November 29, 2016

* The Two!Ears project (http://www.twoears.eu) has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 618075.

Project acronym:	Two!Ears
Project full title:	Reading the world with Two!Ears

Work packages:	WP3
Document number:	D3.5
Document title:	Report on evaluation of the Two!Ears expert system
Version:	1

Delivery date:	30th November 2016
Actual publication date:	30th November 2016
Dissemination level:	Public
Nature:	Report

Editors:	Guy Brown and Ning Ma
Author(s):	Ning Ma, Ivo Trowitzsch, Youssef Kashef, Johannes Mohr, Klaus Obermayer, Christopher Schymura, Dorothea Kolossa, Thomas Walther, Hagen Wierstorf, Tobias May, Guy Brown, Benjamin Cohen-L'Hyver, Patrick Danès, Michel Devy, Thomas Forgue, Ariel Podlubne, Bertrand Vandeportaele
Reviewer:	Bruno Gas

Contents

1 Executive summary

The Two!Ears project aims to develop an intelligent, active computational model of auditory perception and experience that operates in a multi-modal context. Ultimately, the system must identify the acoustic sources that are present in the environment and ascribe meaning to them.

This report presents and evaluates the technologies that have been developed in work package three (WP3) of the Two!Ears project, with a particular focus on the work done in the second and third years of the project. First, the blackboard software architecture is described, which combines rule-based processing and inference using graphical models. This provides a flexible, powerful architecture that fuses traditional rule-based artificial intelligence with statistical machine learning techniques.

Within the software framework of the Two!Ears system, a number of 'knowledge sources' are defined which extract information from the acoustic input and place that information on the blackboard. These include novel methods for segmenting the acoustic input into streams, and a new sound localization approach based on deep neural networks (DNNs). Further novel technologies introduced include a method for tracking acoustic sources, and a state-of-the-art machine learning pipeline for training and testing models for acoustic source identification, number of sources estimation, and joint identification/localization. New methods for audio-visual speech recognition, audio-visual speech enhancement, musical genre identification and attentional processing are also introduced.

The Two!Ears system has been evaluated both within a software simulator (the 'development system') and by embedding the system in a mobile robot (the 'deployment system'). Evaluations of each key technology developed within WP3 are presented in the second half of this report, in some cases showing a comparison with human performance on the same task. Whilst many challenges remain in the field of machine hearing, we show that the performance of the Two!Ears system approaches that of human listeners in some tasks.

2 Introduction

The Two!Ears project aims to develop an intelligent, active computational model of auditory perception and experience that operates in a multi-modal context. At the heart of the project is a software architecture based on a “blackboard system”, which fuses prior knowledge with the currently available audio and visual sensor input. The blackboard system and its associated software frameworks are embedded within a mobile robot, which provides acoustic sensing via an anthropometric binaural head and stereo cameras.

The ultimate goal of the system is to identify the acoustic sources that are present in an environment and ascribe meaning to them. An important aspect of the system is that information at the higher levels of the blackboard (e.g., information about the task or sound sources that are present) can influence lower-level processing (e.g., the weight attributed to particular acoustic features).

Briefly, the key technologies developed in work package three (WP3) include:

- A blackboard software architecture which combines rule-based processing and inference using graphical models, providing a flexible and powerful architecture that leverages both traditional rule-based artificial intelligence and statistical machine learning techniques.

- Methods for segmenting the acoustic input into acoustic streams, including a novel approach that uses a mixture of von Mises distributions to cluster binaural data.

- A new sound localisation method based on a deep neural network (DNN) architecture, which uses multi-condition training and head movements to mitigate front-back localisation errors.

- A novel method for tracking sources with a mobile robot, which allows movements to be selected by minimising the predicted localisation uncertainty.

- A state-of-the-art machine learning pipeline for training and testing models for acoustic source identification, number of sources estimation, identification of segregated streams, joint identification and localization.

- A novel method for improving the robustness of source localisation by exploiting top-down information from source models.

- New methods for audio-visual speech recognition and audio-visual speech enhancement.
- A method for obtaining robust identification models via multi-condition training.
- A novel method of identifying sound types from segregated audio streams.
- A system for musical genre identification.
- An attentional system that is closely integrated with head movements made by the mobile robot.
- Methods for estimating the number of sources in an auditory scene.
- A method for joint identification and localisation using DNNs.

This deliverable focuses on evaluation of the complete Two!Ears system, but also serves as a convenient entry point to reviewers that summarises the achievements of WP3. The remainder of the document is therefore structured as follows. Chapter 3 summarises the key achievements of WP3, referring briefly to work that was reported in previous deliverables and fully explaining new work that was done since the first-year review. Chapter 4 then presents a detailed evaluation of the Two!Ears system. We conclude with some reflections on what has been achieved, and suggestions for future work.

3 Summary of achievements in WP3

3.1 Task 3.1: Architecture

We developed an architecture that integrates experience formation and active behavior from a set of different functional modules. These modules can work on different levels of abstraction, independently from each other or interacting, in a bottom-up or top-down manner. While the targeted domain of the Two!Ears system is defined – auditory and multimodal scene understanding, evaluation and exploration – it involves many different problems, each with many possible solutions. A key feature of our architecture – in the following termed "blackboard system" – is thus its ability to evolve, enabling easy modification, exchange and extension of modules. Software development has focused on implementing dynamic behavior, construction and communications.

The blackboard system allows integration of different functional modules called *knowledge sources*. They define which data they need for execution and which data they produce – the blackboard system provides the tools for requesting and storing this data, but does not care about the actual contents (while the knowledge sources do not need to care about where and how data is stored). The blackboard system has no static knowledge of what types of knowledge sources are available.

The Two!Ears system is an *active* system that does not only work in a signal processing bottom-up manner, but also in a "cognitive" top-down manner. Modules therefore are allowed to change the system setup at runtime through dynamic module instantiation, registration and removal combined with on-the-fly rewiring of the communication links between modules.

Our platform is open to the public in application and further development, being open-source. The system is designed to be usable and easy to configure for different tasks, code units are kept small and understandable, responsibilities between classes are clearly separated, and object-oriented principles are used.

The architecture's main component is the `BlackboardSystem`, which integrates the framework core parts and knowledge sources, constructing and setting up the system. It connects the robot or binaural simulator – whichever is used to provide the ear-signals and command movement. It also integrates the auditory front-end (AFE), responsible for processing the ear-signals into cues (such as interaural time differences) needed by knowledge sources. To enable creation of new knowledge sources that depend on auditory signals without needing to change the Two!Ears system, we provide a utility superclass that enables automatic registration of signal requests. Start-up configuration of the system can either completely be defined by an XML file or by program code.

The `Blackboard` forms the central data repository of the platform. It stores knowledge sources and any shared data, in particular output of the knowledge sources (e.g., estimates of the location of a sound source). It is accessible to all knowledge sources; and it not only stores current data, but keeps track of the history of this data in order to enable knowledge sources to work on time series data. The Blackboard is flexible about data categories, which do not have to be hard-coded into the system.

Knowledge sources are decoupled and have no "knowledge" of each other, instead knowledge sources make requests to be triggered upon firing of particular events. Events in turn can be fired by knowledge sources. The blackboard system is ignorant *a priori* of which events exist. It starts to monitor events upon receiving requests made by knowledge sources, done through the `BlackboardMonitor`.

The `Scheduler` is the component of the blackboard system that actually executes the knowledge sources – after deciding the order in which knowledges sources waiting in the agenda are to be executed. This order is rescheduled whenever conditions determining the order may have changed, or when new knowledge sources may be present in the agenda that are more urgent. Knowledge sources have priorities: knowledge sources with higher priority get executed before knowledge sources with lower priority. Focus (priority) can be put on knowledge sources dynamically, along with the option to propagate this higher priority down along the dependency chain of this knowledge source.

The blackboard system currently integrates the acoustical scene auralization environment, the auditory front-end, and functional modules for auditory object formation. Since the acoustical scene auralization environment is connected to the blackboard system through an interface, it is possible to instead connect the robot to the system (see Deliverable D5.3), which then can – controlled by knowledge sources – actively explore the environment.

3.2 Task 3.2: Pre-segmentation and tracking

3.2.1 Methods for segmentation into acoustic streams

Segmentation based on von Mises distributions

The purpose of sound source segmentation is to assign specific time-frequency units of the auditory feature space to individual sound sources. This can either be done using a hard-assignment via binary masking or by computing soft-masks, which yields a smooth segmentation of the feature space. Soft-masking be interpreted as a probabilistic assignment of time-frequency units to sound sources.

Segmentation serves as the initial processing step for the task of *auditory stream segregation*, aiming at binding sound source location cues with sound identities. The implementation within the TWO!EARS-framework is realised as a knowledge source (KS) named SEGMENTATIONKS, focusing on estimating soft-masks to fit the general approach of TWO!EARS on using probabilistic models. The proposed segmentation framework relies on a two-step approach, comprising:

1. Azimuth clustering, based on a mixture of von Mises (vM) distributions and
2. Probabilistic masking using a Gaussian observation model.

The following paragraphs describe both processing steps in more detail.

Azimuth clustering. The proposed segmentation framework relies on estimates of sound source azimuths, which are provided by the deep neural network (DNN)-based localisation described in Sec. 3.4.1. It utilises a block-based processing scheme to derive statistics of the spatial sound source characteristics over a fixed number of consecutive time frames using a mixture of vM distributions. The DNNLOCATIONKS outputs posterior probabilities w_i for a set of discrete azimuth angles ϕ_i, $i = 1, \ldots, N_\phi$ at each time step k. Accumulating them over a signal block of K_B frames yields a set $\{\phi_j, w_j\}_{j=1}^{(N_\phi \cdot K_\mathrm{B})}$. This representation serves as the basis for a subsequent clustering step. However, conventional clustering techniques like k-means (MacQueen, 1967) or Gaussian mixture models (GMMs) (Dempster *et al.*, 1977) might not be suitable for the problem at hand, since the available observations are azimuth angles, originating from a circular probability distribution bounded in $[-\pi, \pi]$. Therefore, an alternative clustering technique is applied here, which is based on a mixture of vM distributions (Banerjee *et al.*, 2005). Similar approaches have already been proposed in the context of sound source localization and tracking by Traa and Smaragdis (2014) and Markovic and Petrovic (2012). The probability density function (PDF) of a unimodal vM distribution is defined as

$$\mathcal{VM}(\phi,\,|\,\mu,\,\kappa) = \frac{1}{2\pi I_0(\kappa)} \exp\Big\{\kappa\cos(\phi-\mu)\Big\}, \tag{3.1}$$

where $\phi \in [-\pi,\,\pi]$ is an angle, μ is the circular mean, κ is the concentration parameter and $I_i(\,\cdot\,)$ is the modified i-th order Bessel function. However, this representation is not suitable for clustering the set of obtained azimuth angles, as realistic scenarios usually contain more than one simultaneously active sound source, which requires a PDF that is able to model multimodal characteristics. Furthermore, the posterior probabilities w_i, estimated by the DnnLocationKS can be interpreted as individual weighting factors for each azimuth ϕ_i. Hence, this has to be considered when fitting a PDF to the acquired data. Subsequently, of a weighted mixture of vM distributions is used here, which can be derived from Eq. (3.1) as

$$p(\phi_i,\,w_i\,|\,\boldsymbol{\pi},\,\boldsymbol{\mu},\,\boldsymbol{\kappa}) = \sum_{j=1}^{N_\mathrm{c}} \pi_j\Big(\frac{1}{2\pi I_0(w_i\kappa_j)}\exp\Big\{w_i\kappa_j\cos(\phi_i-\mu_j)\Big\}\Big), \tag{3.2}$$

where $\boldsymbol{\pi} = [\pi_1,\,\ldots,\,\pi_{N_\mathrm{c}}]^T$ are the mixture weights satisfying $\sum_{j=1}^{N_\mathrm{c}} \pi_j = 1$, $\boldsymbol{\mu} = [\mu_1,\,\ldots,\,\mu_{N_\mathrm{c}}]^T$ denote the circular means and $\boldsymbol{\kappa} = [\kappa_1,\,\ldots,\,\kappa_{N_\mathrm{c}}]^T$ are the concentration parameters corresponding to each of the N_c mixture components. A similar approach for Gaussian PDFs was recently described by Gebru *et al.* (2016).

The proposed segmentation framework is based on the assumption that the number of active sound sources is known *a priori* (see Section 3.5.2 for a description of how the system can estimate this). Therefore, the number of mixture components N_c is fixed according to this prior knowledge. For a given set of estimated source positions $\boldsymbol{\phi} = \{\phi_i\}_{i=1}^{(N_\phi\,\cdot\,K_\mathrm{B})}$ and corresponding weights $\boldsymbol{w} = \{w_i\}_{i=1}^{(N_\phi\,\cdot\,K_\mathrm{B})}$, the log-likelihood of the PDF introduced in Eq. (3.2) can be expressed as

$$\mathcal{L}(\boldsymbol{\phi},\boldsymbol{w}\,|\,\boldsymbol{\pi},\,\boldsymbol{\mu},\,\boldsymbol{\kappa}) = \sum_{i=1}^{N_\mathrm{s}} \log\Big(\sum_{j=1}^{N_\mathrm{c}} \pi_j\Big(\frac{1}{2\pi I_0(w_i\kappa_j)}\exp\Big\{w_i\kappa_j\cos(\phi_i-\mu_j)\Big\}\Big)\Big), \tag{3.3}$$

where $N_\mathrm{s} = N_\phi \cdot K_\mathrm{B}$ denotes the number of azimuth samples within one signal block. The parameters of Eq. (3.3) are estimated using an expectation maximisation (EM) scheme based on the approach presented in Hung *et al.* (2012). During the expectation step, the responsibilities for the i-th sample and j-th mixture component are computed as

$$\gamma_{ij} = \frac{\pi_j\Big(\frac{1}{2\pi I_0(w_i\kappa_j)}\exp\Big\{w_i\kappa_j\cos(\phi_i-\mu_j)\Big\}\Big)}{\sum_{k=1}^{N_\mathrm{c}}\pi_k\Big(\frac{1}{2\pi I_0(w_i\kappa_k)}\exp\Big\{w_i\kappa_k\cos(\phi_i-\mu_k)\Big\}\Big)} \tag{3.4}$$

The parameter estimates for the circular means and mixture weights can be computed analytically at each maximization step as

$$\mu_j = \operatorname{atan2}\Big(\sum_{i=1}^{N_\mathrm{s}} w_i \gamma_{ij} \sin(\phi_i), \sum_{i=1}^{N_\mathrm{s}} w_i \gamma_{ij} \cos(\phi_i) \Big), \tag{3.5}$$

$$\pi_j = \frac{1}{N_\mathrm{s}} \sum_{i=1}^{N_\mathrm{s}} \gamma_{ij}. \tag{3.6}$$

An estimator for the concentration parameters cannot be derived analytically. However, the required estimation procedure can be expressed as a root-finding problem

$$\sum_{i=1}^{N_\mathrm{s}} w_i \gamma_{ij} \Big(\cos(\phi_i - \mu_j) - \frac{I_1(w_i \kappa_j)}{I_0(w_i \kappa_j)} \Big) \overset{!}{=} 0, \tag{3.7}$$

which can be efficiently solved using Newton's method. The EM algorithm is initialised with model parameters estimated using the circular k-means algorithm described in Banerjee *et al.* (2005).

Probabilistic masking. The probabilistic masking step is based on Gaussian observation models for each frequency channel, relying on two primary binaural cues, namely interaural time differences (ITDs) and interaural level differences (ILDs). The ILD between the left and the right ear signal, denoted as τ_{kl}, is estimated for each time frame k and frequency channel $l = 1, \ldots, L$. ILDs are denoted as δ_{kl}. ITD and ILD features are combined into 2-dimensional binaural feature vectors $\boldsymbol{y}_{kl} = \begin{bmatrix} \tau_{kl} & \delta_{kl} \end{bmatrix}^T$ for each time frame and frequency channel.

The underlying observation models are represented as

$$\boldsymbol{y}_{kl} = \boldsymbol{g}_l(\phi) + \boldsymbol{n}_{kl}, \tag{3.8}$$

where $\boldsymbol{g}_l(\phi)$ is a non-linear mapping function of an azimuth angle ϕ and $\boldsymbol{n}_{kl} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_l)$ is an additive Gaussian noise term with channel-dependent covariance matrix $\boldsymbol{R}_l$. The non-linear mapping function is realised as a regression model

$$\boldsymbol{g}_l(\phi) = \begin{bmatrix} \beta_{l0}^{\tau} + \sum_{n=1}^{N_\mathrm{o}} \beta_{ln}^{\tau} \sin(n \cdot \phi) \\ \\ \beta_{l0}^{\delta} + \sum_{n=1}^{N_\mathrm{o}} \beta_{ln}^{\delta} \sin(n \cdot \phi) \end{bmatrix}, \tag{3.9}$$

which is based on trigonometric functions to account for the circular nature of the source azimuth positions. Herein, N_o is the maximum order of the regression function and $\beta_{ln}^{\{\tau,\delta\}}$ represent the regression coefficients, which are estimated via conventional linear regression using rendered head related impulse responses (HRIRs) with white noise as stimulus signals. The residuals obtained after training are used to estimate the noise covariance matrix $\boldsymbol{R}_l$.

After a model as described in Eq. (3.2) has been fitted to the data within one signal block, each mixture component can be interpreted as a probabilistic representation of the azimuthal position of one sound source. By discarding the mixture weights, the model simplifies to a conventional, unimodal vM distribution for each sound source $p(\phi \,|\, \mu_i,\, \kappa_i)$, where $i = 1, \ldots, N_c$ denotes the number of mixture components in Eq. (3.2), which is equivalent to the number of sound sources. Hence, the soft-mask weighting factor for the i-th source at time step k and frequency index l can be computed as

$$m_{kl}^{(i)} = \frac{p(\boldsymbol{y}_{kl} \,|\, \boldsymbol{g}(\phi_i),\, \boldsymbol{R}_l)}{\sum_{j=1}^{N_\mathrm{c}} p(\boldsymbol{y}_{kl} \,|\, \boldsymbol{g}(\phi_j),\, \boldsymbol{R}_l)}. \tag{3.10}$$

The overall algorithmic structure of the StreamSegregationKS is summarised in Alg. 1.

Algorithm 1 StreamSegregationKS

```
Inputs:
  • Number of active sound sources/mixture components $N_\mathrm{c}$
  • Output from DnnLocationKS $\{\phi_i,\, w_i\}_{i=1}^{(N_\phi \cdot K_\mathrm{B})}$ for current signal block
  • Binaural observations $\boldsymbol{y}_{kl}$ for current signal block
Initialisation: Run circular $k$-means on $\{\phi_i,\, w_i\}_{i=1}^{(N_\phi \cdot K_\mathrm{B})}$ to get initial parameters $\pi_j$,
$\mu_j$, $\kappa_j$ and $\gamma_{ij}$
repeat                                                        ▷ Azimuth clustering
    E-Step:
        Compute responsibilities $\gamma_{ij}$ using Eq. (3.4)
    M-Step:
        Re-estimate circular means $\mu_j$ using Eq. (3.5)
        Re-estimate mixture proportions $\pi_j$ using Eq. (3.6)
        Re-estimate concentration parameters $\kappa_j$ using Newton's method on Eq. (3.7)
    Evaluate log-likelihood using Eq. (3.3)
until log-likelihood $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{w} \,|\, \boldsymbol{\pi},\, \boldsymbol{\mu},\, \boldsymbol{\kappa})$ converges
for $i = 1, \ldots, N_\mathrm{c}$ do                          ▷ Probabilistic masking
        Compute soft-mask for $i$-th sound source using Eq. 3.10
end for
```

Segmentation based on onset and offset detection

As described by Bregman (1990a), common onsets and offsets across frequency represent two important grouping cues that are utilized by the human auditory system to perceptually organize the acoustic input into auditory events. As described in Deliverable D2.3, the AFE provides three processors to extract onset and offset cues, namely the onset processor (`onsetProc.m`), the offset processor (`offsetProc.m`) and the binary onset and offsets map processor (`transientMapProc.m`).

The detection of individual auditory events based on common onsets and offsets is illustrated in Fig. 3.1 for a speech signal convolved with three binaural room impulse responses (BRIRs) from the Surrey database (Hummersone *et al.*, 2010). Specifically, BRIRs with different reverberation times were selected, namely from the anechoic chamber ($T_{60} = 0\,\mathrm{s}$), room A ($T_{60} = 0.32\,\mathrm{s}$) and room D ($T_{60} = 0.89\,\mathrm{s}$). It can be seen that the temporal smearing of the time-domain signal fine structure (left panels) increased with increasing reverberation time. The right panels show the corresponding ratemap representations together with the extracted onset (black vertical bars) and offset (white vertical bars) fronts. When comparing the ratemap representation of room D with the anechoic chamber, a substantially slower decay of energetic components can be observed, which is caused by the presence of reverberation. To improve the robustness of the detected onset and offset fronts in the presence of reverberation, only the most salient events were retained by applying temporal and across-frequency constraints to the auditory onset and offset maps, as specified in Tab. 4.11 in Deliverable D2.3.

In general, the strength of intensity differences decreases in reverberant conditions, effectively reducing the strength of the detected onsets and offsets. As a result, the number and the size of the detected onset and offset fronts reduced with increasing reverberation time. Despite this noticeable impact of reverberation, the majority of onsets can be reliably extracted in all three acoustic conditions. Consequently, the detected onsets and offsets can be combined and used as temporal markers to indicate and separate individual events in the acoustic input.

Segmentation based on amplitude modulation spectrogram (AMS) features

An approach for segregating sources based on amplitude modulation spectrogram (AMS) features was also developed. The segregation system consisted of a feature extraction front-end and a classification back-end (May *et al.*, 2015a, Bentsen *et al.*, 2016), as shown in Fig. 3.2. The target signal was reconstructed by applying the estimated binary mask (EBM) to the subband signals of the noisy speech, as illustrated by the dashed line. Each processing stage is described in detail in the following.

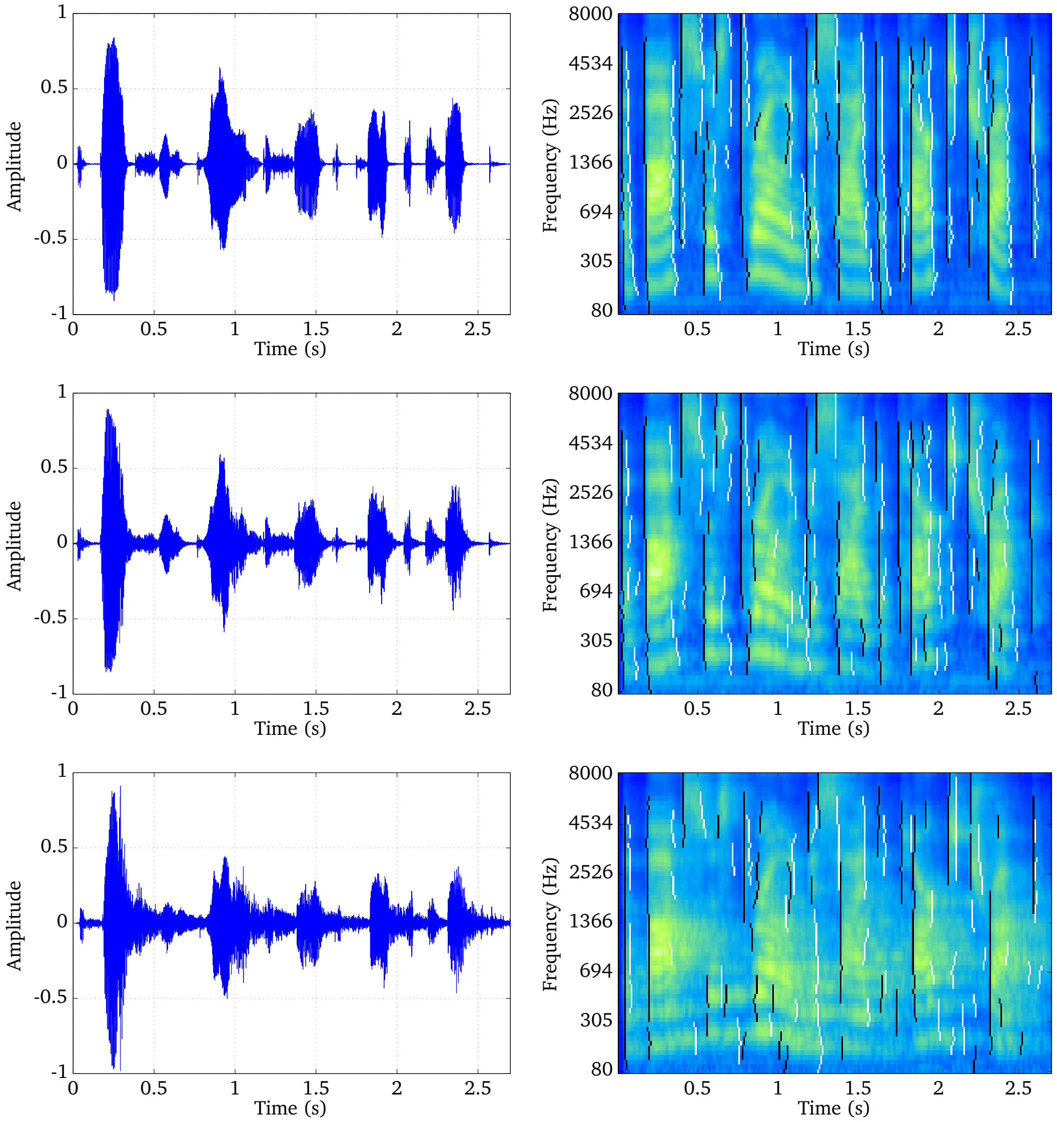


Figure 3.1: Time-domain signals (left panels) and the corresponding ratemap representations (right panels) with overlayed onset (black vertical bars) and offset fronts (white vertical bars) for speech in an anechoic chamber (top panels), room A (middle panels) and room D (bottom panels) from the Surrey database.

The distinct characteristics of speech and noise components were captured by amplitude modulation spectrogram (AMS) features (Kim *et al.*, 2009, May and Dau, 2014a, May *et al.*, 2015a, Tchorz and Kollmeier, 2003). To derive these, the noisy speech was sampled at a rate of 16 kHz and decomposed into 31 frequency channels by a Gammatone filterbank, whose

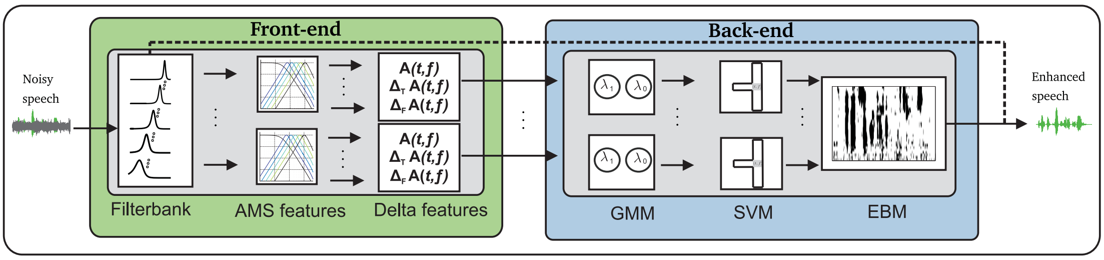


Figure 3.2: Block diagram of the segregation system that shows the main blocks of the feature extraction front-end and the classification back-end. The dashed line illustrates the reconstruction of the target by applying the EBM to the subband signals of the noisy speech.

center frequencies were equally spaced on the equivalent rectangular bandwidth (ERB) scale between 80 and 7642 Hz. The envelope in each subband was extracted by half-wave rectification and low-pass filtering with a cutoff frequency of 1 kHz. Then, each envelope was normalized by its median that was computed over the entire signal, which was shown to improve the generalization to unseen acoustic conditions (e.g., signal-to-noise ratios (SNRs) and room reverberation) (May and Dau, 2014a, May and Gerkmann, 2014). The normalized envelopes were then processed by a modulation filterbank that consisted of one first-order low-pass and five band-pass filters with logarithmically spaced center frequencies and a constant Q-factor of 1. The root mean square (RMS) value of each modulation filter was then calculated across time frames corresponding to 32 ms with 75 % overlap, resulting in a 6-dimensional feature vector for each time-frequency (T-F) unit $\vec{A}(t,f) = \{M_1(t,f), \ldots, M_6(t,f)\}^T$.

Context was explored in the front-end by appending delta features across time (Δ_T) and frequency (Δ_F) (Kim *et al.*, 2009, Han and Wang, 2012, May and Dau, 2013). The final feature vector for each individual T-F unit at time frame t and frequency channel f consisted of $\vec{X}(t,f) = \left[\vec{A}(t,f), \Delta_T\vec{A}(t,f), \Delta_F\vec{A}(t,f)\right]$, where:

$$\Delta_T\vec{A}(t,f) = \begin{cases} \vec{A}(2,f) - \vec{A}(1,f), & \text{if } t = 1 \\ \vec{A}(t,f) - \vec{A}(t-1,f), & \text{otherwise,} \end{cases} \tag{3.11}$$

$$\Delta_F\vec{A}(t,f) = \begin{cases} \vec{A}(t,2) - \vec{A}(t,1), & \text{if } f = 1 \\ \vec{A}(t,f) - \vec{A}(t,f-1), & \text{otherwise.} \end{cases} \tag{3.12}$$

The classification back-end consisted of a two-layer segregation stage (May and Dau, 2014a, May *et al.*, 2015a). In the first layer, a Gaussian mixture model (GMM) classifier was trained to represent the speech and noise-dominated AMS feature distributions ($\lambda_{1,f}$ and $\lambda_{0,f}$) for each subband f. To separate the feature vector into speech- and noise-dominated T-F units, a local criterion (LC) was applied to the *a priori* SNR. The GMM classifier output was given as the posterior probability of speech and noise $P\left(\lambda_{1,f}|\vec{X}\,(t,f)\right)$ and $P\left(\lambda_{0,f}|\vec{X}\,(t,f)\right)$, respectively. The second layer consisted of a linear support vector machine (SVM) classifier (Chang and Lin, 2001), which considered the posterior probability of speech $P\left(\lambda_{1,f}|\vec{X}\,(t,f)\right)$ across a spectro-temporal integration window $\mathcal{W}$ for each subband (May and Dau, 2014a):

$$\vec{\vec{X}}\,(t,f) := \left\{P\left(\lambda_{1,u}|\vec{X}\,(u,v)\right) : (u,v) \in \mathcal{W}\,(t,f)\right\}. \tag{3.13}$$

According to May and Dau (2014a), a causal and plus-shaped window function $\mathcal{W}$ was used here, whereas the window size with respect to time and frequency was controlled by Δt and Δf, respectively.

Other segmentation approaches

Another segmentation approach, blind source separation in the form of independent component analysis, was also tested within the TWO!EARS framework.

For this purpose, we simulated binaural data by convolving two source signals — music and speech — with binaural room impulse responses using the binaural simulator, either under anechoic conditions or with $T_{60} = 320\,\text{ms}$. The following combinations of source DOAs were tested: $[-90°, 90°]$, $[0°, 90°]$, $[45°, 90°]$.

We then applied the JADE algorithm (Cardoso and Souloumiac, 1993) in the short-time Fourier transform (STFT) domain, with permutation correction according to Hoffmann *et al.* (2012) to assess its general applicability to the binaural dataset. Exemplary resulting spectrograms for the reverberant $[0°, 90°]$ setup are shown in Figure 3.3.

As can be seen from the spectrograms, and as informal listening tests revealed, the approach is effective for the binaural data, and could be employed to separate two sources, albeit with some residual interference in the reverberant condition.

However, applying the same ICA algorithm to the AFE features turned out to be unsuccessful. This can be understood based on the non-linear frequency axis of the AFE. Its auditory filterbank leads to separation issues due to the wide bandwidths at the higher frequencies, which invalidate the inherent assumption of convolutive ICA that mixing in each frequency bin can be modeled appropriately by one linear matrix multiplication. It was therefore decided to not pursue the ICA-based segregation approaches further, which, in any case, would have been limited to separating out only two stationary sources.

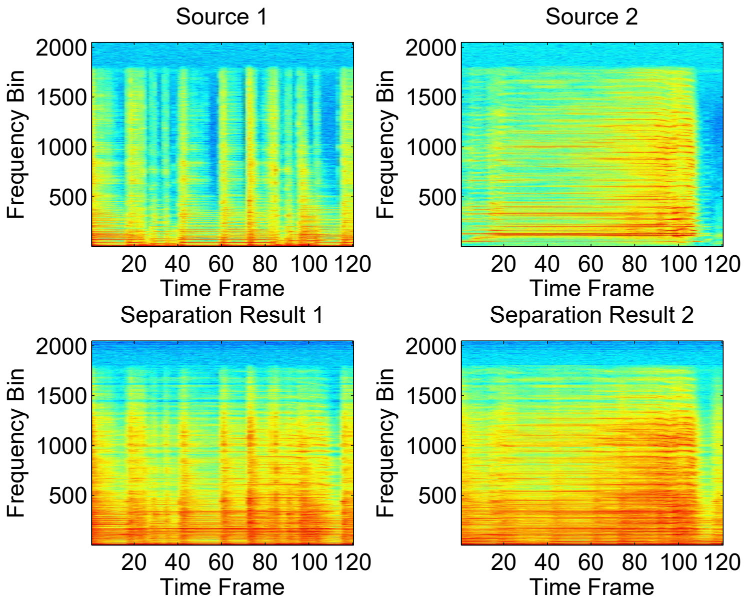


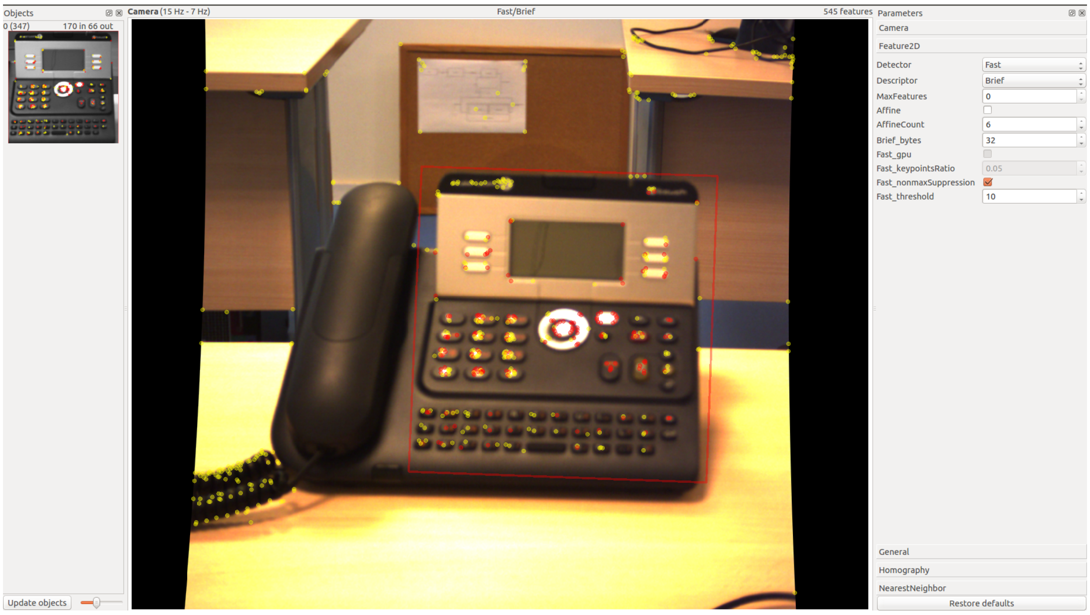
Figure 3.3: Result of independent component analysis applied to binaurally simulated data of one speech and one music signal.

3.2.2 Visual pattern detection

In Deliverable 3.4, methods for vision based labeling were outlined. A subset of them concerned the appearance based detection and tracking, into monocular sequences, of multiple humans standing upright. Running this algorithm on the two cameras composing a stereoscopic rig and combining its results with calibration parameters enables the coarse recovery of their positions. The complementary subset of methods addressed the detection and segmentation of objects. In view of the difficulty of this problem, the first attempts were conducted on the RGB-D point cloud delivered by an active depth sensor (such as the Asus Xtion Pro Live or the Microsoft Kinect). The evaluation of several open source packages led to the selection of the *linemod* algorithm. The initial learning step of this method consists of associating to each object of interest a database of templates made up with color gradients and surface normals extracted from the RGB-D point cloud. The detection stage consists of matching these templates against the incoming point cloud within a multiscale sliding window approach. This method led to good results. It can also apply to the point cloud generated by a stereoscopic sensor, provided it is dense enough.

An anthropomorphic visual setup was designed for the KEMAR head and torso simulator (HATS). It consists of 3D-printed glasses which perfectly fit the KEMAR face and incorporate commercial micro-cameras. To obtain high 3D accuracy after triangulation when tracking humans, lenses with 12 mm focal length were first mounted on the glasses. After further experiments, it appeared that the restricted overlap of the two camera fields of view was unsuited to object perception. Indeed, a difficult pre-scanning of the scene had to be set up by moving the head, so as to define the areas from which a dense RGB-D point cloud should be extracted. Consequently, lenses with smaller focal length were installed. However, these led to a point cloud which was too sparse for a proper functioning of *linemod*. So, appearance-based methods for object detection and segmentation have been investigated since then.

The method relies on a database made up of pre-recorded reference images of each object. To improve the performance of the subsequent detection, these must be taken by the robot at various angles and distances. A first offline process extracts visual descriptors from the reference images by means of an off-the-shelf ROS package. This is a manual phase, which enables the intuitive selection of suitable descriptors and/or the rejection of void images. The online detection of the object is based on the detection of similar visual descriptors in the test images taken from the two cameras. A combination of detectors and descriptors providing enough speed and robustness to perception changes has been determined. The segmentation of the object is expressed in terms of the bounding boxes of the detected features in each camera. Its location is evaluated by triangulation.



Detector	Fast
Descriptor	Brief
MaxFeatures	0
Affine	□
AffineCount	6
Brief_bytes	32
Fast_gpu	□
Fast_keypointsRatio	0.05
Fast_nonmaxSuppression	☑
Fast_threshold	10

Figure 3.4: Extraction of visual descriptors (yellow) in the reference (left) and test (right) image of a telephone. Detection of common visual descriptors within the two images (red).

3.2.3 Tracking

Sound source tracking and active exploration

An essential part of auditory scene analysis (ASA) is the localization of sound sources in the environment (Bregman, 1990b). Recently, this task has been addressed by various studies in the context of robot audition. A significant advantage of robotic agents over static acoustic sensors is the ability to move and actively explore the environment. This has triggered research on algorithms for active listening which incorporate feedback into the audition process. A prominent area of research in this context is active localization. Inspired by the abilities of human listeners to improve the assessment of auditory scenes through head and body motions (Blauert, 1999), this has lead to a variety of approaches for different applications.

An early study on how head movements are utilized by humans to resolve front-back ambiguities in sound localization was introduced in Wallach (1940). Based on these findings, several computational localization models incorporating head-movements have recently been introduced in Ma *et al.* (2015e,a), Schymura *et al.* (2015). Additionally, methods based on whole-robot motion were proposed, using either microphone arrays (Evers *et al.*, 2016) for auditory simultaneous localisation and mapping (SLAM) or binaural sensors (Bustamante *et al.*, 2015, 2016) for active localization of sound sources. The latter work was an (unfunded) contribution to Two!Ears by researchers at LAAS-CNRS, and has been implemented on the 'Jido' mobile robot.

An important question in this context is how robot motion can optimally support localization. For instance, the framework introduced in Bustamante *et al.* (2016) describes an information-based feedback control scheme. It selects controls to maximize information gain of the estimated posterior PDF representing the assumed source location. Similar approaches have also been proposed in the broader context of active exploration, aiming at maximizing the robot's knowledge about the environment (Bourgault *et al.*, 2002, Thrun *et al.*, 2005). The framework proposed here complements the approach of Bustamante *et al.* (2015, 2016), differing from it in two important respects.

Firstly, an extended binaural localization model is introduced, which incorporates azimuth and distance information into the localization process. Binaural models for sound distance estimation based on the direct-to-reverberant ratio (DRR) (Lu and Cooke, 2010) or statistical signal parameters (Georganti *et al.*, 2013) have already been proposed. In this work, sound distance is modelled using the interaural coherence (IC) of reverberant binaural signals, which was originally described in (Vesa, 2009). It was reported that a decreasing DRR results in a decrease of correlation between both binaural channels, which can be approximately represented by the interaural coherence (IC). The proposed framework shows that incorporating distance information improves the localization abilities of the robotic agent in reverberant environments, compared to a conventional bearing-only observation model.

Secondly, a closed-loop feedback control scheme is proposed, aiming at minimizing the entropy of the belief state while approaching a specific goal position. The robot motion will be chosen from a set of pre-defined actions based on a Monte Carlo exploration (MCE) approach (Thrun *et al.*, 2005, Nakhost and Müller, 2009). This allows for the selection of movements by minimizing the predicted localization uncertainty. This approach extends previously proposed methods (Bustamante *et al.*, 2015, 2016) with the possibility for trade-offs between exploratory and goal-direction motions.

System dynamics. The localization model used here assumes a generic nonlinear state space representation

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{x}_{\mathrm{S},k} \\ \boldsymbol{x}_{\mathrm{R},k} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{\mathrm{S},k-1} \\ f(\boldsymbol{x}_{\mathrm{R},k-1},\, \boldsymbol{u}_k) \end{bmatrix} + \boldsymbol{v}_k \tag{3.14}$$

$$\boldsymbol{y}_k = g(\boldsymbol{x}_k) + \boldsymbol{n}_k, \tag{3.15}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ denote the state and observation vectors and $\boldsymbol{u}_k = \begin{bmatrix} u_{\mathrm{L},k} & u_{\mathrm{R},k} \end{bmatrix}$, $u_{\{\mathrm{L},\mathrm{R}\},k} \in [-1, 1]$ represents the control input at the left and right wheel actuators. The system dynamics in Eq. (3.14) is composed of two augmented state vectors $\boldsymbol{x}_{\mathrm{S},k} = \begin{bmatrix} m_{\mathrm{x},k} & m_{\mathrm{y},k} \end{bmatrix}^T$ and $\boldsymbol{x}_{\mathrm{R},k} = \begin{bmatrix} p_{\mathrm{x},k} & p_{\mathrm{y},k} & \theta_k \end{bmatrix}^T$, representing the position of the sound source in Cartesian coordinates and the robots pose including its heading direction θ_k, respectively. The sound source is assumed to be static up to state noise, whereas the robot dynamics is governed by a nonlinear motion model $f(\boldsymbol{x}_{\mathrm{R},k-1},\, \boldsymbol{u}_k)$ (Thrun *et al.*, 2005). Observations are predicted using a nonlinear mapping function $g(\boldsymbol{x}_k)$. Both state and measurement noise characteristics are modeled as additive, zero-mean Gaussian random variables $\boldsymbol{v}_k$ and $\boldsymbol{n}_k$ with corresponding covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$.

Measurement model. The measurement model in Eq. (3.15) introduces a nonlinear mapping function $g(\boldsymbol{x}_k)$ from states $\boldsymbol{x}_k$ to observations $\boldsymbol{y}_k$. As described previously, ITDs, ILDs and IC are used as primary binaural cues for the proposed localization model. ITDs and ILDs are cues that correspond to the relative angle ϕ_k between the sound source and the heading direction of the robot. In addition to that, the IC is used here to model the sound-to-receiver distance d_k. Hence, the system's state is first mapped from Cartesian to polar coordinates. The mapped state is subsequently used to predict binaural observations by a regression model

$$g(\boldsymbol{x}_k) = \boldsymbol{W}^T \boldsymbol{\Phi}(\phi_k(\boldsymbol{x}_k),\, d_k(\boldsymbol{x}_k)), \tag{3.16}$$

where $\boldsymbol{\Phi}(\phi_k(\boldsymbol{x}_k),\, d_k(\boldsymbol{x}_k))$ represents the regressors and $\boldsymbol{W}$ is a matrix of regression coefficients. The latter are computed via multivariate linear regression (Bishop, 2006, Chap. 3) using rendered binaural room impulse responses (BRIRs) and white noise as stimulus signal. A finite Fourier-series representation (Oppenheim *et al.*, 1999) is used to model angle-dependent regressors in $\boldsymbol{\Phi}(\phi_k(\boldsymbol{x}_k),\, d_k(\boldsymbol{x}_k))$, whereas the distance-related regressors are modeled via polynomials.

The residuals obtained after training are used to estimate the measurement noise covariance matrix $\boldsymbol{R}$. This model extends the approach from Schymura *et al.* (2015), which was restricted to azimuth prediction based on a spherical head model. The use of a regression function according to Eq. (3.16) yields a more flexible framework, which can be trained on both measured or simulated BRIRs.

State estimation. The robot position is assumed to be known and deterministic for the current implementation, depending only on the applied control input. Hence, state estimation reduces to recursively computing the posterior PDF or belief of the sound source position $p(\boldsymbol{x}_{\mathrm{S},k} \,|\, \boldsymbol{y}_{1:k},\, \boldsymbol{u}_{1:k})$. This distribution might have multimodal characteristics, as front-back confusions can occur due to the ambiguous nature of binaural cues (Blauert, 1999). This renders Bayesian filtering methods based on unimodal Gaussian assumptions inappropriate for this task.

To overcome these limitations, a Gaussian mixture sigma point particle filter (GM-SPPF) (van der Merwe and Wan, 2003) is used, which represents the posterior PDF as a GMM. However, an analytic evaluation of the entropy is not possible for GMMs. To obtain a measure of uncertainty, an approximation of the belief states entropy is necessary. As described by van der Merwe and Wan (2003), the conditional mean state estimate $\hat{\boldsymbol{x}}_{\mathrm{S},k} = E\{\boldsymbol{x}_{\mathrm{S},k} \,|\, \boldsymbol{y}_{1:k}\}$ and the error covariance matrix $\hat{\boldsymbol{P}}_k = E\{(\boldsymbol{x}_{\mathrm{S},k} - \hat{\boldsymbol{x}}_{\mathrm{S},k})(\boldsymbol{x}_{\mathrm{S},k} - \hat{\boldsymbol{x}}_{\mathrm{S},k})^T\}$ can be utilized to approximate the resulting GMM representation of the belief state by a unimodal Gaussian distribution. Hence, the entropy of the corresponding PDF is defined as

$$H(\boldsymbol{x}_{\mathrm{S},k}) = \frac{1}{2} \log \Big((2\pi e)^D \cdot |\hat{\boldsymbol{P}}_k| \Big), \tag{3.17}$$

where $D = 2$ is the dimensionality of the state vector $\hat{\boldsymbol{x}}_{\mathrm{S},k}$.

Monte Carlo exploration (MCE). MCE has been widely used in the context of robotics as a means to actively gain information about uncertain entities in the environment (Thrun *et al.*, 2005). A prominent application for exploration techniques is simultaneous localisation and mapping (SLAM), where the robot tries to actively explore its environment to reduce uncertainty in the map-building process (Stachniss *et al.*, 2004). However, unrestricted exploration is usually not desired in many applications, for instance, if the robot must reach a specific goal position.

Hence, the implementation in Two!Ears focuses on using MCE to construct a policy $\pi(\boldsymbol{x}_k)$, which allows the robot to select appropriate actions that maximize a specific reward function which trades off exploration and goal-directed movements. Thereby, exploration serves two purposes: using rotational movements to resolve front-back ambiguities and translatory movements to support distance estimation via triangulation.

A generic MCE algorithm as described in (Thrun *et al.*, 2005, Chap. 17) is adopted. It aims at finding the action $\boldsymbol{u}_{k+1}$ that maximizes the expected reward at the subsequent time step. To assure computational tractability, the continuous-valued controls are discretised, yielding a set of N_u actions $\mathcal{U} = \{\boldsymbol{u}_{k+1}^{(1)}, \ldots, \boldsymbol{u}_{k+1}^{(N_u)}\}$.

A control policy is obtained by running Monte Carlo simulations that predict the immediate reward of all actions in the set $\mathcal{U}$, using the system dynamics and measurement model as a black-box simulator. The procedure is initialized by drawing a set of N samples $\tilde{\boldsymbol{x}}_{\mathrm{S},k}^{(i)} \sim p(\boldsymbol{x}_{\mathrm{S},k} \,|\, \boldsymbol{y}_{1:k}, \boldsymbol{u}_{1:k})$, $i = 1, \ldots, N$ from the belief distribution at the current time step. As the robot pose is assumed to be deterministic, the full state samples are represented as $\tilde{\boldsymbol{x}}_k^{(i)} = \begin{bmatrix} \tilde{\boldsymbol{x}}_{\mathrm{S},k}^{(i)} & \boldsymbol{x}_{\mathrm{R},k} \end{bmatrix}^T$. Subsequently, a particle filter update step is conducted for all available actions by sampling observations using the measurement model $\tilde{\boldsymbol{y}}_{k+1} \sim p(\boldsymbol{y}_{k+1} \,|\, \tilde{\boldsymbol{x}}_k^{(i)})$. The update steps generate a set of predicted posterior PDFs $p(\tilde{\boldsymbol{x}}_{\mathrm{S},k+1} \,|\, \tilde{\boldsymbol{y}}_{1:k+1}, \boldsymbol{u}_{1:k+1})$ along with their corresponding entropies, using the approximation introduced in Eq. (3.17). This allows for the calculation of the immediate reward $r(\tilde{\boldsymbol{x}}_{\mathrm{S},k+1}^{(i)}, \boldsymbol{x}_{\mathrm{R},k+1})$ based on the negative entropy.

All obtained immediate rewards are averaged across all Monte Carlo simulations, yielding an approximation of the expected reward $R(\boldsymbol{x}_k, \boldsymbol{u}_{k+1})$. The policy obtained by MCE is a greedy policy, as it exclusively considers the expected reward at the next time step. Hence, action selection can be conducted by evaluating $\pi(\boldsymbol{x}_k) = \arg\max_{\boldsymbol{u}_{k+1}} R(\boldsymbol{x}_k, \boldsymbol{u}_{k+1})$.

Reward function. The proposed framework relies on an immediate reward function that constitutes two possibly conflicting goals: minimizing localization uncertainty by exploratory movements and reaching a specified goal position $\boldsymbol{x}_{\mathrm{G}}$. Hence, a trade-off has to be found which balances exploration and goal-directed actions. This is expressed via the function

$$r(\tilde{\boldsymbol{x}}_{\mathrm{S},k+1}^{(i)}, \boldsymbol{x}_{\mathrm{R},k+1}) = -\Big(\lambda \|\boldsymbol{x}_{\mathrm{R},k+1} - \boldsymbol{x}_{\mathrm{G}}\|_2 + (1-\lambda) H(\tilde{\boldsymbol{x}}_{\mathrm{S},k+1}^{(i)})\Big), \tag{3.18}$$

where $\lambda \in [0, 1]$ is a trade-off parameter that balances the minimization of the Euclidean distance from the robot to the goal position and exploration achieved by considering the entropy predicted for the next time step.

An example of a trajectory generated by MCE is depicted in Fig. 3.5. It shows that the proposed control scheme makes use of translatory and rotational movements to support distance estimation and reduce front-back ambiguities, while approaching the goal position.

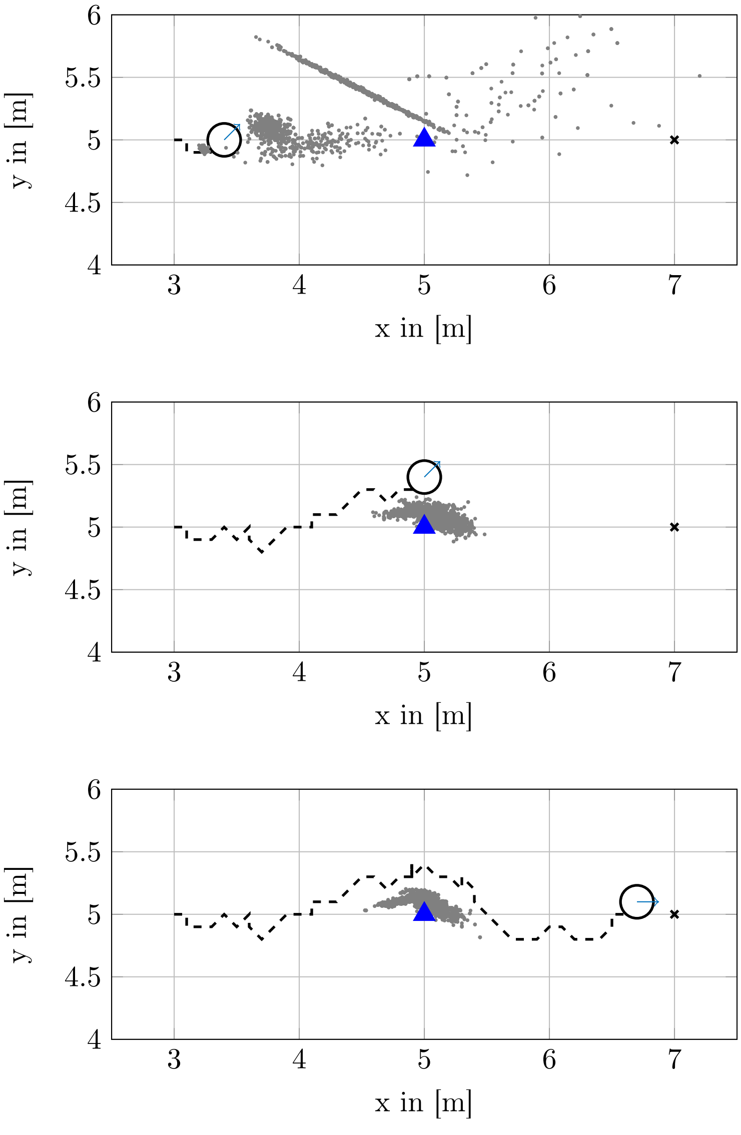


Figure 3.5: Active exploration in reverberant conditions with $T_{60} = 250\,\text{ms}$. The source position (blue triangle) is located between the initial robot position and the goal position (black cross). MCE is performed with $\lambda = 0.5$, generating a trajectory (dashed black line) towards the goal which helps to reduce the uncertainty of the particle set (gray dots).

3.3 Task 3.3: Knowledge-base acquisition

3.3.1 Auditory Machine-Learning Training and Testing Pipeline

We have developed an object-oriented framework for building and evaluating models for auditory sound object annotation and assigning attributes to them. The models were obtained by inductive learning from labeled training data. The framework is called the Auditory Machine-Learning Training and Testing Pipeline (AMLTTP). It is tightly coupled with the Two!Ears system and its components. The AMLTTP consists of two parts, namely generating model training data and the training as well as testing of models, where each part can be broken down into further sub-stages and components. While the pipeline is designed with flexibility in mind and is extendable to new target attributes, data features, or model and training algorithms, it serves the specific purpose of training and evaluation of block-based auditory object-*type*, object-*location*, and *number-of-sources* classifiers using data from simulated auditory scenes generated within the same framework.

Multi-conditional auditory scene simulation

In stage one of the data generation, ear-signals are produced from audio files using the Binaural Simulator from the Two!Ears system. This can be done under various conditions, set up through a configuration object of the AMLTTP:

- An arbitrary number of sources can be created, either diffuse (non-directional), or point-sources with specific positions relative to the head, whose azimuth can be set as well.
- The "head", i.e. the HRIR used, can be exchanged. By default, it is defined as a KEMAR head.
- Sources can be set up to emit white noise or specific audio files.
- Sources can be set up to playback one audio file and then mute, or loop over this audio file, or playback audio files from a set in random order, for a defined duration.
- Ear-signal-level average SNRs between sources can be fixed. They are defined as the ratio of powers between the individual sources' ear-signals.
- Simulated reverb (shoe-box room model) can be defined, or
- BRIRs can be used instead of HRIRs, in which case source positions are defined by the BRIR. Multi-speaker BRIRs allow for scenes with multiple sources.

- The ear-signals average level can be adjusted.

The above conditions together define a scene configuration. An arbitrary number of such scenes can be specified and will be simulated to produce *multi-conditional* training data. While models trained from single-conditional data, i.e. data generated from a single configuration, possibly perform better within this condition because of their specialization, we opted to train models using multi-conditional data. The motivation to use multi-conditional data is to train models that can generalize across different conditions by using data that enforces invariance. Both condition modes were used. Additionally, different scene configurations can be used across training and testing to perform, so called, "cross-testing" of configurations. This way, models can be evaluated under conditions that were not included in their training data, which allows for assessing the models' robustness. Multi-conditional models can still be tested on individual conditions to evaluate them in more detail.

Sample feature generation

After ear-signals have been generated, stage two is about constructing features for all the samples. This stage consists of three subsequent steps: First, basic auditory feature like ratemaps or ILDs are created by the auditory front-end (AFE), which takes the ear-signals as inputs. Then, these basic feature streams get cut into blocks with predefined length and overlap. The AFE operates directly on the ear-signal streams to avoid artefacts on borders of blocks. After block extraction, the features within each block get transformed with statistical or any other functions into sample feature vectors to use as model input. This is performed in the AMLTTP by so-called FeatureCreators, producing one sample vector per block.

FeatureCreators, which inherit from a common base interface and are thus seamlessly usable, are modules to be implemented by the user of the AMLTTP to construct data vectors in a form that is suitable for the model to be trained. Part of implementing the common interface involves defining "AFE-requests" which set up the AFE's output. These requests are made in exactly the same way as in the Two!Ears blackboard system. To be able to later evaluate models on a feature-level, e.g. for dimensionality reduction, FeatureCreators automatically produce a detailed description of each feature dimension. Masks can be used to incorporate results of feature selection methods and only train or test with selected dimensions. The description of each feature also has its use for combining the AMLTTP with external third-party tools such as the Caffe library for training neural networks, as showcased in 3.3.3.

Label creation

Similar to the generation of sample vectors described in the section above, the corresponding target vectors (also called "labels") are produced by "LabelCreators". These labels are used in the supervised training of models, and serve as ground truth in the testing of models. Analogous to FeatureCreators, LabelCreators also implement a common interface for quickly creating new labellers describing different object attributes. Any of these object attributes are derived from annotations at different levels of abstraction (e.g. active sources, source energies, etc.) around the ear-signals streams which were produced in the scene generation stage and are passed through the pipeline. LabelCreators for producing *source class*, *source location* and *number of sources* targets are already available and can be used 'right out of the box', including combinations thereof. Binomial, multinomial, or regression targets can be produced, and combined to form multivariate labels.

Model training algorithms

Consistent with the concept of modularity and extendability, common interfaces exist for *models* and their corresponding *model trainers*. Any algorithm for model construction can be used, and any model type can be constructed and tested. A class inheriting from the interfaces can implement its own technique and can be plugged into the AMLTTP. Along with this, wrapper trainers for conducting stratified cross-validation or hyper-parameter search are provided and can be used with any algorithm. We have already implemented:

- Elastic net with Gaussian or poisson regression;
- Elastic net with binomial or multinomial classification;
- Online gradient descent with logistic, Gaussian, or hinge regression, with or without regularization;
- Support-vector machines.

It is also possible to let the AMLTTP generate and save training samples to file, in order to use external training tools such as the Caffe library for training neural networks as presented in 3.3.3.

Tight coupling with the Two!Ears blackboard-system

The AMLTTP and the Two!Ears blackboad-system are tightly coupled in two central aspects: AMLTTP-trained models can be used from within the blackboard system, and blackboard system knowledge sources can be used from within the AMLTTP.

Models created by the AMLTTP can be plugged into the blackboard-system and be used there without any modification or interface adjustments: they fit into the respective AMLTTP-knowledge source. They automatically employ the right FeatureCreator and model, feeding the system with hypotheses about auditory object attributes. Regardless of whether the ear-signals that are fed into the blackboard system were produced using a binaural simulator or acquired from actual microphones, features are produced in exactly the same manner as in the AMLTTP. The same block length is used, the same AFE requests are made, and the features are constructed using the exact same code.

A particularly interesting feature is that the reverse of this process is also possible: existing knowledge sources and the models that come packaged with them can be included as "data processors" in the AMLTTP through a blackboard-KS wrapper interface. This enables the incorporation of other models' hypotheses about the auditory data, and modifying the data passed to the FeatureCreator or adding to it, or modifying the attribute annotations passed to the LabelCreator. Models that use other models' outputs can be built through this feature, incorporating into the training process the properties of the other models instead of only adapting to ground truth. Four knowledge sources have already been plugged into the AMLTTP through the blackboard-KS wrapper: DnnLocationKS, SegmentationKS, NumberOfSourcesKS and IdentificationKS (see Deliverable D6.1.3 for descriptions of these knowledge sources).

Utility

A lot of work has gone into making the AMLTTP both easy and effective to use. While the first is accomplished through a clean high-level interface, the latter is enabled to a large extent through the following two points:

- All products from intermediate stages, such as the generated ear-signals, or features produced by the AFE, are saved together with their corresponding configurations inside a caching system. Since these stages can be very time-costly for large data sets: repeating trainings with a different model, resuming a process after a crash, etc., are made possible without having to recompute everything again. Also, data are saved in such a way that they will be recombined automatically whenever parts of a configuration have already been computed before. This saves a lot of computation time and helps assigning the produced data to many different experiments' configurations and reproduce experiments more easily.
- The AMLTTP can be run concurrently on many processes and/or machines, whilst working from the same data. File semaphores ensure that processes don't interfere with one another when operating on the same configuration in the same stage.

3.3.2 Feature selection and dimensionality reduction

In chapter 3 of Deliverable D3.4, we investigated the application of feature construction by principle component analysis (PCA) and independent component analysis (ICA) to improve auditory object type classification. Preprocessing with PCA did not yield any clear benefit over using elementary features. Using ICA even decreased performance. Since both preprocessing techniques come with additional computational costs, we did not consider them further.

We analyzed the potential benefits of data-driven feature selection techniques for sound-type classification. We trained type classification models using three schemes: the first scheme employed linear classifiers and support vector learning on all features. The second scheme employed the Least Absolute Shrinkage and Selection Operator (Lasso) technique, a linear (logistic) regression model trained with a regularization term which penalizes a large number of input features (Tibshirani, 1996). The second scheme thus selects informative features while simultaneously constructing the sound-type classifier. The third scheme implements a two-step procedure by combining the first two schemes. First, features are selected using the Lasso technique, second, a linear classifier is constructed using support vector learning on the selected features. This choice of schemes allowed for an assessment of the potential benefits of data-driven feature selection. It also allowed for an assessment of robustness of the results and the potential dependencies of classification performance on different training methods. The above mentioned machine learning schemes were applied to three sets of auditory scenes individually and in combination, and classifiers were constructed for the block-based classification of sound-type.

In summary, we found that the additional feature selection step employed by the third learning scheme usually did not improve average classification performance of a linear SVM compared to a setting where all elementary features are considered. The number of features to be included, however, could be drastically reduced without compromises on classification performance. This also greatly cuts down on computation time necessary for training the classifiers and leads to shorter classification times. Since the Lasso algorithm selects features and trains models in one step and showed performance competitive with the SVM, we concluded that it is a technique well suited for our needs. Each of our models trained with this technique thus uses its own dedicated feature set tailored to the task. Further details are provided in Deliverable D3.4.

3.3.3 Statistical learning

The AMLTTP (see Sec. 3.3.1) served as our framework for building auditory object annotation models through statistical learning. Using it, we trained models that (a) detect auditory objects of specific type in the ear-signals streams, (b) classify auditory object streams segregated according to the objects' locations, (c) jointly classify and localize auditory objects, and (d) estimate the number of active sound sources.

Detection and identification of auditory objects

In order to build models that detect acoustic objects of a particular type, and hence identify them, we used statistical learning in the form of logistic binomial classification with Lasso (Tibshirani, 1996). First results for sound event identification with using Lasso and other classification algorithms have already been reported:

- Support-vector machines, in Deliverable D3.2, section 5.2.4, and D3.4, sections 3.3.2 and 4.2.5
- Gaussian mixture models, in Deliverable D3.2, section 5.2.5
- Mixture of factor analysers, in Deliverable D3.2, section 5.2.7, and D3.4, section 4.2.6
- Logistic classification with Lasso, in Deliverable D3.4, sections 3.3.2 and 4.2.5

Since logistic Lasso had the highest computational efficiency without loss in average classification performance, we concentrated further investigations to this method, utilizing the "GLMNET" package (Friedman *et al.*, 2010, Qian *et al.*, 2013)) in the AMLTTP.

Lasso is a classification method with an embedded feature selection procedure. An important factor in determining the sparsity of the final model is the strength of the L_1 regularization term, which is controlled by the regularization parameter λ. For adjusting its value, we performed a 7-fold cross-validation on the training set for all 100 candidate values along the regularization path. We then chose the value with the best cross-validation performance, and used it with the model trained on the full training set. Since we are interested in obtaining classifiers that work well both on detecting "positives" (auditory objects of the respective type) as well as classifying "negatives" (blocks in which the respective auditory object is not present) as such, we want to optimize both sensitivity and specificity, and used a variant of balanced accuracy (BAC) that not only equally weighs sensitivity and specificity, but also penalizes differences between them, as the performance measure for this optimization:

$$BAC_2 = 1 - \sqrt{\left(\frac{(1 - sensitivity)^2 + (1 - specificity)^2}{2}\right)} \tag{3.19}$$

Data were always split into a training set for model building and a test set for estimating the generalization performance of the classifiers. In order to ensure that a block from the training set and a block from the test set never contained parts of the same sound file, training-test splits as well as cross-validation splits were conducted at the level of the original sound files. Additionally, splits were done stratified with respect to the classes distribution. This means that the set of sound files for each class individually was randomly split into training set and test set, 75% vs 25% in our general setup. Only sounds from the training set were used to generate auditory scenes for building the classification models, and only sounds from the test set were used to generate auditory scenes for evaluating the prediction performance.

For each sound type, we trained a binary classifier in a one-vs.-all scheme, where each classification model decides whether a given auditory stream contains a sound event corresponding to this particular type.

In section 3.4.2, we describe different schemes and features used for training our identification models. In section 4.3, we conduct a detailed analysis of sound event detection with these models. We systematically investigate the impact of superimposed distracting sources and demonstrate how robust models were obtained by including a range of conditions in the training data, a procedure called *multi-conditional training*.

When training with multiple conditions, performance was always evaluated on individual “single conditions”. We also turned back to standard BAC when evaluating our models on the test data for its direct interpretability.

Classification of segregated auditory object streams

The previous section described the statistical model building process of classifiers that detect particular auditory objects in the ear-signal streams. These models operate on the “full” and all streams, with all sources' signals overlapping and superimposed. We followed a second approach of building models that identify objects in *segregated* auditory streams for two reasons:

1. Concurrent auditory objects may be detected and identified with increased precision, if they are separated into individual streams where each stream maximizes information about the object it encloses.
2. If only one source emits sound at a time it is possible to bind the attributes of the auditory objects such as type and location through their co-occurrence in time. This cannot be achieved in scenarios with concurrent sound sources. If however, the ear-signals are segregated into streams corresponding to one source each, this problem can be solved and objects can be formed.

The operating point of statistical models (i.e. the decision thresholds) are tightly optimized to the distributions of the data they operate on (in our case, the features built from auditory representations of the ear-signal streams). The segregation of this data changes these distributions. Therefore, claim (2) made above would most probably only hold if models are *trained on the segregated streams*, rather than applying the full-stream trained models on the segregated streams.

Therefore, we deemed it necessary to *include the segregation model* – see "Segmentation based on von Mises distributions" in section 3.2.1 for a description – into the data generation and training process of the detection models. Consequently, since the segregation model operates based on the azimuths of different sources, the respective azimuth distribution estimation model (see Sec. 3.4.1) had to be incorporated as input to the segregation model. While it would have been possible to use location ground truth as input to generate training data, it seemed reasonable to rather include the respective model and its particular output properties and distribution. Both models were included through the AMLTTP's ability to include blackboard-system knowledge sources (see Sec. 3.3.1, "Tight coupling with the Two!Ears blackboard-system"). Thus, the segregated streams as soft-masked by the segregation model served as auditory representations when generating features for the detection models. Other than that, the models were built in the same way as the "full-stream" models described in the section above.

In section 4.5.1, we analyze these models and evaluate their performance. We compare them to the full-stream models, with respect to the advantage of training on already segregated streams, and with respect to the influence of the estimation of number of active sources.

Joint classification and localization of auditory objects

We developed models that simultaneously identify and localize a set of auditory objects within an ear-signal's stream. This attempts to solve the binding problem, assigning different characteristics to the same object, in a bottom-up fashion. Neural networks were trained end-to-end to jointly optimize for the identification and localization tasks. The two tasks are combined by defining the output of the network as a two-dimensional map, where columns represent azimuth location bins and each row represents a sound type. Each node in the network's output layer represents a classifier trained in a one-vs.-all fashion. Its response is interpreted as the probability of a sound type present at a particular azimuth.

The AMLTTP is used for multi-conditional auditory scene simulation and extraction of features. A LabelCreator is defined that combines both sound type and azimuth location ground truth for each sample, based on the scene configuration and annotated onsets and offsets. The extracted features and their corresponding labels are fed into the Caffe library (Jia *et al.*, 2014)[1] for training neural networks using gradient-based learning. Caffe is an open source library for developing and training deep neural networks. Networks are developed in Caffe by defining their layered graph structure (i.e. how many layers, nodes per layer, connectivity between layers), the operations performed at each layer (e.g. convolution, non-linear transfer function, inner product) and the loss function. We also used Caffe for defining and configuring the learning algorithm. In our case, the parameters of mini-batch gradient descent, referred to in Caffe as the solver. Using the Caffe library has the advantage of accelerating computationally intensive operations by offloading them onto a Graphical Processing Unit (GPU) to quickly leverage the large amount of data used in training. The library handles the execution of feeding input features into the network, propagating them through its layers, computing the loss function and computing the gradients of the weights throughout the network through backpropagation algorithm for the solver to update the network's weights. After several iterations through the training dataset, the network is applied to the test set to assess its generalization performance. A trained network represented by the layer definition and final set of weights are deployed into an IdentityLocationKS of the Two!Ears system.

The procedure we use to train the models is to first simulate the auditory scenes in the AMLTTP. There we also define and extract the set of features. These features are then saved to disk and reformatted[2] to achieve compatibility with the Caffe library. The description that the AMLTTP provides for each individual feature element in the feature vector is essential for resolving the spatial dimensions of the network's input features. The training process provided by Caffe is then configured and launched.

Estimation of number of sources

For computational auditory scene analysis (CASA), estimating the number of active sound sources is an important part in analyzing scenes, not only as a number of interest in itself, but also since other tools like stream segregation (see Sec. 3.2.1) depend on it. We thus built statistical models for estimating this number, and incorporated the means to do so into the AMLTTP. We investigated three learning schemes:

- Treating the estimation as a classification problem and building models with multinomial logistic classification with Lasso;
- Building regression models through Gaussian regression with Lasso;
- Building regression models with underlying Poisson distribution assumption, regularized through Lasso.

[1] Caffe library URL: http://caffe.berkeleyvision.org/

[2] Tools for preparing data for the Caffe library https://github.com/nigroup/nideep

As done for our detection models described above, we used the “GLMNET” package (Friedman *et al.*, 2010, Qian *et al.*, 2013) in the AMLTTP for all three cases, with the same methods of cross-validation, splitting into training and testing sets, etc. However, differing from this description were of course the performance measures used for optimizing λ: for the classification model, we used “multinomial BAC”, which extends balanced accuracy to the multi-class case (weighing all classes’ sensitivities equally); for the regression models, we used (negative) squared error.

The choice of using Lasso for building number-of-sources estimators was also influenced by the fact that we wanted to “offer” a wide range of representations to the model, among them ILDs, ITDs and so-called DUET-features – see description in section 3.5.2 – and thus needed a model training algorithm with effective feature selection; generalized linear models with L_1 regularization are exactly such, as proven in Ng (2004).

In addition to the basic auditory representations, we wanted to present two more abstract and high-level “features” to the models that we suspected to be helpful in determining the number of active sources: (a) the azimuth sources distribution generated by the azimuth distribution estimation model (Sec. 3.4.1), and (b) the detection probability outputs of our models as described in Sec. 3.4.2 and “Detection and identification of auditory objects” above. In the same way as in the case of our detection models working on segregated streams, we thus incorporated those trained models into the data generation and training process of the number-of-sources estimators, accessing their knowledge.

Section 3.5.2 provides a more detailed description of the model and features, with section 4.6 featuring an analysis of them.

3.3.4 Logical rule learning

Several approaches were taken to learning logical rules from imperfect data. In particular, decision trees were used to learn the characteristics of time-frequency regions that are likely to provide reliable information for source localisation. The software architecture includes the facility to interface with an engine for the Prolog programming language, enabling the scheduler to reason using sets of rules expressed as predicates in Prolog.

The use of rough sets theory was envisioned in the original workplan, but as work on the TwoEars architecture progressed this became inappropriate. The graphical-model-based blackboard system is based on the assumption that latent variables exist, denoting the *actual state* of the environment (e.g., states such as ‘speaker one is active’ or ‘there is an emergency’). We aim to estimate the probability distribution of these latent variables, whereas the adoption of rough sets theory implies logical reasoning with variables that are true or false *in part*. Since this idea was at odds with the overall reasoning and knowledge representation framework that has been designed, we decided against the inclusion of this approach and focused on probabilistic methods.

Learning grouping rules

Grouping rules can be learned using a suitable learning technique. In this section we describe work in progress on learning rules for estimating a precedence mask using decision trees. A binaural precedence effect model is proposed to select the reliable time-frequency regions for localisation in the presence of reverberation. The model is derived from a series of binaural features that reflect the transient parts of the direct sound arriving from the parts that are polluted by sound reflections. Decision trees are then applied on the features to estimate a time-frequency mask, which can be used to selectively weight the output of a binaural localisation system.

Precedence mask. The precedence effect is a well-known binaural psychoacoustic effect involved in resolving competition for localisation between a direct sound and its reflections. Such an effect contributes largely to human listeners' ability to localise sound sources in a reverberant environment. In contrast, the performance of sound localisation in machine hearing systems is usually degraded by reverberations. While multi-conditional training can improve the robustness of a localisation system to reverberation, modelling the precedence effect could be a more direct way to bridge the gap between the human and machine sound localisation (Lindemann, 1986, Faller and Merimaa, 2004, Braasch and Blauert, 2011, Braasch *et al.*, 2003).

In this work, a precedence effect model is implemented as a series of decision trees in order to estimate a spectro-temporal mask, a *precedence mask*, that indicates time-frequency regions dominated by direct sound. Such a precedence mask can be employed by a sound localisation system to weight more on the direct-sound dominated regions, thus producing more reliable estimation of sound locations.

Learning rules for mask estimation using decision trees. We employed decision trees to learn rules for mask estimation and selection of localisation features. Fig. 3.6 shows a schematic diagram of the proposed system of binaural sound localisation. At the training stage, binaural features that reflect the activity of direct sound are extracted for the reverberant training speech, and the ideal precedence mask is calculated for the posterior probability of each azimuth as the reference labels. The relationship between the features and the reference labels is learned by decision trees in each auditory channel. In the testing stage, the precedence mask is generated from the decision trees based on the features, and the precedence effect is realised by weighting the contribution of the direct-sound-dominated time-frequency regions.

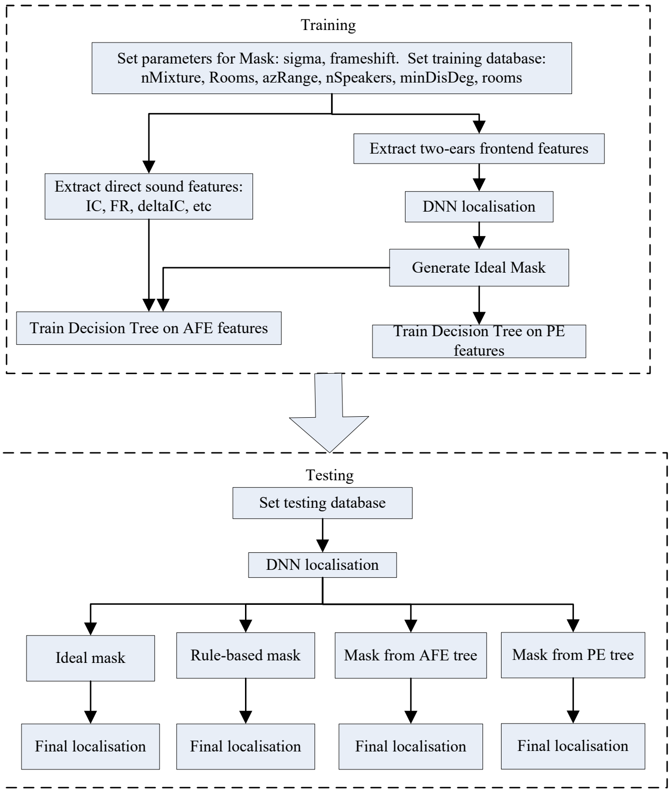

Training
Set parameters for Mask: sigma, frameshift. Set training database: nMixture, Rooms, azRange, nSpeakers, minDisDeg, rooms
Extract two-ears frontend features
Extract direct sound features: IC, FR, deltaIC, etc
DNN localisation
Generate Ideal Mask
Train Decision Tree on AFE features
Train Decision Tree on PE features
Testing
Set testing database
DNN localisation
Ideal mask
Rule-based mask
Mask from AFE tree
Mask from PE tree
Final localisation
Final localisation
Final localisation
Final localisation

Figure 3.6: Schematic diagram of the proposed system, showing steps during training (top) and testing (bottom). During training, the ideal mask is generated based on the posterior probability of each azimuth, which is the output of the DNN. Then a decision tree is trained for each auditory channel to map between the features and the reliability of the time-frequency region, which direct sound mask. In the testing stage, the mask is derived from the feature based on the decision trees.

The precedence mask can be derived from the binaural cues of the input signal according to some rules and models. Three methods of generating the precedence mask have been tested in this work. The first is to generate the mask directly based on some features that reflect the activity of direct sound. The second and third train decision trees for particular feature sets based on training sets in reverberant environments. In the testing, the mask is generated from the decision trees based on the auditory cues of the input signal. In the second method, the features of auditory front end are used directly. In the third method, the features were pre-processed as “precedence features” to enhance the sensitivity to direct sound, and the training and testing were both based on those precedence features.

Precedence features. In (Faller and Merimaa, 2004), the interaural coherence (IC) was used as an indicator for time instants with reliable binaural cues. However, IC only reflects the coherence of two signals, and it may also be high when reverberation is included in the observed segment. Moreover, it is still a problem for a machine listening system to select an appropriate threshold for each auditory channel and environment, while the human auditory system can quickly adapt.

The Franssen effect experiments (Hartmann and Rakerd, 1989, Yost *et al.*, 1997) show that the human auditory system may fail to localise sounds with a constant amplitude and spectrum in reverberant sound fields. As long as no new sound source arrives, the direction of the last localised sound source remains as the perceived direction. This is consistent with the precedence effect if we consider two sounds as the lead and lag signal, and it shows that the lag signal can be neglected during localisation until a new onset appears. It is therefore reasonable to hypothesise that the auditory system separates tones with the same frequency based on their onsets, which tends to merge sounds that do not have a strong onset with the localised source immediately before them.

We propose precedence features based on this explanation of the precedence effect for localisation. The onset of the direct sound causes a peak in the interaural cross-correlation function, and this is followed by stabilised cross-correlation output with the peaks drifting slowly due to arrival of reflections. If the analysis window includes only the direct sound, then the height of the peak in the IC increases abruptly after more direct sound is included in the window. When the analysis window is extended to include more reflections, the IC values start to decrease or fluctuate and the lags begin to decline and drift. This gives us a good indication of the boundary between direct sounds and reflections in each auditory filter. IC values of increasing window lengths are therefore employed as precedence features.

Preliminary results. We trained regression trees to select useful precedence features and estimate a precedence mask. In our preliminary experiments, we used only 10 utterances for training the decision trees. The DNN-based localisation system (Section 3.4.1) was used as the baseline system. The localisation accuracies of the system when employing the ideal and estimated masks are listed in Table 3.1. Systems were tested on mixtures of 1, 2 or 3 speakers in 5 acoustic environments (anechoic, and four rooms from the Surrey database which varied in reverberation time). Bold figures indicate whether the best performance in each condition (for the *no mask* and *decision tree mask* systems only).

The DNN baseline is already a robust localisation system and included multiconditional training to combat the adverse effect of reverberation. When the ideal mask was employed, the improvement was mainly found in the most challenging multi-speaker conditions. The estimated precedence mask shows only a limited benefit, with improved localisation

Table 3.1: Localisation accuracy (%) for three systems in which no time-frequency masking is used (*no mask*), time-frequency regions are selected according to *a priori* knowledge of the ideal mask (*ideal mask*) and a system in which the localisation mask is estimated from auditory features using a decision tree (*decision tree mask*).

	Speakers	Anechoic	Surrey A	Surrey B	Surrey C	Surrey D
No mask	1	**100.00**	99.78	**97.30**	**100.00**	**97.30**
	2	**100.00**	98.15	**94.09**	97.82	**94.89**
	3	99.90	**91.74**	**87.69**	**92.43**	**88.16**
Ideal mask	1	100.00	100.00	100.00	100.00	100.00
	2	100.00	100.00	98.89	100.00	99.89
	3	100.00	98.62	98.19	98.91	97.54
Decision tree mask	1	**100.00**	**100.00**	96.97	**100.00**	**97.30**
	2	**100.00**	**99.10**	93.46	**98.00**	93.97
	3	**100.00**	91.70	86.68	91.43	87.15

accuracy in some conditions and degraded performance in others. This could be largely due to the fact that the decision trees were trained using a small data set. However, the performance when using ideal masks shows this could be a promising approach to robust sound localisation using a precedence model.

Integration with engines for logic programming

A MATLAB interface to the Prolog logic programming language was developed to enable the assessment and evaluation of first order logic rule-bases within the Two!Ears blackboard framework. The interface is based on the SWI-Prolog environment[3], which is embedded into a wrapper function that can be called from within MATLAB. It allows for testing hypothesis on rule bases, formulated in first-order logic. The rule-bases have to be specified as Prolog programs in corresponding .pl-files, which can be handled by the interface. A logical variable is returned upon a hypothesis was tested, being either “True” is the hypothesis was accepted or “False” if a contradiction was found by the Prolog interpreter.

[3] http://www.swi-prolog.org/

3.3.5 Autonomous learning and active data selection

Active data selection on tone in noise experiments data

The motivation of active learning, or active data selection, is to reduce the cost of labeling in supervised learning by automatically avoiding samples that do not add to further training a classifier. These algorithms thus strive to only *query* for labels of samples that supposedly will be most informative about the underlying problem (see Settles (2012) for an overview of the field of active learning).

One of the problems of active learning is that for most schemes even to exhibit theoretical advantages over passive learning, tight constraints concerning data distributions and separability of data have to be fulfilled. In “Agnostic Active Learning Without Constraints” (AALWC) (Beygelzimer *et al.*, 2010), a more generally applicable probabilistic active learning algorithm with a promising approach is described, that produces theoretical advantages over passive learning even for non-separable problems and has an error bound roughly the same as a passive learner after n unlabelled examples in the worst case, and in the best case provides polynomial label complexity improvement.

This algorithm, with a few modifications necessary to be able to practically implement it, is available in the software tool “Vowpal Wabbit” (Langford *et al.*, 2007). Vowpal Wabbit (VW) basically implements an online gradient descent with logistic, hinge or Gaussian regression, and additionally applies the AALWC algorithm to this.

We tested this algorithm through VW on synthetic as well as experimental data. With the synthetic data, we tested sets in different dimensionalities and with different separabilities, the experimental data was taken from tone-in-noise (TiN) experiments as described in Schönfelder and Wichmann (2013). We were able to indeed produce great reductions of label complexity for the synthetic *separable* problems, in low as well as high dimensions. However, this advantage quickly vanished with increased overlap of classes in the data, and would also turn to disadvantages in the form of worse mean error compared to passive supervised learning. This was confirmed with the experimental data, where classes also overlapped: in Fig. 3.7, right panel, results for one batch of TiN data are shown, and in the left panel results for separable data of the same dimensionality are presented. It is clearly observable that in the separable case, the active learning algorithm shows its benefit, while in the non-separable real-life case, it shows disadvanatages over passive learning.

Since in general, there are not many separable problems with real-world data, and we did not suspect data in Two!Ears to be separable, we decided to not follow the active learning approach further.

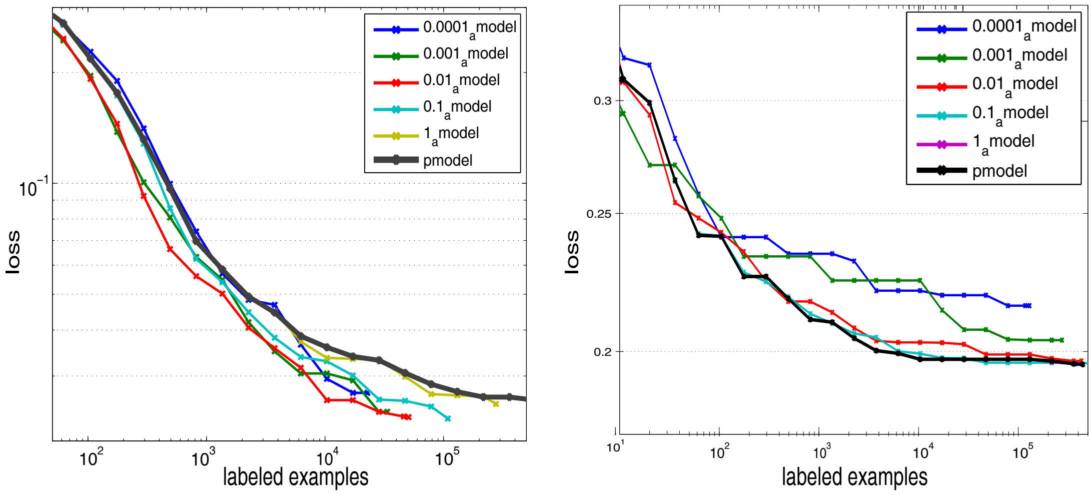


Figure 3.7: Learning curves of active data selection algorithm with different query probabilites on 46-dimensional data. "pmodel" denotes the passive data selection model, i.e. standard supervised with all data labeled. **Left panel:** Artificial, separable data. **Right panel:** Tone in noise experiment data.

3.4 Task 3.4: Semantic labelling and event-expert layer

3.4.1 Source locations

One of the main sound event attributes is its location. Human listeners have little difficulty in localising sounds under conditions where multiple sound sources and room reverberation are present. They are able to decode the complex acoustic mixture that arrives at each ear with apparent ease (Blauert, 1997). In contrast, sound localisation by machine systems is usually unreliable in the presence of interfering sources and reverberation.

We developed a novel machine-hearing system that exploits DNNs for robust localisation of multiple sound sources in such conditions (Ma *et al.*, 2015c). In contrast to many previous binaural hearing systems, the approach proposed here is not restricted to localisation of sound sources in the frontal hemifield; we consider source positions in the full 360 ° azimuth range around the head. In this unconstrained case, the location of a sound cannot be uniquely determined by ITD and ILD; due to the similarity of these cues in the frontal and rear hemifields, front-back confusions occur (Wightman and Kistler, 1999).

System overview

Fig. 3.8 shows a schematic diagram of the system for sound localisation in the full 360 ° azimuth range. During training, clean speech signals were spatialised using HRIRs, and diffuse noise was added before being processed by a binaural model for feature extraction. The noisy binaural features were used to train DNNs to learn the relationship between binaural cues and sound azimuths. During testing, sound mixtures consisting of several talkers are rendered in a virtual acoustic environment, in which a binaural receiver is moved in order to simulate the head rotation of a human listener. Output from the DNN is combined with a head movement strategy to robustly localise multiple talkers in reverberant environments.

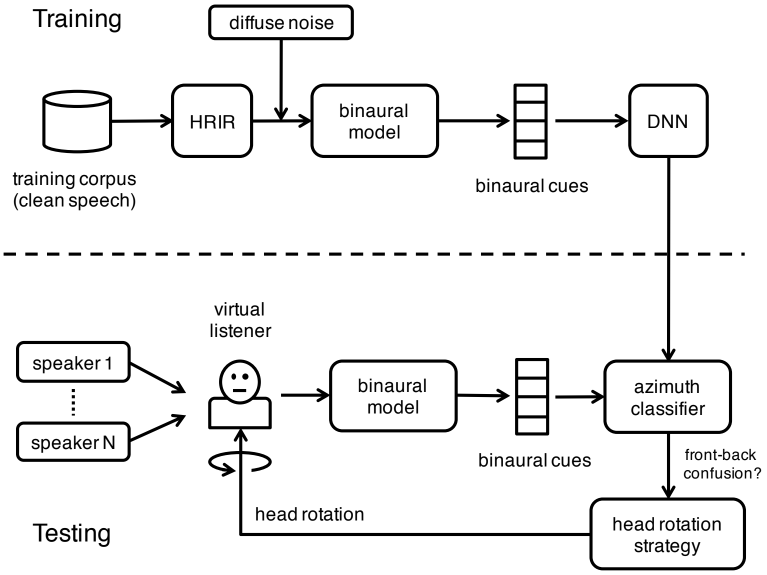


Figure 3.8: Schematic diagram of the sound localisation system, showing steps during training (top) and testing (bottom). During testing, sound mixtures consisting of several talkers are rendered in a virtual acoustic environment, in which a binaural receiver is moved in order to simulate the head rotation of a human listener.

Binaural feature extraction

An auditory front-end (WP2) was employed to analyse binaural ear signals with a bank of 32 overlapping Gammatone filters, with centre frequencies uniformly spaced on the equivalent rectangular bandwidth (ERB) scale between 80 Hz and 8 kHz (Wang and Brown, 2006). Inner-hair-cell processing was approximated by half-wave rectification. Afterwards, a cross-correlation function (CCF) between the right and left ears was computed independently for each frequency band using overlapping frames of 20 ms duration with a shift of 10 ms. The CCF was further normalised by an auto-correlation function at lag zero as described in (May *et al.*, 2011) and evaluated for time lags in the range of ±1.1 ms.

Two binaural features, ITDs and ILDs, are typically used in binaural localisation systems (Blauert, 1997). ITD is estimated as the lag corresponding to the maximum in the cross-correlation function. ILD corresponds to the energy ratio between the left and right ears within the analysis window, expressed in dB. In this study, instead of estimating the ITDs, the entire cross-correlation function was used as localisation features. This approach was motivated by two observations. First, computation of the ITDs involves a peak-picking operation which may not be robust in the presence of noise and reverberation. Second, there are systematic changes in the cross-correlation function with source azimuth (in particular, changes in the main peak with respect to its side peaks). Even in multi-source scenarios, these can be exploited by a suitable classifier.

For signals sampled at a rate of 16 kHz, the CCF with a lag range of ±1 ms produced a 33-dimensional binaural feature space for each frequency band. This was supplemented by the ILD, forming a final 34-dimensional (34D) feature vector.

DNN localisation

DNNs were used to map the 34D binaural feature set to corresponding azimuth angles. A separate DNN was trained for each of the 32 frequency bands. Employing frequency-dependent DNNs was found to be effective for localising simultaneous sound sources. Although simultaneous sources overlap in time, within a local time frame each frequency band is mostly dominated by a single source, according to the principle of 'exclusive allocation' (Bregman, 1990a). Hence, this allows training using single-source data and removes the need to include multi-source data for training.

The DNN consists of an input layer, two hidden layers, and an output layer. The input layer contained 34 nodes and each node was assumed to be a Gaussian random variable with zero mean and unit variance. The 34D binaural feature inputs for each frequency band were Gaussian normalised, and white Gaussian noise (variance 0.4) was added to avoid overfitting, before being used as input to the DNN. The hidden layers had sigmoid activation functions, and each layer contained 128 hidden nodes. The number of hidden nodes was heuristically selected – more hidden nodes increased the computation time but did not improve localisation accuracy. The output layer contained 72 nodes corresponding to the 72 azimuth angles in the full 360 ° azimuth range, with a 5 ° step. The 'softmax' activation function was applied at the output layer. The same DNN architecture was used for all frequency bands and we did not optimise it for individual frequencies.

Given the observed feature set $\vec{x}_{t,f}$ at time frame t and frequency band f, the 72 'softmax' output values from the DNN for frequency band f were considered as posterior probabilities $\mathcal{P}(k|\vec{x}_{t,f})$, where k is the azimuth angle and $\sum_k \mathcal{P}(k|\vec{x}_{t,f}) = 1$. The posteriors were then integrated across frequency to yield the probability of azimuth k, given features of the entire frequency range at time t

$$\mathcal{P}(k|\vec{x}_t) = \frac{P(k)\prod_f \mathcal{P}(k|\vec{x}_{t,f})}{\sum_k P(k)\prod_f \mathcal{P}(k|\vec{x}_{t,f})}, \tag{3.20}$$

where $P(k)$ is the prior probability of each azimuth k. Assuming no prior knowledge of source positions and equal probabilities for all source directions, Eq. 3.20 becomes

$$\mathcal{P}(k|\vec{x}_t) = \frac{\prod_f \mathcal{P}(k|\vec{x}_{t,f})}{\sum_k \prod_f \mathcal{P}(k|\vec{x}_{t,f})}. \tag{3.21}$$

Sound localisation was performed for a signal chunk consisting of T time frames. Therefore the frame posteriors were further averaged across time to produce a posterior distribution $\mathcal{P}(k)$ of sound source activity

$$\mathcal{P}(k) = \frac{1}{T}\sum_{t}^{t+T-1} \mathcal{P}(k|\vec{x}_t). \tag{3.22}$$

The target location was given by the azimuth k that maximises $\mathcal{P}(k)$

$$\hat{k} = \arg\max_k \mathcal{P}(k) \tag{3.23}$$

Localisation with head movements

In order to reduce the number of front-back confusions, the DNN localisation model employs a hypothesis-driven feedback stage that triggers a head movement if the source location cannot be unambiguously estimated (Ma *et al.*, 2015f, May *et al.*, 2015b). A signal chunk is used to compute an initial posterior distribution of the source azimuth using the trained DNNs. In an ideal situation, the local peaks in the posterior distribution correspond to the azimuth of true sources. However, due to early reflections and the similarity of binaural features in the front and rear hemifields, *phantom sources* may also be apparent as peaks in the azimuth posterior distribution. In this case, a random head movement within the range of $[-30°, 30°]$ is triggered to solve the localisation confusion. Other possible strategies for head movement are discussed in (Ma *et al.*, 2015f).

A second posterior distribution is computed for the signal chunk after the completion of the head movement. Assuming that sources are stationary before and after the head movement, if a peak in the first posterior distribution corresponds to a true source position, then it will appear in the second posterior distribution and will be shifted by an amount corresponding to the angle of head rotation. On the other hand, if a peak is due to a phantom source, it will not occur in the second posterior distribution. By exploiting this relationship, potential phantom source peaks are identified and eliminated from both posterior distributions. After the phantom sources have been removed, the two posterior distributions were averaged to further emphasise the local peaks corresponding to true sources. The most prominent peaks in the averaged posterior distribution were assumed to correspond to active source positions. Here the number of active sources was assumed to be known *a priori* (see Sect. 3.5.2).

Multi-conditional training

Previous studies have shown that multi-conditional training (MCT) features can increase the robustness of localisation systems in reverberant multi-source conditions (May *et al.*, 2015b, Woodruff and Wang, 2012, May *et al.*, 2013). Here, localisation models were trained on binaural MCT features created by mixing a target signal at a specified azimuth with diffuse noise at various signal-to-noise ratios (SNRs). The diffuse noise consisted of 72 uncorrelated, white Gaussian noise sources that were placed across the full 360 deg azimuth range in steps of 5 °. Both the target signals and the diffuse noise were spatialised by using the same anechoic HRIR (Wierstorf *et al.*, 2011). More details about the training HRIR can be found in WP1 deliverables.

The training material consisted of speech sentences from the TIMIT database (Garofolo *et al.*, 1993b). A set of 30 sentences was randomly selected for each of the 72 azimuth locations. For each spatialised training sentence, the anechoic signal was corrupted with diffuse noise at three SNRs (20, 10 and 0 dB SNR). Only those features for which the *a priori* SNR between the target and the diffuse noise exceeded −5 dB were used for training. This negative SNR criterion ensured that the multi-modal clusters in the binaural feature space at higher frequencies, which are caused by periodic ambiguities in the cross-correlation analysis, were properly captured.

3.4.2 Source detection and identification

A central attribute of sound sources or auditory objects (depending on level of consideration) is the *type* (also “class” or “identity”, we use all three terms throughout this work), expressing the “contents” of the event or object on an abstract and semantic level. Determining the type of sound events occurring is a central matter of CASA, which we approached through building *detection* or *identification* models: models that decide whether the ear-signals stream contain a particular sound event at the moment, or not. Given models of all available types, the type of sound events can be identified; detection and identification are in this sense directly related. The difficulty of identifying objects in the ear-signals stream in realistic environments is that they very often occur in situations with other superimposed sounds under very different SNRs.

In the following, we describe our system for detecting auditory events with emphasis on detection in realistic scenarios with concurrent sound sources, using statistical learning to find the relationship between auditory features of the ear-signals stream and particular sound events; and give a thorough evaluation in Sect. 4.3. All steps described below were conducted using the AMLTTP (Sec. 3.3.1) with the methods described in Sec. 3.3.3.

Generation of auditory scenes

Since the detection models were to be built using statistical learning, which relies on providing many training examples, and we furthermore strive to attain models working under a large variety of acoustic conditions, we generated training data with the Two!Ears binaural simulator (embedded in the AMLTTP, see Sec.3.3.1) in many scenes with different acoustic settings.

The input to this scene generation were the sounds of the NIGENS database, which we collected with this aim. The database provides audio files of fourteen different event classes: running engine, crash, footsteps, piano, barking dog, phone, knocking, burning fire, crying baby, alarm, female speech, male speech, screams, and a "general" class. Most of those sounds were attained from a commercial stock sound provider (`stockmusic.com`), the speech classes from the GRiD and TIMIT corpora (Cooke *et al.*, 2006a, Garofolo *et al.*, 1993a), and the scream sounds plus a few general sounds from `freesound.org`. The NIGENS database contains 1050 WAV files, 305 of which are in the "general" class – an "anything else but the thirteen provided regular types" class with sounds chosen to exhibit as much variety as possible, e.g. including nature sounds such as wind, rain, or animals, sounds from man-made environments such as honks, doors, or guns, as well as human sounds like coughs. All audio files contain sound events in isolation, i.e. without superposition of ambient sounds or other sources. All sound files were manually annotated for onsets and offsets of sound events.

The binaural auditory scenes were rendered by the binaural simulator in two basic modes: (1) convolving with anechoic HRIRs measured with a KEMAR head, and (2) convolving with BRIRs recorded inside the ADREAM apartment (see Deliverable D1.3), thus including reverberation, at four head × four speaker positions. Figure 4.36 provides a schematic layout of the apartment with the different positions.

We generated the following general types of binaural auditory scenes:

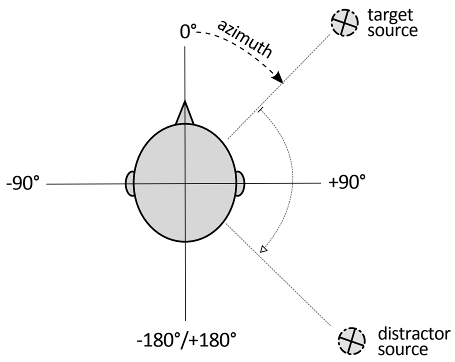


Figure 3.9: Top view of our coordinate system. The head and two sources are shown. The azimuth is always given with respect to the nose of the head in clockwise direction. Target and distractor sources in this example are located at 45°/135°.

1. Anechoic scenes composed of a single point source emitting sounds from all classes at different azimuth angles – we refer to these scenes as "clean" scenes;

2. Ancechoic scenes containing two point sound sources emitting sounds simultaneously (see Fig. 3.9), one of them being fed by sounds from all classes, and the other one only from sounds of the general class;

3. Anechoic scenes containing an ambient (i.e. non-directional) source emitting sounds from all classes;

4. Ancechoic scenes containing two ambient sound sources emitting sounds simultaneously, one of them being fed by sounds from all classes, and the other one only from sounds of the general class;

5. ADREAM apartment scenes containing one to four point sound sources emitting sounds simultaneously, all of them being fed by sounds from all classes;

6. ADREAM apartment scenes containing one to four point sound sources emitting sounds simultaneously, all of them being fed by sounds from all classes, plus an ambient white noise source.

Binaural simulation was conducted for each source separately to allow control over the energy ratio of the sources *at ear-signals level* – this is justified because sound mixing is linear. Therefore, the resulting ear-signals streams for each source were mixed at different ratios of squared amplitudes averaged over both binaural channels and time. The time-averaging only included times of sound activity, such that the ratio does not take periods of silence into account which is a frequent occurrence in general environmental sounds.

Hence, to be precise, what we fixed are the ratios of the energies between the sources when they emit sound simultaneously. We later refer to these ratios as SNRs even when there is no classic "noise" involved.

For the anechoic scenes, the distance of the point sources to the head was fixed at three meters, since the HRIR was recorded at that distance, and we assumed that distance would not be a parameter influencing detection other than through the change of SNR this implies when sources' distances are changed.

Feature extraction

The generated binaural signals were then preprocessed by the AFE of the Two!Ears system to obtain, depending on the experiment, several of the following representations:

ratemaps Auditory spectrograms that resemble auditory nerve firing rates for each 20 ms time frame and individual gammatone frequency channel, computed by smoothing the corresponding inner hair cell signal representation with a leaky integrator (Glasberg and Moore, 1990, Patterson and Holdsworth, 1996, Cooke *et al.*, 2001, May *et al.*, 2012)

spectral features 14 different statistics such as flatness, kurtosis, etc., which summarize the spectral content of the ratemap for each time frame (Peeters *et al.*, 2011, Tzanetakis and Cook, 2002, Jensen and Andersen, 2003, Misra *et al.*, 2004, Lerch, 2012)

onset strengths measured in decibels for each time frame and frequency channel, calculated by the frame-based increase in energy of the ratemap representation (Klapuri, 1999)

amplitude modulation spectrogram Each frequency channel of the inner hair cell representation is analyzed by a bank of logarithmically-scaled modulation filters (Moritz *et al.*, 2011, May and Dau, 2014b). Each time frame was assigned x frequency channels × y modulation filters values

gabor features Spectro-temporal orientation sensitive edge detectors operating on the ratemap representation (Schädler *et al.*, 2012).

Since our detection models operate on blocks of data (in contrast to the continuous stream), these auditory feature representation streams were subsequently decomposed into overlapping blocks, by default of 500 ms length. From the representations of each block we constructed two general different types of feature vectors that were used as input for the sound event detection models:

- *mean-channel features*: AFE representations were averaged over the left and right channels; in addition, the first two deltas (discrete time derivatives) were calculated. Features were then constructed by computing L-statistics[4] (L-mean, L-scale, L-skewness, L-kurtosis) of representations and deltas over time.
- *two-channel features*: Instead of averaging the signals over the two channels, features were constructed for each channel separately. Applying the same procedure as for the mean-channel features resulted in double the feature dimensions.

To create the learning targets and testing ground truth for our models, we used our annotations of sound events' onset and offset times to automatically label all blocks for each sound type (except "general") according to whether this type (the "target" class) was present (+1) or absent (−1) within the block. Blocks for which the target class occupied less than 75% but more than 0%, were excluded from training and testing, as both labels would seem right and assigning a hard "truth" would manipulate both training and evaluation in unclear ways.

Sound event detector training

For each of our sound classes (except "general") we trained a binary classifier in a one-vs-all scheme: each classification model decides whether a given auditory signal block contains a sound event of its target type. For classification, we used Lasso (Tibshirani, 1996), a linear logistic classification model with an L_1 penalty for the regression coefficients. This regularization method leads to a sparse selection of the most informative features for the trained model, disregarding uninformative or unnecessary (because of correlation with others) features. Additional details about the learning method can be found in section 3.3.3.

Since we wanted control over the SNRs and in order to be able to evaluate reasonably, for each detector (that is for each type) we restricted sounds of the target class to only be emitted from one source per scene. This way, the other sources served as "distractor" sources and the SNR was set with the one source that also emits target sounds as reference. Note that the source allowed to emit target sounds also emitted all other classes' sounds (including "general") to provide a large base of counter-examples for the detector model training. In the anechoic scenes, only general class sounds were emitted from the distractor sources. We loosened this constraint in the ADREAM apartment scenes.

[4] L-statistics are given by L-moments, a sequence of statistics used to summarize the shape of a probability distribution Hosking (1990). L-statistics are shown to be more robust than conventional statistics, in particular with respect to the higher moments and when a small amount of data is available (David and Nagaraja, 2003, Ch. 9).

We considered and evaluated two types of training schemes and resulting models:

single-conditional (sc) training : models were trained on data taken from one type and configuration of auditory scene only. These models were expected to excel under their specialized conditions, but degrade when deviating from them.

multi-conditional (mc) training : models were trained on data taken across multiple auditory scenes and different conditions. These models were assumed to feature increased robustness to differing acoustic conditions and varying situations with superimposed sources.

We analyze these schemes and models thoroughly with respect to robustness and dependence of conditions, both for the anechoic and ADREAM settings and, in Sect. 4.3, for the Two!Ears development system and in Sect. 4.7 for the Two!Ears deployment system.

Application in Two!Ears-system

Our sound event type identification models can be plugged into the IdentificationKS of the Two!Ears system. This knowledge source wraps the detection model, the block-based feature extraction and the block extraction, and provides an interface to the blackboard system: it gets access to the Two!Ears system's auditory streams, and puts the model's output probabilities and decisions as hypotheses onto the blackboard. The specifications of this knowledge source is presented in Deliverable D6.1.3.

3.4.3 Speaker recognition

In the previous section we presented work on classification of general sounds. In this section we investigate a specific sound classification problem: speaker recognition.

GMM-UBM

Many widely used speaker identification methods are based on GMMs, which model feature vectors from each speaker with a weighted sum of Gaussian distributions. The basic approach is realised by the log-likelihood ratio test from signal detection theory. GMMs are used for both target and background models. The target model is trained using speech signals from the target speaker. The background model, often referred to as the universal background model (UBM), is trained using speech signals from as many speakers as possible. Given a test signal $\vec{x}$, the log-likelihood ratio between the target model and the UBM is computed as

$$\mathit{\Lambda} = \log \frac{p(\vec{x}|\lambda_s)}{p(\vec{x}|\lambda_{ubm})} \tag{3.24}$$

where $p(\vec{x}|\lambda)$ is the likelihood of $\vec{x}$ given model λ. The log-likelihood ratio is compared to a pre-defined threshold for acceptance of the target speaker.

$$p(\vec{x}|\lambda_s) = \sum_{i=1}^{M} \omega_i f(\vec{x}|\mu_i, \mathit{\Sigma}_i), \tag{3.25}$$

where $\vec{x}$ is a D-dimensional continuous-valued feature vector, $\omega_i, i = 1, \ldots, M$ are the mixture weights, and $f(\vec{x}|\mu_i, \mathit{\Sigma}_i)$ are the i^{th} component Gaussian densities with mean vector μ_i and covariance matrix $\mathit{\Sigma}_i$.

In order to use a GMM, we need to compute the likelihood of a sequence of features given a GMM and estimate the parameters of a GMM given a set of feature vectors. If independence between feature vectors in a sequence is assumed, then the likelihood can be computed as

$$p(\vec{x}_1, \ldots, \vec{x}_N|\lambda_s) = \prod_{n=1}^{N} p(\vec{x}_n|\lambda_s) \tag{3.26}$$

The parameters of UBM are estimated by maximising the likelihood of a set of training feature vectors using the EM algorithm. The target speaker model, however, is often trained by adapting from the universal background model, particularly when there is limited amount of training speech from the target speaker. We employ a popular adaptation method, relevance maximum *a posteriori* (MAP) adaptation, which is a linear interpolation of all mixture components of the UBM that increase likelihood of speech from the target speaker. It is found empirically better to adapt both mean and variance matrices from the UBM.

i-vectors

One of the problems of relevance MAP is that it adapts not only to speaker-specific information but also to channel and other undesired factors. This problem is addressed by the joint factor analysis (JFA) model proposed for the GMM framework (Kenny *et al.*, 2008). JFA has become the basis for the state-of-the-art in speaker identification, and has the following two properties. First, a speech recording of a variable length is represented by a fixed-length supervector. Secondly, and more importantly, such a supervector representation allows explicit modelling of the undesired variability in the speech signal. The speaker supervector can be decomposed into speaker-dependent factors and channel-dependent factors. The values of the speaker factors are assumed to be the same for all recordings of the speaker but the channel factors are assumed to vary from one recording to another. The supervector is constructed by concatenating mean vectors from each GMM component. As a result, the JFA model has to estimate hyper-parameters with a huge dimensionality. Therefore principle component analysis (PCA) is typically applied with JFA.

Dehak *et al.* (2011) showed that the channel factors in the JFA model also contain some speaker information. They proposed the i-vector model based on JFA, which makes no distinction between speaker effects and channel effects in the GMM supervector space. Instead, a total variability space that contains both speaker and channel variabilities is defined. The hyperparameters of the i-vector model can be extracted by using the EM algorithm and have a much lower dimensionality than JFA. Channel compensation can be further applied in the i-vector space using probabilistic linear discriminant analysis (PDLA) (Prince and Elder, 2007).

Given a test speech signal, the test i-vector w is extracted and compared to speaker model i-vectors w_s. Let H_1 indicate the hypothesis that the two i-vectors are from the same speaker, and H_0 be the hypothesis that the two i-vectors are from different speakers. The verification score is computed as the log-likelihood ratio of the same versus different speaker models hypotheses:

$$\varLambda = \log \frac{p(w, w_s|H_1)}{p(w|H_0)p(w_s|H_0)} \tag{3.27}$$

Similar to the GMM-UBM approach, the log-likelihood ratio is compared to a pre-defined threshold to make a decision on the target speaker.

3.4.4 Joint source type and source location determination

In Sect. 3.4.2 above, we described our system for identification of sound events. This system operates directly on the ear-signals stream, hence often on mixtures of superimposed sound sources and accordingly also detects sources concurrently. Ultimately, the goal however is to form auditory *objects*, which means compiling attributes like type and location of sources that belong together in a coherent way. This assignment of different attributes determined directly from the ear-signals' mixture is challenging. In order to solve this so-called "binding"-problem (binding of attributes to form coherent objects), we developed two systems following two approaches: (1) an extension of our identification system to operate on streams *segregated* from the ear-signals stream by through masking time-frequency space according to estimated source locations, and (2) a new system that tries to *jointly* estimate type and location of objects from the ear-signals stream.

Auditory object identification on streams segregated by location

In this section we describe system (1) introduced above; the extension of our identification system to operate on streams segregated from the ear-signals, where ideally each stream can be assigned to one auditory object and contains as much information about the generating source (and as little information about other sources) as possible.

In principle, the detection models developed in Sec. 3.4.2 did not have to be modified in their general functionality, since they still worked on the same kind of data and still should produce the same kind of data – namely the type of sound events occurring, only that in this approach they supposedly would not work on streams of superimposed and overlapping but isolated events. Hence everything explained in the section "Source detection and identification" is valid for this approach too, supplemented by the methods described in "Classification of segregated auditory object streams" of Sec. 3.3.3.

The main difference is the inserted step of stream segregation into the feature generation – inserted after the AFE processing which produces streams of different auditory representations and before the construction of feature vectors from the blocked auditory representations. The segregation is accomplished using the method described in Sec. 3.2.1, which produces a set of so-called probabilistic time-frequency soft-masks, with the number of masks corresponding to the number of sources in the acoustic scene. These masks in our extended system thus were applied to all auditory representations that can be masked in the time-frequency domain, namely ratemaps, onset strength maps, and amplitude modulation spectrograms, and produced one set of these representations per source. Representations that could not be masked like the spectral features were nevertheless used, but hence still represented the original, mixed stream.

This segregation itself step relies on other information: the number of active sources and the location (azimuth) of those sources; both need to be estimated themselves first. We used the azimuth distribution estimation model described in Sec. 3.4.1 and the number of sources estimation model described in Sec. 3.5.2 for this.

Hence, the whole system for object identification on segregated streams consists of the azimuth distribution model, full-stream detection models, number of sources estimation model, segregation model and finally segregated-stream detection models, operating in a connected manner. Figure 3.10 shows a schematic of this framework.

Joint classification and localization of auditory objects

In this section we describe system (2) introduced above. The joint classification and localization task is accomplished using the auditory scenes generated using the sounds and procedures described in Sect. 3.4.2. For the localization part, the 360° azimuth plane

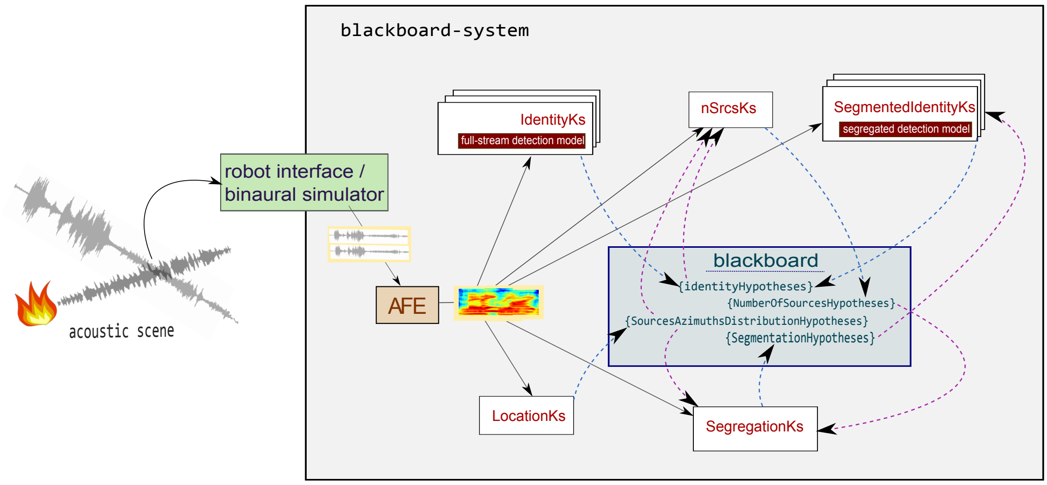

blackboard-system
IdentityKs
full-stream detection model
nSrcsKs
SegmentedIdentityKs
segregated detection model
robot interface / binaural simulator
acoustic scene
AFE
blackboard
{identityHypotheses}
{NumberOfSourcesHypotheses}
{SourcesAzimuthsDistributionHypotheses}
{SegmentationHypotheses}
LocationKs
SegregationKs

Figure 3.10: Schematic of the system for object identification on segregated streams, showing the interaction of the different involved models.

is discretized into 5° azimuth bins, resulting in a total of 72 azimuth bins. The large variation in the durations of the sound files used causes an imbalance between sound type occurrences. Before training, the class occurrences are re-sampled to reach a more uniform class occurrence distribution for reducing this imbalance [5]. Frequently occurring classes are sub-sampled, while rarely occurring instances are oversampled. The general class is handled similarly. Networks are trained on two different types of datasets. The first dataset is composed of clean anechoic sounds. A source is activated to play sounds from the set of 11 classes alarm, baby, crash, dog, engine, female speech, fire, foot steps, knock, phone, piano and an additional general class. The source is placed once in each of the 72 azimuth bins. This mitigates the risk of bias towards any particular locations. The balancing procedure results in a clean sounds training set containing approximately 720,000 blocks of positive instances for each class. The second dataset is composed of mixtures with up to four sources active simultaneously using ADREAM apartment scenes as described in Sect. 3.4.2. The scenes are identical to the ones used in the detection models developed in Sect. 3.4.2. However, samples containing multiple instances of the same sound type are kept and not excluded as done for the detection models.

A convolutional neural network is defined to learn a representation for the joint identification and localization of auditory objects using environmental sounds. Spatiotemporal AFE representations such as ratemaps, amplitude modulation spectrogram (AMS) and ILD between left and right ear representations are extracted. The spatiotemporal dimensions are preserved. It is left to the earlier layers in the network to learn a hierarchical transformation

[5] URL to external tools for balancing the data before training https://github.com/nigroup/nideep

for these representations without further manual intervention.

The convolutional architecture provides implicit regularization by reducing the number of weights that need to be optimized as well as spatial invariance (Lecun *et al.*, 1998). Rectified Linear Units (ReLU) are used as the non-linear transfer function between successive layers. Max pooling is used for selecting the highest response within a local patch to propagate further to the next layer in the network. The filters, subsampling and non-linearity steps make out a convolutional stage as depicted in Fig. 3.11. A technique used for combating overfitting in networks with a large number of learnable weights is the 'dropout' technique. Dropout consists of randomly selecting neurons at various stages in the network and setting their response to zero. This perturbation eliminates these neurons' involvement in the forward as well as backward passes of the backpropagation algorithm and their weights do not get updated during that iteration of the learning algorithm. Dropout has been shown to have a strong regularization impact on neural networks and effectively reduces overfitting (Srivastava *et al.*, 2014). As neurons randomly drop out in different iterations, other neurons from the same layer are forced to contribute to the minimization of the loss function. A weight that has previously been a dominant factor in reducing the error function, will not necessarily be available each iteration. Therefore the network is constantly forced to recruit other neurons to compensate for the neurons that drop out. It has also been interpreted as implicitly training an ensemble of networks, where each new randomly selected routing represents a new network and that over time, the network contains multiple different routings where all are capable of minimizing the loss function (Srivastava *et al.*, 2014). During testing and deployment of the network, all neurons contribute to the network's response.

A schematic of the network architecture is shown in Fig. 3.12. Designing the networks involves the following aspects:

- Feature selection. Varying the AFE representation input to the network. Whether ratemaps only, AMS features only, both or combining them with ILD. In all cases, the representations from the binaural signals are preserved. Keeping left and right representations separate may not be critical for the identification task. However, combining them would eliminate the network's ability in performing the localization task.

- Network depth. Setting the number of layers in a neural network is a design decision that comes with developing any network architecture. Our convolutional neural network is made out of a number of convolutional layers as well as deeper fully connected layers towards the two-dimensional output representing both identity and location of sound types. For the variants that involve different AFE representations, we decide at which stage they are fused together.

The output or decision layers of a network produces a two-dimensional $K * (A + 1)$ response,

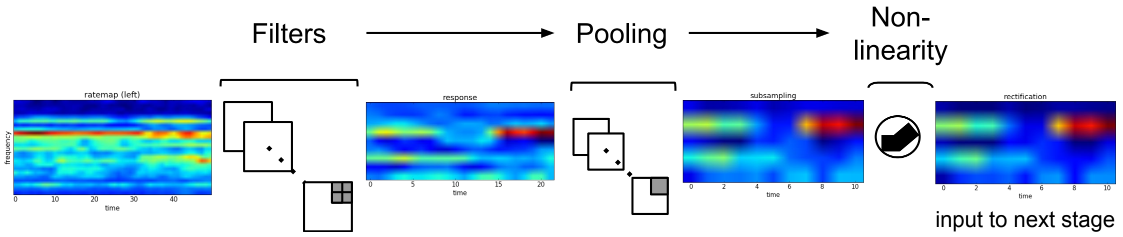


Figure 3.11: A convolutional stage. Input features are transformed by learnable filters to produce more abstract feature maps. Pooling spatially subsamples the feature maps. A non-linearity such as ReLU or sigmoidal function is applied to each element in the feature maps and the response is propagated downstream in the network which can be another convolutional stage or a fully connected layer. The order of pooling and the non-linearity may be changed or even omitted altogether.

where K is the number of sound types and A is the number of azimuth bins. Discretizing the 360° azimuth plane into 5° bins yields 72 azimuth bins. Each output neuron indicates whether a particular class is located at a specific azimuth bin. An additional node is added for each sound type for the network to decide if a particular sound type is absent from the scene. This is referred to as the void bin. Each element of this response matrix is defined as a sigmoidal neuron that acts as a one-vs-all classifier. Therefore, localization is articulated as a classification task rather than a regression task. This is a simple solution to enable the network to respond to multiple sources being present in the auditory scene. This accounts for the following cases in the multi-conditional training data:

- No sources present;
- A single source present;
- Multiple sources of different types present at different azimuth locations;
- Multiple sources of different types present at the same azimuth location;
- Multiple sources of the same type present at different azimuth locations.

The void bin is used to evaluate the network's source identification ability separately from the localization task. Cross entropy is used as the loss function for training the output layer to collectively perform a many-vs-many classification. Mini-batch gradient descent is used for updating the weights of the network. Figure 3.13 illustrates the two-dimensional ground truth.

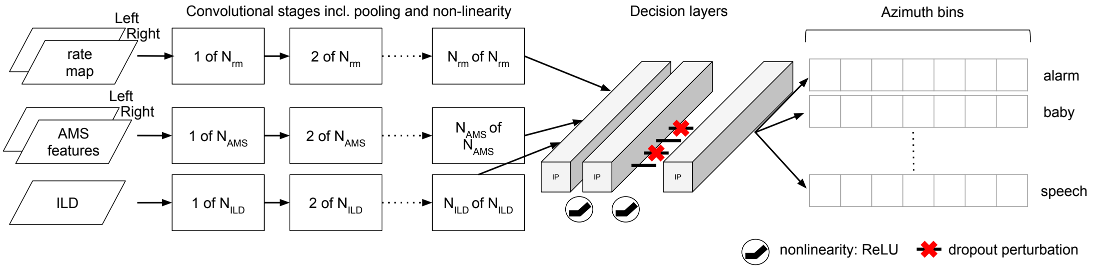


Figure 3.12: Schematic for the convolutional architecture for join classification and localization.

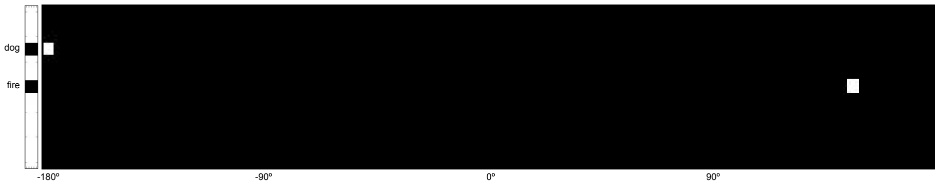


Figure 3.13: The two-dimensional ground truth matrix. In this instance the block is assigned a dog sound at 180° directly behind the head and a second fire source at −145○ to the right behind the head. The vector on the left represents the void bin for each sound type.

Smooth auditory object formation

The methods described in the sections above both operate in a block-based manner, producing output independently for each block, disregarding temporal or spatial context. Since we use blocks of 0.5 s length, output is indicating the very current situation, based on the information included in the latest block of signals only. This on one hand creates quite fluctuating estimates, and on the other hand does not make use of two assumptions emerging from the relation between the auditory object to be formed and the actual sound source represented by it: (a) objects do not change their type suddenly, and (b) objects do not teleport, but either remain at the same place or move about smoothly.

In order to include these two smoothness priors, we added a module integrating object information both over time and location to our systems. This module – called `IntegrateSegregatedIdentitiesKS`, see Deliverable D 6.1.3 – implements the following mechanisms, executed per block:

- Spatially imposing a Gaussian probability distribution onto current blocks' hypotheses from the segregated detection or joint identification and localization system. This Gaussian has a standard deviation of 10° and is multiplied by hypotheses' scores. Summing the resulting probabilities over hypotheses creates a distribution over the whole azimuth range for each class.

- Leaky integrating these classes' probability maps over time, such that the integrated probability map is a result of the distribution created from the current block and of the integrated distribution from the last time step.
- Leaky integrating the number of active sources estimate, aiming rather for a number of objects estimate (objects can be "inactive" temporarily, like when a crying baby takes a breath).
- Averaging integrated probability maps over classes. The resulting type-independent probability map is parsed for the N highest peaks (N being the estimated number of objects), determining the current locations of auditory objects. At these locations, class probabilities in the type-dependent integrated maps are scanned, setting the auditory objects types.

Figure 3.14 visualizes the output of the integrated system, correctly forming two auditory objects at about 65° (fire) and −110° (crying baby). This example scene was analyzed live by the deployment system in the ADREAM appartment.

3.4.5 Musical genre recognition

The assessment of acoustic scenes containing music often requires knowledge about the genre of the musical piece that is presented to the listener. For instance, the evaluation of the quality of a reproduction system is likely to depend on subjective preferences of human listeners based on preferred musical genres. By explicitly including a recognition system of genres in the analysis process, predictive models of quality assessment can benefit from this additional information.

Aucouturier and Pachet (2003) state that the term *genre* suffers from an intrinsic ambiguity, deeply rooted in our dualist view of the world. It can be seen either as

- An intentional concept, where the term genre lies in a linguistic category to be able to talk about music, or
- An extensional concept, where the choice of the genre is solved by analysis.

In both cases, a musical genre can be seen as an effort to order a set of songs, by putting songs that share some common features into classes, thus providing a method of finding similar songs more efficiently.

Within the Two!Ears framework, the task of musical genre recognition is tackled via the extensional approach or, more specifically, by applying methods of machine learning to derive predictions about the most likely musical genre given a set of acoustic features.

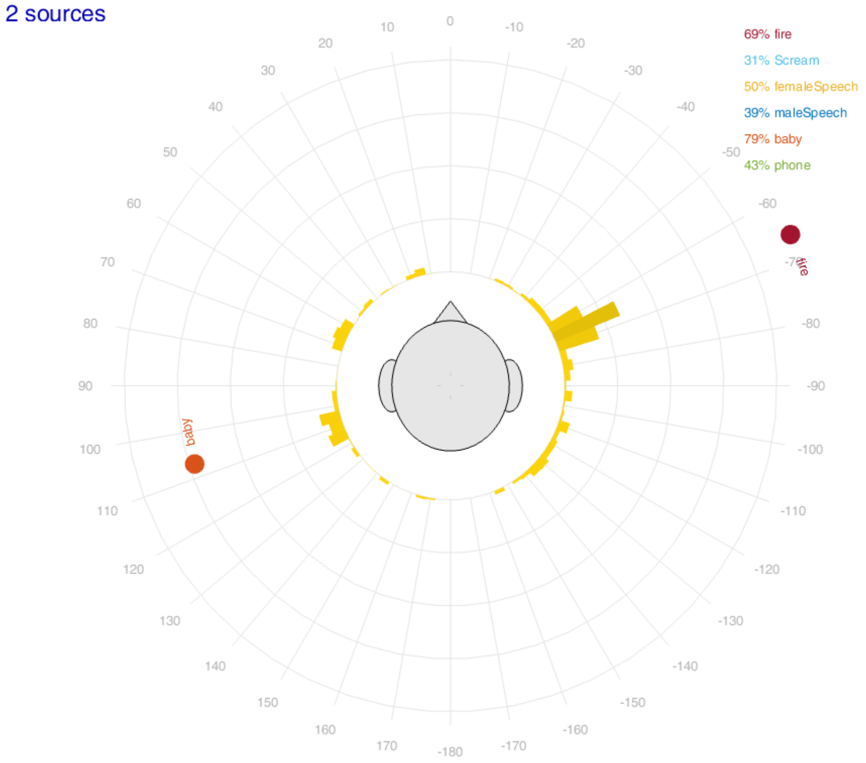


Figure 3.14: Visualization of the blackboard-system output, depicting object formation. Yellow bars around the head indicate source location estimation, scores with labels in the upper right corner state full-stream detection probabilities, and the two labeled balls illustrate formed auditory objects.

The recognition system is encapsulated into a knowledge source named GenreRecognitionKS available within the Two!Ears blackboard system. It takes ratemaps in the log-domain, averaged between both ears, as input features. Additionally, the first and second time derivatives are computed at each frame and concatenated to the raw ratemap features, yielding a $3L$-dimensional feature vector, where L is the number of filterbank channels used in the auditory front-end. Feature vectors are accumulated within blocks of 10 seconds length each. These blocks serve as input to the classification system used to recognize the most likely musical genre.

The classification system is based on linear discriminant analysis (LDA), performed frame-wise within each block of feature vectors. The final decision for the most likely class is derived via a majority-voting scheme over all classification results in one block.

3.5 Task 3.5: Assigning meaning

3.5.1 Sound localisation with top-down source knowledge

Human listeners must answer two questions in order to fully understand an acoustic scene; *what* the sound sources are, and *where* they are. In machine hearing, these two issues have been addressed in many studies via computational approaches for sound source separation, classification and localisation (Wang and Brown, 2006). However, machine systems for answering 'what' and 'where' questions are typically much less tightly-integrated than they appear to be in biological hearing. We addressed this issue by developing a machine system for binaural localisation that exploits top-down knowledge about the source spectral characteristics. Information from source models is used to selectively weight binaural localisation cues, which allows the system to combine top-down and bottom-up information flow within a single computational framework.

Overview

Fig. 3.15 shows the graphical model of the proposed localisation framework that combines top-down source knowledge and bottom-up localisation information. Top-down knowledge of the spectral profile of sources (speech source S_x and noise source S_n) is used to jointly explain the observed noisy ratemap features y_{tf} at each time-frequency bin via hidden source features x_{tf} (speech) and n_{tf} (noise). As we will see below in more details, the combined source models can estimate the probability of each time-frequency bin y_{tf} being dominated by the speech source, w_{tf}, which can be seen as a segmentation of the speech source. The segmentation is used as a weighting factor for selectively weighting the contribution of binaural cues from each time-frequency bin in order to better localise the attended speech source in the presence of the noise source.

Localisation model

The localisation model is the DNN system as described in Section 3.4.1. We introduce a weighting factor w_{tf} for selectively weighting the contribution of binaural cues from each time-frequency bin in order to localise the attended target source in the presence of competing sources. The azimuth posteriors are integrated across frequency to produce the probability of azimuth k given localisation features $\vec{o}_t$ of the entire frequency range,

$$P(k|\vec{o}_t) = \frac{\prod_f P(k|\vec{o}_{tf})^{w_{tf}}}{P(\vec{o}_t)}, \tag{3.28}$$

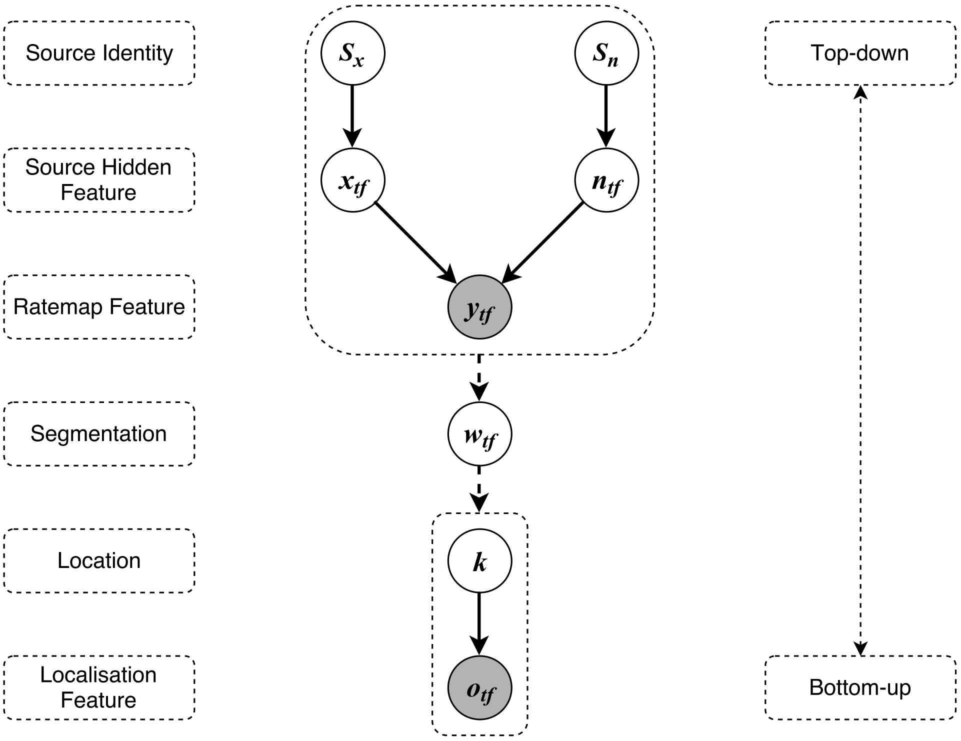


Figure 3.15: Graphical model of the proposed localisation framework that combines top-down and bottom-up information. Top-down knowledge of sources is used to selectively weight binaural localisation cues via an estimated segmentation w_{tf}.

where

$$P(\vec{o}_t) = \sum\nolimits_k \prod\nolimits_f P(k|\vec{o}_{tf})^{w_{tf}}. \tag{3.29}$$

Here w_{tf} is a factor between $[0, 1]$. When w_{tf} is 0 the time-frequency bin is excluded from localisation of the target source. Next we discuss in detail how top-down knowledge from source models can be used to jointly estimate the weighting factors.

Exploiting top-down source models

A set of parameters w_{tf} in Eq. (3.28) are employed to selectively weight binaural cues when performing source localisation. This allows cues that derive from a frequency channel dominated by the target source to be emphasised; or conversely, cues that derive from an interfering source can be penalised. Here, top-down information from source models is combined to jointly estimate these localisation weights.

Let λ_s represent the spectral characteristics of a sound source s in a set of source models

$s = 1, \ldots, \mathcal{S}$. The set of source models are employed to jointly explain the observed ratemap features. In particular, given the observed log-compressed ratemap feature vector $\vec{y}_t = [y_{t1}, \ldots, y_{t32}]^\top$ extracted at time frame t from the binaural signals, we want to determine whether each feature y_{tf} is dominated by the energy of the target source x_{tf} or corrupted by the combined energy of interfering sources n_{tf}. Under the *log-max* approximation (Varga and Moore, 1990) of the interaction function between two acoustic sources, i.e. $y_{tf} \approx \max(x_{tf}, n_{tf})$, the localisation weight w_{tf} can be defined as the probability of y_{tf} being dominated by x_{tf}

$$w_{tf} = P(x_{tf} = y_{tf}, n_{tf} \leq y_{tf} | \vec{y}_t, \lambda_x, \lambda_n), \tag{3.30}$$

where λ_x and λ_n are the models for the target and interfering sources, respectively. Here, the source models λ_s are represented as GMMs with diagonal covariance matrices. Then, λ_n is built by combining all the source models except that of the target source. Alternatively, the above model can be expressed as a large GMM by pooling the Gaussians from all the source models together and multiplying the mixture weights by the corresponding source prior probabilities, so that the resulting mixture weights sum up to one.

Using the expressions for the λ_x and λ_n models in Eq. (3.30), the final expression for the localisation weights w_{tf} is given by

$$w_{tf} = \sum_{m_x, m_n} \frac{\gamma_t^{(m_x, m_n)} p_x(y_{tf}|m_x) C_n(y_{tf}|m_n)}{p_x(y_{tf}|m_x) C_n(y_{tf}|m_n) + p_n(y_{tf}|m_n) C_x(y_{tf}|m_x)}, \tag{3.31}$$

where m_x and m_n are the indices for the mixture components in the target source and interfering sources models, respectively, p_x and p_n denote the Gaussian of the target and competing GMM models, and C_x and C_n are the corresponding Gaussian cumulative distribution functions.

3.5.2 Number of sources estimation

A part of computational auditory scene analysis is the estimation of the number of sources. This information provides a valuable prior for stochastic mixture models, beam formers, speaker tracking and localization systems. The availability of a reliable *a priori* estimate for the number of sources can lead to better calibration of learning problems and often reduces the run-time requirements by orders of magnitude.

The number of sources can be determined by counting the modes in a suitable feature distribution. Therefore, a feature is optimal if it yields a singular and disjunct representation per source. For this we explored several features sets:

1. Short-time Fourier transform (STFT) using the DUET method;

2. ITD and ILD features from the AFE;
3. Cross-correlation, on-set strengths and spectral features from the AFE;
4. An azimuth distribution provided by the localization model described in Sec. 3.4.1;
5. An identification probability per class provided by the identification models described in Sec. 3.4.2.

The features are described in detail below.

Degenerate Unmixing Estimation Technique (DUET)

The DUET method (Rickard, 2007) is a blind source separation technique, specifically addressing the degenerate case of exactly two receivers for an arbitrary amount of sources (e.g. as in binaural hearing). The DUET algorithm formulates a binaural mixture model for n sources in the time-frequency domain:

$$\begin{bmatrix} \widehat{x}_L(\tau,\omega) \\ \widehat{x}_R(\tau,\omega) \end{bmatrix} = \begin{bmatrix} 1 & .. & 1 \\ a_1 e^{-i\omega\delta_1} & .. & a_N e^{-i\omega\delta_N} \end{bmatrix} \begin{bmatrix} \widehat{s_1}(\tau,\omega) \\ .. \\ \widehat{s_N}(\tau,\omega) \end{bmatrix} \tag{3.32}$$

The limiting assumptions are that the mixture is stationary, anechoic, and the sources have disjunct time-frequency signatures (W-disjoint orthogonality), and are thus spatially separable. Under these assumptions, Eq. (3.32) is reduced to,

$$\begin{bmatrix} \widehat{x}_L(\tau,\omega) \\ \widehat{x}_R(\tau,\omega) \end{bmatrix} = \begin{bmatrix} 1 \\ a_j e^{-i\omega\delta_j} \end{bmatrix} \widehat{s}_j(\tau,\omega) \tag{3.33}$$

where exactly one source is responsible for the power in a time-frequency bin. This assumption holds only approximately for wide-band signals in echoic mixtures. The DUET method subsequently estimates the sources by clustering in the space spanned by the mixture parameters a and δ, then builds a mask on the input signal to reconstruct isolated sources and a lossy version of the input mixture.

The feature space relevant for the number of sources estimation is a two-dimensional smoothed weighted histogram over the mixture parameters a and δ for all time-frequency bins of the signal (Fig. 3.16, left). The time-frequency representation is obtained by a short-term Fourier transform (STFT). Time-frequency bins that cluster together in the space spanned by a and δ are attributed to the same source and appear as extrema; the problem is thus reduced to a mode determination in the DUET histogram.

We use a Matlab heuristic 2D extrema finder to search for all sufficiently isolated extrema in the histogram. We normalise the histogram with respect to the most prominent extrema and then use the spectrum of the first seven extremal points in descending order plus the first four L-Moments of the normalised histogram as input for a learning problem.

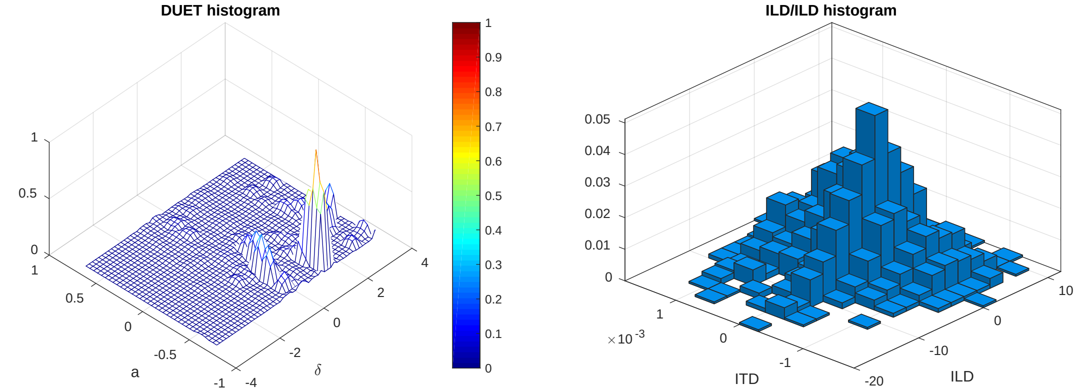


Figure 3.16: A sample DUET histogram (left) and a corresponding histogram built from ILD and ITD (right).

ILD and ITD-based mode determination

The time-frequency representation for the DUET method is obtained by STFT, rather than auditory filters. Since the processing pipeline of the AFE closely mimics the processing in the early sensory stages of the auditory system, we wanted to try to find an AFE-based version of the DUET histogram feature.

The a parameter (symmetric attenuation) of the DUET method corresponds to the ILD of the AFE. The δ parameter (phase delay) of the DUET method corresponds to the ITD. Both features are used to replace the corresponding quantities used for the DUET histogram. We build a two-dimensional histogram from ILDs and ITDs for all time-frequency bins of the signal in the same manner that the DEUT-histogram is built.

For most cases the histogram build this way did not show the clear contrast that was obtained from the DUET histogram (compare Fig. 3.16 left and right).

The two-dimensional histogram based on ILDs and ITDs was processed in the same way as the DUET histogram and produced a similar feature of normalised extrema spectrum and L-Moments that was used as input for a learning problem.

Azimuth distribution based mode determination

The azimuth distribution estimation model described in Sec. 3.4.1 provides an azimuth distribution with a resolution of 5 degrees. The distribution has a considerable front/back confusion, leading to ambiguous estimates for sources sources straight ahead and right behind. We therefore fused the bins of the backward and forward field and used the resulting fused half-field azimuth distribution.

The fused half-field azimuth distribution was processed in the same way as the DUET histogram and produced a similar feature of normalised extrema spectrum and L-Moments that was used as input for a learning problem.

AFE processors

In addition to the above described features we incorporated several features provided by AFE processors. The feature processors used were ILDs, ITDs, onset strengths, spectral features (see Sec. 3.4.2) and cross correlations. For all these features we aggregated the first four L-Moments per frequency channel and averaged the left and right channels if the processors yielded a binaural output.

Evaluation of features

Only the joint combination of all described features has been used for the number of sources estimation. An evaluation of the individual feature groups may still be worth investigating for further studies.

3.5.3 Integration of visual and auditory information

While acoustic signals are the core focus of the Two!Ears system, much can be gained in terms of robustness by using a second modality like vision to extend the available information. This is especially helpful under very noisy or reverberant conditions, such as those considered in the multi-source scenarios. We have considered video information for two purposes here: *recognition of keywords* and *speech enhancement*.

For both cases, we have employed the GRID/CHiME corpus, since matching audio and video recordings were available with sufficient amounts of training and test data, and since the acoustic recordings of speech and noise were given in binaural form.

The audio data is hence taken from the first CHiME challenge (Barker *et al.*, 2013), and it is combined with matching video data from the GRiD corpus (Cooke *et al.*, 2006b). The recordings consist of 1000 sentences spoken by 33 talkers each. All utterances include the annunciation of a letter (A...Z, excluding W) and a digit (0...9), which are hence utilized in evaluating the keyword recognition accuracy.

Audio-visual keyword recognition

Audio-only keyword recognition is still not sufficiently reliable when used under highly noisy or reverberant conditions, and in the presence of multiple sources. Currently the most successful approaches for audiovisual ASR systems, coupled hidden Markov models (HMMs) and turbo decoding, both allow for slight asynchrony between audio and video features, and significantly improve recognition rates in this way. However, both typically still neglect residual errors in the estimation of audio features, so-called *observation uncertainties*. We have therefore considered two methods for including the observation uncertainties into the decoder, and have shown that significant recognition rate improvements are achievable for both coupled HMMs and turbo decoding.

In contrast to standard single-modality data, audiovisual data contains inherent asynchronicities. As speakers may bring articulators into position before the start of phonation, for example at the beginning of an utterance, the information in the visual modality can precede that of the acoustic modality by up to 120 ms (Luettin *et al.*, 2001). Different model topologies have been considered to address this issue. Ideas range from the use of a standard HMM with concatenated features, so-called *feature fusion* (Neti *et al.*, 2000), to a number of so-called *decision fusion* approaches. This decision fusion can take place at different stages of the recognition process. Early integration, for example, fuses the information at the state level, whereas late integration may go as far as recognizing audio and video data separately and then fusing the decisions at the sentence-level (Nefian *et al.*, 2002).

In the following, we consider two methods of classifier integration. Both allow for natural, asynchronous behavior inherent to audiovisual speech, while providing the means to enforce some constraints on synchronicity through the choice of the model topology. The first method, the coupled HMM (CHMM), has been shown to be superior to feature fusion as well as to a number of graphical-model-based approaches (Nefian *et al.*, 2002).

The second method uses turbo decoding to combine classifier information and it has recently been shown to deliver a performance superior to CHMMs by Scheler *et al.* (2012). We have extended both approaches to incorporate the handling of uncertain observations, as described below and as addressed in detail in (Zeiler *et al.*, 2016b).

Coupled HMM decoding

In a coupled hidden Markov model (CHMM) the joint state transition probability is modeled as a linear combination of marginal transition probabilities. An audio stream weight λ_C is used to capture the interactions among audio o_a and video observations o_v and their respective individual observation likelihoods b_a and b_v. For the *joint* audiovisual state likelihood we obtain

$$\mathrm{p}(o_a, o_v|q_a, q_v) = b_a(o_a|q_a)^{\lambda_C} \cdot b_v(o_v|q_v)^{1-\lambda_C}. \tag{3.34}$$

For our experiments we used a token passing decoder to find the best word sequence w^* as the Viterbi path through a network of CHMM word models.

Turbo decoding

Turbo decoding (Berrou *et al.*, 1993) is an information fusion technique, which originated from a breakthrough in digital communication applications. More recently the turbo principle emerged as an alternative decoding scheme in multimodal speech recognition (Shivappa *et al.*, 2008, Receveur *et al.*, 2014) and proved to be useful for other applications such as blind speech separation (Tran Vu and Haeb-Umbach, 2013) and speech enhancement (Hong and Maitre, 2009).

Turbo decoding is based on the iterative exchange of soft information, deduced from state posteriors, between different decoders. This extra information, g_a and g_v, is used like a prior to modify the observation likelihoods b_a and b_v in the forward-backward algorithm (FBA). The modified audio and video likelihoods become

$$\tilde{b}_a(o_a|q_a) = b_a(o_a|q_a) \cdot g_a(q_a)^{\lambda_T \lambda_P}, \tag{3.35}$$

$$\tilde{b}_v(o_v|q_v) = b_v(o_v|q_v) \cdot g_v(q_v)^{(1-\lambda_T)\lambda_P}, \tag{3.36}$$

in which λ_T acts like an audio stream weight and the constant λ_P balances the likelihood and prior probability. From the FBA, we obtain new state posteriors $\tilde{\gamma}$, which subsume the likelihood, the prior probability and the extrinsic probability (Receveur *et al.*, 2014). To find the extrinsic probability $\dot{\gamma}(q_t)$ for state q and frame t, we have to remove all excess information via

$$\dot{\gamma}(q_t) \propto \frac{\tilde{\gamma}(q_t)}{b(o_t|q_t) \cdot g(q_t)}. \tag{3.37}$$

The final step of each such half-iteration is to map the extrinsic probabilities to the state space of the respective other decoder. This is done by a linear transformation.

$$g_a = T_{va}\,\dot{\gamma}_v \qquad \text{audio} \leftarrow \text{video} \tag{3.38}$$

$$g_v = T_{av}\,\dot{\gamma}_a \qquad \text{video} \leftarrow \text{audio} \tag{3.39}$$

The process of modified FBA followed by the deduction of extrinsic probabilities and their transfer to the corresponding state space is iterated for the audio and the video model a few, e.g. 4, times.

Despite objections against the applicability of plain forward-backward inference in loopy graphical models (McEliece *et al.*, 1998), we have experienced no convergence problems in our experiments.

Observation uncertainties

In typical keyword recognition systems, the probabilistic model treats all observations as though they were estimated with complete precision. However, observations that are gathered in noisy, reverberant or multi-source conditions typically come with estimation errors, which, to complicate matters significantly, clearly vary over time. Uncertainty-of-observation approaches attempt to model these residual estimation errors through probability distributions, whose first two moments are estimated and used for focussing the decoder more on the reliable and less on the unreliable sources of information.

As this approach is fully in keeping with the graphical-modelling ideas in the Two!Ears project, we have applied them to the fusion of acoustic and visual information. More specifically, we have considered two uncertainty-of-observation approaches: uncertainty decoding (Deng *et al.*, 2005) and noise-adaptive LDA (Kolossa *et al.*, 2013).

Results for both approaches, integrated into CHMM decoding as well as into the turbo decoding system, will be shown in Section 4.10.1.

Neural-network based audio-visual keyword recognition

Neural networks have led to impressive gains in acoustic speech recognition. We have therefore also considered their application to the task of audio-visual recognition in most recent work. One core question that needed answering in this context is, whether the integration of acoustic and visual streams — supported by observation uncertainties in the HMM/GMM setup — can be learned by a neural network through discriminative training, or whether it should be supported by more explicit strategies for stream weighting.

In the experiments, described below in Section 4.10.1, we show that indeed, discriminative training — even when including dynamic reliability estimates of the feature values — is insufficient for optimally learning the best fusion of both feature streams. Instead, as demonstrated in Meutzner *et al.* (2017), it significantly better to explicitly combine separate acoustic and visual DNNs, e.g. at each HMM state, by computing joint state posteriors in accordance with

$$\log \tilde{p}(\mathbf{o}_t^{\mathrm{AV}}|s) = \lambda_t \log p'(\mathbf{o}_t^{\mathrm{A}}|s) + (1-\lambda_t) \log p''(\mathbf{o}_t^{\mathrm{V}}|s). \tag{3.40}$$

Here, $\lambda_t \in [0,1]$ denotes the time-dependent stream weight of the acoustic feature stream and $\log p'(\mathbf{o}_t^{\mathrm{A}}|s)$ and $\log p''(\mathbf{o}_t^{\mathrm{V}}|s)$ correspond with the log-likelihood of the acoustic feature vector $\mathbf{o}_t^{\mathrm{A}}$ and the visual feature vector $\mathbf{o}_t^{\mathrm{V}}$, respectively, given state index s. The resulting audiovisual log-likelihood $\log \tilde{p}(\mathbf{o}_t^{\mathrm{AV}}|s)$ is then applied in decoding the sequence of bimodal observations $\mathbf{o}_t^{\mathrm{AV}} = (\mathbf{o}_t^{\mathrm{A}}, \mathbf{o}_t^{\mathrm{V}})$. The stream weights λ_t are estimated by a simple logistic regression function, based on the approach introduced by Abdelaziz and Kolossa (2014)

While this combination leads to a more complex architecture, the gains with respect to training a single, audiovisual DNN are notable, leading to the conclusion that at least under dynamically varying conditions, it is quite helpful to model each modality as well as the fusion weight, explicitly rather than relying on the discriminative training approach by itself.

3.5.4 Audio-visual speech enhancement

Models for automatic speech recognition (ASR) contain detailed information about the spectro-temporal characteristics of clean speech signals. Employing these models for speech enhancement has therefore been the target of many past research efforts. In such model-based speech enhancement systems, a powerful ASR is of fundamental importance. We have therefore investigated the use of the visual modality for deriving a more reliable, model-based clean speech estimator. Since acoustic and visual information can be integrated optimally by joint inference in both modalities within the turbo-decoding framework, we have chosen this framework as a basis of audiovisual speech enhancement.

For this purpose, we have made use of the so-called twin-HMM, a statistical model that is endowed with two associated output distributions per acoustic HMM state. One output distribution is used for inferring the acoustic HMM state, and the other is available for speech enhancement. We have extended this architecture by the additional inclusion of visual information, based on the turbo-decoding principle outlined above. Overall, this results in the following architecture, shown in Figure 3.17

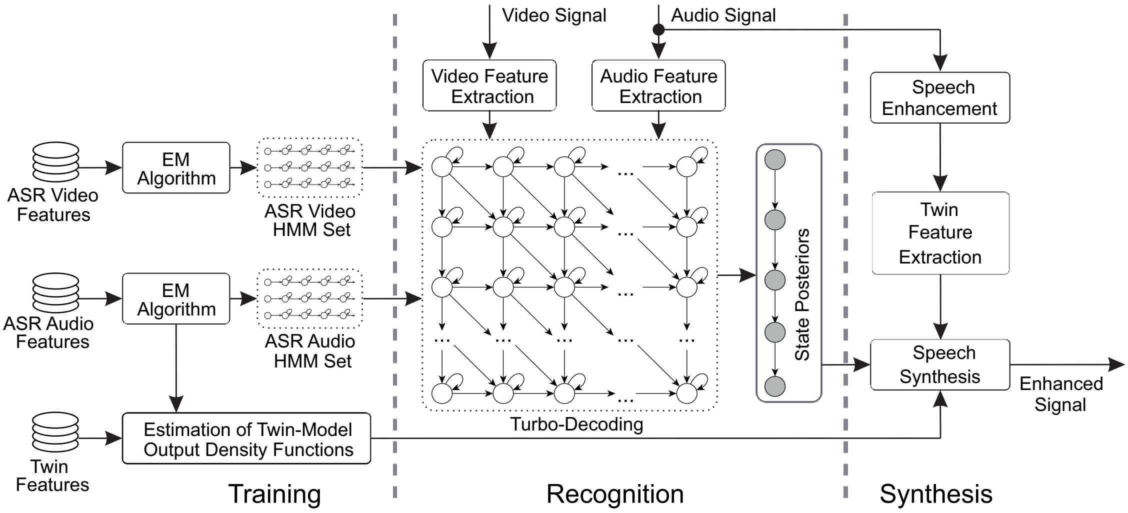


Figure 3.17: Architecture or turbo-twin-HMM-based speech enhancement.

As can be seen, the system, described in Zeiler *et al.* (2016a), unifies the turbo principle for multimodal keyword recognition with the *Twin-HMM* for speech reconstruction. For this purpose, joint inference in the audio and video sequence of an utterance is carried out using turbo decoding. After a few iterations, audio state posteriors are obtained, which serve as a prerequisite for the clean speech estimation. To find an estimate of the clean speech, we attach an additional set of output density functions to the states of the acoustic model, in keeping with the *Twin-HMM* principle (Abdelaziz and Kolossa, 2014). Here we compare two ways of synthesis, all-path (AP) and best-path (BP) synthesis. For AP synthesis, we calculate the minimum-mean square error estimate of the clean speech amplitude spectrum $\hat{x}_{\mathrm{AP}}(t)$ as

$$\mathrm{E}\big(x(t)|o(t)\big) = \sum_{i=1}^{N} p\big(q(t) = i|o(t)\big)\ \mathrm{E}\big(x(t)|q(t) = i\big). \tag{3.41}$$

This corresponds to a sum of synthesis output density function (ODF) mean values μ_i over all states i weighted by the corresponding audio state posterior $\tilde{\gamma}_a(i,t)$ for frame index t,

$$\hat{x}_{\mathrm{AP}}(t) = \sum_{i=1}^{N} \tilde{\gamma}_a(i,t)\ \mu_i. \tag{3.42}$$

For BP synthesis, instead of a weighted sum, we use the most probable state $i^*(t)$ for each frame t, to compute the clean speech signal estimate

$$\hat{x}_{\mathrm{BP}}(t) = \mu_{i^*(t)}. \tag{3.43}$$

We find $i^*(t)$ via a best path search in the state posterior matrix $\tilde{\gamma}$ constrained by the transition structure of the audio model. BP synthesis is done by using only ODFs that belong to states on the best path. Thus, in every frame only a single expectation with unity weight is used to calculate an estimate of the clean speech signal, an approach that is more efficient than the MMSE estimator above, albeit at the cost of making a hard decision on the state identity before synthesis.

Both of these ideas, all-path and best-path synthesis, will be compared in terms of human listening accuracy in Section 4.10.2, below.

4 Evaluation

4.1 Pre-segmentation

4.1.1 System configurations

In order to systemically analyze the impact of spectro-temporal context strategies in the front-end and the back-end of the AMS-based segmentation system described in Sect. 3.2.1, the following four system configurations were tested, as listed in Table 4.1. “No context” denotes the baseline configuration with no delta features in the front-end and no spectro-temporal integration in the back-end, corresponding to setting the window size $\mathcal{W}$ to unity ($\varDelta t = 1, \varDelta f = 1$). “Front-end” includes the delta features while “Back-end” only includes the second-layer classification stage in the back-end ($\varDelta t = 3, \varDelta f = 9$). “Front- & back-end” employs both the front-end and the back-end spectro-temporal context strategies.

Configuration	Front-end		Back-end	
	Delta	Feature	$\mathcal{W}$ size	
	features	dimension	$\varDelta t$	$\varDelta f$
No context	no	6	1	1
Front-end	yes	18	1	1
Back-end	no	6	3	9
Front- & back-end	yes	18	3	9

Table 4.1: Configurations of the speech segregation system

4.1.2 Stimuli

The speech material came from the Danish conversational language understanding evaluation (CLUE) database (Nielsen and Dau, 2009). It consists of 70 sentences in 7 lists for training and 180 sentences in 18 balanced test lists with 10 sentences per list for testing, and is spoken by a Danish male. Noisy speech mixtures with an average duration of 3.6 s were created by mixing individual sentences with a stationary (ICRA1) and a fluctuating

6-talker (ICRA7) noise (Dreschler *et al.*, 2001). A long term average spectrum (LTAS) template was computed based on the CLUE corpus and the LTAS of each noise masker was adjusted to the template LTAS. A randomly-selected noise segment was used for each sentence. The noise segment started 1000 ms before the speech onset and ended 600 ms after the speech offset.

4.1.3 System training and evaluation

The segregation system was trained for each of the two noise types with the noises limited to 10 s in duration[1]. The first layer of the classification back-end consisted of a subband GMM classifier with either 16 (moderate complexity classifier) or 64 Gaussian components (high complexity classifier) and full covariance matrices. The classifiers were first initialized by 15 iterations of the K-means clustering algorithm, followed by 5 (moderate complexity classifier) or 50 (high complexity classifier) iterations of the expectation-maximization algorithm. The classifiers were trained with the 70 training sentences that were each mixed three times with a randomly-selected noise segment at $-5, 0$, and 5 dB SNR. The subsequent linear SVM classifier was trained for each subband with only 10 sentences mixed at $-5, 0$, and 5 dB SNR. Afterwards, a re-thresholding procedure was applied (May and Dau, 2014a, Han and Wang, 2012) using a validation set of 10 sentences, where new SVM decision thresholds were obtained which maximized the HIT - FA rates. Both the first and second-layer classifiers employed a local criterion (LC) of -5 dB in a similar manner as Han and Wang (2012) and May and Dau (2013). The segregation system was evaluated with the 180 CLUE sentences for testing. Each sentence was mixed with the noises at -5 and 0 dB SNR. The trained models were evaluated with the same limited noise 10 s recordings used during training.

4.1.4 Perceptual outcome measures

Four different perceptual outcome measures were used for evaluation, namely the hit rate minus false alarm rate (H-FA) (Kim *et al.*, 2009), the clustering parameter γ (Kressner and Rozell, 2015), and the extended short term objective intelligibility (ESTOI) (Jensen and Taal, 2016), as well as measured intelligibility scores. To compute the H-FA rate, the correctly classified speech-dominated T-F units and incorrectly classified noise-dominated T-F units were derived by comparing the EBM with the ideal binary mask (IBM) unit for unit. The clustering parameter γ was estimated by the graphical model described in Kressner and Rozell (2015). Given a binary mask, the graphical model predicts the

[1] Originally, the ICRA1 consists of a 60 s noise recording and ICRA7 of a 600 s recording (Dreschler *et al.*, 2001)

amount of clustering γ as a single number, where $\gamma = 1.0$ reflects a mask with uniformly and randomly connected T-F units. Larger values (e.g., $\gamma = 2.0$) reflect binary masks with T-F units that are twice as likely to be in the same state as its neighboring units (Kressner and Rozell, 2015). The ESTOI (Jensen and Taal, 2016) is a modified version of short term objective intelligibility (STOI) (Taal *et al.*, 2011) to better account for modulated noise maskers. The STOI metric is based on a short-term correlation analysis between the clean and the degraded speech (Taal *et al.*, 2011), mapped to a value between 0 and 1. In the current study, ESTOI improvements (Δ ESTOI) were reported as the relative difference between the predicted ESTOI values for the processed and the unprocessed noisy speech signal.

4.1.5 Test procedure and subjects

The following 24 conditions were tested: (UN, No integration, Front-end, Back-end, Front- & back-end, IBM) × (ICRA1, ICRA7) × (moderate complexity, high-complexity). As the total number of conditions (24 conditions) exceeded the number of available CLUElists, the experiment was conducted with two subject groups, each with $n = 15$ normal hearing (NH) listeners. The first subject group was exposed to the 12 conditions with only the moderate classifier complexity, and the second group was exposed to the 12 conditions with only the high-classifier complexity.

The listener age was between 20 and 32 years with a mean of 24.5 years. Requirements for participation were: (1) age between 18-40 years (2) audiometric thresholds were less than or equal to 20 dB HL in both ears (0.125 to 8 kHz) and (3) Danish as native language, and (4) no previous experience with hearing in noise test (HINT) or CLUE (Nielsen and Dau, 2009). The total experimental time was around 2 hours including the screening process. Most subjects were paid for the participation except when payment was declined.

4.1.6 Results

First, the results from the listener study are considered and, secondly, the outcome measures are included and compared to the human intelligibility data. The results are presented by considering the effect of system configurations, the effect of system classifier complexity and finally the effect of noise types.

Effect of system configurations

Figure 4.1 shows word recognition scores (WRSs) for the four different system configurations in the two classifier complexities and two noise types. The unprocessed stimuli ("Noisy speech") is included as the baseline and the IBM as a reference for the ideal condition where the speech and the noise signals are known *a priori*. The baseline differs across noise types. Approximately $50 - 55\%$ WRS is observed for the stationary ICRA1 and 65% for the fluctuating ICRA7, presumably as the listeners were able to listen in the dips of the competing talkers noise of ICRA7. The IBMs in all combinations of classifier complexity and noise type reach ceiling levels in WRSs close to 100%. This is expected as the IBM exploits *a priori* information about the speech and the noise.

Across-list averaged WRSs for the four different system configurations in the two classifier complexities and two noise types were used as the response factor to construct a mixed linear model with three fixed factors, namely configuration (4 levels), noise type (2 levels) and complexity (2 levels), and subjects as random factor. Prior to the modeling, a normal scores plot of the intelligibility scores indicated that the ANOVA normality criteria was fulfilled. All fixed effects and all interactions amongst fixed effects were tested in the analysis on a 5% significance level. The p-values are calculated from F statistics where the degrees of freedom of the denominator are based on the Satterthwaite approximation. Least-squares means and CIs were extracted from the mixed linear model. A summary of the three-way ANOVA indicated a highly significant effect of configuration ($F[3, 199] = 69.23, p < 0.001$). The least-squares means and the CIs for levels of configurations, averaged across factors noise type and complexity, indicated the following order with non-overlapping CIs: "Front-end" (67.4% [64.4% 70.5%]) , "Front- & back-end" (56.5% [53.5% 59.5%]), "No context" (48.8% [45.7% 51.8%]) and "Back-end" (41.5% [38.4% 44.5%]). Therefore, the "Front-end" strategy led to significantly higher scores than all other configurations and the "Back-end" resulted in the lowest scores of all system configurations.

To asses speech intelligibility improvements, Paired Student's t-tests between the noisy speech and each of the system configurations were made. As a reference, maximal improvements over noisy speech are given by IBM improvements and are approximately 50% for ICRA1 and 35% for ICRA7. The t-tests indicated only significant (on a 5% significance level) improvements for the high-complexity classifier of 64 GMM components for configurations "No Context" ($t[14] = -2.16, p = 0.02$), "Front-end" ($t[14] = -4.29, p =< 0.001$) and "Front- & back-end" ($t[14] = -2.82, p = 0.007$) for the ICRA1 noise and only "Front-end" ($t[14] = -7.44, p =< 0.001$) for the ICRA7 noise.

Figure 4.2 shows WRS improvements, H-FA rates, γ values and ESTOI improvements over noisy speech for the four different system configurations in the two noise types and classifier

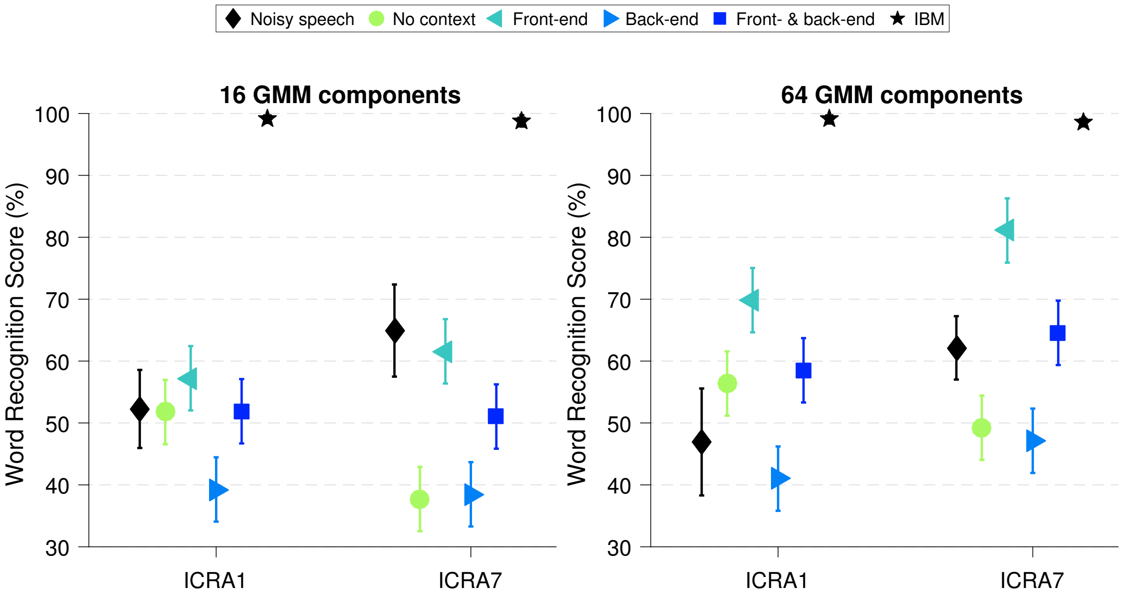


Figure 4.1: Intelligibility scores for the four different system configurations for the moderate complexity classifier (left panel) and the high complexity classifier (right panel) in the two noise types at -5 dB SNR. The unprocessed "Noisy speech" has been included as a baseline and the IBM has been added as a reference. For the baseline and the reference, sample means across subjects were computed and 95% Student's t-based CIs of the mean were computed using 14 degrees of freedom. For the other data points, the least square means and 95% CIs from a fitted mixed linear model were used.

complexities. The improvements in WRS over noisy speech from the Paired Student's t-tests are shown in the first panel in Figure 4.2. WRS improvements are derived from the Paired Student's t-tests and significant improvements (on a 5% significance level) are marked with an asterisk.

The second panel shows sample mean H-FA rates. For both noise types and classifier complexities, the lowest HIT - FA rates were observed for the "No context" configuration and the highest H-FA rates for the "Front- & back-end" configuration. Also, larger HIT - FA rates were obtained for the "Back-end" than the "Front-end" configuration. The comparison of the H-FA rates to the WRSs indicated a mismatch as the "Front-end" configuration led to the highest intelligibility scores, but not the highest segregation performance.

The third panel shows the computed γ. Reported values of γ from the "No context" configuration are consistent with previous results (Kressner and Rozell, 2015, 2016). Furthermore, Figure 4.2 reveals that the IBM itself contains a certain amount of structure, presumably due to the compact representation of speech-dominated T-F units forming

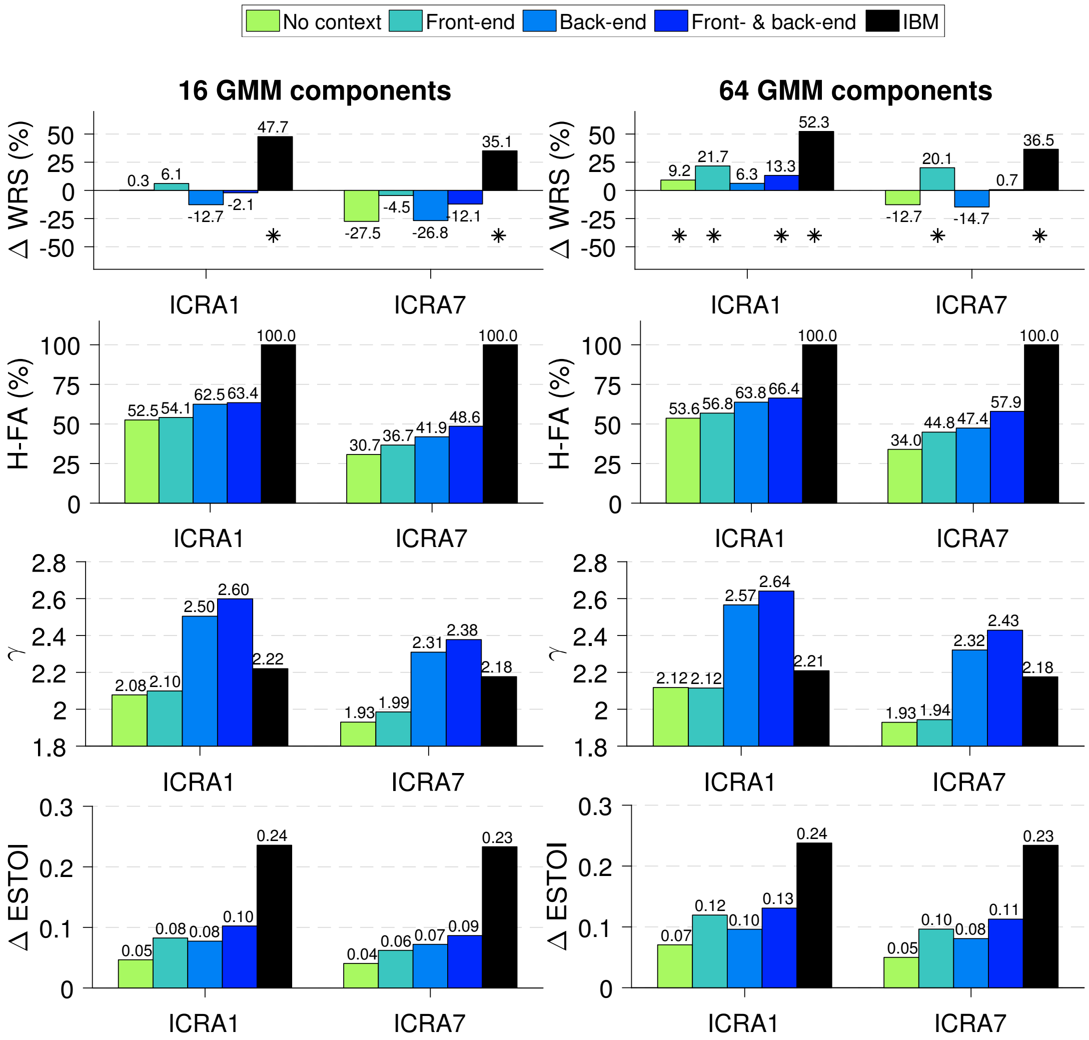


Figure 4.2: WRS improvements, H-FA rates, γ values and ESTOI improvements for the four different system configurations in the two classifier complexities (16 and 64 GMM components) and two noise types (ICRA1 and ICRA7). The IBM has been included as a reference for the ideal condition where the speech and the noise signals are known *a priori*. WRS improvements are derived from the Paired Student's t-tests and significant improvements (on a 5% significance level) are marked with an asterisk. The other outcome measures are averaged across the 180 sentences for a SNR of $-5\,\text{dB}$.

glimpses of the target signal. Most importantly, the γ values from configurations that exploited spectro-temporal context through the SVM classifier in the back-end ("Back-end" and "Front- & back-end") are consistently larger than those from configurations "Front-end" where the SVM classifier did not incorporate contextual information across adjacent T-F units ("No context"). On the contrary, the delta features alone do not seem to increase the amount of clustering in the mask.

The fourth panel in Figure 4.2 shows the ESTOI improvement over the unprocessed noisy mixture baselines. For the baselines, ESTOI predicted on average 0.59 for ICRA1 and 0.55 for ICRA7. The IBM marker in Figure 4.2 indicates the largest possible intelligibility improvement that the system configurations can achieve. Overall, the largest improvements were predicted for the configuration "Front- & back-end". However, the results for the remaining configurations were inconsistent across the noise types and classifier complexity.

Effect of the classifier complexity

The three-way ANOVA found a significant effect of complexity ($F[1, 28] = 24.38, p < 0.001$). Furthermore, significant interactions between complexity and configuration ($F[3, 196] = 3.04, p = 0.03$) and between complexity and noise type ($F[1, 196] = 6.99, p = 0.009$) were observed. The least-squares means and the CIs for levels of complexity, averaged across factors noise type and configuration, indicated with non-overlapping CIs that the high-complexity classifier of 64 GMM components resulted in scores (58.5% [55.5% 61.4%]) significantly higher than the 16 GMM components (48.6% [45.7% 51.5%]). From Figure 4.1, it is evident that the ranking of the configurations remained unchanged across classifier complexity.

Comparing the left panels with the right panels in Figure 4.2, higher H-FA rates were obtained using the high-complexity classifier over the moderate complexity classifier. Furthermore, it is interesting that an increase of 2.7% in H-FA leads to an increase in WRS of 15.6% with increasing classifier complexity. Also, larger ESTOI improvements over noisy speech were generally observed using the high-complexity classifier over the moderate complexity classifier. However, γ remained unchanged, suggesting that the structure of the mask was not affected by the classifier complexity in the segregation system.

Effect of noise type

The three-way ANOVA indicated a non-significant effect of noise type ($F[1, 196] = 0.27, p = 0.6$). A significant effect of interactions between noise type and configuration ($F[3, 196] = 8.47, p < 0.001$) was observed. The interaction amongst all three fixed factors was not significant ($F[3, 196] = 0.71, p = 0.55$). From Figure 4.1, it is also here evident that the ranking of the configurations remained unchanged across noise type. Averaged H-FA rates in Figure 4.2 indicate that the system in general produced higher H-FA rates in the presence of the stationary ICRA1 noise than for the ICRA7 noise, presumably because it was more difficult to separate the speech modulations from the non-stationary 6-talker babble noise. Also, γ values were on average larger for ICRA1 than ICRA7. In general, ESTOI predicted larger intelligibility improvements for ICRA7 than ICRA1.

4.1.7 Discussion and conclusion

In the present study the impact of different spectro-temporal context strategies in a computational speech segregation system was investigated systematically in the front-end, by computing the delta features, and in the back-end, by using a second layer SVM. The listener study in 4.1.6 showed that exploring spectro-temporal context in the front-end strategy led to significantly higher intelligibility scores than using the strategy in the back-end. Furthermore, the back-end strategy actually resulted in the lowest scores of all system configurations. The predictions of the outcome measures in Figure 4.2 indicated that the back-end strategy increased the H-FA rates but, at the same time, the amount of clustering of the mask. Previously, it has been shown that clustering of the two error types reduces the intelligibility scores in comparison to the randomly distributed errors (Kressner and Rozell, 2015), which is likely to explain the detrimental effect of the back-end strategy on the present intelligibility scores. Furthermore, from Figure 4.2 it was evident that the ranking of the system configurations remained unchanged across classifier complexity and noise type. This emphasizes these findings. Therefore, in the considered segregation system, a better spectro-temporal strategy is therefore to compute delta features of the AMS features rather than employing the selected spectro-temporal integration strategy, despite the fact that the H-FA rate does not increase as much as when exploiting contextual knowledge through a SVM classifier.

4.2 Source localisation

In this section we evaluate sound source localisation in a cocktail party scenario (DASA-1, WP6). Both bottom-up and top-down systems are at play in the perceptual organisation of sound, via a process termed ‘auditory scene analysis’ (ASA) by Bregman (1990a). In the cocktail party scenario, the voice of the target speaker and interfering (masker) voices will originate from different locations. Hence, binaural cues will differ for the target and maskers, providing a means to identify them. In addition to this bottom-up cue, top-down knowledge can also be applied. In the cocktail party, this includes information about the vocal characteristics of the target and masker voices, and also knowledge about their spatial positions. In the latter regard, listeners could potentially exploit the fact that the spatial locations of the masker voices are known.

Indeed, a recent psychophysical study has shown that listeners are able to exploit prior knowledge of the masker locations in a cocktail party scenario. Kopco *et al.* (2010) investigated the ability of listeners to localise a female target voice in the presence of four male masking voices. They found that listeners were better able to localise the target when the spatial locations of the masker voices were cued before the task.

We conducted listening experiments to investigate whether listeners are able to exploit prior information about the masker locations in Kopco et al.'s task when listening over headphones, where binaural cues are limited to those present in the HRIRs used to spatialise the signals for headphone listening. In headphone listening, head movements are not available and room characteristics can be carefully controlled; hence, we also investigate whether prior knowledge of the masker locations can assist localisation in both anechoic and reverberant conditions.

The approaches described in previous chapters (DNN localisation, Section 3.4.1; source segregation and localisation with top-down knowledge, Section 3.5.1) are integrated in a computational system and evaluated in the same task. In particular, we ask whether the sources of knowledge available to listeners in this scenario – speaker characteristics and masker locations – can be successfully exploited in a computational system for sound localisation. Such information is not typically used in machine listening systems for source localisation (Ma *et al.*, 2015b).

The remainder of the section is structured as follows. First, we describe a listening test which broadly follows the method of Kopco et al., but uses headphone listening and assesses listener performance under both reverberant and anechoic conditions. A computational model is then described, which exploits speaker models and prior information of masker locations within a DNN architecture. Finally, a comparison of listener and model performance on the same localisation task is presented.

4.2.1 Methods

Eleven normal-hearing listeners participated in the listening test, including four females and seven males between the ages of 22 and 50 years.

Speech materials were taken from a corpus of monosyllabic words recorded at Boston University's Hearing Research Centre (Kidd *et al.*, 2008), as used in (Kopco *et al.*, 2010). The target was a female voice speaking the word ‘two’. The four maskers were male voices speaking non-digit words, drawn randomly from a set of 32 words. Speech material was recorded at a sample rate of 44.1 kHz with an average duration of 0.4 s.

Participants listened to the stimuli via headphones in a simulation of binaural localisation. Two listening sessions were included. For the anechoic session, binaural speech signals were created by convolving monaural signals with HRIRs recorded from the Knowles Electronic

Manikin for Acoustic Research (KEMAR) dummy head (Wierstorf *et al.*, 2011, WP1). For the reverberant session, the BRIR of Room A from the Surrey BRIR database (Hummersone *et al.*, 2010) was used to simulate reverberant room conditions. The Surrey database was captured using a Cortex head and torso simulator (HATS). Room A has a reverberation time (T_{60}) of 0.32 s and a DRR of 6.1 dB. The room has dimensions 5.7 × 6.6 × 2.3 m (width × length × height), and the BRIR was measured at a head height of 1.78 m and a distance of 1.5 m between the circular loudspeaker array and the HATS.

Binaural mixtures of five competing talkers (one female target, four male maskers) were created by spatialising each talker separately before adding them together in each of the two binaural channels. For both anechoic and reverberant sessions, each masker was equal in level to the target.

Listeners participated in the experiment in a sound-attenuating booth using a computer running MATLAB. Stimuli were presented over a pair of Sennheiser HD 600 headphones. A graphical user interface (GUI) was used to record participants' responses. Their task was to report the location of the female-voice target either in isolation (control runs), or in the presence of four male-voice maskers (masker runs). Participants indicated their response by selecting a loudspeaker location in a loudspeaker array shown on the computer screen using a computer mouse. There was also a button in the GUI that listeners could press to indicate that no target was heard.

The listening tests were administered across two sessions that were completed on different days. In one session anechoic stimuli were used while in the other session reverberant stimuli were used. At the beginning of each session, a practice run was included in which the participants listened to the female-voice target in isolation from all target locations. After that, 12 runs were presented following a similar procedure adopted in (Kopco *et al.*, 2010). The first and last of these were no-masker control runs, each of which consisted of 55 trials (5 trials per target location). In the masker runs the maskers were presented in one of five masker patterns. There were five runs where the masker pattern was kept fixed for the duration of the run (Fixed), and five runs where the masker pattern was randomly chosen on each trial (Mixed). Each masker run consisted of 60 trials including five catch trials, in which the target was replaced by another random male masker. The catch trials were included in order to monitor false alarm rates. The type of masker runs was indicated at the beginning of each run by presenting a recording of the phrase 'fixed maskers' sequentially at each of the four masker locations for the Fixed runs, and a recording of the phrase 'mixed maskers' for the Mixed runs. The Fixed and Mixed runs were interleaved.

4.2.2 Configuration of the Two!Ears system

We now show how the Two!Ears system can model listener performance in this task. The system uses DNNs to learn the relationship between binaural cues and source azimuth (Section 3.4.1). It also exploits top-down knowledge about the spectral characteristics of the target source, and the prior knowledge of masker positions when available (Section 3.5.1).

The auditory front-end (WP2) consisted of a bank of 32 overlapping Gammatone filters, with centre frequencies uniformly spaced on the ERB scale between 80 Hz and 8 kHz (Wang and Brown, 2006). Inner-hair-cell processing was approximated by half-wave rectification. Following this, the cross-correlation between the right and left ears was computed independently for each frequency channel using overlapping frames of 20 ms duration with a shift of 10 ms. As in (Ma *et al.*, 2015d), the system used the whole cross-correlation function, instead of ITD, as localisation cues. When sampled at 16 kHz, the cross-correlation function with a lag range of ± 1 ms produced a 33-dimensional binaural feature vector for each frequency channel. This was supplemented by the ILD, forming a final 34-d feature vector.

The relationship between binaural cues and source azimuth in each frequency channel was learned by a DNN. The DNN consists of an input layer, 4 hidden layers, and an output layer. The input layer contained 34 nodes and each node was assumed to be a Gaussian random variable with zero mean and unit variance. The hidden layers had sigmoid activation functions, and each layer contained 128 hidden nodes. The output layer contained 51 nodes corresponding to 51 azimuth angles between -50 deg and 50 deg with an azimuth resolution of 2 deg. The 'softmax' activation function was applied at the output layer.

The DNNs were trained using speech signals from the GRID corpus (Cooke *et al.*, 2006c), spatialised using an anechoic HRIR measured with a KEMAR dummy head (Wierstorf *et al.*, 2011). Diffuse noise were added during training and there was no retraining using the matching BRIR for this study.

Source spectral characteristics were modelled using ratemap features (Brown and Cooke, 1994a). A ratemap is a spectro-temporal representation of auditory nerve firing rates, extracted from the inner hair cell output of each frequency channel by leaky integration and downsampling. Ratemaps were computed for each ear, averaged across the two ears, and finally log-compressed (cf. the log-max approximation noted above). The stimuli from the practice run were used to estimate source model parameters for the target talker, and the signals of the catch trials were used to estimate the masker model, i.e. the system used the same information that was available to the listeners.

In the Fixed runs prior knowledge about masker locations was available. Such information can be exploited in the system by reducing the probability for azimuth ϕ where a masker is located. However, since a target and a masker can be co-located, by doing so the probability at the true target location is also reduced. The proposed system therefore only reduced the probabilities at a masker location ϕ in the time-frequency region considered to be dominated by the maskers.

4.2.3 Results and discussion

Results of listening tests

Figure 4.3 plots RMS errors averaged across participants in the no-masker control condition as a function of the target location. For the anechoic case, RMS errors grew approximately with target laterality from about 11° to 22°. The V-shape trend is consistent with that reported in (Kopco *et al.*, 2010) where localisation was performed in a real room, but the localisation accuracy is substantially lower for headphone listening and the V shape is also less pronounced.

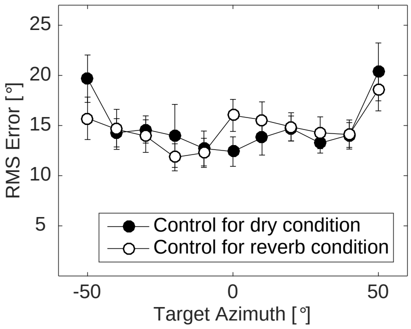


Figure 4.3: Localisation performance in the no-masker control condition. Across-participant averages (±1 SEM) of the RMS error are plotted as a function of the target location for both anechoic (dry) and reverberant (reverb) sessions.

For the reverberant case, however, control data did not show a dip at the central locations, with RMS errors ranging between 12° to 17°. This is most likely due to the effect of reverberation on perceived location. Many participants reported that the target speech appeared to emirate from above them, presumably due to reflections from the ceiling in which the BRIRs were recorded. Another possible explanation is that the mismatch between the listeners' HRIRs and the one used to simulate binaural listening in this study could disrupt speech localisation more when reverberation is present, as shown in (Begault, 1992).

Figure 4.5 shows the effect of maskers on RMS errors for each target location and each masker pattern, by subtracting control RMS errors for each participant from those in the different masker conditions. The effect of masking on RMS errors depended in a complex way on various parameters manipulated in this study. First, the presence of maskers resulted in a larger increase in error in the more challenging reverberant condition, in particular at more lateral target locations. In the anechoic condition, however, the RMS errors tended to increase most at more central locations between -20 deg and 20 deg. This is in contrast to findings by Kopco *et al.* (2010), where the RMS errors tended to increase most at target locations that corresponded to masker locations.

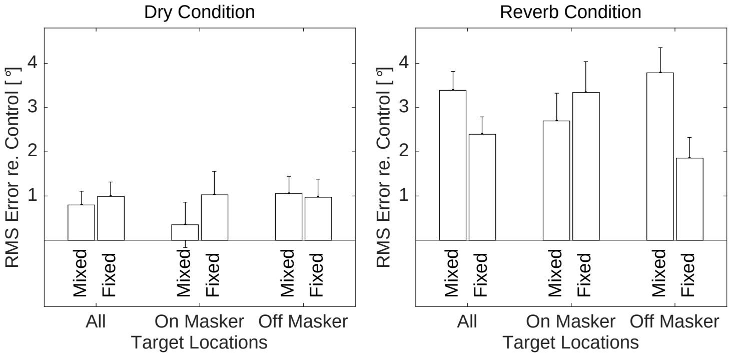


Figure 4.4: Effect of maskers on localisation accuracy shown as the increase in RMS error (*re.* the no-masker control condition).

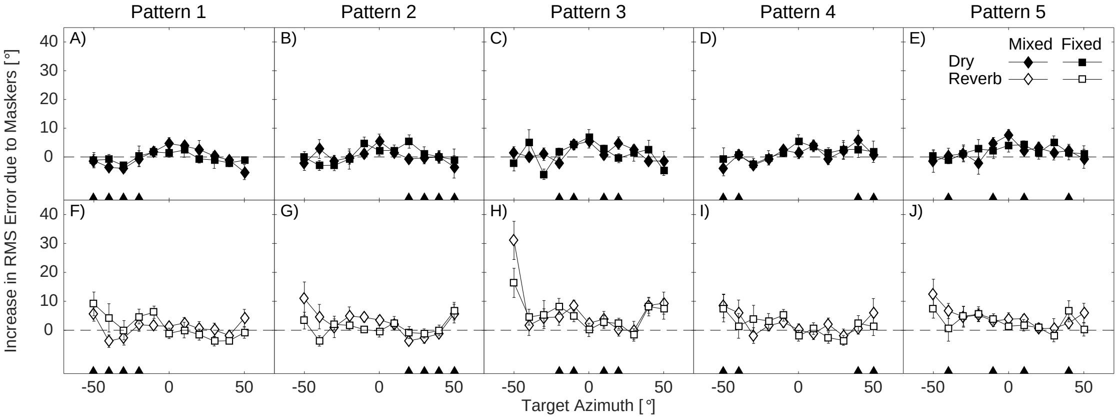


Figure 4.5: Across-participant average (± SEM) of the increases in RMS errors (*re.* the no-masker control condition) as a function of the target location. Masker locations are indicated by the filled triangles along the abscissa.

The effect of maskers on RMS errors also depended on whether the masker locations were fixed or mixed within a run. This is better illustrated in Figure 4.4, which shows the increase in the RMS error averaged across all masker patterns and across either all target locations (all), across the target locations where the target was presented with a co-located masker (on-masker), or across the target locations where the target was not co-located with a masker (off-masker). The availability of *a priori* information about masker locations had a main effect in the reverberant condition. Averaged across all target locations, the RMS error reduction in the Fixed condition compared to the Mixed condition was 1.1° (or 31%). The effect of *a priori* knowledge was even larger when only the off-masker target locations were considered, reducing the RMS error by 2° (or 51%). On the other hand, the *a priori* information had a modest effect on the on-masker targets, increasing the RMS error by 0.6°. The effect is also modest under the anechoic condition. A repeated measures analysis of variance (ANOVA) confirms that the *a priori* information only has a significant main effect on localisation accuracy in the reverberant condition when all target locations are considered $[F = 5.56, p < .05]$ or only the off-masker locations are considered $[F = 13, p < .005]$. No significant main effect was found for the on-masker data and in the anechoic condition.

Results of model simulation

The proposed model achieved 100% target location accuracy in the no-masker control runs under both anechoic and reverberant conditions, compared to an average of 15° error by listeners.

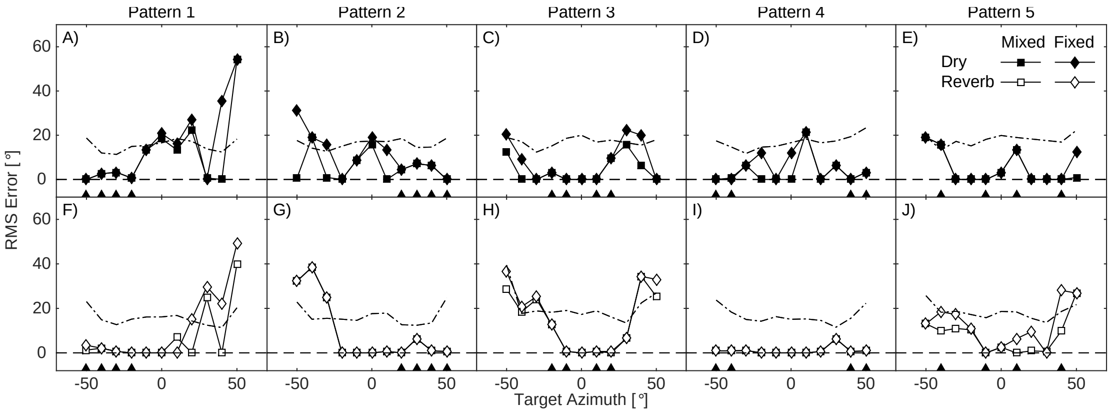


Figure 4.6: RMS errors of the proposed model as a function of the target location. Average listeners' RMS errors are plotted as dotted lines in each panel for comparison.

Model RMS errors in masker conditions are plotted in Figure 4.6, which also shows the average listener data for comparison. The machine system outperformed listeners in many conditions. This can be largely attributed to the use of source models in the system (Section 3.5.1), without which performance was poor in this relatively challenging localisation task[2].

The system did not perform well when the target was not co-located with a masker, especially for masker patterns 1–3 in which the maskers were more clustered. Apparently, in such conditions the maskers disrupted the localisation cues for the target more than when maskers were distributed in space, and the DNN failed to indicate a high probability of a sound source occurring at the target location.

Figure 4.7 shows that the machine system also benefitted from prior knowledge about the masker locations. The average error reduction in the Fixed condition compared to the Mixed condition was 3.2° in the anechoic condition and 2° in the reverberant condition (both significant with $p < .005$). As in listeners' data, this error reduction became larger when only the off-masker locations were considered.

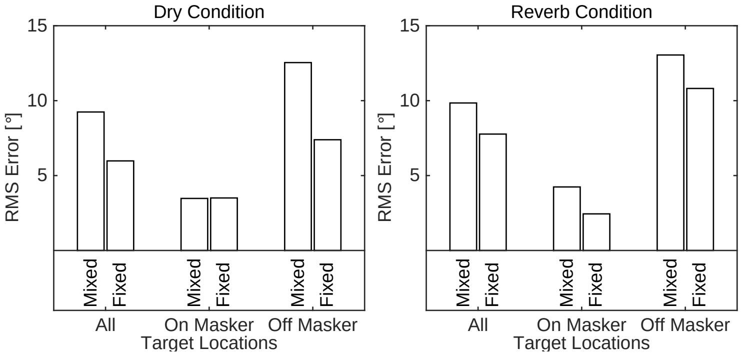


Figure 4.7: Effect of maskers on localisation accuracy of the proposed model shown as RMS error.

4.3 Source identification

In this section, we evaluate our identification system described in Sect. 3.4.2 on the development system, i.e. on simulated auditory scenes. We first show in Sec. 4.3.1 how the performance depends on scene conditions and that models can be very sensitive to changes of conditions under which they were trained, that is we evaluate the robustness of detection. Then, we show how we solved this issue in Sec. 4.3.2 through "multi-conditional" training; and finally present our evaluation results for this kind of models under realistic reverberant conditions in Sec. 4.3.3.

[2] Each masker was equal in level to the target, which is equivalent to about -12 dB target-to-masker-ratio with four maskers.

4.3.1 Robustness of single-conditional source detectors

In this study, we used simulated auditory scenes to evaluate our single-conditional sound type classification models (Sec. 3.4.2) obtained with the Lasso classifier (see Sec. 3.3.3) by inductive learning. In particular, we investigated the impact of different angular source configurations and signal-to-noise ratios. We chose these configurations in such a way that we could also investigate the effect the orientation of the head (relative to the sources) has on the detection of environmental sounds (see Deliverable D4.3, Sec. 4.2.6). By building machine learning models on training data under particular conditions and evaluating these models both under the same (*iso*) and under different (*cross*) conditions, we quantified how differences between the conditions affect generalization performance.

Furthermore, we showed that model robustness can be improved (i) by augmenting the training data of the sound type classifiers by superimposing general environmental sounds, and in the next section described, (ii) by using multi-conditional training, in which training data for model building is composed from many different scene configurations, in which signal-to-noise ratios and/or angular source directions are varied.

Data and model construction

As with training, for all evaluations of sound event detection we used the NIGENS database, with its $13 + 1$ specific event classes of environmental sound types. Due to the high computational demand of the following analysis of single-conditional models (tens of thousands of cross-tests), we used only four of our classes as detection *targets* in this study: alarm, crying baby, female speech, and fire. All sounds of the other classes still provided counter-examples for model training and testing. The very diverse set of sounds from the general class served both as counter-examples for the classifiers and as distractor signals.

Auditory scenes were generated as described in Sec. 3.4.2, with the following specifics in this case:

1. Scenes composed of a single point source emitting sounds from all classes (including the general class) at 5 azimuth angles $\{0°, 22.5°, 45°, 67.5°, 90°\}$ (all azimuths are given with respect to the direction of the nose of the binaural head, see Fig. 3.9).

2. Scenes containing two point sources emitting sound simultaneously: A *target source* emitting sounds from all classes, and a *distractor source* emitting only sounds from the general class. Target and distractor sound sources were located at 17 combinations of azimuth angles: {0°/0°, 0°/45°, 45°/0°, 22.5°/−22.5°, 67.5°/112.5°, 0°/90°, 22.5°/112.5°, 45°/135°, 90°/180°, 22.5°/−67.5°, 45°/−45°, 90°/0°, 0°/180°, 22.5°/−157.5°, 45°/−135°, 67.5°/−112.5°, 90°/−90°}. The SNR between the target

and distractor source was set (with the method described in 3.4.2) to the values −20 dB, −10 dB, 0 dB, or 10 dB.

In the following we refer to the sounds of scene 1 both as *clean sounds* or as scenes with an SNR of inf dB, mixing terminology of scenes 1 and 2.

In the study of single-conditional models' generalization, we used both mean-channel and two-channel feature sets, based on ratemaps and onset strength maps with 16 frequency channels, spectral features based on 32 frequency channels (all three framed in 20 ms time bins), and amplitude modulation spectrograms with 32 ms time frames over 8 frequency channels with 9 modulation filters. Regardless of the number, frequency channels always ranged from 80 Hz to 8 kHz.

Training of models was conducted using logistic Lasso in the AMLTTP (see Sec. 3.4.2) using data from exactly one acoustic scene with one SNR and azimuth configuration only. The training set amounted to roughly $75k$ feature vectors (1082-dimensional for mean-channel, and 2164-dimensional for two-channel feature sets) and corresponding labels.

Evaluation method

Data were split into a training set for model building and a test set for estimating the generalization performance of the classifiers. In order to ensure that a block from the training set and a block from the test set never contained parts of the same sound file, training-test splits as well as cross-validation splits were conducted at the level of the original sound files. This means that the set of sound files for each class was randomly split into training set (75%) and test set (25%). Only the sounds from the training set were used to generate the auditory scenes for building the classification models, and only the sounds from the test set were used to generate the auditory scenes for evaluating the prediction performance. All evaluations were done on three different training-test splits.

Performance was always evaluated on individual "single conditions", either

- on test data chosen from a scene and combination of SNR and azimuth configuration included in the training data (*iso-testing*), or
- on test data chosen from a scene and combination of SNR and azimuth configuration excluded from the training data (*cross-testing*).

Below, we will use the term *iso-azimuth* to refer to generalization performance estimates where the same azimuth configuration was used for testing and training, but where the SNR value was chosen randomly from the 5 SNR values, independently for training and testing.

Likewise, we will use the term *iso-SNR* to refer to generalization performance estimates where the same SNR was used for testing and training, but where the azimuth configuration was chosen randomly from the 17 simulated azimuth configurations, independently for training and testing.

The *balanced accuracy* (BAC) is used as performance measure, which is defined as the arithmetic mean of sensitivity, i.e. the true positives (TP) divided by the size of the positive class (PC), and specificity, i.e. the true negatives (TN) divided by the size of the negative class (NC):

$$BAC = \frac{1}{2}\left(\frac{TP}{PC} + \frac{TN}{NC}\right) \tag{4.1}$$

Generalization from clean to noisy sounds

In the first analysis, we investigated how well sound categorization models trained on clean sounds generalize to noisy situations where an additional distractor source is present. To this purpose, we evaluated our models trained on clean sounds (sc_{inf}) under iso-testing (identical SNR (inf) and azimuth configuration on training and test set), as well as on the scenarios where a distractor source from the general class was superimposed on the target source at signal-to-noise ratios (SNR) of 10 dB, 0 dB, −10 dB and −20 dB (cross-testing; iso-azimuth). Target and distractor sound sources were placed at various azimuth configurations (set 2).

Fig. 4.8 shows the test set model performance values pooled over all four target classes, all angular configurations, and all data set splits. The figure shows two groups of box-plots corresponding to mean-channel features (left) and two-channel features (right). The first box-plot in each group corresponds to iso-testing (trained on clean sounds, tested on clean sounds, same azimuth configuration), whereas the next four plots correspond to cross-testing (trained on clean sounds, tested on noisy sounds; same azimuth configuration) with increasing noise level. On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The notches represent the 95% confidence interval for the true median, while the whiskers indicate the minimum and maximum values.

While models trained on clean sounds perform well under iso-conditions, their median cross-test performance, even when only a relatively quiet distractor source is added to the scene (SNR of 10 dB), is much worse, and decreases further with increasing noise level (decreasing SNR). For models trained on clean sounds, there is no significant difference between the median performance of the mean-channel and two-channel feature sets.

This finding gave rise to the question whether the bad generalization of clean models to noisy

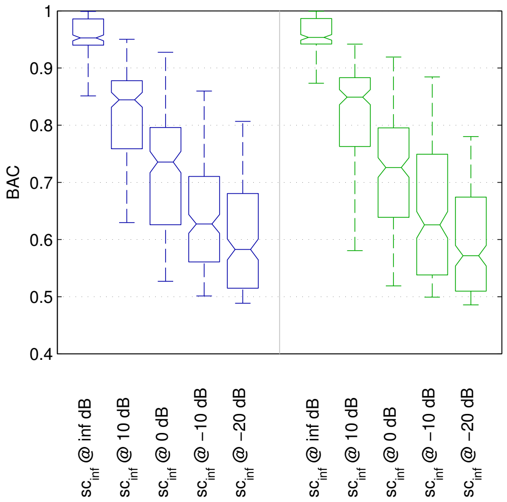


Figure 4.8: Performance of the sc_{inf} model, separately for mean- (left) and two-channel features (right), including all dataset splits, classes and iso-azm configurations.

data was due to the intrinsic increased difficulty of the task, i.e. a better performance could not be expected, or whether models trained on clean sounds might be sub-optimal when applied in a setting where multiple sources are present simultaneously.

To answer this question, we conducted a second analysis, where we not only considered classification models trained on clean sounds, but also classification models trained on scenes where a distractor source from the general class was superimposed on the target sound. SNR and angular configuration were set to the same values that were later used for testing (*iso*-testing). The corresponding performances are shown in Fig. 4.9, pooled over the $17 \times 3 \times 4$ model tests for each box (azimuths, dataset splits, target classes).

We continue to observe a drop in performance with increasing noise levels – however, it is much less steep compared to Fig. 4.8. We can see that for each SNR, the iso models perform significantly better than the models trained only on clean sounds, with differences up to 15%. We also see that the intrinsic performance decrease with SNR is shaped differently from the performance decrease of the clean models, namely going from a small drop to higher ones, instead of the other way around.

For models trained on set 2, the iso-test performances are significantly higher for the two-channel than for the mean-channel feature set, the difference growing with increasing SNR. We hence assume that models using the two-channel feature set can make better use of angular separation between target and distractor sources.

These results suggest that models trained on clean sounds do not generalize well to realistic acoustic environments occupied by additional distractor sound sources. Instead, models

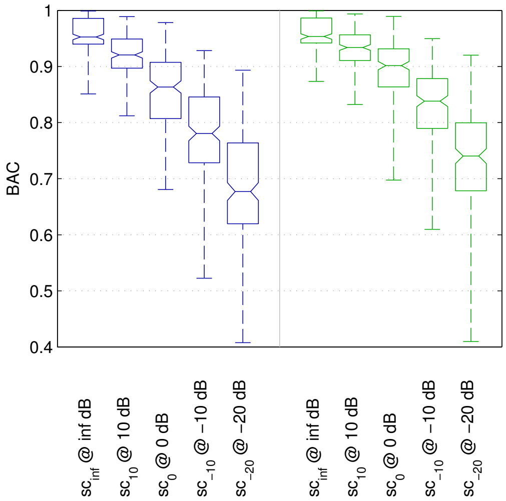


Figure 4.9: Performances of sc_{iso} models, separately for mean- (left) and two-channel features (right). The boxplots include all dataset splits, classes and iso-azm configurations.

specialized to the particular SNR condition at training time were shown to yield significantly better performance at test time. One likely explanation is that the models trained on clean data over-fit on features that are well discriminating in the noise-free case, but no longer in the noisy scenario. However, if distractors are already superimposed during training, models obtain a higher robustness against such noise.

Note that the general sound class used for generating the distractor signals contains a very diverse set of environmental sounds, hardly showing any similarities other than being a sound at all, and that neither distractor nor target sounds used for testing have been involved in the training process at any point. Therefore, the models do not adapt to a particular type of distractor signal and filter it out, but rather learn to be robust against a wide spectrum of possible distractors by finding the features that uniquely discriminate the target class from all others.

Generalization across conditions

In the previous analysis, we had looked at the generalization of models from clean sounds to sounds with a superimposed environmental sound source as additional distractor. In the following analysis, we addressed the more general question of how well models trained under a particular condition predict when applied to a different condition than the one used for training. In particular, we were interested in how far the performance of such cross-tests depends on the similarity between training and test conditions.

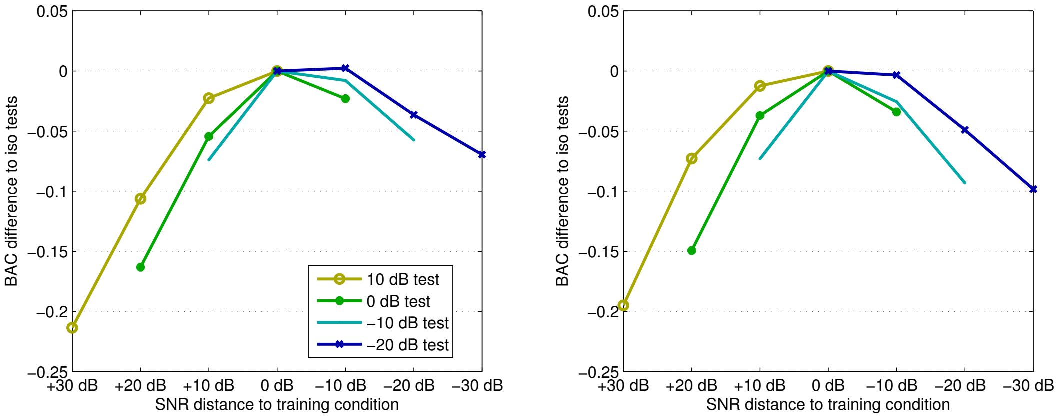


Figure 4.10: Performance difference of cross- and iso-tests at particular SNR distances ($d = SNR_{te} - SNR_{tr}$, e.g. tested at 0 dB, trained at -20 dB equals a distance of +20 dB), averaged over all iso-azimuth configurations, dataset splits, and sound classes. Left: mean-channel feature set, right: two-channel feature set.

In the scenes of set 2 with two simultaneously present sound sources two factors describe the conditions: (i) the signal-to-noise ratio (SNR) between the two sources, and (ii), the azimuth angles of the two sources relative to the nose of the head. In order to understand how cross-testing the two factors affects generalization, we modulated each factor separately giving *iso-azimuth* and *iso-SNR* conditions as before.

The effects of SNR differences between testing and training on the generalization performance are depicted in Fig. 4.10, summarizing the *iso-azimuth* tests. The curves show the difference in balanced accuracy between *cross-SNR/iso-azimuth* and the respective *iso-SNR/iso-azimuth* tests as a function of the difference between the SNR at test time and the SNR at train time. The results were averaged over all azimuth configurations, data set splits and target sound types and plotted separately for mean-channel and two-channel feature sets. The results from testing or training on the 'clean" sounds (inf dB condition) were not included in this analysis.

By definition, the difference at iso-testing (0 dB distance) is zero. As the test SNR deviates further away from the training SNR, the performance loss as compared to iso-models increases monotonically, across all testing SNRs, and in both directions of deviation. However, there is one exception: if tested at -20 dB, mean-channel models (Fig. 4.10) trained at -10 dB perform as well as iso-trained models. If we focus on the range from $+10$ dB to -10 dB, we see an indication that models trained at a lower SNR than at which they were tested degrade less compared to models that have been trained at a higher SNR.

In order to assess the effect of changing the angular configuration of target and distractor

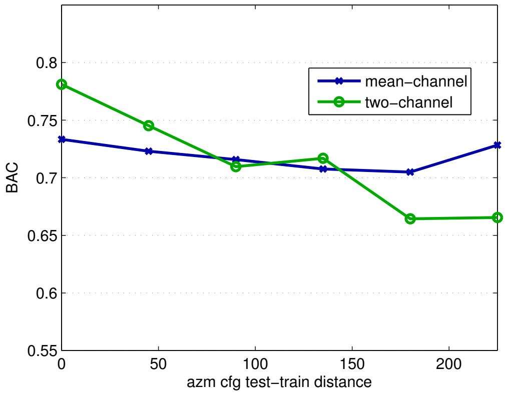


Figure 4.11: Performance of single-condition training cross-azimuth tests, averaged over all data set splits, classes, and iso-SNR configurations. The x-axis groups cross tests by "azimuth distance" of the cross configurations.

source, we defined a simple (geometric) distance measure for two angular source configurations (ϕ_1, ψ_1) and (ϕ_2, ψ_2) for conditions c_1, c_2. If the smallest angular distance $\alpha(\phi_1, \phi_2) \in [0°, 180°]$ between the two azimuths $\phi_1, \phi_2 \in (-180°, 180°]$ of one source in conditions c_1 and c_2 is defined as

$$\alpha(\phi_1, \phi_2) = \min(\beta(\phi_1, \phi_2), 360° - \beta(\phi_1, \phi_2)) \tag{4.2}$$

$$\text{with } \beta(\phi_1, \phi_2) = \max(\phi_2, \phi_1) - \min(\phi_2, \phi_1), \tag{4.3}$$

then the distance $\delta(c_1, c_2) \in [0°, 360°]$ between the two angular source configurations is defined as

$$\delta(c_1, c_2) = \alpha(\phi_1, \phi_2) + \alpha(\psi_1, \psi_2). \tag{4.4}$$

Fig. 4.11 shows the average balanced accuracy of our *iso-SNR* tests as a function of this distance δ between testing and training configurations for both the mean-channel and the two-channel feature set. The average includes all data set splits, target classes, and iso-SNR conditions.

Three effects can be observed:

- For the mean-channel feature set, the performance decreases only slightly with the distance in angular source configuration. The increase at the end of the distance curve can be explained by the geometric nature of our distance measure, which does not account for symmetry effects in binaural auditory perception, as there is a certain degree of symmetry between sounds coming from the front and the back of the head.

- For the two-channel feature set, the performance drops much more steeply as δ increases. For a δ of 180°, the drop in average BAC is larger then 10%.
- For $\delta = 0°$ (*iso-azimuth*), the two-channel feature set models exhibit a much higher average performance than the mean-channel models. The two curves cross at a distance of about 100° to 150°. This shows that sound type classification models on the two-channel feature set are better at making use of directional separation of sources, but are at the same time more strongly affected by deviations of the true angular source configuration from the training situation.

4.3.2 Multi-conditional in comparison to single-conditional detection models

Section 4.3.1 showed that the application of models is very sensitive to deviations of the environment from training conditions. Practical applications are confronted with two problems:

1. Training specialized models for every possible combination of conditions would be extremely demanding, if not prohibitive.
2. Inferring the current conditions is very difficult, in particular for the SNR, which also varies strongly over time. Unfortunately, model performance is particularly sensitive to the deviation of the SNR from the training situation.

In order to obtain classifiers that are robust against varying conditions, we suggested the use of *multi-conditional* (MC) training, in which the data for training the classification model is pooled over different conditions.

Three sets of scenes were added to the ones used with the single-condition (SC) models in Sec. 4.3.1:

2.b Scenes containing two point sources emitting sound simultanously: A "*target source*" emitting sounds from all classes, and a "*distractor source*" emitting only sounds from the general class. Target and distractor sound sources were located at 16 combinations of azimuth angles: {0°/0°, 0°/45°, 45°/135°, 45°/−135°, −45°/−45°, −45°/−90°, 90°/180°, 90°/−90°, −90°/−45°, −90°/90°, 135°/−135°, 135°/45°, −135°/−90°, −135°/45°, 180°/180°, 180°/0°}. The SNR between the target and distractor source was set (with the method described in 3.4.2) to the values −20 dB, −10 dB, 0 dB, or 10 dB.

3. Scenes containing an ambient (or diffuse, i.e. non-directional) "*target source*" emitting sounds from all classes.

4. Scenes containing two ambient sources emitting sounds simultaneously: an ambient "*target source*" emitting sounds from all classes and an ambient "*distractor source*" emitting sounds from the general class.

Sets 2 from Sec. 4.3.1 (in the following: set 2.a) and 2.b from this analysis differ in the distribution of azimuths: in 2.a, the target source azimuth resolution was higher, but the target source was always located between 0° and 90° in all 17 configurations (the distractor locations were distributed also outside this quadrant). This set was used for all *tests*. In set 2.b, the target and distractor sources were distributed uniformly around the circle, at the cost of resolution. This set was used to *train* our multi-conditional models in order to avoid a bias in performance towards quadrant one.

We investigated two types of multi-condition models:

1. mc_{amb}: Models trained by combining data from the ambient source configurations from the sets of scenes 3 and 4 described above at all SNRs. Since there are only ambient (or diffuse) sources present, these models cannot learn any directional information or head-related changes of the signals through the training data; they can also not adapt to a particular SNR because all SNRs are equally present in the data.

2. mc_{ps}: Models trained by combining data from all point source configurations from set 2.b at all SNRs. In this set, target and distractor source angles were uniformly distributed around the circle. These models cannot adapt to a particular SNR either, but get information about head orientation-related changes of the signal through training data.

In Fig. 4.12 we compare the generalization performance of mc_{amb} and mc_{ps} models to the generalization performance of single-conditional models tested in the following paradigms:

- sc_{iso}: training and testing under the same conditions, both with respect to SNR and azimuth.

- sc_{isoAzm}: training and testing at the same azimuth configuration, but at all SNR combinations (including iso). This displays a situation in which there is no information about the true SNR and thus a (in this respect) random model is chosen.

- sc: training and testing at arbitrary iso- and cross-configurations, resembling a situation in which there is no information about the true SNR and azimuth configuration.

The multi-condition models are tested on the same data as the single-condition models, using sets 1 and 2.a. The figure shows box-plots of the balanced accuracy for each model/test situation. Results were pooled over all azimuth configurations on the test scenario, all data set splits, and all target sound classes, both for the mean-channel feature set (left-hand side) and the two-channel feature set (right-hand side). On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The notches represent the 95% confidence interval for the true median, while the whiskers indicate the minimum and maximum value.

The best median performance is obtained for the sc_{iso} models. This is not surprising, as these models are specialized to the particular angular source configuration and SNR present in the test set. The sc_{isoAzm} models, which are only specialized to the correct angular source configuration but trained at an arbitrary SNR, perform on average much worse (about 10%). For the "arbitrary" single-condition model, performance drops even further, by about 2% on the mean-channel feature set and by about 4% on the two-channel feature set (recall from Sec. 4.3.1 that the two-channel models suffer from stronger degradation in cross-azimuth situations).

The multi-conditional ambient (mc_{amb}) models, which were not trained on directional information at all, (i) perform on average slightly better than the sc_{isoAzm} models, although the latter were specialized on the correct angular configuration, and (ii) clearly outperform the single-condition models trained at an arbitrary SNR and azimuth configuration. This means that in the absence of *reliable* information about the testing conditions, which would allow a suitable choice of trained model, the mc_{amb} models offer a stronger performance – without the need to train many models and predetermine conditions.

However, the multi-conditional ambient models are clearly outperformed by the multi-conditional point source models (mc_{ps}), which were trained on a uniformly distributed set of target and distractor source angles under varied SNR. The median performance of the mc_{ps} models lies close to the performance of sc_{iso} models that were trained at the true angular source distribution and SNR. Note that this is the case even though most of the azimuth configurations from set 2.a that have been used for testing have *not* been in the training set for the mc_{ps} models, which underlines the strong generalization of these models. All models except the mc_{ps} models exhibited a higher median performance on the two-channel feature set than on the mean-channel feature set.

Fig. 4.13 shows how the mc_{amb} and mc_{ps} models compare to single-conditional *iso-azimuth* models that were trained at specific SNRs and tested at all SNR conditions. It shows the average balanced accuracy as a function of the SNR for tests using the mean-channel feature set (a), and the two-channel feature set (b).

We observe the following effects:

- The performance of the mc_{ps} model always lies close to the *iso-SNR/iso-azimuth* SC performance, and even exceeds it on the mean-channel feature set for low SNRs.

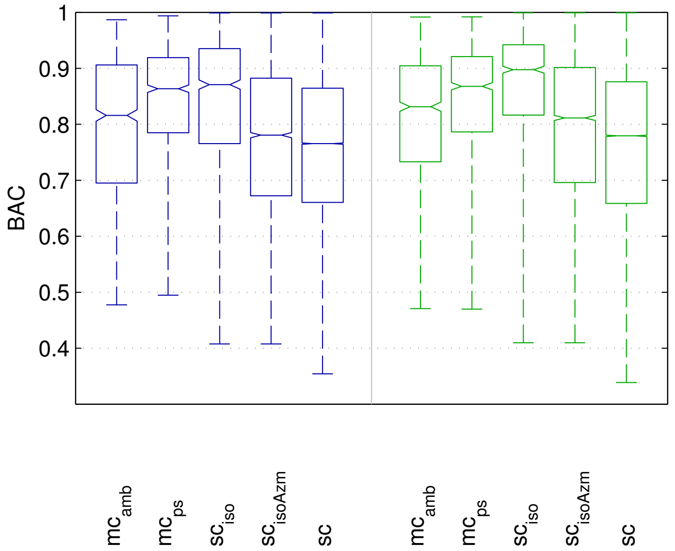


Figure 4.12: Performances of 5 different model/test-paradigms for each feature set (mean-channel left, two-channel right), each box pooling over the SNRs $\{inf, 10, 0, -10, -20\}$, all corresponding iso- and/or cross-azimuth configurations, all data set splits and sound classes.

Obviously, when not being able to use the features of both channels to better separate spatially distributed sources (in the case of two-channel feature set the single conditional models perform better in iso-SNR/-iso-azimuth tests for all SNRs), the mc_{ps} models generalize better to low SNRs than the specialized sc_{iso} models.

- The mc_{amb} models do not quite reach the performance of the *iso-SNR/iso-azimuth* SC models, but outperform many *cross-SNR/iso-azimuth* models. This is without making use of any directional information during the training phase.

Inspecting the SC models again, we see that the performance curves of the sc_{inf} models are convex, whereas the curves of other SC models are mostly concave. A possible explanation for this finding is that the models trained on clean sounds have not learned to accommodate the presence of simultaneous distractors at all, and are more strongly affected even by small amounts of noise than the other SC models. For these other SC models, we observe saturation effects at SNRs that are higher than the SNR at which they were trained. One reason for this could be that these models specialize on features that do not not get more discriminative with increasing SNR.

In summary, our results show that multi-conditional training by including different SNRs in the training data produces robust models that perform well over a wide range of SNRs. Although this also works if training is conducted with ambient sources, close-to-optimal performance can be achieved when the multi-conditional training is conducted

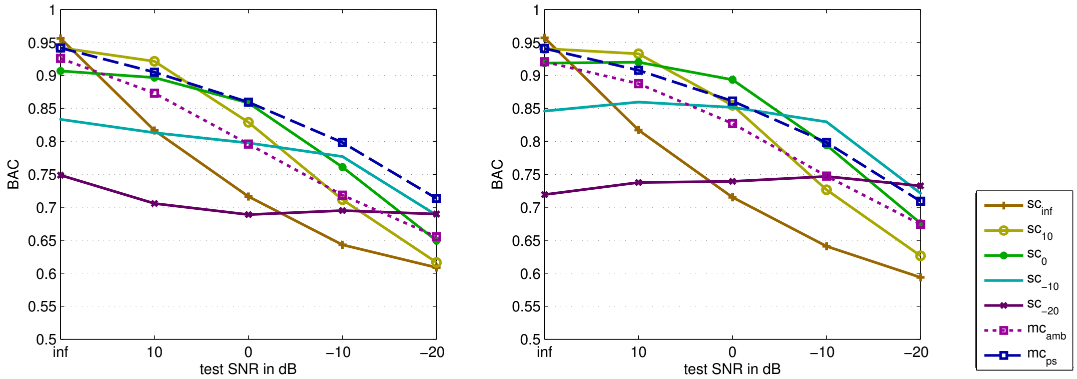


Figure 4.13: Performance of different model groups at particular SNRs, averaged over all (iso, for *sc* models) azimuth configurations, data set splits, and sound classes. Left: mean-channel feature set, right: two-channel feature set.

on point sources varying not only in SNR but also including multiple angular source configurations. We can thus conclude that multi-conditional point-source (mc_{ps}) models are the best choice for practical applications: only a single model needs to be trained for each target sound class, inferring the conditions *a priori* is not required, and the resulting models are robust with respect to scene conditions and even reach a close-to-optimal performance.

4.3.3 Robust detection models trained with BRIRs

The detection models as evaluated in section “Multi-conditional in comparison to single-conditional detection models” suggest a method to be able to identify sound events even in adverse, changing and differing from the training conditions. However, the system was still restricted to anechoic scenes with two sources. We thus also considered a more challenging acoustic condition, and built an identification system trained and evaluated on reverberant scenes with up to four sources plus up to one diffuse white noise source.

ADREAM apartment multi-conditional data

The general method concerning data generation, feature extraction, and model training stayed the same as described in sections 3.3.3, 3.4.2 and 4.3.2. Binary sound identification models were trained for each of the 13 classes found in the NIGENS database. Instead of the so far used anechoic scenes, for these models we used the ADREAM apartment scenes through applying the respective BRIRs in the binaural simulator. Figure

4.36 shows the layout of these scenes with the different available positions of head and sources.

We generated a set of 100 scenes, one half of type (5) (without diffuse white noise) described in Sec. 3.4.2 and one half of type (6) (with diffuse white noise). The number of sources was distributed equally across these scenes, accordingly resulting in 25% with each condition (one to four sources). For all scenes, we randomly chose

- the head position (uniformly from one to four),
- the head's orientation (uniformly from all measured BRIR's orientations)
- the point source positions (uniformly from one to four),
- the SNRs between sources (uniformly between −20 and +20),
- and, if applicable, the diffuse white noise's SNR (uniformly between −5 and +20).

The random sampling of the scene parameters was due to the explosion of the parameter space because of it's increased dimensionality, and inspired by Bergstra and Bengio (2012) where the authors show that it is more efficient to randomly sample a large hyperparameter space than doing grid search or even careful manual search.

Since the evaluation of section 4.3.2 showed that with the multi-conditonal point source models the two-channel feature set did not yield advantages, we used the mean-channel feature set as defined in 3.4.2 only in this study. We tested three variants:

$fc2$ Based on ratemaps with 24 frequency channels, spectral features based on 32 frequency channels (both framed in 25 ms time bins), and amplitude modulation spectrograms with 128 ms time frames over 16 frequency channels with 8 modulation filters.

$fc3$ Same as $fc2$, but additionally based on Gabor filter responses (calculated from 24 frequency channels), framed in 25 ms time bins.

$fc3_{1s}$ Same as $fc3$, but constructed from 1.0 s blocks instead of the default 0.5 s.

Regardless of the number, frequency channels always ranged from 80 Hz to 8 kHz.

Anechoic versus ADREAM schemes

First, we asked whether results from the anechoic two-source multi-conditional system carry over to the realistic reverberant environment with up to four sources. To examine this matter, we compared multi-conditional models of the alarm, baby, female and fire classes, (a) trained in the mc_{ps} scheme described in Sec. 4.3.2 and (b) trained in the ADREAM scheme with $fc2$ as described above. In Figure 4.14, the mean balanced accuracy (averaged over classes) of both systems is plotted over SNR. Because of the many different and randomly sampled parameters in the ADREAM scenes, the ADREAM model's curve is a running average over 20 scene configurations sorted by "aggregate" SNR, defined as the power ratio between the source emitting target sounds and all others, including diffuse white noise. It is obvious that our results from the anechoic case, even though considered "easier" in several aspects, match the results with the ADREAM models quite well – not only qualitatively, but even quantitatively. This affirmed our assumption that reverberation (at least for the condition here) does not substantially influence the identification of objects, and that the random sampling of scene parameters serve the purpose of multi-conditional training well.

Evaluating multi-conditional models in the ADREAM scenes

Since the approach of multi-conditionally trained models proved to still be valid under realistic conditions, we went on to compare the three different feature sets described above and describe the behavior of the system under different conditions. For all following results, models targeting all 13 NIGENS classes have been trained and tested.

Figure 4.15 shows the mean balanced accuracies, sensitivities and specificities of our models trained on the ADREAM scenes and their dependencies on aggregate SNR (as defined above) (left panel) and number of sources (right panel). The three different feature sets are plotted in different colors. The following can be observed:

- The performance difference between the feature sets are small but consistent. The $fc3_{1s}$ models perform best and $fc2$ models worst (considering the small difference, least best may be more suitable). The average advantage over all scenes of $fc3_{1s}$ models over $fc2$ models is 0.0123, and with the difference between the $fc2$ and $fc3$ models being the smaller part of this advantage, we conclude that the effect is mainly due to the longer blocksize and not due to increased information with the Gabor features. The graphs show that the longer blocks yield a higher specificity throughout all SNR or number of sources levels, and help with finding targets (sensitivity) only for lower SNRs.

- Sensitivity shows a very strong and unambiguous dependence on SNR. It peaks at about 95 % detected targets, stays almost uninfluenced until about 7 dB, and below that starts to fall into a linear descend of about 10 % per 5 dB.

- The specificity is more variable with regard to SNR; it seems to be more directly dependent on the number of sources – in the right plot, specificity and number of sources show very proportional decline. In any case, the accuracy of labeling blocks without target event as negatives is much more stable than the sensitivity.

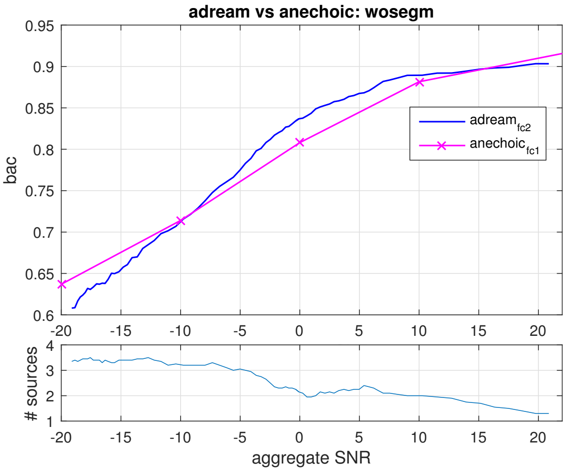


Figure 4.14: Mean performance of detection models built with anechoic vs ADREAM schemes (averaged over sound types alarm, baby, female speech, fire), depending on SNR in dB between source emitting target sounds and all other sources (determined at ears). The "adream" curve is constructed as running average over 20 scene configurations sorted by aggregate SNR, "anechoic" curve shows average over 16 scene configurations at particular SNRs (-20, -10, 0, 10, inf). Bottom panel shows average number of sources in the ADREAM scenes (excluding diffuse white noise) depending on the aggregate SNR.

Observations two and three are intuitive: whether an event is detected strongly depends on the energy this event exhibits compared to all other sources. Whether false positives arise, strongly depends on how many sources are active and thus how many different sounds are emitted concurrently. However, as can be noticed in the lower panels, average SNR and average number of sources correlate and are not independent. To validate the interpretation above, we thus additionally plotted mean performances over SNR and number of sources *conditioned* on particular values or ranges of the correlated variable. The upper row in Figure 4.17 shows the graphs with number of sources fixed to two (left) and three (right), and the bottom row shows the graphs with the SNR fixed to ranges of -10 dB to 0 dB (left) and -5 dB to $+5$ dB (right). Indeed, our observations are confirmed here: with fixed number of sources, specificity shows very small dependence on SNR. With SNR fixed in a smaller range, sensitivity exhibits small (and ambiguous) dependence on number of sources.

Finally, we investigated the individual classes' performances of the $fc3_{1s}$ models, Figure 4.18 showing the results in the same form of graphs as above and Fig. 4.16 a version with boxplots summarizing all scene results per class. For a placement of the median performance values shown in the boxplots, consider that the median SNR over all 100 scenes was -2.2 dB. Knock models' performance is superior to all others, followed by male and female speech and then fire, dog, footsteps, and phone. The one class standing out negatively is crash with a median far below all others. Having a closer look at sensitivities and specificities, we

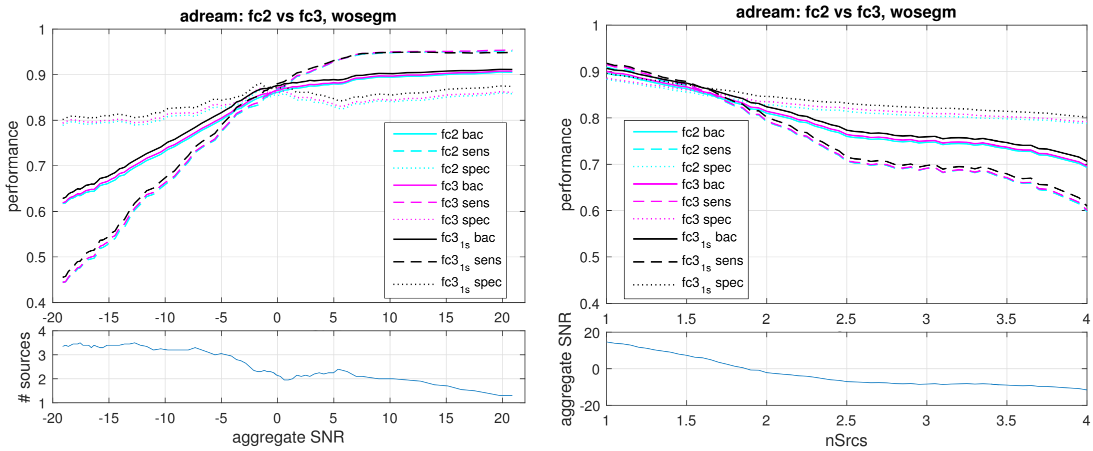


Figure 4.15: Mean performance of detection models built with different feature sets (averaged over all 13 sound types). Curves are constructed by running averages over 20 scene configurations sorted by aggregate SNR. Left panel shows performance depending on aggregate SNR in dB between source emitting target sound and all other sources (determined at ears), right panel shows performance depending on average number of sources (excluding diffuse white noise).

note the following: crash, engine, baby and fire seem to be types that more often produce false positives, that is, they seem to be less specific. Crash, alarm, baby and scream are the classes that are least easiest to detect.

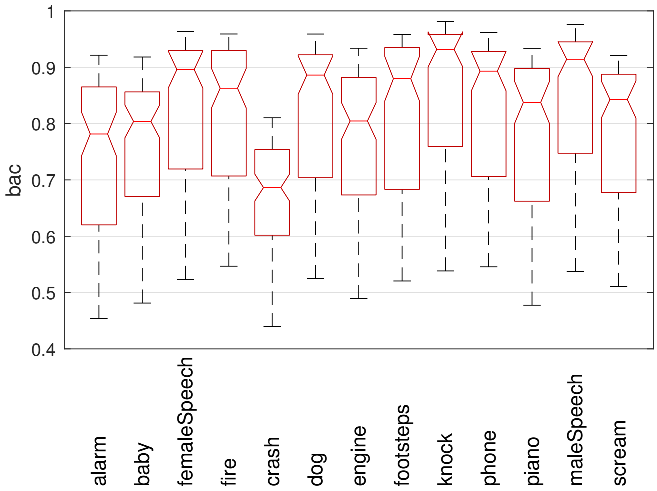


Figure 4.16: Performance of detection models showing differences between sound types. Boxplots include performances of all 100 scene configurations, showing the median (central line), max and min values (upper and lower whiskers), 25% and 75% percentiles (lower and upper end of boxes).

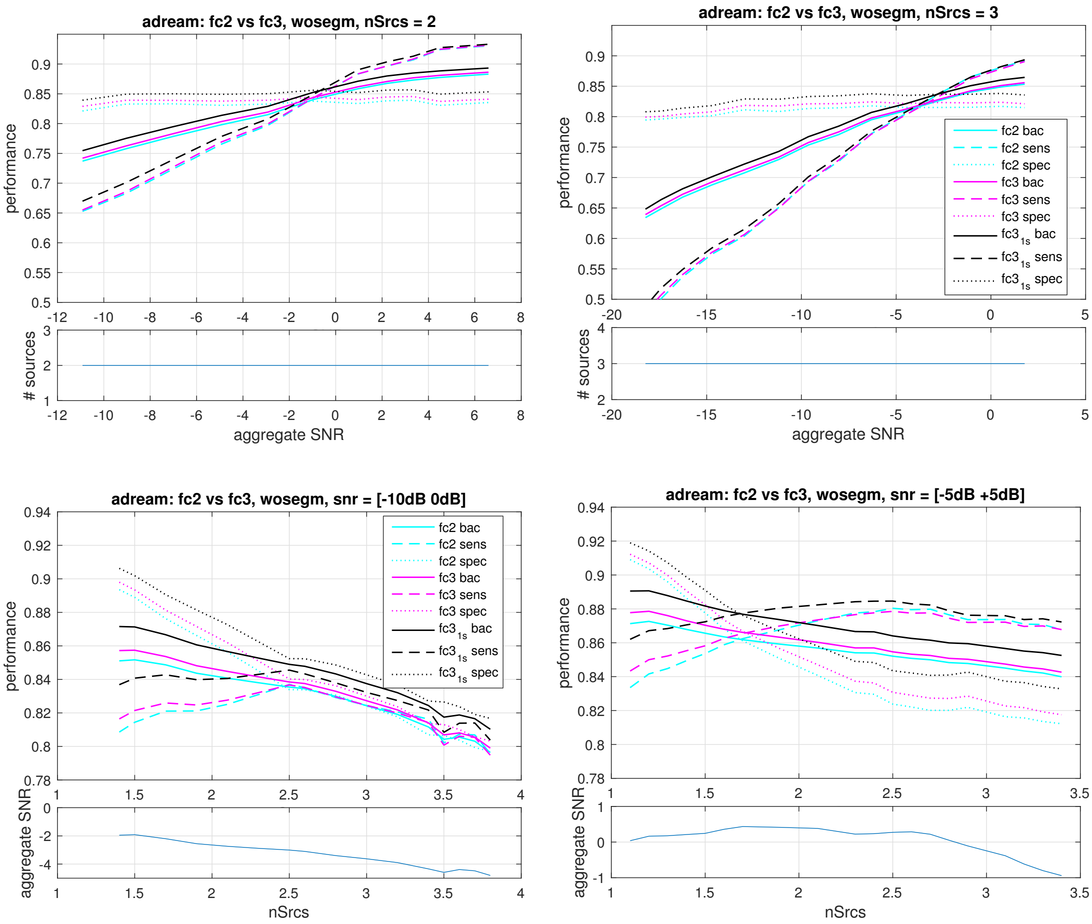


Figure 4.17: Mean performance of detection models built with different feature sets (averaged over all 13 sound types). Curves are constructed as running averages over 10 scene configurations sorted by aggregate SNR (top row) and number of sources (bottom graphs). In the top row, only scenes with two (left panel) and three (right panel) sources are evaluated; and in the bottom row, only scenes with aggregate SNR between −10 dB and 0 dB (left panel) and between −5 dB and +5 dB (right panel) sources are evaluated.

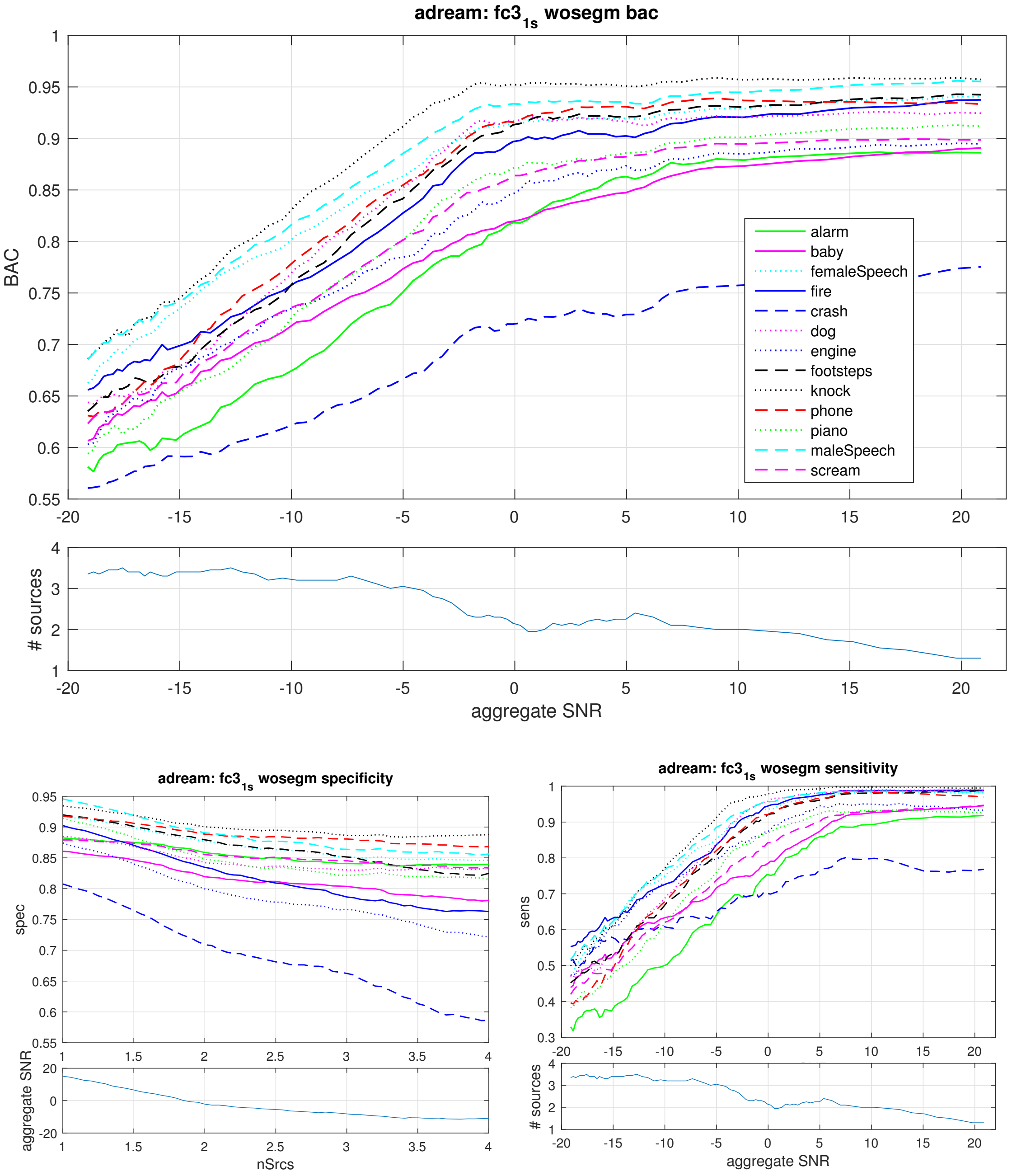


Figure 4.18: Balanced accuracy (top), sensitivity (bottom right) and specificity (bottom left) of detection models depending on aggregate SNR in dB between source emitting target sound and all other sources (determined at ears). Curves are constructed as running averages over 20 scene configurations sorted by aggregate SNR. Bottom panel shows average number of sources (excluding diffuse white noise) depending on the aggregate SNR.

4.4 Speaker recognition

In this section, we evaluate our speaker recognition system described in Section 3.4.3 on the development system.

4.4.1 Experimental setup

We used speech materials from the GRID corpus (Cooke *et al.*, 2006b) for evaluation. The corpus consists of 1000 simple command-like sentences spoken by each of 34 speakers. There are 18 male speakers and 16 female speakers in the corpus. The monaural speech signals were spatialised by convolution with a set of BRIRs measured with a KEMAR dummy head in the ADREAM apartment at the LAAS in Toulouse (See Deliverable D1.3). The set of BRIRs consists of four listeners positions and at each listener position BRIRs were recorded for four fixed source positions (Fig. 4.36).

For each of the 34 GRID speakers, and each of the 16 BRIR conditions, 20 different sentences were randomly selected from the GRID corpus. Three quarters of the signals were used as the training set (320 sentences per speaker) and the remaining signals were used as the test set (80 sentences per speaker).

4.4.2 Feature extraction

The auditory front-end in the Two!Ears system was used to extract ratemap features from a bank of 32 gammatone filters with centre frequencies equally spaced on the ERB scale between 80 and 8000 Hz (Wang and Brown, 2006). Ratemaps are spectral features that represent a map of auditory nerve firing rate (Brown and Cooke, 1994b). In the binaural setting of Two!Ears, ratemaps were extracted for both ears. The *average* ratemaps were constructed by taking the average ratemap value of each pair of time-frequency cells between the two ears, before compression. The *binaural* ratemaps were constructed by concatenating the left ear and right ear ratemaps to form 64D feature vectors. The features were then log-compressed.

We also estimated the first derivatives of each ratemap feature dimension as a way of converting dynamic, contextual behaviour into a relatively fixed point in the new feature space. Delta features were computed using a 7-frame window (10 ms frame rate) and they are used to supplement the static ratemap features.

Many state-of-the-art speaker recognition systems employ MFCC features. To make the ratemap features more orthogonal, we also tested similar features by applying a discrete cosine transform (DCT) to the ratemap features, known as the gammatone filter cepstral coefficients (GFCCs). The first 12 cepstral coefficients were used together with the energy term, producing a 13-dimensional feature space. Cepstral mean normalisation was then applied. The average GFCCs were computed from the average ratemap features and the binaural GFCCs were constructed by concatenating GFCCs of individual ears. Similar to the ratemap features, the first and second derivatives of the GFCCs were computed and used to supplement the static GFCCs, producing final 39-dimensional feature vectors.

4.4.3 Model setup

The UBM was trained using the training set from all the 34 speakers using a 64-component GMM. For the GMM-UBM model, each speaker model was then produced by adapting means, variances and mixture weights from the UBM using the MAP relevance adaptation. The i-vector model used the same UBM to train the total variability subspace (100-dimension) and extract the sentence i-vectors. Final model i-vectors were obtained by taking the average of the extracted i-vectors. A Gaussian PLDA model was trained with the extracted i-vectors to score the testing trials.

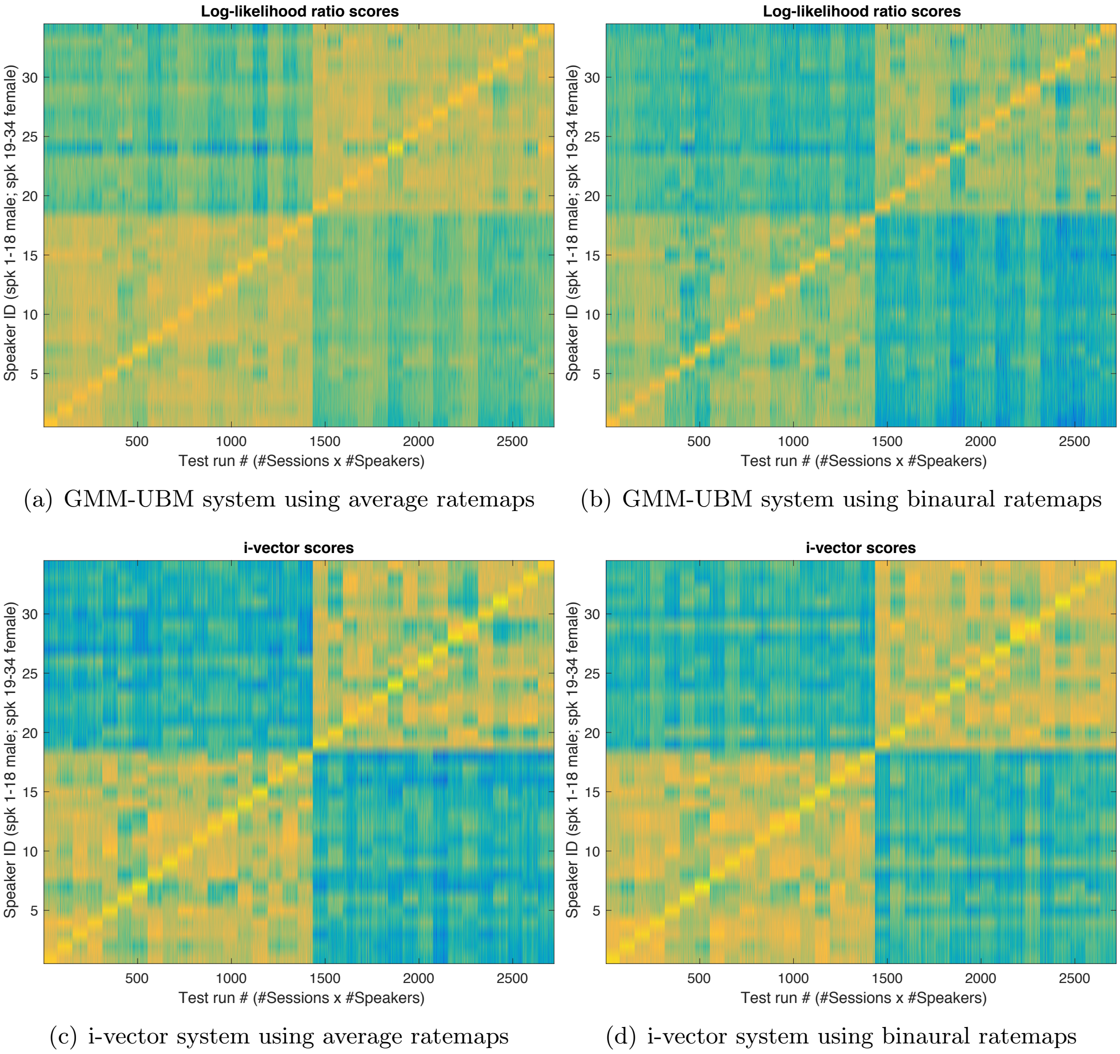
4.4.4 Results

The speaker recognition accuracies of both models using various feature sets were listed in Table 4.2. Both models produced very good accuracies for this simple speaker recognition task. For the GMM-UBM system, the best performing features are the binaural ratemaps while for the i-vector system the best performing features are the average ratemaps. The GFCC features produced inferior results for both systems when compared to ratemap features, especially for the i-vector system. This is likely due to an optimisation issue as we did not extensively optimise the parameters of the models.

Table 4.2: Speaker recognition accuracies (%) of various speaker recognition systems.

Model	Ratemap		GFCC	
	Average	Binaural	Average	Binaural
GMM-UBM	98.38	99.08	98.24	98.57
i-vector	**99.15**	99.08	96.80	95.11

Figures 4.19 and 4.20 show the log-likelihood ratio scores of all speaker models plotted as confusion matrices for various systems. In each figure, the diagonals (bottom left to top right) show correct speaker model. The speaker ID axis is organised in such a way that the 18 male speakers were in a sequence before the 16 female speakers. It is clear that among all the systems, the speakers of the same gender tend to have more



(a) GMM-UBM system using average ratemaps

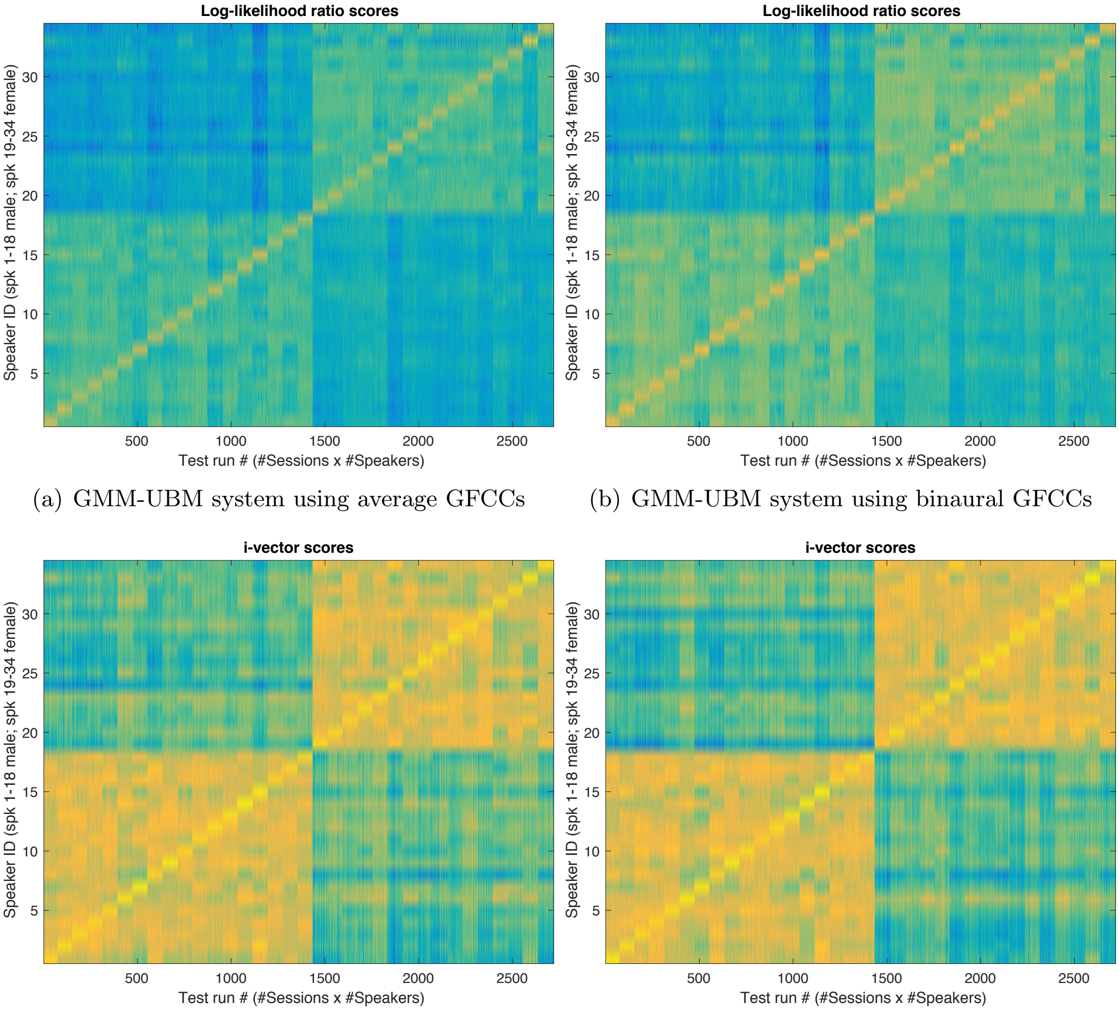
(b) GMM-UBM system using binaural ratemaps

(c) i-vector system using average ratemaps

(d) i-vector system using binaural ratemaps

Figure 4.19: Log-likelihood ratio scores of all speaker models plotted as confusion matrices for various systems using ratemap features.

confusions than those of the opposite gender. Visually, the i-vector system using average ratemaps (Figure 4.19c) shows the largest contrast between the scores of the correct speaker model and the other speaker models. The i-vector systems using GFCC features show more confusions among the speakers of the same gender, partly explaining their poorer recognition performance.



(a) GMM-UBM system using average GFCCs

(b) GMM-UBM system using binaural GFCCs

(c) i-vector system using average GFCCs

(d) i-vector system using binaural GFCCs

Figure 4.20: Log-likelihood ratio scores of all speaker models plotted as confusion matrices for various systems using GFCC features.

4.5 Joint Identification and Localisation

The sections above in this chapter, 4.2 and 4.3, presented the evaluation results of the independently from each other working localization and identification systems. As described in section 3.4.4, we additionally built systems to *jointly* determine location and type of sources in order to be able to form auditory objects. In the following two sections, we evaluate those systems.

4.5.1 Auditory object identification on segregated streams

Building onto the successful construction of robust multi-conditional detection models operating in realistic environments (see Sec. 4.3.3), we designed a system to identify localized sources, detailed in Sec. 3.4.4, and depicted in Fig. 3.10. We trained this system using the ADREAM scenes with the methods elaborated on in Section 3.3.3. Feature set $fc2$ (defined in Sec. 4.3.3) was used with this system, as the Gabor features employed in $fc3$ were not segregatable with the time-frequency soft-masks produced by the segregation model.

We present results concerning three questions in this evaluation: (i) can the detection models trained on the "full' ear-signals stream (Sec. 3.4.2) be applied to the segregated streams directly, (ii) how does unprecise estimation of active number of sources influence the system, and (iii) how does the joint system perform in comparison to the full-stream identification system?

Figure 4.21 gives an answer to question one by showing the performances of (a) our identification system trained on unsegregated streams but applied to the segregated streams, versus (b) our identification system trained on segregated streams and tested on segregated streams. The models trained on unsegregated streams cannot deal well with the segregated streams' data and show a strongly degrading sensitivity with increasing number of sources (i.e. with segregating into more streams).

Due to the fact that the number-of-active-sources estimation model (see Sections 3.5.2 and 4.6) was not ready to be used at the time of building the segregated-streams identification models, we used ground truth for this variable in training. However, when this estimator became available for use, we *tested* the trained system with it, hence evaluating how the uncertainty in the estimation of the number of active sources impacts the identification of the segregated streams. Figure 4.22 depicts (a) the mean performance of segregated identification models with number-of-sources ground truth and (b) the mean performance of segregated identification models with estimated number-of-sources. Since the two curves almost coincide, we conclude that the estimated number of active sources provides a well-behaving input to the segregation system and accordingly, that the segregated identification

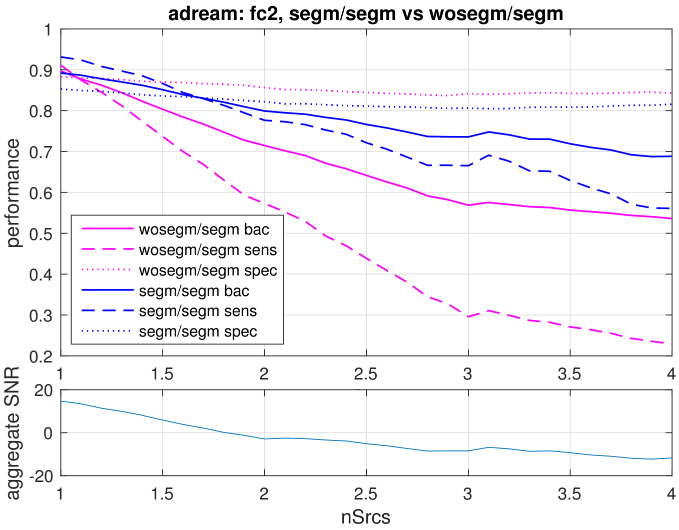


Figure 4.21: Mean performance of detection models (averaged over all 13 sound types) tested on segregated streams but trained on unsegregated streams versus trained on segregated streams, depending on average number of sources (excluding diffuse white noise). Curves are constructed as running averages over 20 scene configurations sorted by number of sources. Bottom panel shows aggregate SNR depending on average number of sources.

system can be applied with good performance expectations under realistic conditions without *a priori* knowledge of the number of sources being active.

We finally compared the performance of the segregated identication system to the performance of the full-stream identification system, and display the results in Figure 4.23 averaged over classes with dependency to scene conditions, and in Figure 4.24 for the individual classes, summarized over scenes. As clearly notable from the first figure, the two systems exhibit almost the same performance curves as a function of the number of sources. Investigating the performance differences between the two systems on a class-wise basis, we observe that for most classes, the identification success is more or less independent of the two systems, with the particular exception of the alarm class, which seems to suffer from segregation more than the others, and the class phone, which seems to profit from segregation.

Concluding, we can say that our segregated identification system detects sound events almost as well as the full-stream one, however in addition it can attribute the event's type also to a source location and hence provides an important step in forming coherent auditory objects.

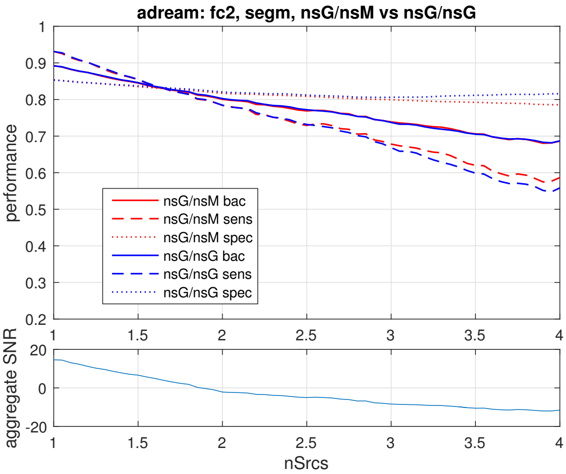


Figure 4.22: Mean performance of detection models trained on segregated streams (averaged over all 13 sound types) tested one time on streams segregated with number of sources ground truth and one time on streams segregated with number of sources estimated by a model, depending on average number of sources (excluding diffuse white noise). Curves are constructed as running averages over 20 scene configurations sorted by number of sources.

4.5.2 Joint identification and localization with Convolutional Neural Networks

Multiple Convolutional Neural Networks (CNNs) were trained to perform joint bottom-up identification and localization. The following presents the key experiments conducted on multiple networks whilst investigating the effect of network design decisions on the network's performance on either task.

Fig. 4.25 visualizes the response of a network for different input blocks sampled randomly from the test set. The response consists of multiple vectors. One vector for each class and each node representing the probability of the respective sound type being located within an azimuth bin. The probabilities of all nodes are all independent. The network's ability at solving the joint identification and localization task is evaluated by measuring the classification performance of each node in the response matrix over the test set composed of 100 configurations from the ADREAM scenes.

To account for the large imbalance that each node encounters during training and testing we continue to use balanced accuracy (BAC) as defined in 4.1. Given the much steeper imbalance encountered by the localization node, we add the $F1$ score as an additional measure to increase the resolution over a node's performance. The high imbalance also requires the selection of a threshold at which we draw a decision from the sigmoidal response of each node (i.e. selecting the classifier's operating point). The BAC is computed for each

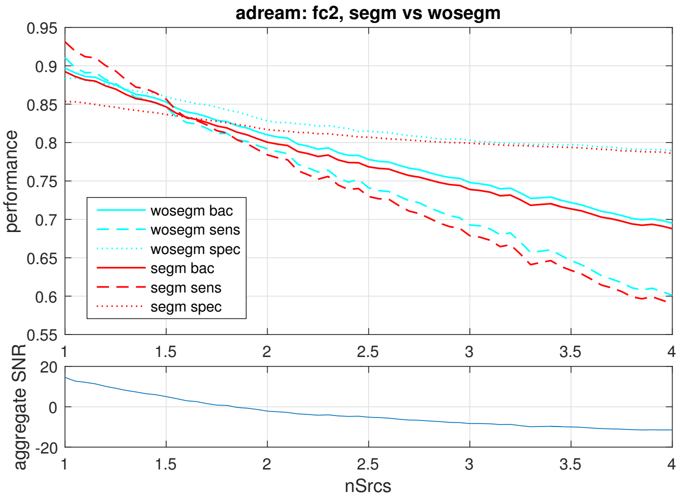


Figure 4.23: Mean performance of detection models (averaged over all 13 sound types) working on segregated versus unsegregated streams (training and testing), depending on average number of sources (excluding diffuse white noise). Curves are constructed as running averages over 20 scene configurations sorted by number of sources. Bottom panel shows aggregate SNR depending on average number of sources.

classifier at different thresholds. For each classifier the threshold value that maximizes its respective BAC is selected for further evaluation as well as deployment. In 4.5.2 the evaluation of a single output node is presented ncluding the procedure for selecting its operating point. 4.5.2 builds on this measurement and summarizes the performance of the models for the identification and localization tasks by pooling over the performance measures calculated for their respective classifiers. This evaluation in 4.5.2 is performed on models trained using clean anechoic sounds. It is repeated for ADREAM scenes in 4.5.2.

Networks were trained using mini-batch gradient descent with a batch size of 512, learning rate α=0.001, momentum μ=0.9 and weight decay=0.005.

Individual classifier performance measure

Each output node is a sigmoidal neuron. It can either be a neuron that indicates the presence of a particular sound type at a specific azimuth location, or a neuron that responds to the complete absence of that sound type from the auditory scene. To account for the imbalance between positive and negative examples that each output node encounters, we analyze the performance of each individual classifier in order to select its operating point that maximizes its performance. The operating point is selected by generating the receiver operating characteristic (ROC) for each classifier and calculating the BAC at the threshold

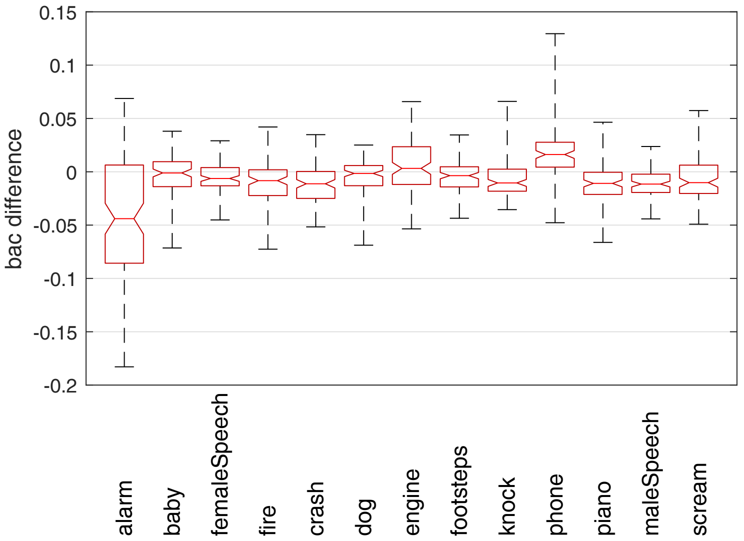


Figure 4.24: BAC differences between full-stream identification and segregated identification models. Boxplots include performance differences at all 100 scene configurations, showing the median (central line), max and min values (upper and lower whiskers), 25% and 75% percentiles (lower and upper end of boxes).

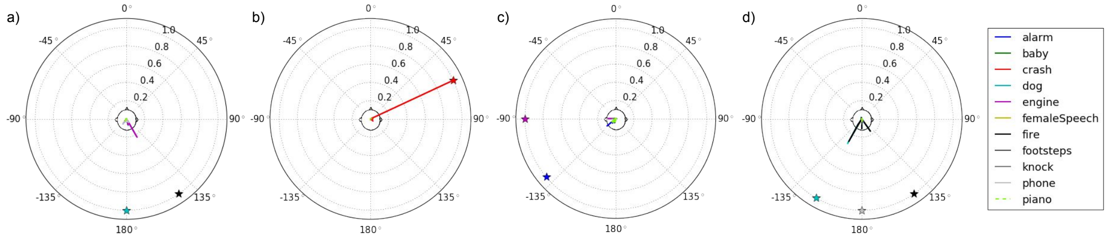


Figure 4.25: Examples of network responses. The color of a star represents the sound type. The stars are positioned at the ground truth azimuth. The radial access represents the raw probability produced by the localization nodes. a) two misidentified, correctly localized sources, b) correct identification and localization of a sound in a single source scene, c) correct classification and localization of sounds in a scene with two sources d) 3-source scene with partially correct and incorrect identification and localization estimates.

points used for generating the ROC curve. Figure 4.26 illustrates a ROC curve for a single classifier responsible for identifying the presence of alarm sounds in a scene. The threshold that maximizes BAC along the ROC curve is selected as the classifier's operating point. For purpose of illustration, we select here the classifiers that respond to the absence of a sound type from a scene, or presence of the sound type, if we choose to invert the response. Figure 4.27 illustrates an example of analyzing the ROC curves of sound type identification neurons, the corresponding BAC along this curve and the threshold that maximizes each individual classifier's BAC. Classifiers for localizing a sound type at a particular azimuth are treated the same way.

A model's sound type identification performance is assessed by pooling the results of its identification classifiers' maximum BAC. The model's localization performance is measured by pooling over the maximum BAC of its localization nodes.

Performance on clean sounds

The set of models was trained on the clean sounds dataset to assess the effects of (i) feature groups (i.e. ratemap, AMS, or both) and (ii) the effect of the number of decision layers have on this joint task.

dataset	features	left-right convolutions	decision layers	model code
clean anechoic	ratemap	combined	2	rmCombLR2out
clean anechoic	ratemap	combined	4	rmCombLR4out
clean anechoic	ratemap	combined	5	rmCombLR5out
clean anechoic	ratemap	separate	1	rmSepLR1out
clean anechoic	AMS	separate	1	amsSepLR1out
clean anechoic	ratemap AMS	separate	1	rm+amsSepLR1out

Figure 4.28(a) depicts model identification performance on clean anechoic sounds. The maximum BAC of identification nodes are pooled together for each network. Both tasks benefit from increased model complexity. However, the variance of performance between different classes increases as well. As expected, the localization task becomes easier than the identification task when only a single or no source is active at a time. Keeping two separate tracks for the left and right channels with shared convolutional weights is inferior to feeding the channels directly into the first convolutional layer. The networks with combined left and right input seem to resolve channel invariance implicitly. The need for designing this explicitly into the architecture by re-using the weights of left and right convolutions tracks becomes unnecessary. The overall performance for the localization task improves when using both rate maps and AMS than only using a single AFE representation. However, the combination of both AFE representations leads to an overfitting for the identification task.

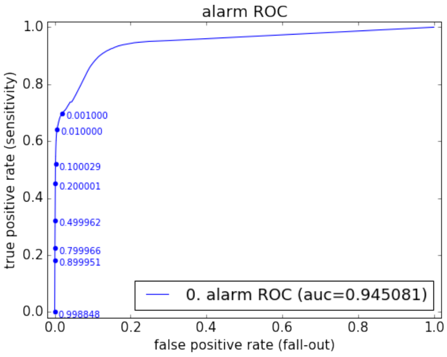


Figure 4.26: Example of an ROC curve with markings of the different thresholds.

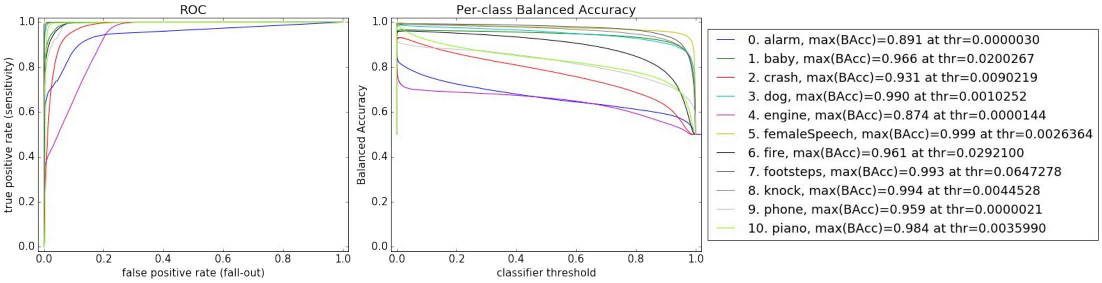
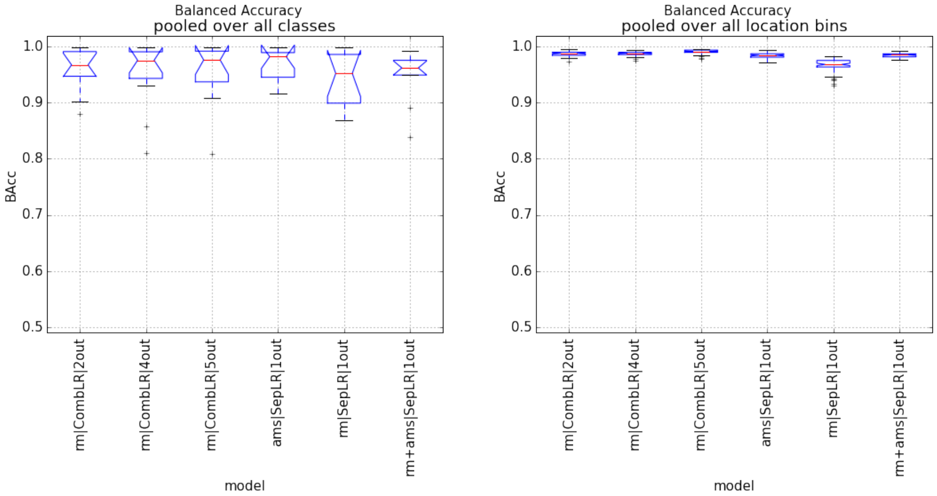


Figure 4.27: Example of ROC curves (left) measured for identification neurons and resulting BAC curves (right). Per-classifier thresholds that maximize BAC are depicted in the legend.

Performance on ADREAM scenes

The set of models was trained on ADREAM scenes with up to 4 multiple active sources to assess (i) the effects feature groups (i.e. ratemap, AMS, ILD, or all) have on this joint task as well as (ii) the number of convolutional stages and (iii) the number of decision layers. All models employ a combined convolution of AFE representations.



(a) identification

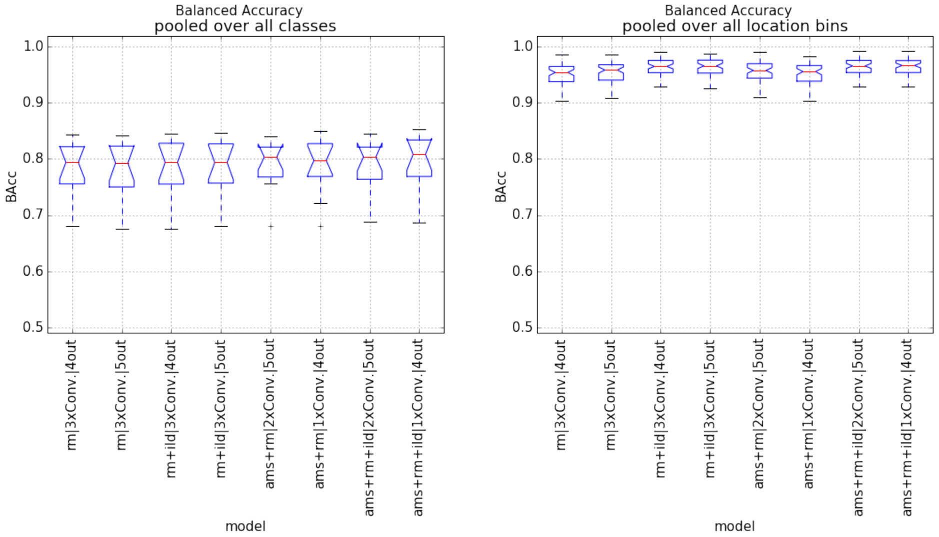
(b) localization

Figure 4.28: BAC of identification and localization nodes

dataset	features	conv. stages	decision layers	model code
ADREAM	ratemap	3	4	rm3xConv4out
ADREAM	ratemap	3	5	rm3xConv5out
ADREAM	ratemap AMS	2	5	rm+ams2xConv5out
ADREAM	ratemap AMS	2	4	rm+ams2xConv4out
ADREAM	ratemap AMS ILD	2	5	rm+ams+ild2xConv5out
ADREAM	ratemap AMS ILD	1	4	rm+ams+ild2xConv4out

Figure 4.29(a) depicts model performance on BRIR sounds in the ADREAM apartment. The decision threshold for each sigmoidal output node is calibrated to maximize BAC. The performance of nodes for each task are then pooled together to compare different networks with each other. We observe that performance for either task is insensitive to the network complexity, except that inter-sound-type variance is least for the deepest network which uses both AMS and ratemap, two convolutional stages and 5 decision layers. We conclude this for either task. Performance is affected strongly by the type of AFE representation that

is fed into the network. Adding ILD yields significant improvement on a network's ability to localize multiple sources. Including AMS benefits the identification task by decreasing the variance in inter-class performances. ILD provides the strongest indicator for source location. Without the ILDs, the network is still able to find cues of inter-channel differences within the ratemap or both the ratemap and AMS that correlate with location. However, the normalization of the left and right channels for these representations would render localization solely based on ratemaps and/or AMS more difficult. A possible explanation is that the cues selected by the network are due to the impulse response shaping the frequency bands of both left and right channels differently.



(a) identification

(b) localization

Figure 4.29: BAC of identification and localization nodes in ADREAM scenes

An additional test separates samples from the test set according to the number of active sound sources. Results for testing on these meta-conditions are presented in Fig. 4.30. At first we confirm an imbalance of data points for each condition and that performance correlates with this to some degree which could possibly indicate that the networks picked up on this bias. The first condition with zero number of sources only reflects specificity. This condition simply does not contain any positive examples. Together with the next condition, in which a single source is present, both conditions make out the "clean" subset of the test set. Although a direct comparison is not completely justified, we notice that performance is lower than that observed on networks trained to specifically on clean data in Fig. 4.28(a). Observing that the relation between model performances across conditions is similar is an indicator that different models did not specialize to any of these 5 conditions. A similar observation can be made in Fig. 4.31(a), where conditions are sorted according to aggregate SNR instead of number of sources directly. Here we find that the performance of all networks is nearly constant until it drops below an aggregate SNR of 5 dB and drops almost linearly with decreasing SNR. This gives us additional resolution over which conditions are more favorable for which networks. In particular we observe how networks employing ILDs have a slight advantage over the other networks in the < -10 dB range.

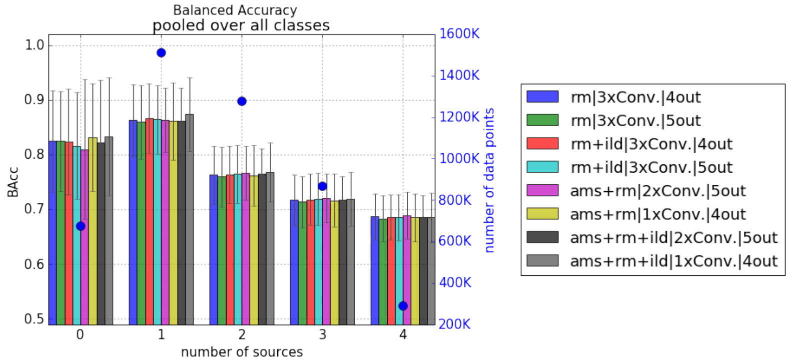


Figure 4.30: BAC of identification nodes in ADREAM scenes with different number of active sounds. The dots indicate the number of data points for each condition

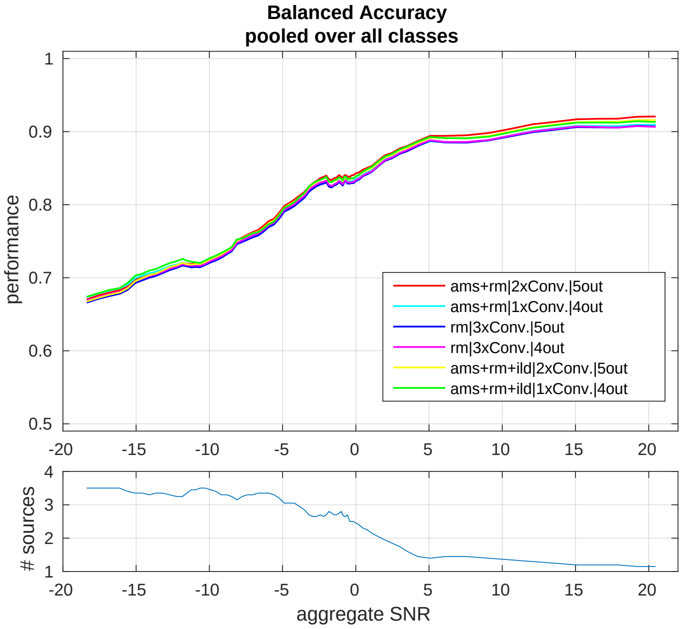
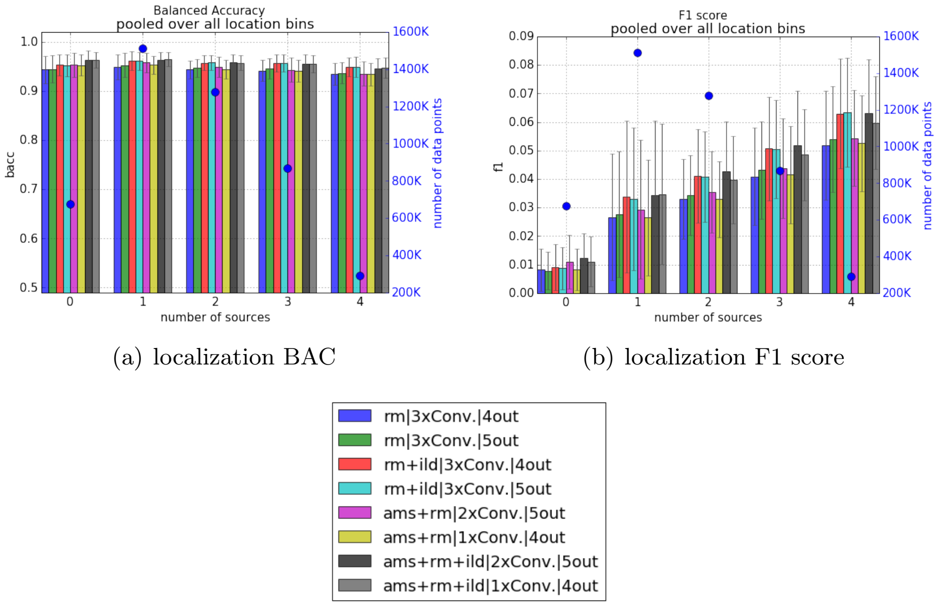


Figure 4.31: BAC of identification nodes in ADREAM scenes with respect to aggregate SNR. Curves are constructed as running averages over 20 scene configurations sorted by aggregate SNR. The bottom panel shows the average number of sources (excluding diffuse white noise) depending on the aggregate SNR.

Splitting the test set into bins according to the number of active sources is repeated for the localization task. Here we find the F1 source to be a performance measure that is more sensitive to high imbalance encountered by the localization nodes. We no longer observe a bias with respect to the number of data points for the different conditions. Inter-model differences appear consistent across all conditions, which is also indicative that the different networks did not specialize themselves towards a particular condition. Finally the results reiterate the advantage of ILD for the localization task.



(a) localization BAC

(b) localization F1 score

Figure 4.32: BAC and F1 scores of localization nodes in ADREAM scenes with different number of active sounds. The dots indicate the number of data points for each condition

4.6 Number of sources

The following presents an evaluation of the models trained for estimating the number of sound sources within a scene. Auditory scenes were simulated using the BRIRs measured in the ADREAM apartment. The configuration of the auditory scenes used in the following experiments is described in Sec. 3.4.2.

The combined feature set described in Sec. 3.5.2 was trained using a penalized generalized linear model from the multinomial and Poisson families. The models were trained using the GLMNET package (Qian *et al.*, 2013) with a ℓ_1-Norm penalty (Lasso). A third model was trained from the Poisson family that additionally received input from identification

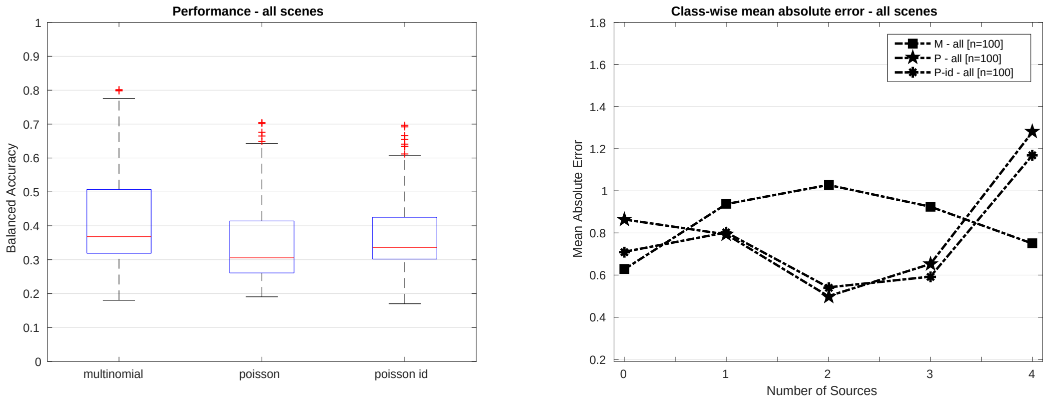


Figure 4.33: BAC_M (left) and MAE^l_μ (right) for all models and all scenes.

knowledge sources, as described in Sec. 3.3.3. Fig. 4.34 shows confusion matrices for all models and all scenes.

The chance level baseline for all scenes is 0.2 for the five classes representing zero to four sources present in the scene. The (macro) balanced accuracy[3] (BAC_M) for all scenes was determined at 0.4346 (median 0.3679) for the multinomial model, 0.3540 (median 0.3056) for the Poisson model and 0.3795 (median 0.3363) for the Poisson model with identification input (Fig. 4.33, left).

We further investigated the class-wise (micro) mean absolute error (MAE^l_μ)[4] for the models. The MAE^l_μ quantifies the extent of error per class, to better understand the errors made by the models. The BAC_M performance suggests a class-wise (macro) mean average error of > 0.5, which can be interpreted as estimates bleeding into the surrounding classes (Fig. 4.33, right). The MAE^l_μ will quantify the magnitude of that bleeding effect.

Although the multinomial model scores a higher BAC_M, and therefore makes fewer errors, the Poisson models have a lower error magnitude MAE^l_μ for one, two and three sources while also showing higher MAE^l_μ for zero and four sources. This effect is consistent for both models from the Poisson family. The addition of the identification input results in a slight quantitative improvement without qualitative change. The Poisson models have a tendency to underestimate for three and four sources, but have overall better calibration in the critical regime of one, two and three sources.

[3] BAC_M: (macro) balanced accuracy: $\frac{\sum_{i=1}^{L}\frac{tp_i+tn_i}{tp_i+fn_i+fn_i+tn_i}}{L}$, L the class count

[4] MAE^l_μ: class-wise (micro) mean absolute error: $\frac{\sum_{i=1}^{N} \mid \lfloor y^i_{pred} \rceil_l - y^i_{true} \mid}{N}$, N the sample count

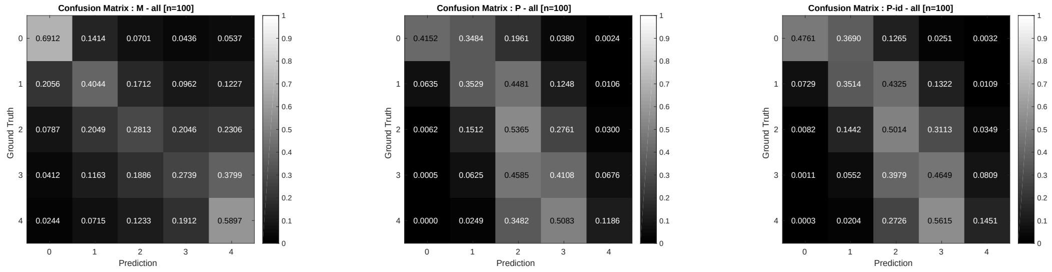


Figure 4.34: Confusion matrices for multinomial (left) and Poisson (center) and Poisson with identification input (right) models over all scenes.

Performance with respect to scene conditions

To increase the resolution of the evaluation we measured model errors under various scene conditions. The scene conditions under investigation were the presence and strength of noise, the mean absolute source SNR and the head position of the KEMAR head dummy in the scene. These scene conditions were analysed with the MAE^l_μ. Only those scenes where the condition under consideration was true were included in the respective analysis. Since both models from the Poisson family show qualitatively comparable results, we only show the Poisson model with identification input for this part of the analysis.

Noise condition. This condition stratifies for the presence, and if present, strength of an ambient white noise process in the scene. Scenes were roughly split in half by the noise condition. The scenes with an ambient white noise process were further divided according to the strength of the noise. (cf. Fig. 4.35, top row)

	"w/o noise"	"w/ noise"	"noise SNR≤0"	"noise SNR>0"
count	n=48	n=52	n=15	n=37

The ambient white noise process has a consistent impact on the prediction performance. It reduces the error for both zero and single source and only slightly confounds the predictions for two and more sources.

SNR condition. This condition stratifies for the mean absolute SNR[5] of sources in the scene. (cf. fig. 4.35, center row)

	"SNR 1"	"SNR 2"	"SNR 3"	"SNR 4"
count	n=24	n=25	n=25	n=25
interval	[0,6]	]6,11.2858]	]11.2858,16.2121]	[16.2121,29.2980]

[5] $maSINR$: mean absolute SNR: $\frac{1}{N}\sum_i^N |20\, log_{10} \frac{pr_i^{S_1}}{\sum_{j\neq i} pr_j^{S_1}}|$, $pr_i^{S_1}$ the power ratio w.r.t S_1

	"head pos idx 1"	"head pos idx 2"	"head pos idx 3"	"head pos idx 4"
count	n=20	n=18	n=28	n=34

The SNR condition rates the scene from easy (1) to hard (4). Most of the scenes in the "SNR 4" condition can be considered hard for human listeners. The error scales approximately with the difficulty of the SNR condition, the case of zero and one sources is especially affected.

Head Position. This condition stratifies for the position of the KEMAR head within the ADREAM apartment in the scene. There are four predefined positions at which the head can be positioned while also having an arbitrary azimuthal angle (see Fig 4.36). The azimuthal angle of the KEMAR head was not considered for this analysis. Position 1 is notable: this might be explained by the fact it is the position at which the sources are closest to each other with respect to azimuth, thus possibly creating less pronounced peaks in the used DUET and ITD/ILD histograms and accordingly rendering the problem more difficult. (cf. Fig. 4.35, bottom row)

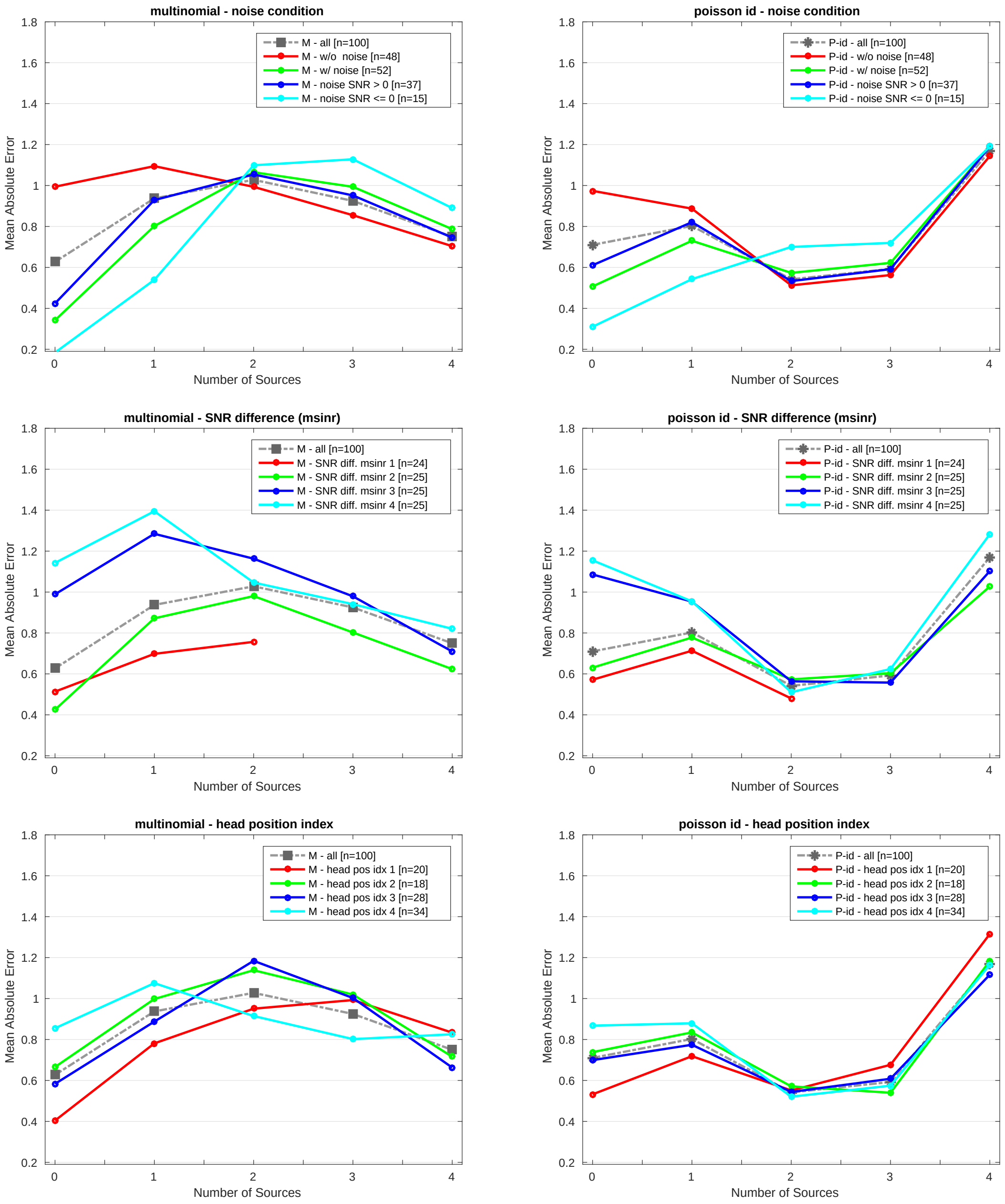


Figure 4.35: Mean absolute error per class for multinomial (left column) and Poisson with identification input (right column) under various scene conditions. The gray curve shows the error for all scenes for the respective model. The first row shows the error stratified by noise conditions. The second row shows the error stratified by SNR conditions. The third row shows the error stratified by KEMAR head position.

4.7 Evaluation on recorded scenes

Using the binaural simulator along with BRIRs enabled the generation of hundreds of realistic acoustic scenes in strongly varying configurations, as for example described in Section "Generation of auditory scenes", Sec. 3.4.2.

However, realistic synthesized scenes are not real scenes, hence we *recorded* scenes with the deployment system in the ADREAM apartment. The purpose of this data collection and the experiments built on top of them was to identify possible discrepancies between model performance measured on generated data and data recorded through the robot's ears. It also serves as a validation to how robust models are in less controlled environments and acoustic scenes with unfamiliar conditions.

Collecting these recordings allowed for a systematic evaluation of models by employing them in their respective knowledge sources in the blackboard-system (cf. Deliverable D6.1.3) and provided us overview over the models' (inter-) operation in the deployment system.

4.7.1 Setup

As audio source files, NIGENS sounds from the *test set* were used. The sounds included in this test set and hence in these recordings were a subset, but otherwise the same as used in evaluations on generated data, and had not been involved in model training. Audio files were normalized to an RMS of one and concatenated in time with a random gap of silence between $[200, 3000]$ ms length, and saved as a mono channel sequence audio file. Single sequence audio files as well as mixtures of up to four sequence files were put together, with each sequence occupying one channel. Short sequences were looped until covering a predetermined duration. In the case of mixtures, the power between channels was adjusted such that all channels had the same mean power. A chirp signal was prepended and appended to the mixture to help in the alignment of sound event annotations.

The scene was setup as depicted in Fig. 4.36 similar to the setup of the ADREAM BRIR recordings. Speakers were placed at positions one to four, volume being set to the same level for all speakers. Each channel of the sound sequence files was assigned to one speaker. Recordings were made while the robot was (i) positioned approximately at one of the four BRIR recording positions, (ii) within the adream apartment but not at a BRIR recording position, or (iii) moving about outside as well as inside the ADREAM apartment. Table 4.3 lists all recordings.

robot position	speaker 1	speaker 2	speaker 3	speaker 4	duration [mm:ss]
1,2,4	alarm				03:17,03:07*
2,3	alarm	general	footsteps	fire	11:29
1,2,4		baby			05:42,05:37*
1,2,5	baby	maleSpeech	femaleSpeech	scream	05:42
2			scream		02:48
1,2,3		scream		baby	05:42
moving,2,3,5			femaleSpeech		01:25
2			femaleSpeech	baby	05:42
2,4				fire	11:29,11:16*
moving,1,2,5	fire	alarm	baby	femaleSpeech	05:42
1,2				footsteps	05:13
2	general				24:25
1,2	general	femaleSpeech			01:25
2,2	general		femaleSpeech	maleSpeech	01:25
2,3		maleSpeech			01:16
2,3	maleSpeech			femaleSpeech	01:25
2			alarm	general	03:17
2		femaleSpeech	baby	general	05:42
2	baby	fire			11:29
1,2,3	alarm	baby	scream	fire	11:29
2	fire	alarm	scream		11:29
moving,4		fire	dog	baby	01:44*
4	baby			piano	02:58*
4			dog		01:44*
4	baby	fire	dog	piano	01:44*

Table 4.3: Description of the recordings denoting the robot's position and which sound type is played on which speaker for how long. *Recordings at position 4 differ from the rest in duration of the entire sequence played as well as the silent gaps between sounds within a sequence.

4.7.2 Sound type identification system results

We tested our sound type identification systems described in Sec. 3.4.2 (full-stream) and Sec. 3.4.4 (segregated) on all recorded scenes, and compared with our corresponding results from evaluations in Sec. 4.3.3 and Sec. 4.5.1. We used the same methods as in those sections, but using the deployment system ADREAM recorded scenes instead of the binaural simulator-generated scenes. In Fig. 4.37, for the two full-stream system variants with feature sets $fc2$ and $fc3_{1s}$ and the segregated system, the resulting performances both on the development and deployment system are displayed over the number of sources in the scene. We note the following effects:

- The qualitative behavior of development and deployment system with regard to

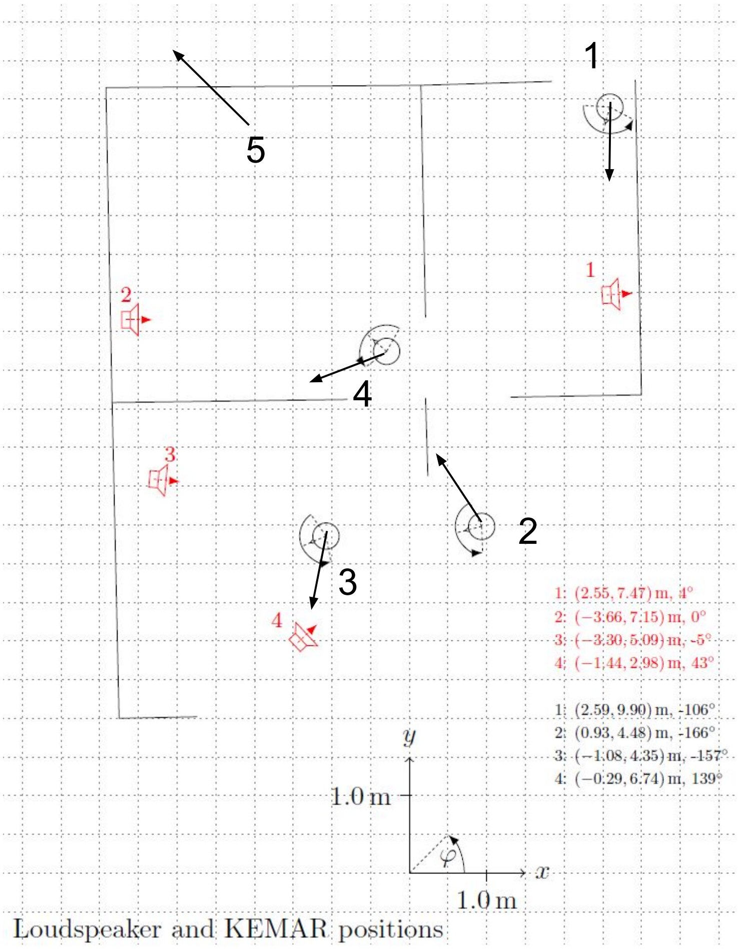


Figure 4.36: Layout of positions and orientations of the robot and the loud speakers in the adream apartment used during recordings. The arcs around the positions indicate the orientation ranges at which the BRIR were recorded.

dependence on number of sources and difference between models is similar.

- The tests on the deployment system result in roughly 0.05 lower balanced accuracies, leveling even for scenes with four sources. A slightly lower performance was expected, as in the recordings there are additional disturbing sources such as computer fans, the robot's noises, human noises, etc. However, the quantitative comparison is in any case difficult and may be misleading, since the tests were not performed on exactly the same scenes – in particular, the distribution of sound types played back concurrently through the speakers was different than in the generated scenes, as only a subset of types was used for the recordings.

- The advantage of the $fc3_{1s}$ models is slightly higher with the deployment system.
- The segregated identification on the recorded scenes performs slightly better than the full-stream $fc2$ variant, while with the development system and generated scenes, it is the other way around.

The following table shows mean performance *differences* between robot positions (but for each combination with the same models and same mixtures), averaged over mixtures and different models. A positive value at row x, column y, indicates that performance with the robot on position x was better than on position y. The effects are small, but we note two consistent results: position three is worse than all others, and position five is better than all others.

position	1	2	3	5	moving
1		0.006	0.068	−0.002	−0.014
2	−0.006		0.004	−0.016	0.018
3	−0.068	−0.004		−0.123	−0.118
5	0.002	0.016	0.123		0.014
moving	0.014	−0.018	0.118	−0.014	

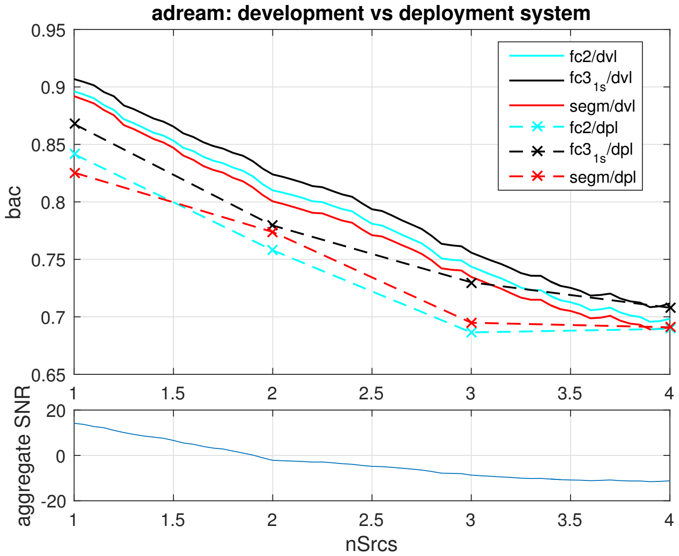


Figure 4.37: Mean balanced accuracies of detection models built with different schemes and from different scenes (averaged over all sound types), depending on number of sources. “dvl” curves are constructed as running average over 20 scene configurations sorted by number of sources, “dpl” curves show average over about 10 scenes with particular number of sources. Bottom panel shows aggregate SNR depending on number of sources.

4.8 Active exploration

4.8.1 Sound database

A collection of speech and non-speech sounds was obtained using the dataset provided with the "sound event detection in synthetic audio" task of the detection and classification of acoustic scenes and events (DCASE) challenge 2016[6]. The database consists of isolated recordings from 11 sound categories with 20 samples per category.

4.8.2 Experimental setup

BRIRs of four rooms, comprising reverberation times (T_{60}) of 250 ms, 500 ms, 750 ms and one anechoic room, were created using HRIRs from Algazi *et al.* (2001) and the image-source method introduced by Allen and Berkley (1979). Supervised training of the measurement model, Eqn. (3.15), was conducted using pre-rendered BRIRs on a grid with 25 cm distance and 5° azimuth spacing. During evaluation, BRIRs were rendered online at each time step. The source position was always placed at the center of the room.

The experimental procedure is based on a 4-fold cross-validation approach, where BRIRs of three rooms were used for training. The remaining BRIRs were used for evaluation, by running 50 simulations with random initial robot poses and goal positions. Sound samples were randomly selected from the database and the initial conditions were consistent over all cross-validation folds. Each simulation was restricted to a maximum simulated duration of 60 s. All experiments were repeated for different settings of the trade-off parameter $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$. An additional experiment with a bearing-only measurement model was also conducted.

The system performance was assessed by the root mean square (RMS) error of the estimated Cartesian sound source position, averaged over the corresponding sets of simulations. To better account for outliers that sporadically occur due to the stochastic nature of the particle filter, the median of the achieved localization performance is reported in Tab. 4.4.

Table 4.4: Median localization errors (m) for all investigated acoustic conditions (T_{60}, denoted in the top row). Experiments using a measurement model based on azimuth and distance (AD) were conducted. For comparison, a bearing-only measurement model (A) was evaluated with $\lambda = 0.25$.

Obs.	λ	Anec.	250ms	500ms	750ms	Avg.
AD	0.00	0.58	0.72	1.63	2.82	1.44
AD	0.25	**0.51**	**0.63**	**0.70**	**0.77**	**0.65**
AD	0.50	0.63	0.67	1.05	1.29	0.91
AD	0.75	0.87	0.91	1.18	1.44	1.10
AD	1.00	1.09	1.08	1.36	1.37	1.23
A	0.25	0.82	0.84	0.87	0.89	0.86

Table 4.5: Time-to-goal (TTG) in (s) and percentage of goal positions reached (GPR) within the available simulation time, averaged across all acoustic conditions for four investigated values of λ (top row). As $\lambda = 0$ does not steer the robot towards the goal position, it is not explicitly shown here.

Metric	0.25	0.5	0.75	1.00
TTG	38.31	14.49	9.94	**9.44**
GPR	60.00	98.00	**100.00**	**100.00**

4.8.3 Results and discussion

The results depicted in Tab. 4.4 show that MCE improves localization performance compared to a fixed trajectory towards the goal ($\lambda = 1$). The best localization accuracy is achieved for $\lambda = 0.25$. However, this comes at the cost of an increased time to reach the goal position, as depicted in Tab. 4.5. Therefore, λ must be chosen appropriately for specific applications.

An interesting outcome of the conducted experiments is that localization performance is significantly degraded for $\lambda = 0$. This can be explained by the fact that the robot is able to move around freely, without being restricted to approach the goal position. The policy obtained in this case tends to steer the robot very close to the assumed source position. This disturbs the observed binaural cues due to the near-field effect of the HRIRs and large jumps in azimuth between consecutive time steps. As this is not explicitly covered by the measurement model, the particle filter is not able to appropriately predict these effects.

[6] `http://www.cs.tut.fi/sgn/arg/dcase2016/`

The comparison of the proposed azimuth and distance-dependent measurement model with the bearing-only approach shows, that modeling distance information using the IC helps to further improve the localization capabilities of the system. The improvements are statistically significant according to a t-test conducted with $p < 0.01$.

4.9 Selective attention

4.9.1 Evaluation of HTMKS

The Head Turning Modulation (HTM) model described in Deliverable D4.3 can be evaluated by considering the two behaviours the robot exhibits when it is driven by this model. Even though the HTM model is an online learning algorithm that is not supposed to ever stop learning, the progressive change in the *behaviour* of the robot can be observed. This consists mainly of those head movements that can be triggered by the two modules which constitute the HTM model: the Dynamic Weighting (DW) module and the Multimodal Fusion and Inference module (MFI), both described in Deliverable D4.3.

Simulation setup

The goal of the experiments is to validate whether the HTMKS is able to learn all the categories present in the environment, without any *a priori* knowledge. To achieve that, the HTMKS was provided with a larger amount of emulated audio and visual classifiers than actually contained in the simulated scenes. Specifically, 30 audio and 15 visual experts were emulated for a total of 40 possible combinations, labeled as correct audiovisual pairings. However, this labelling is not taken into account in any HTMKS computations and is only used to validate the model based on ground-truth data.

In all the following, one *time step* corresponds to a frame length (500ms). Thus, 1000 time steps simulations would correspond in real experiments to 500 seconds of data processed (≈ 8.3 minutes). On this basis, simulated environments were only populated with 4 to 10 different audiovisual categories, with respect to the number of audiovisual objects that would be present in the scenarios used for the demonstrations. Thus, the HTMKS is still provided with outputs from classifiers that should exhibit high probabilities. This enables us to introduce some classification errors and some wrong audiovisual pairings in order to test the correction abilities of the proposed model.

A rate of $\epsilon^a = \epsilon^v = 35\%$ of frame-based errors was used to mimic realistic audio and visual classifiers. Since the HTMKS does use azimuth information about current objects present in the environment but is not supposed to correct any output from the dedicated KS, the

Table 4.6: Labels used in the simulation

Visual labels	Audio labels	Audiovisual categories $\mathcal{C}$
male	`speech`	*male* `speech`
female	`coughing`	*male* `coughing`
baby	`crying`	*male* `crying`
bird	`eating`	*male* `eating`
dog	`piano`	*male* `piano`
cat	`violin`	*male* `violin`
door	`laughing`	*male* `laughing`
siren	`singing`	*female* `coughing`
train	`flute`	*female* `crying`
car	`screaming`	*female* `eating`
glass	`harp`	*female* `singing`
bed	`whistling`	*female* `laughing`
wind	`flying`	*female* `flute`
switch		*female* `screaming`
		female `harp`
		baby `crying`
		baby `laughing`
		baby `screaming`
		bird `whistling`
		bird `flying`
		dog `barking`
		dog `panting`
		cat `meaowing`
		cat `scratching`
		door `knock`
		door `closing`
		siren `beeping`
		siren `alert`
		train `alert`
		train `braking`
		train `accelerating`
		car `braking`
		car `accelerating`
		car `drifting`
		glass `breaking`
		glass `clinking`
		bed `squeaking`
		wind `blowing`
		rain `falling`
		switch `switched`

emulated DnnLocationKS was designed as a perfect localizer in order not to interfere with the primary goal of the HTMKS. Every source in the environment was linked to a given audiovisual object. This creates a realistic environment, where sources exhibit temporally discontinuous behaviour (e.g., sources can stop emitting for a while then start emitting again). The HTMKS should be able to understand the environment through this notion of physical objects.

Fig. 4.38 shows an illustration of the simulated environment together with some of the results of the computations made by the HTMKS. The simulated audio and visual labels are presented in Table 4.6.

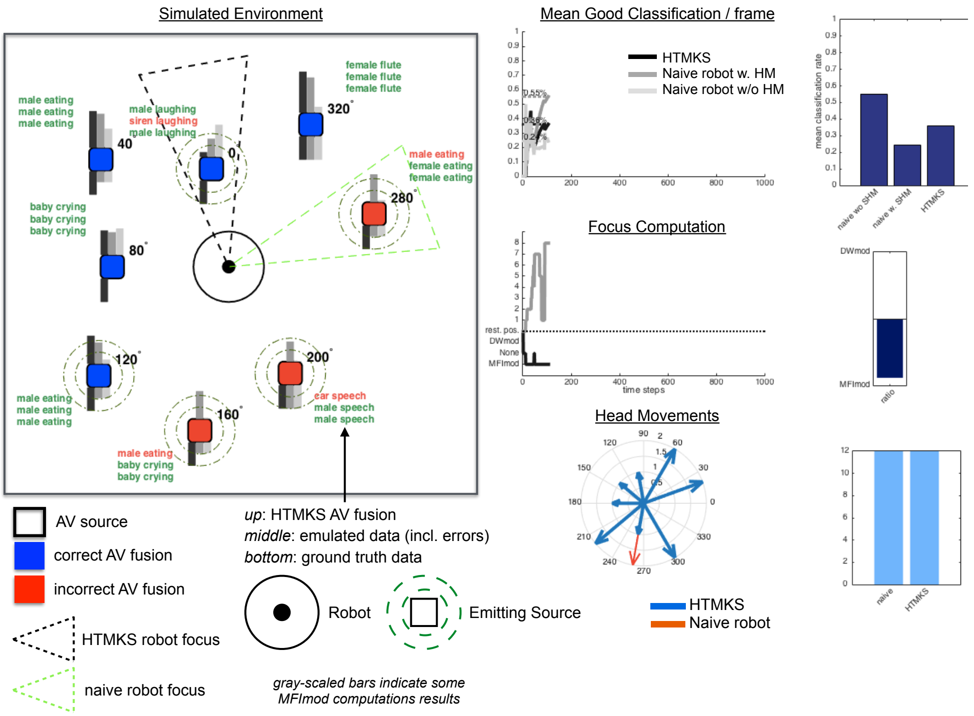


Figure 4.38: Illustration of the simulated environment together with some of the results of the HTMKS computations.

Evaluation of the audiovisual fusion

The first evaluation is concerned with the ability of the HTMKS to correct wrong audiovisual pairings exhibited by potentially erroneous audiovisual classification experts.

Fig. 4.39 shows how the fusion of audio and visual classifiers computed by the HTMKS outperform a naive fusion based only on the output of these experts, and without any *a priori* knowledge about the environment. Fig. 4.40 is the average results of 15 runs

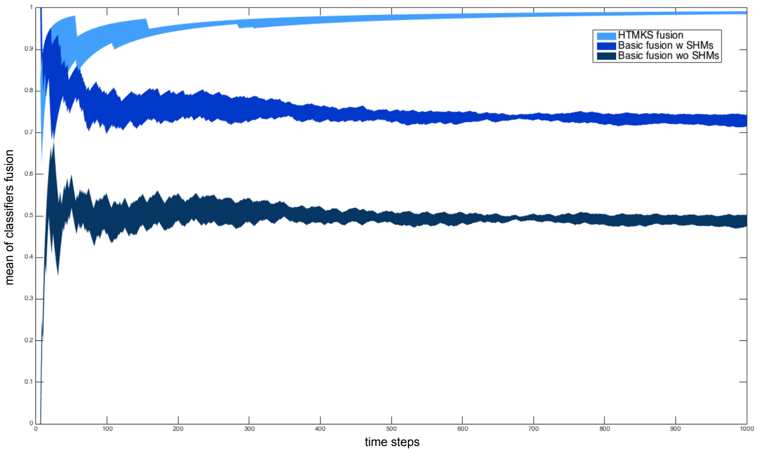


Figure 4.39: Average results of classifier fusion on five runs for an environment populated with 5 sources and only three different audiovisual objects: pairs $n^{1,9,17}$ (see Table 4.6).

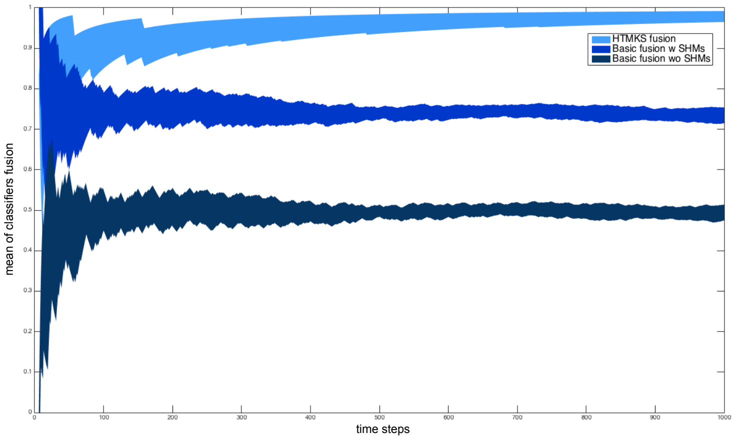


Figure 4.40: Average results of classifier fusion on five runs for three different environments (a total of 15 runs).

Table 4.7: Correction performances over 1000 random vectors

Category $\mathcal{C}$	α^v	α^a
$\mathcal{C}_1 =$(`speech`,*male*)	44.52 %	73.15 %
$\mathcal{C}_2 =$(`coughing`,*female*)	96.05 %	45.80 %
$\mathcal{C}_3 =$(`barking`,*dog*)	92.89 %	27.17 %
$\mathcal{C}_4 =$(`knock`,*door*)	73.96 %	25.38 %
$\mathcal{C}_5 =$(`beeping`,*siren*)	83.15 %	75.96 %
$\mathcal{C}_6 =$(`breaking`,*glass*)	64.90 %	77.42 %
$\mathcal{C}_7 =$(`blowing`,*wind*)	85.45 %	57.24 %

on 3 different environments populated with respectively 5 or 8 sources, and simulating different combinations of audiovisual objects. Once again, even when dealing with more complex environments, the HTMKS outputs hypotheses about the audiovisual objects in the environment with $\approx 97\%$ accuracy. One has also to take into account the fact that these results include the process of progressive inhibition of head movements (see next section) computed by both the MFI and the DW modules.

The MFI module can estimate the category of an audio-visual frame, even in the presence of audio or visual classifiers errors. This has been assessed by introducing errors in the input vectors $((\mathbf{P}^a)^T, (\mathbf{P}^v)^T)^T$ and by looking at the consecutive audiovisual category $\widehat{\mathcal{C}}$ computed by the HTMKS. More precisely, for a fixed audio (visual) vector $\mathbf{P}^a$ ($\mathbf{P}^v$ resp.), 1000 artificially designed audio-visual frames $\mathbf{P}$ are tested to evaluate the correction rate α^v (α^a resp.) brought by the audio-visual fusion for 7 categories $\mathcal{C}$.

Table 4.7 shows very good correction rates for certain categories (up to 96.05% for $\mathcal{C}_2$), but also poorer rates for some others (25.38% for $\mathcal{C}_4$). This can be explained by the number of observed occurrences of each category during the exploration. Indeed, 20% of the simulated objects belongs to the category $\mathcal{C}_2$, while only 9% are from $\mathcal{C}_4$. But note that statistically speaking, the random nature of the audio or visual component in $\mathbf{P}$ for this evaluation corresponds to a mean correct categorization rate of 14%. The MFI module thus outperforms by 2 to 7 times a random choice among the 7 considered audio or visual categories.

Evaluation of the progressive inhibition of the number of head movements

Here, the resulting behaviour of a robot driven by the HTMKS is compared to the behaviour of a naive robot that turns its head toward the direction of every sound source appearing in the environment. This comparison is of interest since the goal of the HTMKS is to enable the robot to learn how to inhibit irrelevant head turns.

The MFI module is active whenever the system (and namely the M-SOM, see D4.3) gauge that it is unable to process correctly the audio-visual categories detected. Progressively, the system will be able to trust the results of its computations resulting in the inhibition previously required head movements.

The DW module is active only when the performance of category inference is above the K_{head} threshold defined beforehand with respect to the scenario (D4.3). Whenever this performance is not high enough, the MFImod triggers the head movements. On the other hand, once the performance for a given audiovisual category is high enough, the *congruence* of this category given the environment currently explored can be computed.

This evaluation has been computed for an environment populated with 7 sources and only 2 different audiovisual objects. The number of different audiovisual objects has been set to be that low on purpose, so that the impact of the MFImod and of DWmod computations can be observed easily.

In addition, we chose this scenario to highlight the main and important difference between the Dynamic Weighting module and other systems of attention modulation based on *saliency*. Indeed, the simulated environment used here is populated with two kinds of audiovisual objects: (*male* `speech`) and (*baby* `crying`), with less speaking males that crying babies. The goal of the DWmod is to explore and understand unknown environments without any *a priori* knowledge about it. Thus, if the robot is in a nursery, babies crying would not be of importance (in the scope of low-level head-movement-related attention), given the high probability of this event to occur. The goal of this simulation is to measure how the MFImod and DWmod have been able to inhibit irrelevant head movements while still exhibiting good audiovisual fusion results. Results are shown in Fig. 4.41.

Evaluation of the effect of the K_{head} criterion

This evaluation is computed on a simplified scenario in order to assess the impact of changes in the K_{head} criterion (see Deliverable D4.3). The environment consists of 8 sources, with a maximum of 5 simultaneous sound sources, during a 1000 discrete time steps simulation, that is 40 audiovisual events occurring. Three values of K_{head} were tested: $[0, 0.5, 1]$. This criterion only concerns MFImod computations. Thus, in order to assess the role of this criterion in the global behavior of the MFImod, the following set up exclude the results of the DWmod computations.

Fig. 4.42 shows the two impacts the K_{head} criterion has on the robot's behavior:

(i) the results of mean correct audiovisual fusion: the lower the K_{head} criterion, the more wrong hypotheses are made about audiovisual labels of the objects populating

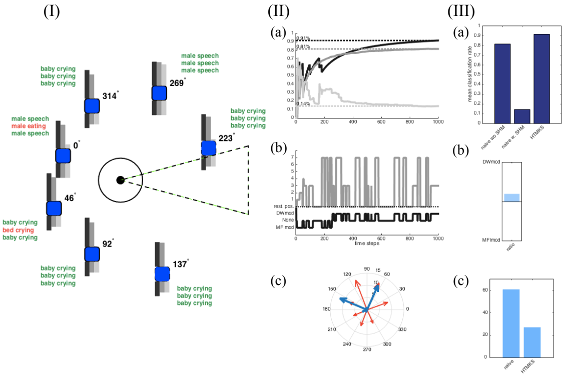


Figure 4.41: Impact of the Multimodal Fusion & Inference module together with the Dynamic Weighting module in head movements inhibition. This simplified simulated scenario has been set up in order to observe the results of MFImod and DWmod computations more easily. Subfigures (II.a)&(III.a) shows the results of the audiovisual fusion performed by the MFImod. Subfigure (II.b) shows how the DWmod becomes quickly responsible for every head movements while the MFImod stops triggering head movements. Subfigures (II.c)&(III.c) shows how many head movements have been inhibited compared to a naive robot.

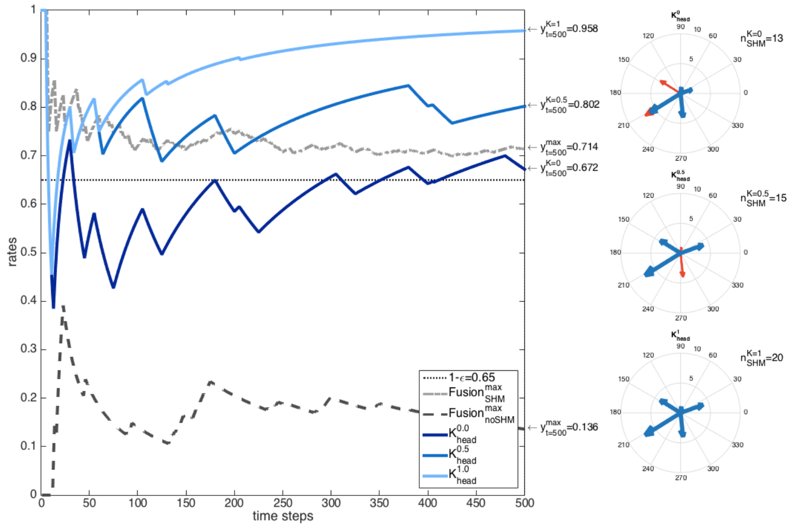


Figure 4.42: Impact of different K_{head} criterion values ($[0, 0.5, 1]$) on *(left)* mean good audiovisual fusion, and *(right)* on the number of head movements triggered, compared to a naive robot.

the environment. On the other hand, with a high criterion, fusion and inference are better.

(ii) the number of head movements: the lower K_{head}, the fewer movements are made, while the closest to K_{head} to 1, the less head movements inhibitions are made.

Fig 4.43 shows another representation of the results shown in Figure 4.42 emphasizing which object has been focused on by the robot. Similarly to what has been stated above, for a high K_{head} every object is systematically focused while for a lower value of K_{head}, a more audiovisual events are not, or only for a short period of time, taken into consideration by the robot.

Hence, this criterion has to be adjusted according to the primary goal of the robot: if it is in a S&R scenario, the priority is to move rapidly towards a victim rather than fully explore the environment; on the other hand, in a pure exploration scenario, the accurate understanding of the environment being the priority, the criterion can be set to its maximum thus allowing the robot to get the most information possible about the environment.

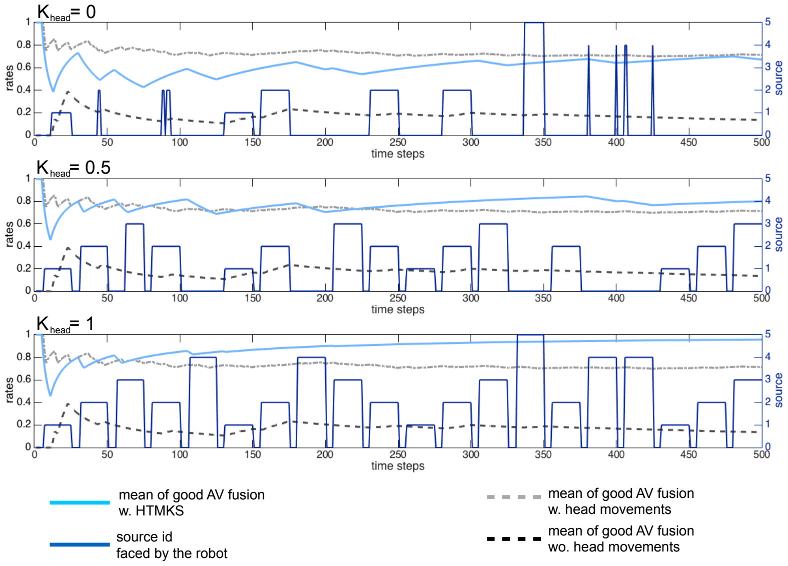


Figure 4.43: Other visualization of results shown on Figure 4.42 exhibiting a more detailed representation of focused objects.

4.10 Audio-visual integration

Audiovisual integration is evaluated for two purposes, namely audiovisual keyword recognition and audiovisual speech enhancement. Ultimately, as expected, this leads to an enhanced intelligibility, reflected in improved machine recognition accuracies as well as in better intelligibility of enhanced speech for human listeners.

4.10.1 Audio-visual keyword recognition

The turbo-principle and the coupled HMM were compared for the task of keyword recognition in noisy scenarios, where we have newly incorporated the use of observation uncertainties to deal with the time-varying reliability of the available acoustic information.

Experimental setup

Audio data from the first CHiME challenge (Barker *et al.*, 2013) in combination with matching video data from the GRiD corpus (Cooke *et al.*, 2006b) was used for our experimental evaluation.

The recordings consist of 1000 sentences spoken by 33 talkers each. All utterances include the annunciation of a letter (A…Z, excluding W) and a digit (0…9). Audio and video files are not start/endpoint-aligned between the CHiME and the GRiD corpus. We therefore performed a start/end-point matching via the word alignment files provided on http://spandh.dcs.shef.ac.uk/gridcorpus/.

Results for GMM-based recognition

The entire set of training data was used in the initial model training. Subsequently, development data of the first five speakers was used to adjust all free parameters, i.e. the audio stream weights for the CHMM and the TD decoder. Table 1 shows the corresponding values that we obtained for the four different types of decoders. We set $\lambda_P = 0.1$ for all TD experiments and $D' = 37$ for GDN.

We have measured the success of each of the considered recognition setups in terms of the keyword accuracy, corresponding to the percentage of correctly identified letters and digits. Table 4.8 shows the results that were achieved.

As can be seen, each of the acoustic recognizers was run in four different modes of

Method	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	avg.
Video	69.98	69.98	69.98	69.98	69.98	69.98	69.98
Audio Full	65.09	75.04	80.83	86.73	90.34	92.54	81.76
Audio Diag	71.90	79.05	82.56	87.75	91.64	91.60	84.08
Audio UD	72.99	77.57	81.60	88.73	91.49	91.74	84.02
Audio NALDA	74.00	78.94	85.19	90.93	92.40	93.30	85.79
CHMM Full	76.67	82.56	87.30	89.74	92.37	93.88	87.09
CHMM Diag	84.72	85.81	88.68	90.47	91.22	92.09	88.83
CHMM UD	83.63	84.59	87.77	88.97	91.18	90.64	87.80
CHMM NALDA	84.13	87.59	90.28	92.40	93.36	93.43	90.20
Turbo Full	81.79	86.41	90.11	91.53	93.98	**95.32**	89.86
Turbo Diag	85.75	88.58	90.45	92.16	93.68	93.52	90.69
Turbo UD	84.34	87.57	89.67	91.48	93.60	92.71	89.89
Turbo NALDA	**87.21**	**89.48**	**92.08**	**93.09**	**95.26**	95.12	**92.04**

Table 4.8: Results of acoustic-only, visual-only, and audio-visual keyword recognition.

operation:

- using diagonal-covariance Gaussian probability density functions (diag)
- full-covariance Gaussian probability density functions (full)
- uncertainty decoding according to Deng *et al.* (2005) (UD)
- or noise-adaptive linear discriminant analysis introduced in Kolossa *et al.* (2013) (NALDA)

Results for DNN-based recognition

For the neural-network based system, Kaldi was employed as the recognition engine and its own training scripts were applied, cf. Meutzner *et al.* (2017). However, the Kaldi feature extraction was replaced by Two!Ears gammatone filterbank features. Results in Table 4.9 show the clear superiority of the use of stream weighting (SW) as compared to the additional use of observation uncertainties within a DNN training.

Method	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Avg.
Audio	71.64	76.55	84.83	88.45	92.07	93.28	84.47
Video	72.50	72.50	72.50	72.50	72.50	72.50	72.50
AV	86.21	87.24	90.00	91.12	91.12	91.21	89.48
AV + Unc.	85.09	86.12	88.79	91.12	91.21	92.07	89.07
AV + SW	**86.72**	**88.88**	**91.90**	**93.19**	**94.40**	**95.78**	**91.81**

Table 4.9: Keyword accuracies (%) using the ratemap acoustic features. In addition to acoustic, visual and audiovisual recognition, we compare the use of observation uncertainties (Unc.) with the use of explicit, frame-wise dynamic stream weight adaptation.

4.10.2 Audio-visual speech enhancement

Since the turbo-principle was hence clearly more successful in the recognition of keywords, it was employed for model-based speech enhancement within the same task. As a measure of success, we now measured human recognition accuracy on the enhanced data, by carrying out crowdsourcing experiments. Each test participant was asked to transcribe 14 audio signals, covering two settings of our enhancement strategy, using best-path synthesis (AVSE BP) and all-path synthesis (AVSE AP) in addition to log-MMSE signal enhancement and noisy speech. We created test sets of 4 different SNRs (-9 dB, -6 dB, -3 dB, and 0 dB) and used crowd-sourcing tests at CrowdFlower to collect all results. Table 4.10 shows the achieved accuracies of the test participants.

Signal	-9 dB	-6 dB	-3 dB	0 dB	Avg.
Noisy	48.65	60.23	70.47	77.24	64.15
log-MMSE	41.53	53.88	61.41	71.08	56.97
AVSE AP	73.43	77.28	81.58	82.65	78.74
AVSE BP	74.82	79.35	81.43	84.64	80.06

Table 4.10: Listening test results. Each score is based on averaging about 260 unique utterances.

As expected, log-MMSE speech enhancement was not successful in improving the intelligibility of the noisy speech. In clear contrast, both the AP and the BP setting of audio-visual speech enhancement yielded significant improvements, with a slight (and unexpected) gain for best-path in contrast to all-path synthesis. Hence, it is apparently of greater value to re-synthesize in accordance with the best state sequence rather than averaging over multiple sequences. As this setting is also computationally more efficient, it is the clear best choice for this task of audiovisual model-based speech enhancement.

4.11 Meaning assignment

One specific example of meaning assignment is evaluated here, which relates closely to a technology developed in WP3. Further examples of meaning assignment by the Two!Ears system are given in Deliverable D4.3, since those involve top-down feedback (the topic of WP4).

4.11.1 Genre recognition

Training and evaluation of the genre classifier was performed using the GTZAN genre collection[7] dataset, which contains music signals from 10 different genres, namely "blues", "classical", "country", "disco", "hiphop", "jazz", "metal", "pop", "reggae" and "rock". The classifier's performance was evaluated using 5-fold cross-validation on the dataset. Cepstral modulation ratio regression (CMRARE) and mel-frequency cepstral coefficients (MFCCs) (Martin and Nagathil, 2009) parameters – both are features which are commonly used in the context of musical genre recognition – were chosen for performance comparison. As all classes are balanced within the dataset, classification error was used as a performance metric.

log-RM	MFCC	CMRARE
16.25	22.40	47.33

Table 4.11: Averaged classification errors achieved on the GTZAN genre collection dataset.

Results of the performance comparison are depicted in Table 4.11. The log-ratemap features derived from the Two!Ears auditory front-end outperform both MFCC and CMRARE features. The performance increase is statistically significant according to a t-test conducted with $p < 0.01$. The confusion matrices depicted in Fig. 4.44 show, that all musical genres are predicted with sufficient accuracy for log-ratemap and MFCC features. In contrast, CMRARE parameters yield an overall higher classification error and show difficulties in correctly classifying the musical genres "disco", "hiphop" and "rock". The reason for this most likely stems from the block-based processing scheme which is adopted here, as CMRARE features normally rely on longer signal segments to gain optimal predictive performance. However, the Two!Ears framework requires the capabilities to analyse an incoming stream of audio signals online, hence the block-based processing is required for this task.

[7] `http://marsyasweb.appspot.com/download/data_sets/`

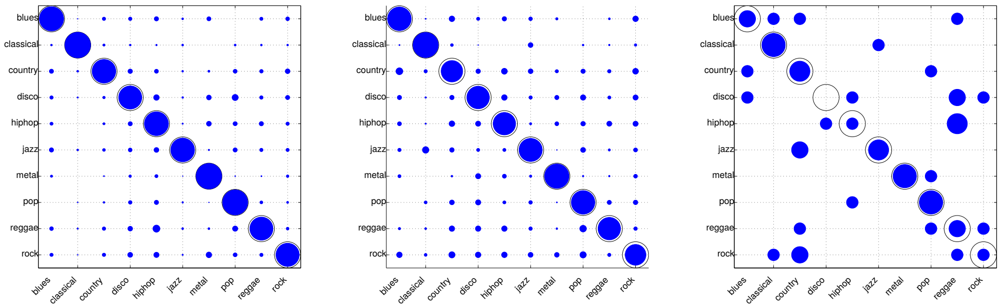


Figure 4.44: Confusion matrices for (from left to right) log-RM, MFCC and CMRARE features, derived from averaging classification results over all cross-validation folds.

5 Summary and discussion

This report has described the key achievements of WP3, with a strong emphasis on the work done in the second and third years of the Two!Ears project. The project has made substantial progress against its ambitious goal: we have described a software framework, embedded in a mobile robot, that is able to reason about acoustic scenes. In complex multi-source environments, the system is able to determine the type of sound sources that are present, localize them and track them. Additionally, the characteristics of sounds sources can be determined, such as the genre of a piece of music, or the identity of a speaker. The localization and attentional systems are able to direct the head of the robot in order to resolve ambiguities. In some cases, such as when determining the azimuth of a target voice in a mixture of four other voices, the performance of the system approaches human performance.

In many of these cases, top-down feedback in the system has been shown to be vital. For example, the performance of source localization can be significantly improved if top-down feedback is used about the spectral characteristics of the sound sources that are present. Similarly, top-down information gleaned about the number of sources in the environment, which can be done from acoustic models, can provide important cues to lower-level processing such as stream segmentation.

An notable aspect of the system is the way in which it leverages statistical machine learning within a blackboard architecture. Various aspects of the Two!Ears system make use of graphical models and deep neural networks, which provide powerful tools for exploiting the acoustic data available. Multi-condition training across a wide range of acoustic conditions has been shown to improve the performance of the system in a number of respects, including source localization and classification of sound type.

Whilst much progress has been made within the project, there are of course many challenges still remaining. The focus of Two!Ears has primarily been on acoustic scenes; while audio-visual integration has been addressed to some extent, for example in our work on audiovisual speech recognition, many facets of audiovisual integration in humans have yet to be explored. For example, our system might be tested with stimuli designed to show the McGurk effect. Similarly, there are many exciting prospects for using the tools that we have developed and made available to the research community. The ability to replace a human listener with a robotic system in listening experiments opens up many possibilities. For example, head tracking could be used to record the movement of a subject's head during a localization experiment, allowing their head movements to be 'played back' on the robot system. A novel analysis of the head movement strategies used by listeners could therefore be conducted, and compared against various machine algorithms.

Acronyms

AFE	auditory front-end
AMS	amplitude modulation spectrogram
AP	all-path
ASR	automatic speech recognition
BP	best-path
BRIR	binaural room impulse response
CHMM	coupled hidden Markov model
DNN	deep neural network
EBM	estimated binary mask
ERB	equivalent rectangular bandwidth
FBA	forward-backward algorithm
GMM	Gaussian mixture model
HATS	head and torso simulator
H-FA	hit rate minus false alarm rate
IBM	ideal binary mask
IC	interaural coherence
CI	confidence interval
KS	knowledge source
LC	local criterion
CLUE	conversational language understanding evaluation

LTAS	long term average spectrum
NH	normal hearing
HINT	hearing in noise test
PDLA	probabilistic linear discriminant analysis
RMS	root mean square
SLAM	simultaneous localisation and mapping
SNR	signal-to-noise ratio
STOI	short term objective intelligibility
ESTOI	extended short term objective intelligibility
SVM	support vector machine
T-F	time-frequency
WRS	word recognition score
WP3	work package three

Bibliography

Abdelaziz, A. H. and Kolossa, D. (**2014**), "Dynamic Stream Weight Estimation in Coupled-HMM-based Audio-visual Speech Recognition Using Multilayer Perceptrons," in *Proc. Interspeech*. (Cited on pages 65 and 66)

Algazi, V. R., Duda, R. O., Thompson, D. M., and Avendano, C. (**2001**), "The CIPIC HRTF database," in *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the*, pp. 99–102. (Cited on page 125)

Allen, J. B. and Berkley, D. A. (**1979**), "Image method for efficiently simulating smallâĂŽÄ?-room acoustics," *The Journal of the Acoustical Society of America* **65**(4), pp. 943–950. (Cited on page 125)

Aucouturier, J.-J. and Pachet, F. (**2003**), "Representing musical genre: A state of the art," *Journal of New Music Research* **32**(1), pp. 83–93. (Cited on page 54)

Banerjee, A., Dhillon, I. S., Ghosh, J., and Sra, S. (**2005**), "Clustering on the Unit Hypersphere Using Von Mises-Fisher Distributions," *J. Mach. Learn. Res.* **6**, pp. 1345–1382. (Cited on pages 7 and 9)

Barker, J., Vincent, E., Ma, N., Christensen, H., and Green, P. (**2013**), "The PASCAL CHiME speech separation and recognition challenge," *Computer Speech and Language* **27**(3), pp. 621–633. (Cited on pages 61 and 135)

Begault, D. R. (**1992**), "Perceptual effects of synthetic reverberation on three-dimensional audio systems," *Journal of the Audio Engineering Society* **40**(11), pp. 895–903. (Cited on page 80)

Bentsen, T., May, T., Kressner, A. A., and Dau, T. (**2016**), "Comparing the influence of spectro-temporal integration in computational speech segregation," in *17th Annual Conference of the International Speech Communication Association*, San Francisco, USA, pp. 3324–3328. (Cited on page 11)

Bergstra, J. and Bengio, Y. (**2012**), "Random search for hyper-parameter optimization," *Journal of Machine Learning Research* **13**(Feb), pp. 281–305. (Cited on page 96)

Berrou, C., Glavieux, A., and Thitimajshima, P. (**1993**), "Near Shannon limit errorcorrecting coding and decoding: Turbo-codes," in *Proc. ICC*, Geneva, vol. 2, pp. 1064–1070. (Cited on page 63)

Beygelzimer, A., Langford, J., Tong, Z., and Hsu, D. J. (**2010**), "Agnostic active learning without constraints," in *Advances in Neural Information Processing Systems*, pp. 199–207. (Cited on page 36)

Bishop, C. M. (**2006**), *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA. (Cited on page 18)

Blauert, J. (**1997**), *Spatial hearing - The psychophysics of human sound localization*, The MIT Press, Cambride, MA, USA. (Cited on pages 37 and 39)

Blauert, J. (**1999**), *Spatial Hearing: The Psychophysics of Human Sound Localization*, MIT Press, Cambridge, MA. (Cited on pages 17 and 19)

Bourgault, F., Makarenko, A. A., Williams, S. B., Grocholsky, B., and Durrant-Whyte, H. F. (**2002**), "Information based adaptive robotic exploration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 540–545. (Cited on page 17)

Braasch, J. and Blauert, J. (**2011**), "Stimulus-dependent Adaption of Inhibitory Elements in Precedence Effect Models," in *Forum Acusticum.* (Cited on page 32)

Braasch, J., Blauert, J., and Djelani, T. (**2003**), "The precedence effect for noise bursts of different bandwidths. I. Psychoacoustical data," *Acoustical Science and Technology* **24**(5), pp. 233–241. (Cited on page 32)

Bregman, A. (**1990**a), *Auditory Scene Analysis*, MIT Press, Cambridge, MA. (Cited on pages 11, 39, and 76)

Bregman, A. S. (**1990**b), *Auditory Scene Analysis: The Perceptual Organization of Sound*, MIT Press, Cambridge, MA. (Cited on page 17)

Brown, G. and Cooke, M. (**1994**a), "Computational auditory scene analysis," *Comput. Speech. Lang.* **8**, pp. 297–336. (Cited on page 79)

Brown, G. J. and Cooke, M. P. (**1994**b), "Computational auditory scene analysis," *Computer Speech and Language* **8**(4), pp. 297–336. (Cited on page 102)

Bustamante, G., Danès, P., Forgue, T., and Podlubne, A. (**2016**), "Towards information-based feedback control for binaural active localization," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, Shanghai, China, March 20–25, 2016*, pp. 6325–6329. (Cited on pages 17 and 18)

Bustamante, G., Portello, A., and Danès, P. (**2015**), "A three-stage framework to active source localization from a binaural head," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, pp. 5620–5624. (Cited on pages 17 and 18)

Cardoso, J.-F. and Souloumiac, A. (**1993**), "Blind beamforming for non-Gaussian Signals," in *IEEE-Proceedings-F*, IEEE, vol. 140, pp. 362–370, http://sig.enst.fr/ cardoso/. (Cited on page 14)

Chang, C. C. and Lin, C. J. (**Software is available at `www.csie.ntu.edu.tw/~cjlin/libsvm` 2001**), "LIBSVM: A library for support vector machines," Software is available at `www.csie.ntu.edu.tw/~cjlin/libsvm`. (Cited on page 14)

Cooke, M., Barker, J., Cunningham, S., and Shao, X. (**2006**a), "An audio-visual corpus for speech perception and automatic speech recognition," *The Journal of the Acoustical Society of America* **120**(5), pp. 2421–2424. (Cited on page 42)

Cooke, M., Barker, J., Cunningham, S., and Shao, X. (**2006**b), "An audio-visual corpus for speech perception and automatic speech recognition," *Journal of the Acoustical Society of America* **120**(5), pp. 2421–4. (Cited on pages 62, 102, and 135)

Cooke, M., Barker, J., Cunningham, S., and Shao, X. (**2006**c), "An audio-visual corpus for speech perception and automatic speech recognition," *J. Acoust. Soc. Am.* **120**, pp. 2421–2424. (Cited on page 79)

Cooke, M., Green, P., Josifovski, L., and Vizinho, A. (**2001**), "Robust automatic speech recognition with missing and unreliable acoustic data," *Speech Communication* **34**(3), pp. 267 – 285, URL `http://www.sciencedirect.com/science/article/pii/S0167639300000340`. (Cited on page 44)

David, H. A. and Nagaraja, H. N. (**2003**), *Order Statistics*, Wiley, 3 ed. (Cited on page 45)

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., and Ouellet, P. (**2011**), "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing* **19**(4), pp. 788–798. (Cited on page 48)

Dempster, A., Laird, N., and Rubin, D. (**1977**), "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Society, Series B* **39**(1), pp. 1–38. (Cited on page 7)

Deng, L., Droppo, J., and Acero, A. (**2005**), "Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion," *IEEE Trans. Speech and Audio Processing* **13**(3), pp. 412–421. (Cited on pages 64 and 136)

Dreschler, W. A., Verschuure, H., Ludvigsen, C., and Westermann, S. (**2001**), "ICRA noises: Artificial noise signals with speech-like spectral and temporal properties for hearing instrument assessment," *Audiology* **40**(3), pp. 148–157. (Cited on page 70)

Evers, C., Moore, A. H., and Naylor, P. A. (**2016**), "Acoustic Simultaneous Localization and Mapping (a-SLAM) of a moving microphone array and its surrounding speakers," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, Shanghai, China, March 20–25, 2016.* (Cited on page 17)

Faller, C. and Merimaa, J. (**2004**), "Source localization in complex listening situations: Selection of binaural cues based on interaural coherence," *The Journal of the Acoustical Society of America* **116**(5), pp. 3075–3089. (Cited on pages 32 and 34)

Friedman, J., Hastie, T., and Tibshirani, R. (**2010**), "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software* **33**(1), pp. 1. (Cited on pages 27 and 31)

Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallett, D., Dahlgren, N., and Zue, V. (**1993**a), "TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1," *Web Download, Philadelphia: Linguistic Data Consortium* . (Cited on page 42)

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., and Dahlgren, N. L. (**1993**b), "DARPA TIMIT Acoustic-phonetic continuous speech corpus CD-ROM," *National Inst. Standards and Technol. (NIST)* . (Cited on page 41)

Gebru, I., Alameda-Pineda, X., Forbes, F., and Horaud, R. (**2016**), "EM Algorithms for Weighted-Data Clustering with Application to Audio-Visual Scene Analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* . (Cited on page 8)

Georganti, E., May, T., van de Par, S., and Mourjopoulos, J. (**2013**), "Sound Source Distance Estimation in Rooms based on Statistical Properties of Binaural Signals," *IEEE Transactions on Audio, Speech, and Language Processing* **21**(8), pp. 1727–1741. (Cited on page 17)

Glasberg, B. R. and Moore, B. C. (**1990**), "Derivation of auditory filter shapes from notched-noise data," *Hearing research* **47**(1-2), pp. 103–138. (Cited on page 44)

Han, K. and Wang, D. L. (**2012**), "A classification based approach to speech segregation," *Journal of the Acoustical Society of America* **132**(5), pp. 3475–3483. (Cited on pages 13 and 70)

Hartmann, W. M. and Rakerd, B. (**1989**), "Localization of sound in rooms IV: The Franssen effect," *The Journal of the Acoustical Society of America* **86**(4), pp. 1366–1373. (Cited on page 34)

Hoffmann, E., Kolossa, D., Köhler, B.-U., and Orglmeister, R. (**2012**), "Using Information Theoretic Distance Measures for Solving the Permutation Problem of Blind Source

Separation of Speech Signals," *EURASIP Journal on Audio, Speech, and Music Processing* . (Cited on page 14)

Hong, S. and Maitre, H. (**2009**), "Turbo Iterative Signal Processing," in *Proc. DSP/SPE*, pp. 495–500. (Cited on page 63)

Hosking, J. R. (**1990**), "L-moments: analysis and estimation of distributions using linear combinations of order statistics," *Journal of the Royal Statistical Society. Series B (Methodological)* , pp. 105–124. (Cited on page 45)

Hummersone, C., Mason, R., and Brookes, T. (**2010**), "Dynamic precedence effect modeling for source separation in reverberant environments," *IEEE Transactions on Audio, Speech, and Language Processing* **18**(7), pp. 1867–1871. (Cited on pages 11 and 78)

Hung, W.-L., Chang-Chien, S.-J., and Yang, M.-S. (**2012**), "Self-updating clustering algorithm for estimating the parameters in mixtures of von Mises distributions," *Journal of Applied Statistics* **39**(10), pp. 2259–2274. (Cited on page 8)

Jensen, J. and Taal, C. H. (**2016**), "An Algorithm for Predicting the Intelligibility of Speech Masked by Modulated Noise Maskers," *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **24**(11), pp. 2009–2022. (Cited on pages 70 and 71)

Jensen, K. and Andersen, T. H. (**2003**), "Real-time beat estimationusing feature extraction," in *International Symposium on Computer Music Modeling and Retrieval*, Springer, pp. 13–22. (Cited on page 44)

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (**2014**), "Caffe: Convolutional Architecture for Fast Feature Embedding," *arXiv preprint arXiv:1408.5093* . (Cited on page 30)

Kenny, P., Ouellet, P., Dehak, N., Gupta, V., and Dumouchel, P. (**2008**), "A study of inter-speaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing* **16**(5), pp. 980–988. (Cited on page 47)

Kidd, G. J., Best, V., and Mason, C. R. (**2008**), "Listening to every other word: Examining the strength of linkage variables in forming streams of speech," *Journal of the Acoustical Society of America* **124**(6), pp. 3793–3802. (Cited on page 77)

Kim, G., Lu, Y., Hu, Y., and Loizou, P. C. (**2009**), "An algorithm that improves speech intelligibility in noise for normal-hearing listeners," *Journal of the Acoustical Society of America* **126**(3), pp. 1486–1494. (Cited on pages 12, 13, and 70)

Klapuri, A. (**1999**), "Sound onset detection by applying psychoacoustic knowledge," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, IEEE, vol. 6, pp. 3089–3092. (Cited on page 44)

Kolossa, D., Zeiler, S., Saeidi, R., and Fernandez Astudillo, R. (**2013**), "Noise-Adaptive LDA: A New Approach for Speech Recognition Under Observation Uncertainty," *IEEE Signal Processing Letters* **20**(11), pp. 1018–1021. (Cited on pages 64 and 136)

Kopco, N., Best, V., and Carlile, S. (**2010**), "Speech localization in a multitalker mixture," *Journal of the Acoustical Society of America* **127**(3), pp. 1450–1457. (Cited on pages 77, 78, 80, and 81)

Kressner, A. A. and Rozell, C. J. (**2015**), "Structure in time-frequency binary masking errors and its impact on speech intelligibility," *Journal of the Acoustical Society of America* **137**(4), pp. 2025–2035. (Cited on pages 70, 71, 73, and 76)

Kressner, A. A. and Rozell, C. J. (**2016**), "Cochlear implant speech intelligibility outcomes with structured and unstructured binary mask errors," *Journal of the Acoustical Society of America* **139**(2), pp. 800–810. (Cited on page 73)

Langford, J., Li, L., and Strehl, A. (**2007**), "Vowpal wabbit online learning project," . (Cited on page 36)

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (**1998**), "Gradient-based learning applied to document recognition," *Proceedings of the IEEE* **86**(11), pp. 2278–2324. (Cited on page 51)

Lerch, A. (**2012**), *An introduction to audio content analysis: Applications in signal processing and music informatics*, John Wiley & Sons. (Cited on page 44)

Lindemann, W. (**1986**), "Extension of a binaural cross correlation model by contralateral inhibition. II. The law of the first wave front," *The Journal of the Acoustical Society of America* **80**(6), pp. 1623–1630. (Cited on page 32)

Lu, Y.-C. and Cooke, M. (**2010**), "Binaural Estimation of Sound Source Distance via the Direct-to-reverberant Energy Ratio for Static and Moving Sources," *Trans. Audio, Speech and Lang. Proc.* **18**(7), pp. 1793–1805. (Cited on page 17)

Luettin, J., Potamianos, G., and Neti, C. (**2001**), "Asynchronous Stream Modelling for Large Vocabulary Audio-Visual Speech Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 169–172. (Cited on page 62)

Ma, N., Brown, G. J., and Gonzalez, J. A. (**2015**a), "Exploiting Deep Neural Networks and Head Movements for Binaural Localisation of Multiple Speakers in Reverberant Conditions," in *16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10*, pp. 3066–3070. (Cited on page 17)

Ma, N., Brown, G. J., and Gonzalez, J. A. (**2015**b), "Exploiting top-down source models to improve binaural localisation of multiple sources in reverberant environments," in *Proc. Interspeech*, Dresden, Germany, pp. 160–164. (Cited on page 77)

Ma, N., Brown, G. J., and May, T. (**2015**c), "Robust localisation of of multiple speakers exploiting deep neural networks and head movements," in *Proc. Interspeech'15*. (Cited on page 37)

Ma, N., Brown, G. J., and May, T. (**2015**d), "Robust localisation of multiple speakers exploiting deep neural networks and head movements," in *Proc. Interspeech*, Dresden, Germany, pp. 3302–3306. (Cited on page 79)

Ma, N., May, T., Wierstorf, H., and Brown, G. J. (**2015**e), "A machine-hearing system exploiting head movements for binaural sound localisation in reverberant conditions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2699–2703. (Cited on page 17)

Ma, N., May, T., Wierstorf, H., and Brown, G. J. (**2015**f), "A machine-hearing system exploiting head movements for binaural sound localisation in reverberant conditions," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2699–2703. (Cited on page 40)

MacQueen, J. B. (**1967**), "Some methods for classification and analysis of multivariate observations," in *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297. (Cited on page 7)

Markovic, I. and Petrovic, I. (**2012**), "Bearing-only tracking with a mixture of von Mises distributions," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 707–712. (Cited on page 7)

Martin, R. and Nagathil, A. (**2009**), "Cepstral modulation ratio regression (CMRARE) parameters for audio signal analysis and classification," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 321–324. (Cited on page 138)

May, T., Bentsen, T., and Dau, T. (**2015**a), "The role of temporal resolution in modulation-based speech segregation," in *16th Annual Conference of the International Speech Communication Association*, Dresden, Germany, pp. 170–174. (Cited on pages 11, 12, and 13)

May, T. and Dau, T. (**2013**), "Environment-aware ideal binary mask estimation using monaural cues," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA. (Cited on pages 13 and 70)

May, T. and Dau, T. (**2014**a), "Computational speech segregation based on an auditory-inspired modulation analysis," *Journal of the Acoustical Society of America* **136**(6), pp. 3350–3359. (Cited on pages 12, 13, 14, and 70)

May, T. and Dau, T. (**2014**b), "Computational speech segregation based on an auditory-inspired modulation analysis," *The Journal of the Acoustical Society of America* **136**(6), pp. 3350–3359. (Cited on page 44)

May, T. and Gerkmann, T. (**2014**), "Generalization of supervised learning for binary mask estimation," in *International Workshop on Acoustic Echo and Noise Control*, Juan les Pins, France, pp. 154–187. (Cited on page 13)

May, T., Ma, N., and Brown, G. J. (**2015**b), "Robust localisation of multiple speakers exploiting head movements and multi-conditional training of binaural cues," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.* (Cited on pages 40 and 41)

May, T., van de Par, S., and Kohlrausch, A. (**2011**), "A probabilistic model for robust localization based on a binaural auditory front-end," *IEEE Transactions on Audio, Speech, and Language Processing* **19**(1), pp. 1–13. (Cited on page 39)

May, T., van de Par, S., and Kohlrausch, A. (**2012**), "A binaural scene analyzer for joint localization and recognition of speakers in the presence of interfering noise sources and reverberation," *IEEE Transactions on Audio, Speech, and Language Processing* **20**(7), pp. 2016–2030. (Cited on page 44)

May, T., van de Par, S., and Kohlrausch, A. (**2013**), "Binaural Localization and Detection of Speakers in Complex Acoustic Scenes," in *The technology of binaural listening*, edited by J. Blauert, Springer, Berlin–Heidelberg–New York NY, chap. 15, pp. 397–425. (Cited on page 41)

McEliece, R. J., MacKay, D., and Cheng, J. (**1998**), "Turbo decoding as an instance of Pearl's belief propagation algorithm," *IEEE Journal on Selected Areas in Communications* **16**(2), pp. 140–152. (Cited on page 64)

Meutzner, H., Ma, N., Nickel, R., Schymura, C., and Kolossa, D. (**2017**), "Improving Audio-Visual Speech Recognition using Deep Neural Networks with Dynamic Stream Reliability Estimates," in *submitted for Proc. ICASSP.* (Cited on pages 65 and 136)

Misra, H., Ikbal, S., Bourlard, H., and Hermansky, H. (**2004**), "Spectral entropy based feature for robust ASR," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, IEEE, vol. 1, pp. I–193. (Cited on page 44)

Moritz, N., Anemüller, J., and Kollmeier, B. (**2011**), "Amplitude modulation spectrogram based features for robust speech recognition in noisy and reverberant environments," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 5492–5495. (Cited on page 44)

Nakhost, H. and Müller, M. (**2009**), "Monte-Carlo Exploration for Deterministic Planning," in *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI'09, pp. 1766–1771. (Cited on page 18)

Nefian, A. V., Liang, L., Pi, X., Liu, X., and Murphy, K. (**2002**), "Dynamic Bayesian Networks for Audio-Visual Speech Recognition," *EURASIP Journal on Applied Signal Processing* **11**, pp. 1274–1288. (Cited on page 62)

Neti, C., Potamianos, G., Luettin, J., Matthews, I., Glotin, H., Vergyri, D., Sison, J., Mashari, A., and Zhou, J. (**2000**), "Audio-visual speech recognition," Tech. Rep. WS00AVSR, Johns Hopkins University, CLSP. (Cited on page 62)

Ng, A. Y. (**2004**), "Feature selection, L 1 vs. L 2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning*, ACM, p. 78. (Cited on page 31)

Nielsen, J. B. and Dau, T. (**2009**), "Development of a Danish speech intelligibility test," *International Journal of Audiology* **48**(10), pp. 729–741. (Cited on pages 69 and 71)

Oppenheim, A. V., Schafer, R. W., and Buck, J. R. (**1999**), *Discrete-time Signal Processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA. (Cited on page 19)

Patterson, R. D. and Holdsworth, J. (**1996**), "A functional model of neural activity patterns and auditory images," *Advances in speech, hearing and language processing* **3**(Part B), pp. 547–563. (Cited on page 44)

Peeters, G., Giordano, B. L., Susini, P., Misdariis, N., and McAdams, S. (**2011**), "The timbre toolbox: Extracting audio descriptors from musical signals," *The Journal of the Acoustical Society of America* **130**(5), pp. 2902–2916. (Cited on page 44)

Prince, S. J. and Elder, J. H. (**2007**), "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE ICCV*, pp. 1–8. (Cited on page 48)

Qian, J., Hastie, T., Friedman, J., Tibshirani, R., and Simon, N. (`http://gts.sourceforge.net/` **2013**), "Glmnet for Matlab," `http://gts.sourceforge.net/`. (Cited on pages 27, 31, and 116)

Receveur, S., Scheler, D., and Fingscheidt, T. (**2014**), "A Turbo-Decoding Weighted Forward-Backward Algorithm for Multimodal Speech Recognition," in *Proc. of 5th Int. Workshop on Spoken Dialog Syst.*, Napa, California. (Cited on page 63)

Rickard, S. (**2007**), *The DUET Blind Source Separation Algorithm*, Springer Netherlands, Dordrecht, pp. 217–241, URL `http://dx.doi.org/10.1007/978-1-4020-6479-1_8`. (Cited on page 59)

Schädler, M. R., Meyer, B. T., and Kollmeier, B. (**2012**), "Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition," *The Journal of the Acoustical Society of America* **131**(5), pp. 4134–4151. (Cited on page 44)

Scheler, D., Walz, S., and Fingscheidt, T. (**2012**), "On iterative exchange of soft state Information in two-channel automatic Speech Recognition," in *Proc. ITG Facht. Sprachkomm.* (Cited on page 62)

Schönfelder, V. H. and Wichmann, F. A. (**2013**), "Identification of stimulus cues in narrowband tone-in-noise detection using sparse observer models," *The Journal of the Acoustical Society of America* **134**(1), pp. 447–463. (Cited on page 36)

Schymura, C., Winter, F., Kolossa, D., and Spors, S. (**2015**), "Binaural Sound Source Localisation and Tracking Using a Dynamic Spherical Head Model," in *16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10*, pp. 165–169. (Cited on pages 17 and 19)

Settles, B. (**2012**), "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning* **6**(1), pp. 1–114. (Cited on page 36)

Shivappa, S., Rao, B. D., and Trivedi, M. M. (**2008**), "Multimodal information fusion using the iterative decoding algorithm and its application to audio-visual speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2241–2244. (Cited on page 63)

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (**2014**), "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research* **15**, pp. 1929–1958, URL `http://jmlr.org/papers/v15/srivastava14a.html`. (Cited on page 51)

Stachniss, C., Hahnel, D., and Burgard, W. (**2004**), "Exploration with active loop-closing for FastSLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 1505–1510. (Cited on page 19)

Taal, C. H., Hendriks, R. C., Heusdens, R., and Jensen, J. (**2011**), "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Transactions on Audio, Speech and Language Processing* **19**(7), pp. 2125–2136. (Cited on page 71)

Tchorz, J. and Kollmeier, B. (**2003**), "SNR estimation based on amplitude modulation analysis with applications to noise suppression," *IEEE Transactions on Audio, Speech, and Language Processing* **11**(3), pp. 184–192. (Cited on page 12)

Thrun, S., Burgard, W., and Fox, D. (**2005**), *Probabilistic Robotics*, The MIT Press. (Cited on pages 17, 18, 19, and 20)

Tibshirani, R. (**1996**), "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)* , pp. 267–288. (Cited on pages 26, 27, and 45)

Traa, J. and Smaragdis, P. (**2014**), "Multiple speaker tracking with the Factorial von Mises-Fisher Filter," in *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. (Cited on page 7)

Tran Vu, D. H. and Haeb-Umbach, R. (**2013**), "Blind Speech Separation Exploiting Temporal and Spectral Correlations Using Turbo Decoding of 2D-HMMs," in *Proc. EUSIPCO*, URL `http://nt.uni-paderborn.de/public/pubs/2013/TrHa2013_01.pdf`. (Cited on page 63)

Tzanetakis, G. and Cook, P. (**2002**), "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing* **10**(5), pp. 293–302. (Cited on page 44)

van der Merwe, R. and Wan, E. (**2003**), "Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models," in *IEEE International Conference on Acoustics, Speech, and Signal Processing.* (Cited on page 19)

Varga, A. and Moore, R. (**1990**), "Hidden Markov model decomposition of speech and noise," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 845–848. (Cited on page 58)

Vesa, S. (**2009**), "Binaural Sound Source Distance Learning in Rooms," *IEEE Transactions on Audio, Speech, and Language Processing* **17**(8), pp. 1498–1507. (Cited on page 17)

Wallach, H. (**1940**), "The role of head movements and vestibular and visual cues in sound localization," *Journal of Experimental Psychology* **27**(4), pp. 339–368. (Cited on page 17)

Wang, D. L. and Brown, G. J. (Eds.) (**2006**), *Computational Auditory Scene Analysis: Principles, Algorithms and Applications*, Wiley/IEEE Press. (Cited on pages 38, 56, 79, and 102)

Wierstorf, H., Geier, M., Raake, A., and Spors, S. (**2011**), "A free database of head-related impulse response measurements in the horizontal plane with multiple distances," in *Proc. 130th Conv. Audio Eng. Soc.* (Cited on pages 41, 78, and 79)

Wightman, F. L. and Kistler, D. J. (**1999**), "Resolution of front–back ambiguity in spatial hearing by listener and source movement," *Journal of the Acoustical Society of America* **105**(5), pp. 2841–2853. (Cited on page 37)

Woodruff, J. and Wang, D. L. (**2012**), "Binaural localization of multiple sources in reverberant and noisy environments," *IEEE Transactions on Audio, Speech, and Language Processing* **20**(5), pp. 1503–1512. (Cited on page 41)

Yost, W. A., Mapes-Riordan, D., and Guzman, S. J. (**1997**), "The relationship between localization and the Franssen effect," *The Journal of the Acoustical Society of America* **101**(5), pp. 2994–2997. (Cited on page 34)

Zeiler, S., Meutzner, H., Abdelaziz, A. H., and Kolossa, D. (**2016**a), "Introducing the Turbo-Twin-HMM for Audio-Visual Speech Enhancement," in *Proc. Interspeech.* (Cited on page 66)

Zeiler, S., Nickel, R., Ma, N., Brown, G., and Kolossa, D. (**2016**b), "Robust audiovisual speech Recognition Using Noise-Adaptive Linear Discriminant Analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* (Cited on page 62)