BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

MICROPHONE ARRAYS FOR SPEAKER RECOGNITION
MIKROFONNÍ POLE PRO ROZPOZNÁVÁNÍ ŘEČNÍKŮ

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR **LADISLAV MOŠNER**
AUTOR PRÁCE

SUPERVISOR **Doc. Dr. Ing. JAN ČERNOCKÝ**
VEDOUCÍ PRÁCE

BRNO 2017

Master's Thesis Specification/19199/2016/xmosne01

Brno University of Technology - Faculty of Information Technology

Department of Computer Graphics and Multimedia Academic year 2016/2017

Master's Thesis Specification

For:	**Mošner Ladislav, Bc.**
Branch of study:	Computer Graphics and Multimedia
Title:	**Microphone Arrays for Speaker Recognition**
Category:	Speech and Natural Language Processing

Instructions for project work:

1. Get acquainted with the principles of speaker recognition (SRE) based on i-vectors.
2. Get acquainted with the principles of microphone arrays and acoustic beam-forming.
3. Based on standard NIST data-sets and room impulse response generator, prepare a simulated data-set.
4. Evaluate the SRE on original and beam-formed data.
5. Suggest ways to improve the SRE system for such data (adaptation, re-training of different building blocks, etc.), implement them and evaluate.
6. Create a short video and/or poster documenting your work.

Basic references:

- according to supervisor's advice.

Requirements for the semestral defense:

Items 1 to 4 of the assignment.

Detailed formal specifications can be found at http://www.fit.vutbr.cz/info/szz/

The Master's Thesis must define its purpose, describe a current state of the art, introduce the theoretical and technical background relevant to the problems solved, and specify what parts have been used from earlier projects or have been taken over from other sources.

Each student will hand-in printed as well as electronic versions of the technical report, an electronic version of the complete program documentation, program source files, and a functional hardware prototype sample if desired. The information in electronic form will be stored on a standard non-rewritable medium (CD-R, DVD-R, etc.) in formats common at the FIT. In order to allow regular handling, the medium will be securely attached to the printed report.

Supervisor:	**Černocký Jan, doc. Dr. Ing.**, DCGM FIT BUT
Beginning of work:	November 1, 2016
Date of delivery:	May 24, 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

Jan Černocký
Associate Professor and Head of Department

Abstract

This thesis addresses the problem of remote speaker recognition. The accuracy of standard speaker recognition decreases considerably in the presence of far-field data, therefore, we devised two strategies to improve the results. First, we employed a microphone array (purposely positioned set of microphones) that is able to steer a virtual “beam” to the position of the speaker. We also performed system adaptation of different parts of the system (PLDA scoring and i-vector extraction). We have synthesized our training and test data from the standard NIST 2010 data by room simulation and we have shown that both techniques and their combination significantly improve the results. We have also dealt with joint speaker identity and position estimation. While the results in simulated outdoor environment (reverberation-free) are encouraging, the results from interiors (with reverberation) are mixed and require further investigation. Finally, we were able to test our system on a limited amount of real re-transmitted data. While the results for male speakers match the simulation, the results for females are not convincing and need further analysis.

Abstrakt

Tato diplomová práce se zabývá problematikou vzdáleného rozpoznávání mluvčích. V případě dat zachycených odlehlým mikrofonem se přesnost standardního rozpoznávání značně snižuje, proto jsem navrhl dva přístupy pro zlepšení výsledků. Prvním z nich je použití mikrofonního pole (záměrně rozestavené sady mikrofonů), které je schopné nasměrovat virtuální “paprsek” na pozici řečníka. Dále jsem prováděl adaptaci komponent systému (PLDA skórování a extraktoru i-vektorů). S využitím simulace pokojových podmínek jsem syntetizoval trénovací a testovací data ze standardní datové sady NIST 2010. Ukázal jsem, že obě techniky a jejich kombinace vedou k výraznému zlepšení výsledků. Dále jsem se zabýval společným určením identity a pozice mluvčího. Zatímco výsledky ve venkovním simulovaném prostředí (bez ozvěn) jsou slibné, výsledky z interiéru (s ozvěnami) jsou smíšené a vyžadují další prozkoumání. Na závěr jsem mohl systémem vyhodnotit omezené množství reálných dat získaných přehráním a záznamem nahrávek ve skutečné místnosti. Zatímco výsledky pro mužské nahrávky odpovídají simulaci, výsledky pro ženské nahrávky nejsou přesvědčivé a vyžadují další analýzu.

Keywords

Speaker recognition, microphone arrays, beamforming, speaker localization, i-vector, room impulse response

Klíčová slova

Rozpoznávání mluvčího, mikrofonní pole, beamforming, lokalizace mluvčího, i-vektor, impulsní odezva místnosti

Reference

MOŠNER, Ladislav. *Microphone Arrays for Speaker Recognition*. Brno, 2017. Master’s thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Černocký Jan.

Microphone Arrays for Speaker Recognition

Declaration

Hereby I declare that this master's thesis was prepared as an original author's work under the supervision of Jan Černocký. The supplementary information was provided by members of BUT Speech@FIT research group. All the relevant information sources, which were used during preparation of this thesis, are properly cited and included in the list of references.

....
Ladislav Mošner
May 23, 2017

Acknowledgements

I would like to express my sincere thanks to my supervisor Jan Černocký for his valuable advices, shared expertise and positive attitude. I would also like to thank members of BUT Speech@FIT group, especially Oldřich Plchot, Pavel Matějka and Ondřej Glembek for being willing to advise and Kamil Chalupníček for performing retransmission of the data.

Contents

List of Figures

Chapter 1

Introduction

Nowadays, the systems for speaker recognition (SRE) achieve accuracies that merit the attention. It is mainly due to a great effort that was put into the research of such systems. Over the years, many techniques were invented to mitigate the influence of unwanted information present in speech audio recordings, such as a background noise, characteristics of a microphone, coding, etc. As the result, the mathematically sophisticated but usable systems that are based on the i-vector representation of recordings came into existence. Speaker recognition is exploited in multiple areas of human activity. For example forensic applications, search in audio archives and biometric systems rely on the SRE technologies.

However, these systems mostly require signals acquired by close-talk microphones, which restrains systems from being used in far-field scenarios. In such cases, the accuracy of common speaker recognizers decreases substantially. It is a consequence of disturbing room noise and reverberation. To enhance the audio signal that comes from a specific location, one microphone is not enough. A microphone array is a good choice. In a nutshell, the microphone array is a purposely positioned set of microphones. It is capable of steering its look direction and attenuating sounds impinging on the sensors from directions.

Solving the outlined problem is the motivation for this work. First, we will explore how much the SRE accuracy deteriorates with the remote microphones. Then, the possibilities for improvement will be discussed. We will make use of microphone arrays. Another approach to the accuracy improvements is the adaptation of the system to new conditions. Moreover, we will attempt to use the microphone arrays also to localize a speaker in an interior.

To give an overview of current speaker recognition systems, chapter 2 will cover the basic theory associated with particular levels of the processing chain. Microphone arrays along with two methods for steering a look direction will be addressed in chapter 3. In this chapter, a brief summary of speaker localization will be given as well. Chapter 4 will introduce utilized dataset and state a need for data simulation. The next parts will be dedicated to performed experiments. An assessment of accuracy deterioration when audio is recorded with remote microphones will be given in 5. Chapter 6 summarizes application of beamforming and retraining of the system components in order to improve the accuracy. Different approaches to speaker localization and recognition will be introduced in chapter 7. The last chapter 8 deals with a question how much simulated and real data correlate.

Chapter 2

Speaker recognition based on i-vectors

In this work, we use the current state-of-the-art approach to speaker recognition. It is well established and it has been used with some modifications for a few years and it still yields supreme results. The whole system can be seen as a chain consisting of major “blocks”, namely *feature extraction*, *Gaussian mixture universal background model* (UBM), *i-vector extraction*, *probabilistic linear discriminant analysis*, which produces final *scores*. The presented sequence is depicted in Figure 2.1. A basic theoretical background of used methods will be described in this chapter.

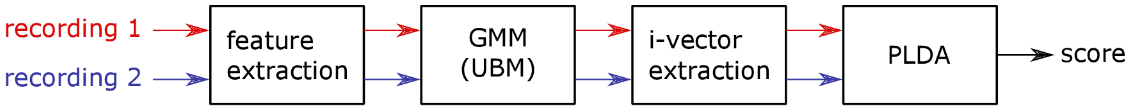


Figure 2.1: Block diagram of the used speaker recognition processing chain.

2.1 Feature extraction

As it is inconvenient to work directly with the raw audio signal, there is a need for conversion to a suitable float-vector representation with lowered dimension, which is the objective of feature extraction. The recordings of our interest can differ in length and contain much more information than just the speech. Therefore, the feature extraction methods are designed to reduce the dimensionality and produce the feature vectors, typically one vector per 10 ms step. The feature vectors must preserve the relevant information for recognition [17].

In speech processing, Mel-Frequency Cepstral Coefficients (MFCC) are very common features. Originally the MFCC features were inspired in human perception experiments, while they also work well in recognition experiments [30]. The extraction may be summarized in the following steps:

- windowed segments (10 ms) of the signal are transformed to the frequency domain using discrete Fourier transform (DFT),

- the spectrum is filtered with band-pass filters distributed uniformly along the Mel-Frequency scale; Mel frequency $M(f)$ is computed from frequency f as follows [30]

$$M(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right), \tag{2.1}$$

- power of each frequency band is computed and logarithm is applied to samples,
- the inverse discrete cosine transform is used for the transformation to Mel quefrency which leads to the acquisition of feature coefficients.

2.2 Gaussian Mixture Modeling

The feature vectors acquired in the previous step are modeled with *Gaussian Mixture Model* (GMM). This is not relevant only to speaker recognition tasks but also to language identification, LVCSR[1], etc. [9].

Gaussian Mixture Model consists of C weighted normal probability distribution functions (PDF). An example of the GMM for 2D features is shown in Figure 2.2. As the feature vectors usually comprises tens of features [25], the normal distribution of the components has to be multivariate. Assuming that $\mathbf{o}$ is an F-dimensional random vector, the PDF value of a single F-dimensional multivariate normal distribution is given as:

$$\mathcal{N}(\boldsymbol{o}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) = \frac{1}{(2\pi)^{\frac{F}{2}}|\boldsymbol{\Sigma}_c|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{o}-\boldsymbol{\mu}_c)'\boldsymbol{\Sigma}_c{}^{-1}(\boldsymbol{o}-\boldsymbol{\mu}_c)}, \tag{2.2}$$

where $\boldsymbol{\mu}_c$[2] is a vector of the *mean*, $\boldsymbol{\Sigma}_c$ is the *covariance matrix*. Based on the previous text, we can now formulate the parameters of the GMM. First, weights form vector $\boldsymbol{w}$. Supervector $\boldsymbol{\mu}$ denotes a supervector obtained by concatenation of per-component mean vectors $\boldsymbol{\mu}_c$, where $c = 1, \ldots, C$. The last parameter $\boldsymbol{\Sigma}$ is a block diagonal matrix, in which the diagonal consists of the covariance matrices $\boldsymbol{\Sigma}_c$ for each component. For convenience, all parameters will be referred to as $\theta = (\boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then the probability density function of the GMM for a feature vector $\boldsymbol{o}$ is given as [9]:

$$\mathcal{G}(\boldsymbol{o}|\theta) = \sum_{c=1}^{C} w_c \mathcal{N}(\boldsymbol{o}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c). \tag{2.3}$$

The GMM can be seen as a *generative probabilistic model* [9], which serves as a generator of features – firstly, component is selected randomly while taking weights w_c into account (prior probabilities of components), then the feature is generated from the corresponding normal distribution (component c). However, in case of the evaluation, the identity of the component is unknown (*hidden variable*), therefore marginalization over components – as in (2.3) – must be performed.

Universal Background Model

The GMM may right serve as a generative classifier. In speaker verification, an *assymetrical* procedure utilizing the GMMs may be used [9]. It implies the need for one model for each

[1] LVCSR stands for Large-Vocabulary Continuous Speech Recognition.

[2] Note that (2.2) is a general formula for PDF of a multivariate normal distribution and subscript c was added to distinguish between parameters of individual mixture components.

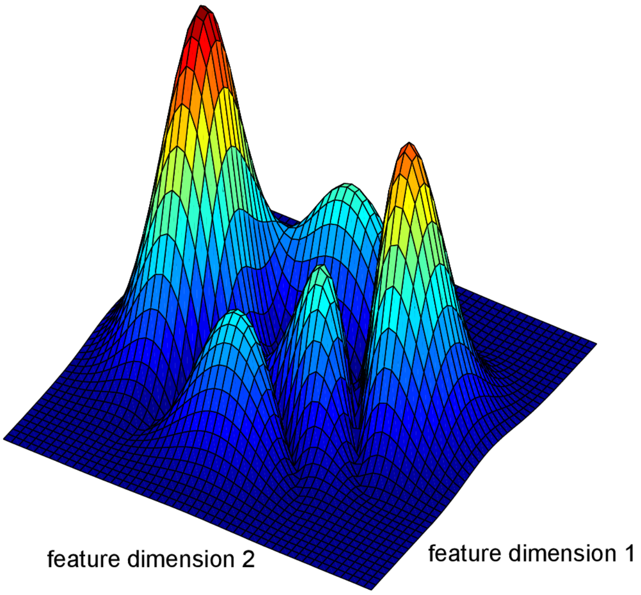


Figure 2.2: An example of 2D GMM. Adapted from [9].

speaker (obtained during *enrollment* phase) and a model that represents "any" speaker – *universal background model* (UBM). Therefore, the UBM is trained in maximum likelihood (ML) [25] manner using *Expectation-maximization* (EM) algorithm 2.2 and a huge amount of data. A speaker dependent GMM is usually obtained by *Maximum a posteriori* (MAP) adaptation of the UBM [23].

In this thesis, however, the GMM will not be used directly for speaker verification (or recognition), but the need for the UBM holds. It will be used in subsequent phases of the processing pipeline (namely during i-vector extraction 2.3) for the statistics extraction.

Expectation-maximization algorithm for training of the universal background model

In this section a brief overview of the *Expectation-maximization* algorithm will be given. A thorough explanation is provided for instance by [25].

Given the parameters of the UBM and the feature vector $\boldsymbol{o}_i$, a posterior probability $p(c|\boldsymbol{o}_i)$, where c denotes the cth mixture component, referred to as $\gamma_{c,i}$ [9] is given by

$$\gamma_{c,i} = \frac{w_c \mathcal{N}(\boldsymbol{o}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{C} w_c \mathcal{N}(\boldsymbol{o}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}. \tag{2.4}$$

It is convenient to work with *sufficient statistics*. Assuming that N feature vectors are extracted (section 2.1) from the utterance, i.e. $\boldsymbol{o}_i$, $i = 1, \dots, N$, the sufficient statistics are computed as [9]

$$N_c = \sum_{i=1}^{N} \gamma_{c,i}, \tag{2.5}$$

$$\boldsymbol{f}_c = \sum_{i=1}^{N} \gamma_{c,i} \boldsymbol{o}_i, \tag{2.6}$$

$$\boldsymbol{S}_c = \sum_{i=1}^{N} \gamma_{c,i} \boldsymbol{o}_i \boldsymbol{o}_i'. \tag{2.7}$$

N_c is a single number called zero-order statistic, vector $\boldsymbol{f}_c$ and matrix $\boldsymbol{S}_c$ are first- and second-order statistics, respectively. EM algorithm for ML estimate is then expressed in the following steps:

Initialization Parametres θ are initialized. Different approaches may be applied (for example K-means [25]).

E step Calculation of the likelihood of the data $\boldsymbol{O}$ (or log-likelihood in practice) based on the actual UBM parameters θ_0. Data matrix $\boldsymbol{O}$ comprises N feature vectors, i.e. $\boldsymbol{O} = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_N]$. The likelihood of $\boldsymbol{O}$ is given by

$$p(\boldsymbol{O}|\theta_0) = \prod_{i=1}^{N} \mathcal{G}(\boldsymbol{o}_i|\theta_0). \tag{2.8}$$

Sufficient statistics, as defined by (2.5), (2.6) and (2.7) are computed of all available data (note that N is not limited to only one recording in this case).

M step Utilizing sufficient statistics, new model parameters are estimated as follows:

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{N_c} \boldsymbol{f}_c, \tag{2.9}$$

$$\hat{\boldsymbol{\Sigma}}_c = \frac{1}{N_c} \boldsymbol{S}_c - \hat{\boldsymbol{\mu}}_c \hat{\boldsymbol{\mu}}_c', \tag{2.10}$$

$$\hat{w}_c = \frac{N_c}{N}. \tag{2.11}$$

Check the convergence Based on the likelihood or the number of steps, it is decided whether to continue with the next iteration (E step again) or stop.

2.3 Supervectors of the GMM and i-vectors

As was mentioned in chapter 2.2, a concatenation of GMM's mean vectors creates the vector $\boldsymbol{\mu}$, which is referred to as *supervector*. Every recording can be represented by the supervector. It includes information not only about the speaker but also about the channel (including conditions occurring during recording). Due to this fact and a high dimensionality (CF, where C is a number of GMM components and F stands for the dimensionality of the GMM space), they are not suitable features.

Joint Factor Analysis (JFA) [12] was a state-of-the-art method by 2008. It tended to model speaker-dependent and channel-dependent spaces while expressing supervector with speaker and channel factors. Later, it was shown that also channel subspace includes information about speakers, which laid the foundation of a new approach.

I-vectors are low dimensional vectors (in comparison to supervectors), which represent variable length recordings in a uniform, fixed-length way. In the next part of the chapter, the i-vector approach will be briefly summarized.

Model and total variability space

A supervector $\boldsymbol{m}$ containing speaker and channel information can be expressed as

$$\boldsymbol{m} = \boldsymbol{\mu} + \boldsymbol{T}\boldsymbol{\phi}, \tag{2.12}$$

where $\boldsymbol{\mu}$ denotes the supervector of the UBM, $\boldsymbol{T}$ is a $CF \times I$ matrix (I is the dimension of i-vectors). It defines I-dimensional subspace of the supervector space called *total variability space* [7]. $\boldsymbol{T}$ matrix is sometimes referred to as *i-vector extractor* [9]. Finally, $\boldsymbol{\phi}$ is a random vector of size I.

The distribution of supervectors $\boldsymbol{m}$ is normal, i.e. $\boldsymbol{m} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{T}\boldsymbol{T}')$. Given the recording r a random vector $\boldsymbol{\phi}$ has normal distribution $\boldsymbol{\phi} \sim \mathcal{N}(\hat{\boldsymbol{\phi}}_r, \boldsymbol{L}_r^{-1})$, where the mean vector $\hat{\boldsymbol{\phi}}_r$ is an estimate of the *i-vector*. $\boldsymbol{L}_r$ is the precision matrix of the posterior distribution of $\boldsymbol{\phi}$ [9] given as

$$\boldsymbol{L}_r = \boldsymbol{I} + \boldsymbol{T}'\boldsymbol{N}_r\boldsymbol{\Sigma}^{-1}\boldsymbol{T}, \tag{2.13}$$

where $\boldsymbol{\Sigma}$ is a block-diagonal matrix defined in section 2.2. $\boldsymbol{N}_r$ is also a block-diagonal matrix specific for the recording r and is given as

$$\boldsymbol{N}_r = \begin{bmatrix} N_{1,r}\boldsymbol{I} & 0 & \dots & 0 \\ 0 & N_{2,r}\boldsymbol{I} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & N_{C,r}\boldsymbol{I} \end{bmatrix}, \tag{2.14}$$

where $N_{c,r}$ is zero-order statistic defined in equation (2.5) for the recording r. As it was indicated, i-vectors are feature vectors; each i-vector represents one recording regardless of the duration. They are computed as follows

$$\hat{\boldsymbol{\phi}}_r = \boldsymbol{L}_r^{-1}\boldsymbol{T}'\boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{f}}_r \tag{2.15}$$

where $\bar{\boldsymbol{f}}_r$ is a vector

$$\bar{\boldsymbol{f}}_r = \begin{bmatrix} \boldsymbol{f}_{1,r} \\ \boldsymbol{f}_{2,r} \\ \vdots \\ \boldsymbol{f}_{C,r} \end{bmatrix} + \boldsymbol{N}_r\boldsymbol{\mu}. \tag{2.16}$$

The symbol $\boldsymbol{f}_{c,r}$ stands for the first-order statistic from (2.6) computed for recording r. Derivations of previous equations are available in [29, 9]. In [29] Zhang also describes the procedure to train the i-vector extractor that employs iterative EM algorithm.

2.4 Probabilistic Linear Discriminant Analysis

Probabilistic linear discriminant analysis (PLDA) is a method similar to joint factor analysis (JFA) [12]. Originally it was proposed by Prince and Elder [21] for the task of face recognition. Then it was successfully adopted in a field of speaker recognition [13]. More specifically, it is applied to perform classification in the space of i-vectors.

Model

Because PLDA is based on JFA, the model is very similar to that of JFA. The i-vector $\hat{\boldsymbol{\phi}}_{r,s}$ of recording r in which utterance of speaker s is present is modeled as [5, 26]:

$$\hat{\boldsymbol{\phi}}_{r,s} = \boldsymbol{\mu}_{ivec} + \boldsymbol{V}\boldsymbol{y}_s + \boldsymbol{U}\boldsymbol{x}_{r,s} + \boldsymbol{z}_{r,s}, \tag{2.17}$$

where $\boldsymbol{\mu}_{ivec}$ is i-vectors' mean vector, $\boldsymbol{V}$ is a loading matrix defining a subspace of i-vectors' space characterizing speaker variability. $\boldsymbol{U}$ is also a loading matrix, but its columns define a subspace of channel variability. Hidden variables $\boldsymbol{y}_s$, $\boldsymbol{x}_{r,s}$ represent the speaker and the channel, respectively. Variable $\boldsymbol{z}_{r,s}$ (which is not hidden) denotes a residual noise. For all variables, normal prior distributions are expected:

$$\boldsymbol{y}_s \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{2.18}$$

$$\boldsymbol{x}_{r,s} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{2.19}$$

$$\boldsymbol{z}_{r,s} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{D}^{-1}), \tag{2.20}$$

where $\boldsymbol{D}$ is a diagonal precision matrix. It is convenient to work with centered i-vectors [5]. Then the model is simplified as follows:

$$\hat{\boldsymbol{\phi}}_{r,s} = \boldsymbol{V}\boldsymbol{y}_s + \boldsymbol{U}\boldsymbol{x}_{r,s} + \boldsymbol{z}_{r,s}. \tag{2.21}$$

According to [25], (2.21) may be divided into two parts: $\boldsymbol{s}_s = \boldsymbol{V}\boldsymbol{y}_s$ and $\boldsymbol{c}_{r,s} = \boldsymbol{U}\boldsymbol{x}_{r,s} + \boldsymbol{z}_{r,s}$. Then their priors are

$$\boldsymbol{s}_s \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{V}\boldsymbol{V}'), \tag{2.22}$$

$$\boldsymbol{c}_{r,s} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{U}\boldsymbol{U}' + \boldsymbol{D}^{-1}). \tag{2.23}$$

Matrix $\boldsymbol{V}\boldsymbol{V}'$ is referred to as across-class covariance $\boldsymbol{\Sigma}_{AC}$ and $\boldsymbol{U}\boldsymbol{U}'$ as within-class covariance $\boldsymbol{\Sigma}_{WC}$ [9]. The objective of the PLDA training is to estimate the parameters of the model, i.e matrices $\boldsymbol{V}$, $\boldsymbol{U}$ and $\boldsymbol{D}$. A complete descriptions of an EM procedure with derivations are for instance in [5, 26].

Trial scoring

Let there be two i-vectors $\hat{\boldsymbol{\phi}}_1$, $\hat{\boldsymbol{\phi}}_2$ in a trial. A trial score is defined as a log-likelihood ratio of two hypotheses:

- $\mathcal{H}_1$: i-vectors belong to the same speaker,
- $\mathcal{H}_2$: i-vectors correspond to two different speakers.

The score is mathematically expressed as [9]:

$$s(\hat{\boldsymbol{\phi}}_1, \hat{\boldsymbol{\phi}}_2) = \log \frac{p(\hat{\boldsymbol{\phi}}_1, \hat{\boldsymbol{\phi}}_2 | \mathcal{H}_1)}{p(\hat{\boldsymbol{\phi}}_1, \hat{\boldsymbol{\phi}}_2 | \mathcal{H}_2)}. \tag{2.24}$$

We can reformulate the hypotheses so that they are expressed in terms of two models. The first model (corresponding with the hypothesis $\mathcal{H}_1$) represents the situation in which both the i-vectors share speaker dependent hidden variable $\boldsymbol{y}_{12}$. The channel dependent variables may differ, thus we will refer to them as $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The second model represents the fact that speakers differ by two distinct variables $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. Then the hypotheses taking the models into account may be rewritten as

$$\begin{aligned}
\mathcal{H}_1 : \begin{bmatrix} \hat{\boldsymbol{\phi}}_1 \\ \hat{\boldsymbol{\phi}}_2 \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\mu}_{ivec} \\ \boldsymbol{\mu}_{ivec} \end{bmatrix} + \begin{bmatrix} \boldsymbol{V} & \boldsymbol{U} & \boldsymbol{0} \\ \boldsymbol{V} & \boldsymbol{0} & \boldsymbol{U} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_{12} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix}, \\
\mathcal{H}_2 : \begin{bmatrix} \hat{\boldsymbol{\phi}}_1 \\ \hat{\boldsymbol{\phi}}_2 \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\mu}_{ivec} \\ \boldsymbol{\mu}_{ivec} \end{bmatrix} + \begin{bmatrix} \boldsymbol{V} & \boldsymbol{0} & \boldsymbol{U} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{V} & \boldsymbol{0} & \boldsymbol{U} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix}.
\end{aligned} \tag{2.25}$$

Employing the formula for probability $P\Big(\begin{bmatrix} \hat{\boldsymbol{\phi}}_1 \\ \hat{\boldsymbol{\phi}}_2 \end{bmatrix} | \mathcal{H}\Big)$ defined in [20] and substituting it to (2.24) we will obtain resulting equation:

$$\begin{aligned}
s(\hat{\boldsymbol{\phi}}_1, \hat{\boldsymbol{\phi}}_2) = \log \mathcal{N}\Bigg(\begin{bmatrix} \hat{\boldsymbol{\phi}}_1 \\ \hat{\boldsymbol{\phi}}_2 \end{bmatrix} | \begin{bmatrix} \boldsymbol{\mu}_{ivec} \\ \boldsymbol{\mu}_{ivec} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{AC} + \boldsymbol{\Sigma}_{WC} & \boldsymbol{\Sigma}_{AC} \\ \boldsymbol{\Sigma}_{AC} & \boldsymbol{\Sigma}_{AC} + \boldsymbol{\Sigma}_{WC} \end{bmatrix} \Bigg) - \\
\log \mathcal{N}\Bigg(\begin{bmatrix} \hat{\boldsymbol{\phi}}_1 \\ \hat{\boldsymbol{\phi}}_2 \end{bmatrix} | \begin{bmatrix} \boldsymbol{\mu}_{ivec} \\ \boldsymbol{\mu}_{ivec} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{AC} + \boldsymbol{\Sigma}_{WC} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{AC} + \boldsymbol{\Sigma}_{WC} \end{bmatrix} \Bigg).
\end{aligned} \tag{2.26}$$

Chapter 3

Microphone arrays and enhancement of speech signal

When it comes to processing of speech signals captured remotely, utilization of a single microphone is inconvenient, as it records noises and other unwanted speech coming from different directions. In this case, the accuracy of (not only) speaker recognition decreases. To handle the problem, microphone arrays are a good choice.

A microphone array can be interpreted as a *spatial filter*, which is capable of enhancing a signal coming from a specific direction, while it attenuates the noise and competitive speech coming from other directions [14]. The filter is described by *directivity pattern* [18] or *beam pattern* [14], which specifies the array response as a function of frequency and direction of arrival. For a uniform linear array (ULA)[1] and a specific frequency, the directivity pattern is depicted in Figure 3.1. The lobe around the maximum is called the *main lobe*, other lobes are *sidelobes* [18].

Also, a different point of view exists. It can be considered that the microphone array samples a continuous *passive aperture* [18]. The aperture is a spatial region, which transmits or receives waves. In the case of passive aperture, it only receives them. Hence, analogy to Shannon sampling theorem exists for a spatial domain as well. *Spatial sampling theorem* is defined as

$$d < \frac{\lambda_{min}}{2}, \tag{3.1}$$

where λ_{min} is a minimum wavelength present in a signal and d is a spacing between microphones.

So far only a uniform linear array was mentioned. However, not only the linear shape is used. Also, spherical[2], circular or planar arrays exist. Additionally, McCowan suggests a non-uniform array comprising sub-arrays for particular frequency bands [19].

Among different applications of microphone arrays, there are two of them that are of our interest. Namely *beamforming* and *speaker localization*. Those will be briefly described in the following sections.

[1] Uniform linear array stands for a microphone array that consists of microphones positioned on a line. The spacing between neighboring microphones is uniform.

[2] For instance [14] compares linear and spherical arrays in terms of WER – word error rate).

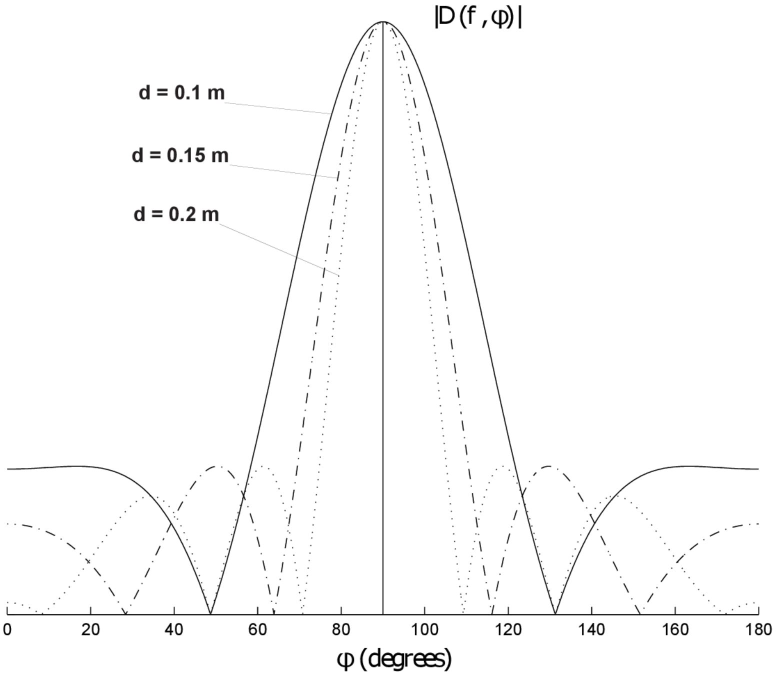


Figure 3.1: Directivity patterns for a uniform linear array. Spacing d between microphones affects the shape. Adapted from [18].

3.1 Beamforming

Generally, *beamforming techniques* are methods that implement *shaping* and *steering* of a directivity pattern in order to enhance the sound produced by the desired source of audio. By steering, we mean re-positioning of the main lobe to a specific angle. In other words, beamforming techniques try to steer *look direction* to the source of interest.

In the following text, we will consider the array of arbitrary geometry consisting of M microphones. Let $S(j\omega)$ be the discrete-time Fourier transform (DTFT) of a source signal. For every microphone m, $m = 1, \ldots, M$, the channel between it and the source forms a filter, thus can be described by impulse response as well as by frequency characteristic $G_m(j\omega)$. Considering an additional noise $N_m(j\omega)$, which can differ for all the microphones, the signal impinging on each microphone is given as [27]:

$$Y_m(j\omega) = G_m(j\omega)S(j\omega) + N_m(j\omega) = X_m(j\omega) + N_m(j\omega). \tag{3.2}$$

The aim is to recover the signal component of so-called *reference microphone* m_0, i.e. $X_{m_0}(j\omega)$. This is achieved by applying a linear filter $\boldsymbol{h}_{m_0}(j\omega)$ to the vector of observed signals $\boldsymbol{y}(j\omega) = \begin{bmatrix} Y_1(j\omega) & Y_2(j\omega) & \ldots & Y_M(j\omega) \end{bmatrix}'$. The result of filtering is given as

$$Z(j\omega) = \boldsymbol{h}_{m_0}^H(j\omega)\boldsymbol{y}(j\omega), \tag{3.3}$$

where superscript ^{H} stands for transpose-conjugation.

Many beamforming techniques has been proposed, but all of them are entirely described by $\boldsymbol{h}_{m_0}^H(j\omega)$ [14]. Two well-known methods – *delay-and-sum* (DS) and *minimum variance distortionless response* (MVDR) – will be introduced in the next sections.

Delay-and-sum

Delay-and-sum is the simplest and most intuitive beamforming method. It performs well in case that the source signal is not disrupted, as it makes no assumptions about the noise. Despite its simplicity, it yields reasonable results.

It utilizes the fact that due to a propagation delay, the original sound wave arrives at different instants of time to each microphone. When a *time difference of arrival* (TDOA) is known, then signals recorded by microphones can be shifted accordingly. It causes the alignment of the desired signal, while other components included in the audio remain unaligned, thus will be attenuated. The principle is is displayed in Figure 3.2. In practice, beamforming is performed in the frequency domain. Hence, the time delay is achieved by phase shift. It leads to the definition of the *manifold vector* [14]:

$$\boldsymbol{v}(j\omega) = \begin{bmatrix} e^{-j\omega\tau_1} & e^{-j\omega\tau_2} & \dots & e^{-j\omega\tau_M} \end{bmatrix}', \tag{3.4}$$

where τ_m, $m = 1, \dots, M$ is a time delay imposed on the signal from mth microphone. The linear filter $\boldsymbol{h}_{m_0}^{(DS)}(j\omega)$ is given as:

$$\boldsymbol{h}_{m_0}^{(DS)}(j\omega) = \frac{\boldsymbol{v}(j\omega)}{M}. \tag{3.5}$$

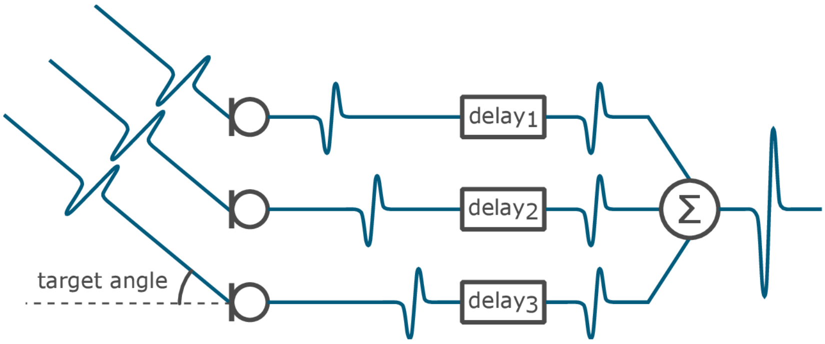


Figure 3.2: The principle of beamforming. Signals that come from the direction of interest are aligned and hence amplified.

Minimum variance distortionless response

Minimum variance distortionless response is a beamforming method that is meant to suppress spatially correlated noise [14], hence it improves the performance of DS beamformer. Linear filter $\boldsymbol{h}_{m_0}^{(MVDR)}(j\omega)$ is, in this case,a result of an optimization problem which minimizes the residual noise of the beamformer output subject to the constraint that prevents from distortion of the desired signal (so-called *distortionless constraint*). The derivation can be found in [27]. The resulting filter is given as follows:

$$\boldsymbol{h}_{m_0}^{H(MVDR)}(j\omega) = \frac{\boldsymbol{\Sigma}_N^{-1}\boldsymbol{v}(j\omega)}{\boldsymbol{v}^H(j\omega)\boldsymbol{\Sigma}_N^{-1}\boldsymbol{v}(j\omega)} \tag{3.6}$$

where $\boldsymbol{v}(j\omega)$ is defined the same way as in (3.4), $\boldsymbol{\Sigma}_N = \mathcal{E}\{\boldsymbol{n}(j\omega)\boldsymbol{n}^H(j\omega)\}$ is the noise covariance matrix. Vector $\boldsymbol{n}(j\omega)$ comprises spectra of additional noises from (3.2): $\boldsymbol{n}(j\omega) = \begin{bmatrix} N_1(j\omega) & N_2(j\omega) & \dots & N_M(j\omega) \end{bmatrix}'$.

3.2 Speaker localization

Microphone arrays can also be used for the estimation of speaker location. To do so, two main approaches exist:

- aimed at finding a position providing maximum *steered response power* (SRP),
- based on *time difference of arrival* (TDOA).

The first approach – also known as beamforming or maximum likelihood – defines a function, which assigns each spatial point a value. The position of the speaker is then given as the location with maximum assigned value [16]. An important algorithm implementing this approach is *steered response power with phase transform* (SRP-PHAT) [8].

The latter group of methods is based on TDOAs. It means that there must be a phase preceding location estimation, i.e. estimation of time differences. Note that estimation of TDOA is also necessary in the case of DS and MVDR. Many TDOA estimation methods exist but they differ in computational complexity and accuracy. An overview of well-known approaches is given in [6]. The *cross-correlation* is the most straightforward. As its extension, a *generalized cross-correlation* using different types of weighting, such as phase transform (PHAT), was developed. When TDOAs are known, algorithms from this category can be utilized. Spherical intersection, spherical interpolation [14] or linear intersection [3, 4] are examples of existing methods.

Linear intersection

Linear intersection [3] is a closed-form location estimator, which provides suboptimal localization data. In return, it allows for real-time processing. It makes use of TDOAs; hence they must be estimated beforehand. Let $\boldsymbol{m}_{i1}$ and $\boldsymbol{m}_{i2}$ be 3D coordinates of a microphone pair for $i = 1, \ldots, N$. Then the correct time difference of arrival associated with the source with coordinates $\boldsymbol{s}$ and the pair is given as

$$T(\{\boldsymbol{m}_{i1}, \boldsymbol{m}_{i2}\}, \boldsymbol{s}) = \frac{|\boldsymbol{s} - \boldsymbol{m}_{i1}| - |\boldsymbol{s} - \boldsymbol{m}_{i2}|}{c}, \tag{3.7}$$

where c is the speed of sound. However, in practice, TDOA estimate of τ_i does not necessarily equals the ideal $T(\{\boldsymbol{m}_{i1}, \boldsymbol{m}_{i2}\}, \boldsymbol{s})$. Therefore, this method computes multiple alternative positions that are then merged respecting their likelihoods.

The linear intersection algorithm expects that microphones are placed in a far field of the source; hence wavefront can be approximated by a plane. Then, for the pair of microphones $\{\boldsymbol{m}_{i1}, \boldsymbol{m}_{i2}\}$, we can assume that the source is located somewhere on a surface of a cone whose vertex is at the midpoint of the pair. Its axis of symmetry equals the line connecting of the two microphones. The apex angle corresponds to delay τ_i associated with $\{\boldsymbol{m}_{i1}, \boldsymbol{m}_{i2}\}$. Having two pairs of microphones $\{\boldsymbol{m}_{i1}, \boldsymbol{m}_{i2}\}$, $\{\boldsymbol{m}_{j1}, \boldsymbol{m}_{j2}\}$ whose connecting lines are orthogonal and mutually bisecting, two cones sharing the vertex can be constructed. The intersection of the cones forms two lines. One of them is unrealistic due to the fact, that the microphones are placed on a wall. The remaining line is called *bearing line* and should be steered to a sound source. It is parametrically expressed as

$$\boldsymbol{l}_{ij} = r_{ij}\boldsymbol{a}_{ij} + \boldsymbol{m}_{ij}, \tag{3.8}$$

where r_{ij} is a parameter, $\boldsymbol{a}_{ij}$ is a slope and $\boldsymbol{m}_{ij}$ is a center of the quartet of microphones. When we have two bearing lines l_{ij} and l_{fg}, we can determine two points that are closest to each other and each of them lies on a different line. There is rarely an intersection because it is unlikely that lines will cross in one point. Then the approximate intersection is obtained from the result of an overdetermined system

$$r_{ij}\boldsymbol{a}_{ij} - r_{fg}\boldsymbol{a}_{fg} = \boldsymbol{m}_{fg} - \boldsymbol{m}_{ij} - d_{ij,fg}(\boldsymbol{a}_{ij} \times \boldsymbol{a}_{fg}), \tag{3.9}$$

where $d_{ij,fg}$ is the distance between the closest points. Depending on the number of available microphone quartets, the same number of potential locations is determined. The final estimation is computed as a weighted average of them, while weights are defined as a value of a Gaussian function for the difference between τ_i and the delay associated with estimated position. Figure 3.3 visually demonstrates estimation of the position given approximate intersections.

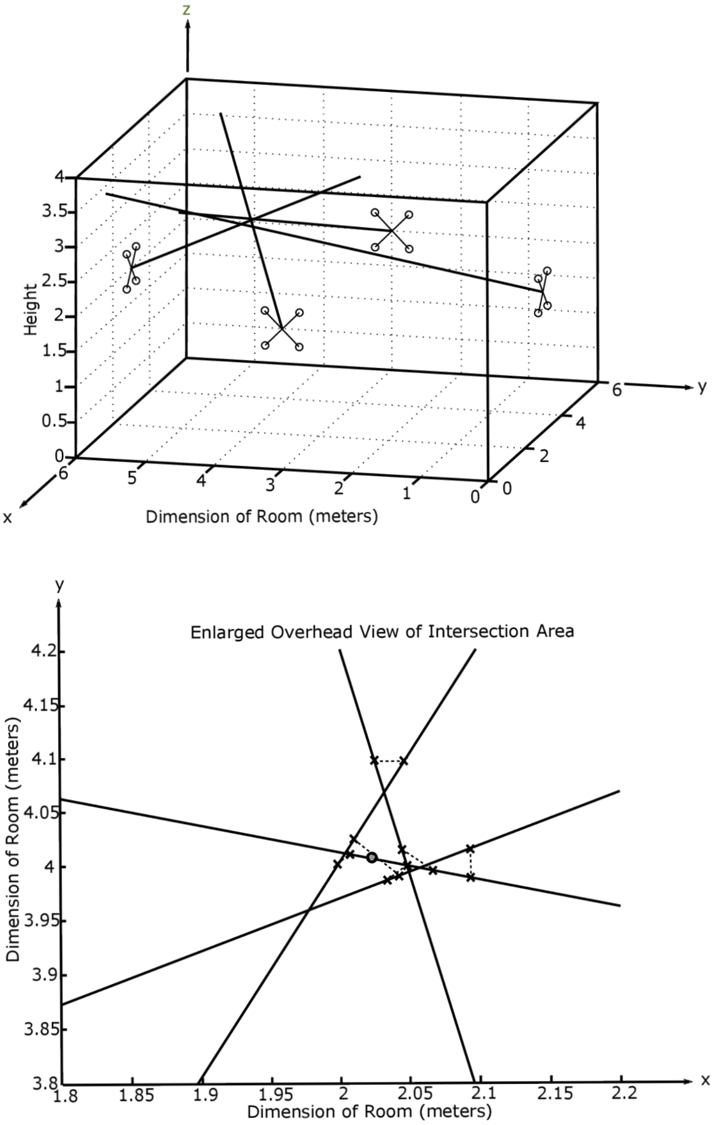

z
Height
4
3.5
3
2.5
2
1.5
1
0.5
0
x
6
5
4
3
2
1
0
Dimension of Room (meters)
y
0
2
4
6
y
Enlarged Overhead View of Intersection Area
Dimension of Room (meters)
4.2
4.15
4.1
4.05
4
3.95
3.9
3.85
3.8
1.8
1.85
1.9
1.95
2
2.05
2.1
2.15
2.2
x
Dimension of Room (meters)

Figure 3.3: Visualization of approximate intersections of bearing lines associated with quartets of microphones. Adapted from [3].

Chapter 4

Dataset

In this chapter, the datasets that were used throughout the experiments will be introduced. Moreover, we will state, why data simulation was needed. There are naturally multiple techniques that can be employed. Therefore, some of the advantages and disadvantages will be introduced. Finally, the method of our choice will be mentioned.

4.1 NIST Year 2010 Speaker Recognition data

As we study the influence of distortion introduced by room acoustics, the need for both clean recordings and their noisy counterparts is obvious. However, there is no available multichannel SRE dataset. We, therefore, decided to use the data released for NIST Year 2010 Speaker Recognition evaluations. The dataset comprises many conditions of training and test recordings, including close-talk microphones and phone calls. However, not all of them are suitable for our problem. Overall, 9 evaluation conditions are defined [1] – these represent subsets of the trials in the core test. For our purposes, condition 1 was chosen – all trials involving interview speech from the same microphone in training and test. We chose it because the data should be somewhat clean, as they will undergo further processing in order to obtain their noisy versions. Since the duration of all the test data is about 200 hours, their retransmission would require resources we do not have at our disposal. Hence, simulation of the rooms is required. Even though the data for evaluation are from the NIST Year 2010 Speaker Recognition task dataset, the SRE system was trained with the data from other seasons of NIST evaluations.

All the recordings are sampled at 8 kHz frequency. They are stored as 16-bit μ-law signals.

4.2 Simulation of the data

The previous section explained the necessity of the test data simulation. There is one more reason. In the experiments, we also attempted to adapt the SRE system to new conditions. To do so, we needed to augment the training data with distorted recordings. The SRE training datasets are typically huge ($>$ 1000 hours). Thus, a real retransmission (or even recording) would not be feasible in a reasonable time at all. Therefore, the data were obtained by simulation of room acoustics as well.

To perform the simulation the tool *RIR Generator* [10] – room impulse response generator – by E. Habets was employed. It is based on the principle of *image method* [2].

Image method relies on ray acoustics, hence uses rays instead of sound waves and omits interference and diffraction. It is significant simplification that also results in less demanding computations compared to physically based approaches, such as Finite Element Method (FEM) or Boundary Element Method (BEM) [10]. On the other hand, more sophisticated methods get closer to real-world conditions.

Chapter 5

Experiments with clean and reverberated data

In this chapter, an overview of used speaker recognition (SRE) system along with its parameters will be given. This model was used to process the reference data and also the trials including modified audio recordings. We will also present experiments that were meant to explore the effect of the room acoustics on the SRE accuracy.

5.1 Baseline

The speaker recognition system that we use comprises all the components specified in chapter 2. 60-dimensional MFC coefficients were used as the features. With them, the UBM comprising 2048 components was trained. Subsequently, the UBM was used for sufficient statistics computation. In the next phase, an i-vector extraction based on the statistics took place. The 600-dimensional vectors were projected to the 200-dimensional space using *Linear Discriminant Analysis* (LDA). The latent variables of PLDA model are of the same dimension.

As a part of the thesis, the scripts for the UBM, i-vector extractor, and PLDA training were written in MATLAB. They were compared with their python equivalents used by BUT Speech@FIT group. For a smaller amount of data, the parameters of trained UBMs were the same, but our solution used likelihoods instead of log-likelihoods, which could lead to numerical instability. Thus, the python script was used for the training of the SRE system UBM. As far as the i-vector extractor estimation is considered, our solution does not include the minimum divergence (MD) step, hence convergence is slower. For convenience we used the python solution again. Results of the PLDA training were comparable even for a large amount of data. Therefore, the SRE system comprises matrices trained with our script.

To be able to compare results obtained in experiments, the reference ones are needed. They were obtained on condition one of the NIST Year 2010 Speaker recognition task for the reasons stated in section 4.1. As a metric, we will use the *equal error rate* (EER) and *minimum detection cost functions* defined for NIST Speaker Recognition Evaluations in years 2008 (DCF_{08}^{min}) and 2010 (DCF_{10}^{min}) [1]. In terms of these metrics, the performance of the system is shown in Table 5.1.

Table 5.1: Reference results – condition 1 of NIST Year 2010 Speaker Recognition evaluation (clean data).

	females	**males**
EER [%]	2.070	0.607
$\mathbf{DCF}_{08}^{min}$	0.100	0.044
$\mathbf{DCF}_{10}^{min}$	0.350	0.182

5.2 Impact of the simulated room

Using RIR Generator, we simulated the acoustics of a hall with dimensions 8 × 10 × 5 m and a room: 4 × 4.5 × 3 m. We will refer to them as “the hall” and “the room”, respectively. The source was situated at the position (7, 9, 2) m in “the hall” and at (2, 2.25, 1.7) m in “the” room (the origin of a coordinate system is in one of the lower corners). Then we used a modeled microphone array of 8 hypercardioid microphones that were distributed along two parallel two meters long lines 0.67 m apart. It was placed on the smaller wall closer to the origin. The simulated rooms are displayed in Figures 5.1 and 5.2. Obtained impulse responses (one for every microphone) were convolved with the data needed for the NIST 2010 SRE condition 1 evaluation. This way we acquired the signals that should correlate with those recorded in the specified rooms.

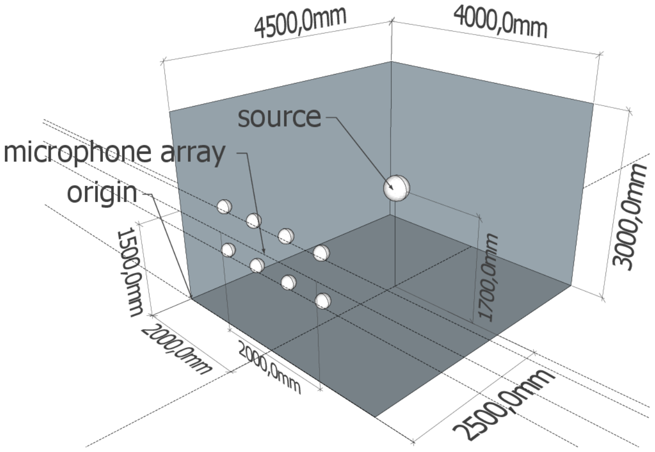


Figure 5.1: Model of the small simulated room (4 × 4.5 × 3 m).

In the first experiment we wanted to discover the deterioration when the original SRE is used for far-field recordings. We, therefore, prepared the test dataset in this way: for every recording of the original dataset a random microphone (out of eight) was chosen and its simulated output was added to the set of recordings to test. Using the SRE system specified in 4.1, we evaluated the condition 1 with those data. The results are given in Tables 5.2 and 5.3. Two situations were taken into consideration. In the first one, the enrollment and test conditions matched. In the second, enrollment recordings were clean. We can see significant degradation of the accuracy. Moreover, the larger the room is, the worse results are obtained. In the next chapter, we will explore various possibilities to make the difference between results of presented evaluations smaller.

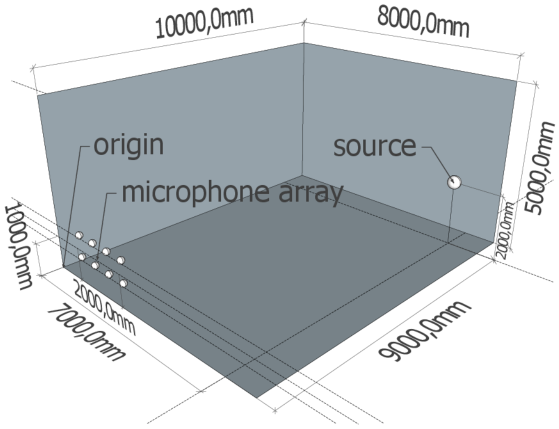


Figure 5.2: Model of the large simulated room (8 × 10 × 5 m).

Table 5.2: NIST Year 2010 Speaker Recognition condition 1 evaluation results when simulated outputs of randomly chosen microphones were used. Enrollment conditions matched test conditions.

	the room		**the hall**	
	females	**males**	**females**	**males**
EER [%]	16.251	7.887	19.741	10.925
$\mathbf{DCF}_{08}^{min}$	0.724	0.393	0.879	0.580
$\mathbf{DCF}_{10}^{min}$	0.968	0.843	0.979	0.922

Table 5.3: NIST Year 2010 Speaker Recognition condition 1 evaluation results when simulated outputs of randomly chosen microphones were used. Enrollment recordings were clean.

	the room		**the hall**	
	females	**males**	**females**	**males**
EER [%]	10.824	6.933	10.622	6.471
$\mathbf{DCF}_{08}^{min}$	0.513	0.312	0.511	0.294
$\mathbf{DCF}_{10}^{min}$	0.939	0.734	0.937	0.702

Chapter 6

Experiments with beamforming and model adaptation

In this chapter, we will roughly divide a processing chain of speaker recognition presented in Figure 2.1 into two parts. The first one will include preprocessing that is applied only to input recordings, the second will consist of the rest of the system. The reason for the division is that we will explore consequences of modifications of these parts separately. Later on, we will also change them both at the same time. In the former case, we will apply beamforming methods to raw recordings. Thus, this step can be understood as a preprocessing of audio before feature extraction. The latter case is more complex as it incorporates many stages. The adaptation of the system for reverberated data will be our aim.

6.1 Delay-and-sum beamforming

To perform delay-and-sum beamforming, we created our own implementation in MATLAB. The steps that the algorithm performs in order to get a single output given multiple input signals are shown in Figure 6.1. In the next part, we will describe the pipeline in more detail:

Division of the recordings into frames

In practice, one cannot assume that the time shift between signals from the microphone array remains the same during the whole recording. It is either consequence of the speaker movement or a need for buffered (real-time) processing. Therefore, we split input recordings into overlapping frames. The length of the frames is 500 ms and the overlap is 250 ms. In order to cut out a frame of the signal, we use Hann (sometimes called Hanning) window. The choice of this type of window is twofold: first, we will transform the windowed frame into frequency and the frequency characteristic of Hann window has better properties than that of the rectangular window. Moreover, in the last phase of the algorithm, we will sum the frames. A useful property of Hann window is that when the shift equals the half of a window length, the sum of non-zero values from function range of two shifted windows is one.

Transformation into frequency domain

We recall that in section 3.1, the linear filter that describes delay-and-sum was defined in the frequency domain. In the case of delay-and-sum, it is also possible to perform

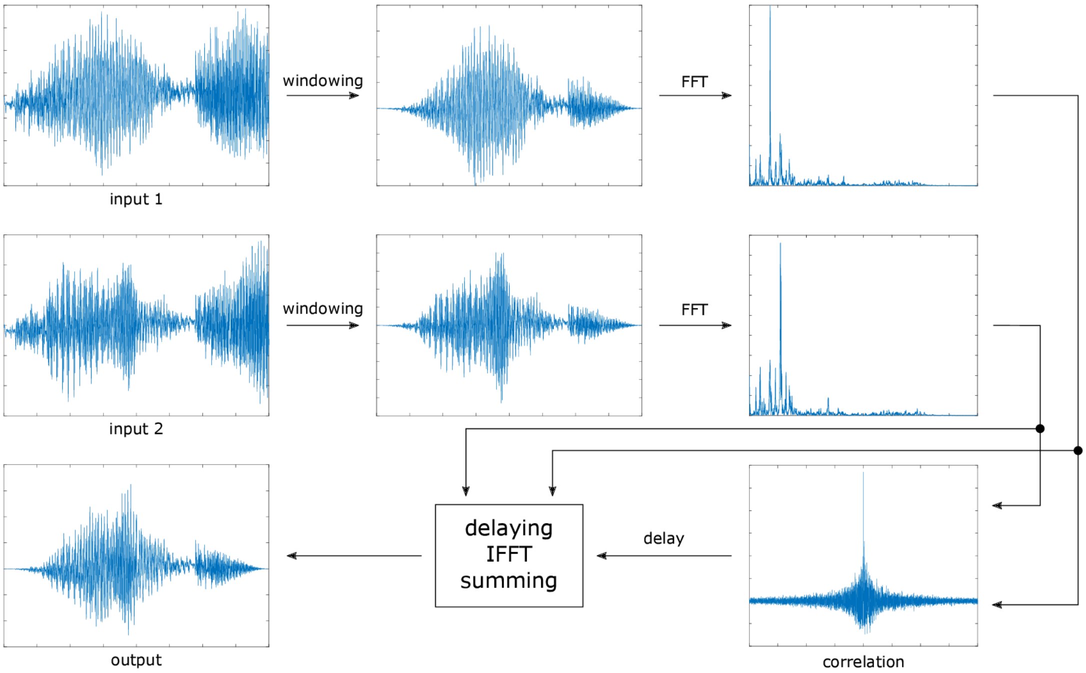


Figure 6.1: Steps performed during delay-and-sum beamforming.

computations in the time domain. However, due to subsequent delay estimation, it is more convenient to work with spectra. Hence, we apply fast Fourier transform (FFT) to every frame.

Time difference of arrival estimation

The position of the speaker of interest correlates with time differences of a sound wave arrivals at the microphones. In order to perform delay-and-sum, we must estimate shifts of signals that are caused by TDOAs. As we work with simulated rooms and know the exact positions of the microphones and the speaker, the computation is easy. However, in a real-world scenario, the information about a sound source is unknown; hence shifts need to be estimated. In our approach, we choose one microphone – the reference one. The shift between the reference signal and the signal from another microphone corresponds to the maximum of a function that is defined as the inverse Fourier transform of generalized cross-correlation with phase transform (GCC-PHAT). Let $Y_{m_0}(f)$ and $Y_i(f)$ be the Fourier transform of the reference and ith signal, respectively. Then GCC-PHAT is given as

$$GCC\text{-}PHAT = \frac{Y_{m_0}(f)Y_i^*(f)}{|Y_{m_0}(f)Y_i^*(f)|}, \tag{6.1}$$

where * denotes the complex conjugate.

Applying delays and summation

At this stage, the approximate delays $\tau_1, \ldots, \tau_M$ are known, thus we can create manifold vector (3.4). Then, filter (3.5) can be applied to the input signals. Thereafter, the output is transformed to the time domain using inverse Fourier transform. This way we obtain 500 ms long signal that is added to the end of the resulting signal with appropriate overlap.

Preprocessing the test data by delay-and-sum

We used our script that performs delay-and-sum beamforming to process multichannel data from the microphone array that was simulated in two types of interior – "the room" and "the hall". We also wanted to make use of the information about the exact positions of the microphone array and the speaker. It can help us realize how errors introduced by incorrect delay estimation affect the accuracy of recognition. To distinguish between the two delay-and-sum options, we will refer to them as follows:

DS delay-and-sum that uses GCC-PHAT to estimate shifts between recordings,

DS_known_pos delay-and-sum that makes use of known positions of the microphones and the source.

We also wanted to investigate the effect of the acoustic conditions during the enrollment. Therefore, we will show two results in all subsequent experiments. In the first one, enrollment and test conditions will match. In the second one, the enrollment data will be clean – data from NIST 2010 speaker recognition task.

The results in terms of EER are shown in Figures 6.2 and 6.3. Note that neither *DS* nor *DS_known_pos* affects the accuracy when the test data are clean. It is due to a fact, that the SRE system remained the same and only preprocessing of multichannel data was performed. Hence, single-channel clean recordings were not affected. Regarding enrollment conditions, we can see that a match introduces more significant deterioration of the accuracy. Also, the larger the room is, the more harmful the impact seems to be. It means that when using only one far-field microphone, it is natural to advise not to record the enrollment data in the room. However, when it comes to beamforming, we can see the opposite behavior. In all test conditions, EER is lower when also the enrollment data are recorded in the room with the microphone array and processed by delay-and-sum. The last outcome of this experiment relates to known positions of the microphones and the source. It seems that in smaller rooms (in our case 4 × 4.5 × 3 m), this special information is not advantageous as the accuracy is about the same. However, it is more interesting when the recording is performed in spacious interiors (in our case 8 × 10 × 5 m). The accuracy is lower when no additional information about locations is available (*DS*) in comparison to *DS_known_pos*. After further analysis, we conclude that in large rooms, the reverberation is really strong and GCC-PHAT fails many times. It falsely considers one of the early reflections as directly propagated sound. In Figure 6.4, simulated room impulse responses for pairs of the reference microphone and the source located in "the room" and "the hall" are shown. For the sake of visual clarity, the sampling frequency was set to 32 kHz even though 8 kHz is considered in experiments. We can see that early reflections in "the hall" are stronger than in "the room".

We also experimented with MVDR beamformer, for which the noise covariance matrix needs to be estimated. A common way to obtain it is to use voice activity detector (VAD) for noisy parts of audio estimation. It leads to suppression of noise that comes from a particular direction (is correlated). In our simulated conditions, a noise is a part of the original recording, thus it comes from the same direction as a speech signal. This made impossible to obtain better results in comparison with delay-and-sum as an enhancement and attenuation were performed simultaneously. MVDR allows not only directional attenuation [11]. When assuming diffuse noise field, no VAD is needed [22]. We also attempted to perform MVDR beamforming with Multi-channel speech enhancement system tool[1] that

[1] https://github.com/DistantSpeechRecognition/mcse

can compute covariance matrix for diffuse noise fields. Since nor diffuseness is satisfied in our conditions, we did not achieve improvement. The results are summarized in appendix B.

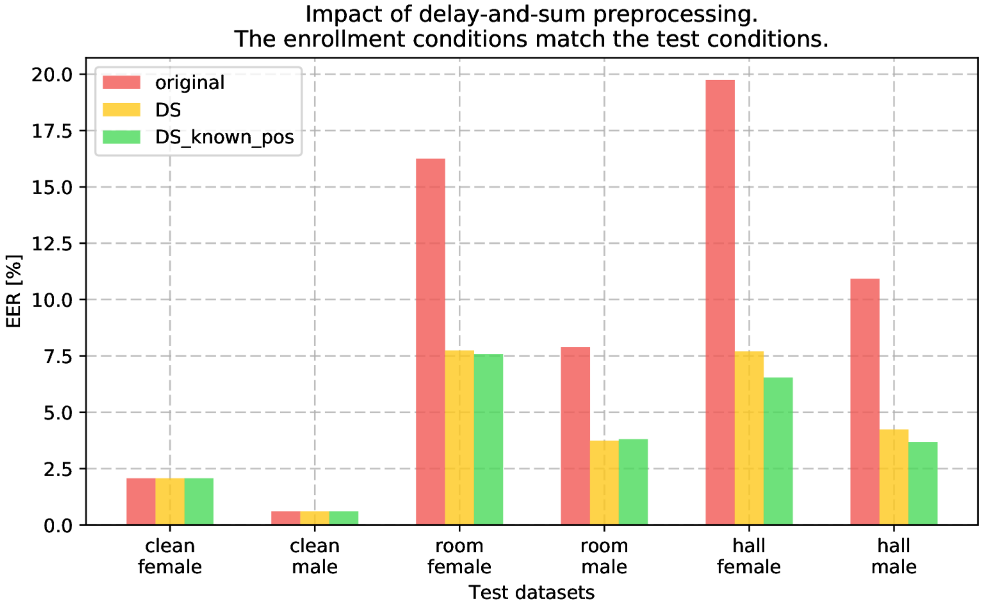


Figure 6.2: Impact of the delay-and-sum preprocessing on the recognition accuracy in terms of equal error rate (the lower the better). The enrollment and test conditions are matching.

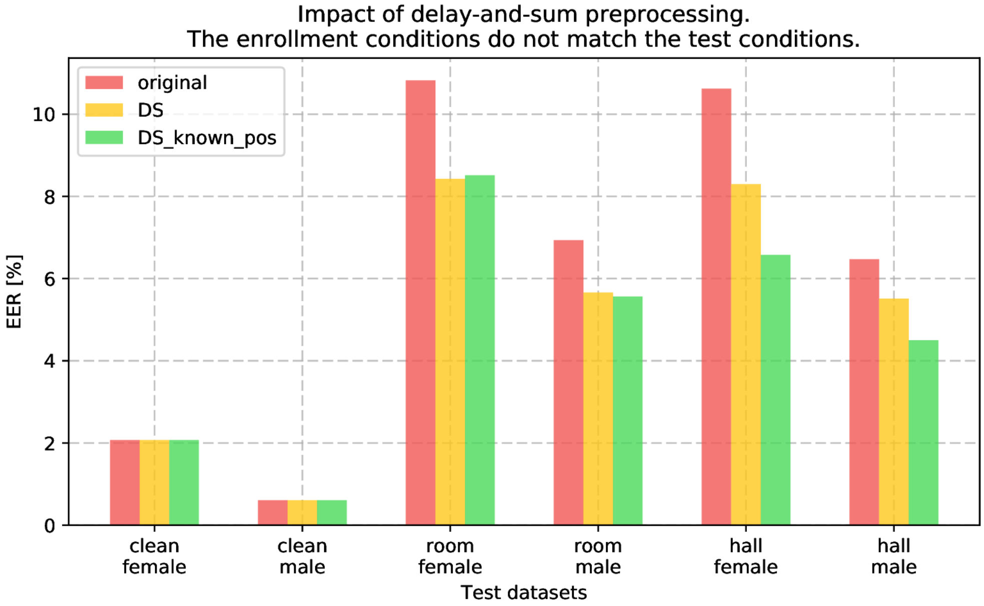


Figure 6.3: Impact of the delay-and-sum preprocessing on the recognition accuracy in terms of equal error rate (the lower the better). The enrollment and test conditions are non-matching.

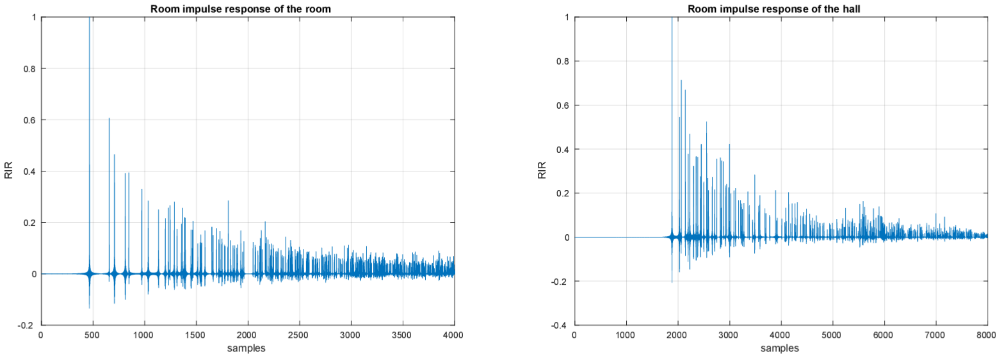


Figure 6.4: Room impulse responses for pairs of the reference microphone and the source in "the room" (left) and "the hall" (right). The sampling frequency is 32 kHz.

Difference between clean and noisy i-vectors

In our experiment, we have shown that delay-and-sum can improve the SRE system accuracy even if it remains unchanged. It means that PLDA classifies i-vectors in the same manner without any knowledge of the input data character change. We, therefore, expect that i-vectors extracted from beamformed multichannel data get close to the i-vectors of the original clean recordings at least in the speaker subspace. To explore whether a positive effect of delay-and-sum is obvious when the whole i-vectors are compared, we performed the following analysis: for female part of the test data, we computed the Euclidean distance between two i-vectors at a time. The first one was extracted from the original clean recording. The second was extracted from the appropriately simulated counterpart that either was or was not processed by delay-and-sum. We considered only "the room" (not "the hall"). The distributions of Euclidean distances are shown in Figure 6.5. Obviously, the histogram that corresponds to the beamformed data is shifted towards zero in comparison to the unprocessed far-field histogram. This behavior was expected, but the change of the distribution is not that significant when we consider the difference between the recognition accuracy when the beamforming is incorporated.

6.2 Model adaptation

In this section, we will focus on retraining of particular parts of the speaker recognition system. It means that we will modify certain elements of the baseline system – augmented datasets will be used for training. However, there are multiple options. Originally the training datasets contained a certain amount of recordings. Therefore, the first question is, how much data should be added to the original training datasets or should they be (at least partially) replaced? Another concern relates to the way the data should be augmented. This applies to the setting of parameters of the simulation, room sizes, and other audio processing. To answer all arising questions, many experiments would need to be performed. Since tackling all the possibilities is not tractable in a reasonable time period, we will cover just some of them.

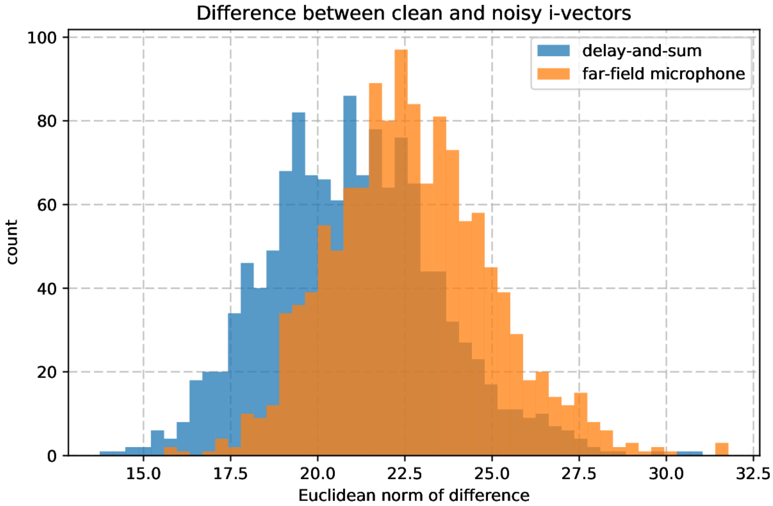


Figure 6.5: Distributions of Euclidean distances between clean and noisy i-vectors. The noisy i-vectors were extracted from the recordings that were simulated in “the room” (not “the hall”). They were further processed by delay-and-sum (blue histogram) or were let unprocessed (orange histogram).

PLDA retraining

Regarding the channel of additional training data, it would be proper to perform retraining incorporating either data disturbed by room acoustics or data additionally preprocessed by beamforming and finally both of them. We decided to experiment only with beamformed simulated recordings. At first, we simulated data recorded in the rooms depicted in Figures 5.1 and 5.2. It means that room dimensions and microphones’ arrangement equal testing conditions. As it is not possible to assume the exact shape of a room and microphone array beforehand, the second (more realistic) set of data was created by simulation of random rooms. Lower and upper bounds of wall dimensions are given in Table 6.1. The position of microphone array was also determined randomly. We aimed at exploring the impact of the amount of data as well. Overall, we prepared three training datasets. Two of them differ in the number of i-vectors, whereas the other pair is equally sized but represents different simulation conditions. We will refer to them as follows:

original contains original training data,

2_rooms_2 contains original training data + one modified copy (“the hall” Fig. 5.2 simulation) + one modified copy (“the room” Fig. 5.1 simulation),

2_rooms_1 contains original training data + a modified half of the original dataset (“the hall” simulation) + a modified half of the original dataset (“the room” simulation),

rand_rooms_1 contains original training data + one modified copy (random room simulation).

We can see that sets *2_rooms_1* and *rand_rooms_1* comprise the same amount of data and both of them are twice as large as the *original* set. In contrast, *2_rooms_2* contains three times as much data as the *original* set.

Table 6.1: Dimension limits for random room simulations.

	room dimension minimum	room dimension maximum	margin closer to the origin	margin further from the origin
x [m]	2.0	10.0	0.4	0.4
y [m]	4.0	12.0	0.4	0.4
z [m]	2.3	5.0	1.0	0.4

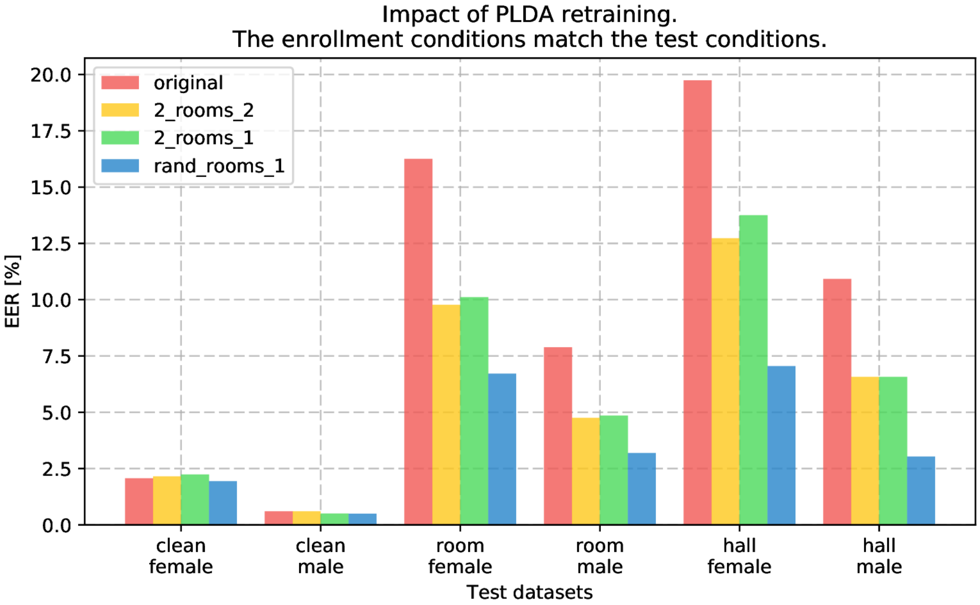


Figure 6.6: Impact of PLDA retraining on the recognition accuracy in terms of equal error rate (the lower the better). The enrollment and test conditions are matching.

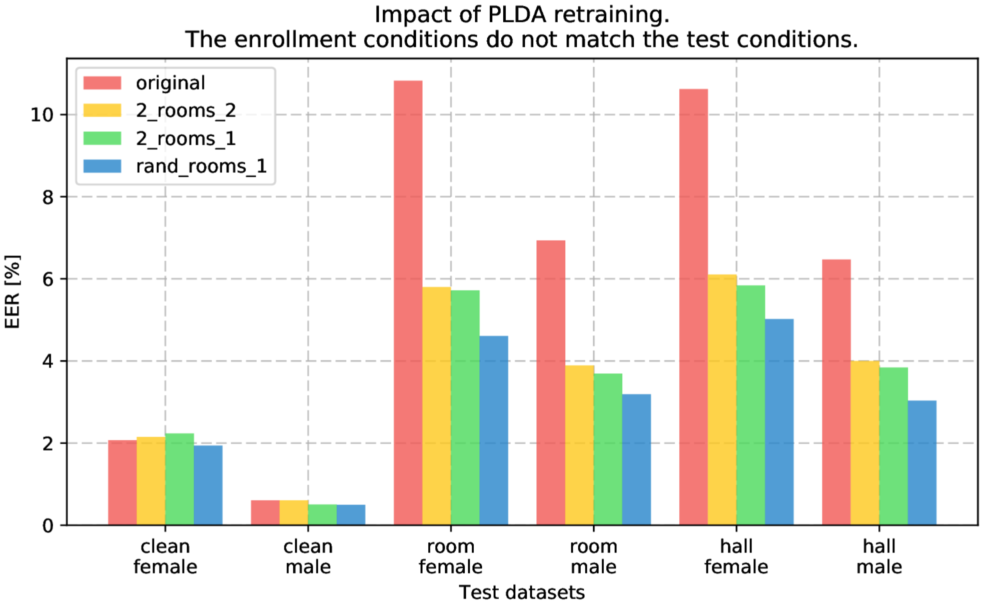


Figure 6.7: Impact of PLDA retraining on the recognition accuracy in terms of equal error rate (the lower the better). The enrollment and test conditions are non-matching.

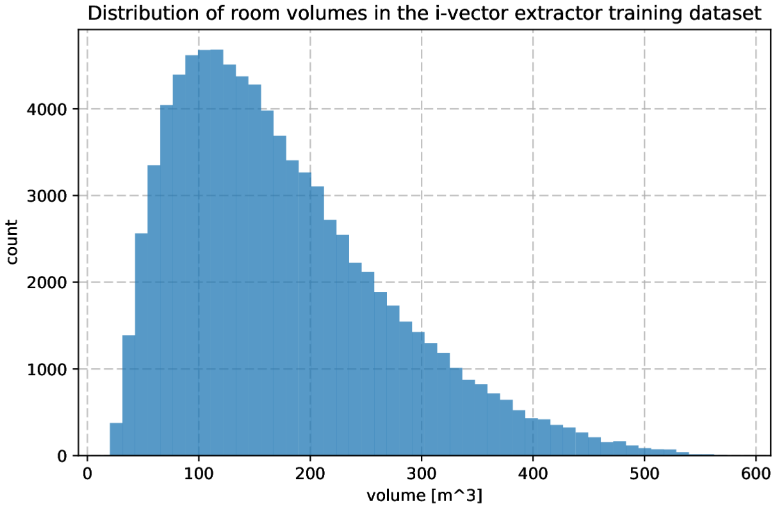


Figure 6.8: Volume distribution of rooms in which the i-vector extractor training data were simulated.

The results of the PLDA retraining are shown in Figures 6.6 and 6.7. We can see that all types of training data augmentation helped to improve the system accuracy for the test data that were obtained in simulated rooms. The training datasets that contain recordings from the same rooms in which test audio was recorded were expected to be more convenient due to the condition match. However, it has emerged that the training dataset (*rand_rooms_1*) containing more variable samples helped to improve the accuracy of the system even more. Moreover, it made the system more robust as EER decreased for all clean test datasets. On the other hand, PLDA retraining, during which *2_rooms_2* was employed, lead to slightly worse results for undistorted test audio. Next, we wanted to know, whether the amount of added training data will affect the accuracy significantly. From yellow and green bars in graphs, we can see that utilization of *2_rooms_1* set resulted in a bit worse accuracy when the enrollment and test conditions matched. The deterioration was not that substantial when we consider the fact that *2_rooms_1* contains one third less data than *2_rooms_2*, which leads to a reduction of time needed for data generation and training. When enrollment data were clean and *2_rooms_2* was used for the PLDA retraining, the accuracy even slightly increased for the room and hall test conditions. Another outcome of this experiment is the effect of the enrollment conditions. According to graphs, it seems to be inconvenient to record the enrollment audio by one far-field microphone even though it matches the test conditions. Recordings are then really unclear and the features extracted from them does not describe the speaker well.

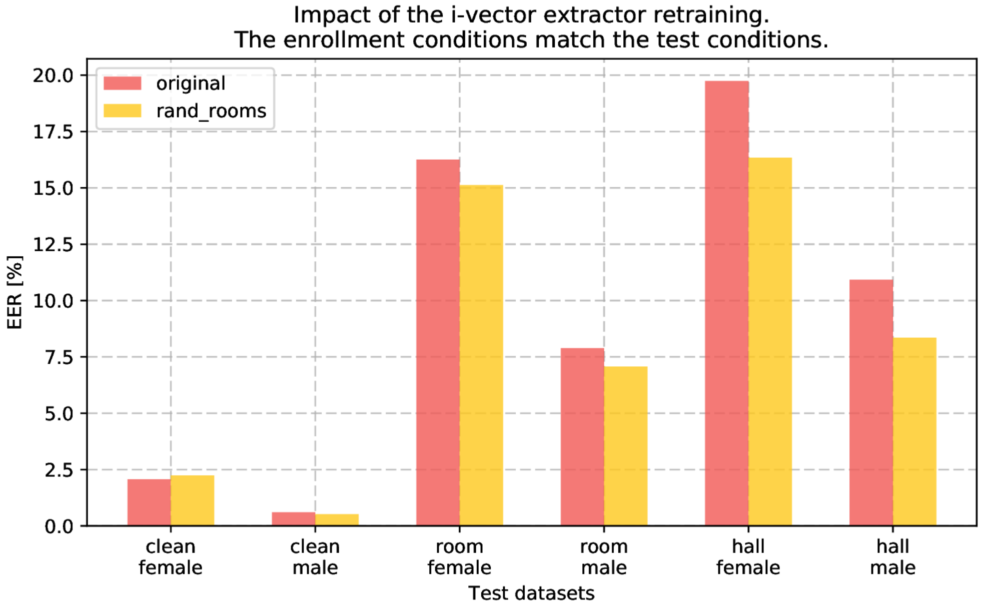


Figure 6.9: Impact of the i-vector extractor retraining on the recognition accuracy in terms of equal error rate (the lower the better). The enrollment and test conditions are matching.

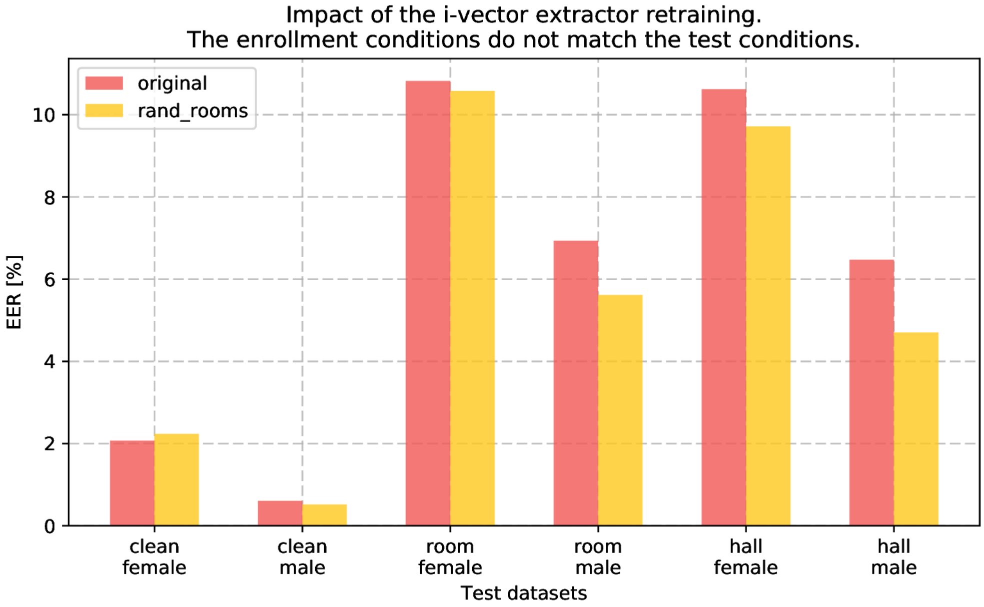


Figure 6.10: Impact of the i-vector extractor retraining on the recognition accuracy in terms of equal error rate (the lower the better). The enrollment and test conditions are non-matching.

I-vector extractor retraining

In the next experiment, we focused on the block preceding PLDA – i-vector extractor. We made use of findings of training datasets stated in the previous section. As the PLDA retraining incorporating a varied data led to the best results, we decided to prepare the training set the same way. It is required to retrain PLDA after modification of the i-vector extractor as well. When testing, a new i-vector extractor extracts feature vectors that are classified by PLDA, so it should know about the new total variability space. Because we have four training datasets for the PLDA training, every new i-vector extractor would lead to four experiments. We, therefore, prepared only one augmented set. In this section we wanted to focus only on the effect of i-vector extraction part of the pipeline, hence we used the original data for the PLDA learning. Bearing this in mind, we will show the accuracy changes with respect to two training datasets:

original i-vector extractor training dataset,

rand_rooms that contains the original training data + one modified copy (random room simulation).

As in the previous experiments, match and mismatch of the enrollment and test data was also explored. The obtained results are shown in Figures 6.9 and 6.10. We can see that, as in case of the PLDA retraining, the worse accuracy which is a result of matching test and enrollment conditions remains worse even after the i-vector extractor retraining. There are other two trends in this experiment to mention. First, improvement of the accuracy is more significant in spacious interiors ("the hall") regardless of the enrollment and test conditions. We expected rather different behaviors according to rooms' volume distribution in the training dataset (Figure 6.8). There are more samples that are closer to the volume of "the room" (54 m^3) than to the volume of "the hall" (400 m^3). The shape of the histogram is a result of multiplication of the dimensions that are individually drawn from uniform distributions. The second observable trend is that the accuracy improvement is greater for male test data. It holds in both types of rooms.

6.3 Combinations of beamforming and system adaptation

Until now, one single change at a time was considered – only beamforming or only the i-vector extractor retraining or only the PLDA retraining. Even though the application of the individual techniques led to improvements in the overall accuracy of the SRE system, there is still room for further improvements. We, therefore, decided to combine them with hope to achieve a greater amelioration of the results. More specifically, we examined retraining of the i-vector extractor and PLDA using all the combinations of the training datasets presented above. It is worth mentioning that when we obtained a new i-vector extractor and PLDA training followed, i-vectors needed in the process of the loading matrices estimation were extracted with that new extractor. It results in an effective extension of the classifier training data variability because this way we obtain different PLDAs for the same recordings in the dataset. In addition, we made use of microphone arrays and delay-and-sum beamforming. We will also preserve consistency with the previous experiments. Therefore, the results of two types of experiments will be shown. In the first type, the enrollment and test conditions match, whereas in the second they differ.

As we will present many outcomes at once, we introduce a color convention that we will use to enhance readability and convenience. One color (Figure 6.11) is assigned to each discussed technique – beamforming, i-vector extractor retraining, PLDA retraining. Thus, when we will refer to the type of beamforming or the training dataset, the color will clarify which part of the SRE system is considered at the time.

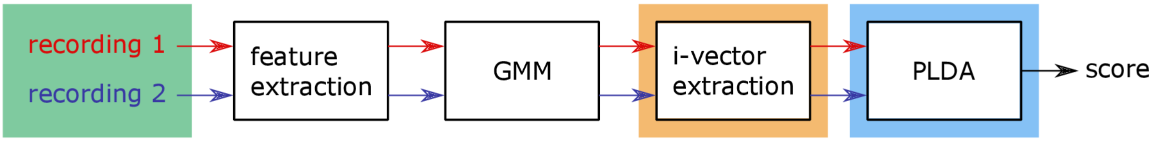


Figure 6.11: Color convention for the SRE system components. Green corresponds to audio preprocessing (beamforming), orange and blue correspond to i-vector extractor and PLDA training data, respectively.

In this section, results of the experiments will not be expressed in terms of EER. The accuracy of the original system, when tested on the clean data, will be considered as the reference – the best accuracy. When the same system is used to evaluate the data that were captured by one far-field microphone in the interior, we notice the deterioration of the accuracy (EER rises). These results are the worst. The techniques we have already discussed and their combinations should improve the recognition and in the best case, the original accuracy should be achieved. We, therefore, introduce a new measure that expresses relative improvement in percent – *recovery from error* (RFE). In compliance with the previous description, RFE is given by

$$RFE(x,g,r) = \left(\frac{a_{room}(g,r) - x}{a_{room}(g,r) - a_{clean}(g)} \right) \cdot 100, \tag{6.2}$$

where g and r specify test conditions – g is gender, r is a type of room. $a_{clean}(g)$ is the accuracy of the original system for the clean test data of gender g in terms of EER. $a_{room}(g,r)$ is analogous, but the test data from room r are considered (single far-field microphone). The symbol x refers to the new accuracy obtained by applying modifications of the system or preprocessing, whereas conditions match those specified by g and r. When RFE equals zero, it means that the new system does not help to recover from errors. On the other hand, 100 % is achieved in case that a particular technique helps to reach the original accuracy.

In Table 6.2, all the obtained results are summarized in terms of RFE. Both single-change experiments that were presented in more details before and combinations of multiple techniques are included. For every setup, two rows are displayed. In upper ones (with white background), matching enrollment and test conditions are considered. Gray rows show RFEs for the cases when the clean enrollment data were used. In almost every experiment, the gray number is lower than its white background counterpart. It is because the accuracy deterioration was more significant when the enrollment and test conditions matched while evaluating signals captured by one far-field microphone (worst case). Therefore, there was more room for improvements.

When a single technique is applied at a time, the PLDA retraining seems to yield the best results as it makes the final decision. Also, delay-and-sum can bring noticeable improvements. We can see that knowledge of the position may be favorable when performing beamforming. However, not in all cases it leads to better result in comparison with delay-and-sum that uses GCC-PHAT to estimate TDOAs. It is positive that more realistic scenario is comparable to the one that is artificial.Regarding single changes, the i-vector extractor adaptation introduced the least significant improvements and rather unstable behavior (2.8 % to 30.2 % RFE).

On the other hand, the combination of multiple techniques proved to be advantageous. By applying beamforming, i-vector extractor and PLDA adaptation, we achieved the best recovery.

Table 6.2: Recovery from errors (RFE) achieved when applying beamforming and the system adaptation. Both single modification and combinations of different techniques are included. Results in gray rows were obtained when the enrollment data were clean. Enrollment data match is considered in rows with a white background.

	4 × 4.5 × 3 m		**8 × 10 × 5 m**	
	females	**males**	**females**	**males**
original, **2_rooms_2**	45.7	43.1	39.7	42.2
	57.4	48.1	52.8	42.2
original, **2_rooms_1**	43.3	41.7	33.9	42.2
	58.3	51.3	55.9	44.8
original, **rand_rooms_1**	67.3	64.5	71.8	76.5
	71.0	59.1	65.5	58.6
DS, **original**, **original**	60.0	56.9	68.1	64.8
	27.4	20.1	27.2	16.4
DS, **original**, **2_rooms_2**	80.1	70.8	83.3	75.5
	74.5	62.4	73.4	52.6
DS, **original**, **2_rooms_1**	80.6	71.4	83.0	76.6
	75.0	64.8	73.9	52.6
DS, **original**, **rand_rooms_1**	77.4	73.6	83.0	79.4
	75.9	68.0	75.4	60.3
DS_known_pos, **original**, **original**	61.2	56.1	74.7	70.2
	26.4	21.7	47.3	33.6
DS_known_pos, **original**, **2_rooms_2**	80.6	70.8	82.5	78.4
	74.0	63.1	73.9	50.0
DS_known_pos, **original**, **2_rooms_1**	80.7	72.2	83.5	76.5
	75.1	64.8	75.4	53.4
DS_known_pos, **original**, **rand_rooms_1**	77.3	73.6	82.3	78.4
	76.5	**67.4**	79.5	62.1
rand_rooms, **original**	7.9	11.2	19.3	25.0
	2.8	20.9	10.6	30.2
rand_rooms, **2_rooms_2**	51.6	45.8	59.2	57.9
	47.0	24.1	48.3	27.6

rand_rooms, 2_rooms_1	49.7 48.5	45.3 28.1	56.7 50.3	57.5 30.1
rand_rooms, rand_rooms_1	68.5 71.1	62.5 54.5	75.0 70.0	76.5 61.2
DS, rand_rooms, original	63.4 27.4	56.9 34.4	68.9 33.2	72.1 32.8
DS, rand_rooms, 2_rooms_2	**83.4** 65.2	75.0 43.3	82.5 58.9	75.5 34.4
DS, rand_rooms, 2_rooms_1	82.5 65.7	76.3 44.9	82.8 62.9	78.4 37.9
DS, rand_rooms, rand_rooms_1	81.5 **77.9**	73.6 63.2	83.0 78.0	79.4 62.1
DS_known_pos, rand_rooms, original	63.4 27.8	57.0 34.5	75.2 52.3	72.6 42.2
DS_known_pos, rand_rooms, 2_rooms_2	83.0 65.2	75.0 45.7	82.0 54.8	73.7 31.0
DS_known_pos, rand_rooms, 2_rooms_1	82.5 66.1	**76.4** 45.7	83.3 56.3	74.5 33.6
DS_known_pos, rand_rooms, rand_rooms_1	81.3 78.5	75.0 64.0	**83.5** **80.0**	**80.4** **62.9**

All the results of our experiments were expressed terms of EER or RFE. These metrics, however, do not describe the behavior of the system completely. Therefore, we present a few DET curves in Figure 6.12 to show the effect of applied changes in more detail. The graph is meant to describe the original system, the worst case when the test data are captured by one microphone in an interior, and the system yielding the best-achieved results. Regarding the distorted test data, we used female recordings that were simulated in "the room". The blue curve correspond to the system which comprises retrained i-vector extractor (*rand_rooms* training dataset) and PLDA (*rand_rooms_1* training dataset). Also, delay-and-sum beamforming was applied to the multichannel test data. We can see that in terms of miss probability, the improvement is significant. In false alarm probability region, the best results we achieved are still far from those obtained with the original system but this region is usually not very relevant in practical scenarios.

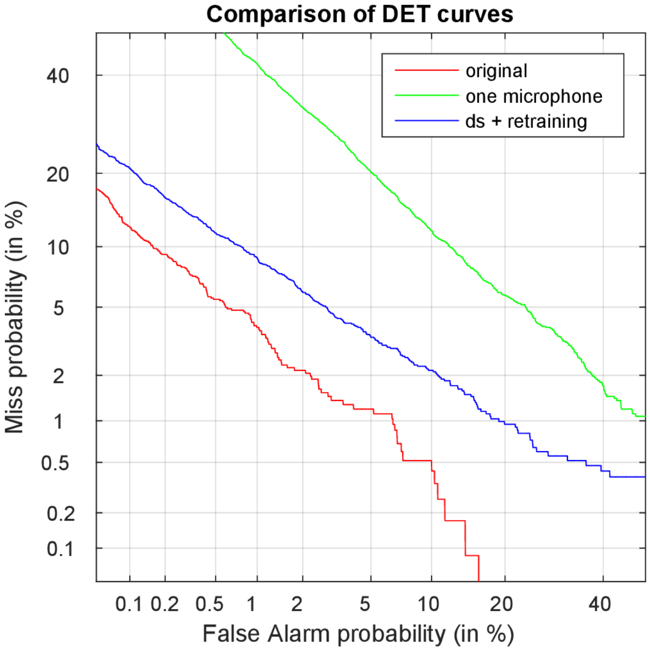


Figure 6.12: Comparison of DET curves. For the evaluation, female test data were used. Regarding simulated test data (green and blue curves), “the room” was employed.

Chapter 7

Estimation of speaker identity and position

When a speaker is allowed to speak in a room without further cooperation with the SRE system, not only his identity can be estimated – other valuable information is his position. In this chapter, we will consider two methodologies:

separate estimation In this approach, we will first estimate the speaker position and then perform beamforming taking this information into account. One could think that it is nothing more than just beamforming as was presented beforehand. However, earlier we assumed that GCC-PHAT is accurate enough to estimate the time difference of the sound wave arrival for pairs of microphones correctly. Hence, all the delays correspond to exactly one point in 3D space in a near-field scenario or two exactly defined angles (azimuth, elevation) in a far-field scenario. In fact, all the disturbing noises and reverberation that are present in interiors disallow to estimate TDOA correctly. Then, the delays may not be correlated. We, therefore, want to gain the approximate position utilizing noisy TDOAs. When the position is known, we can re-estimate TDOAs and use them while performing beamforming. At this stage, we know the approximate position and have the signal in which speaker voice should be enhanced. Then the recording can be passed to the SRE system to obtain the identity.

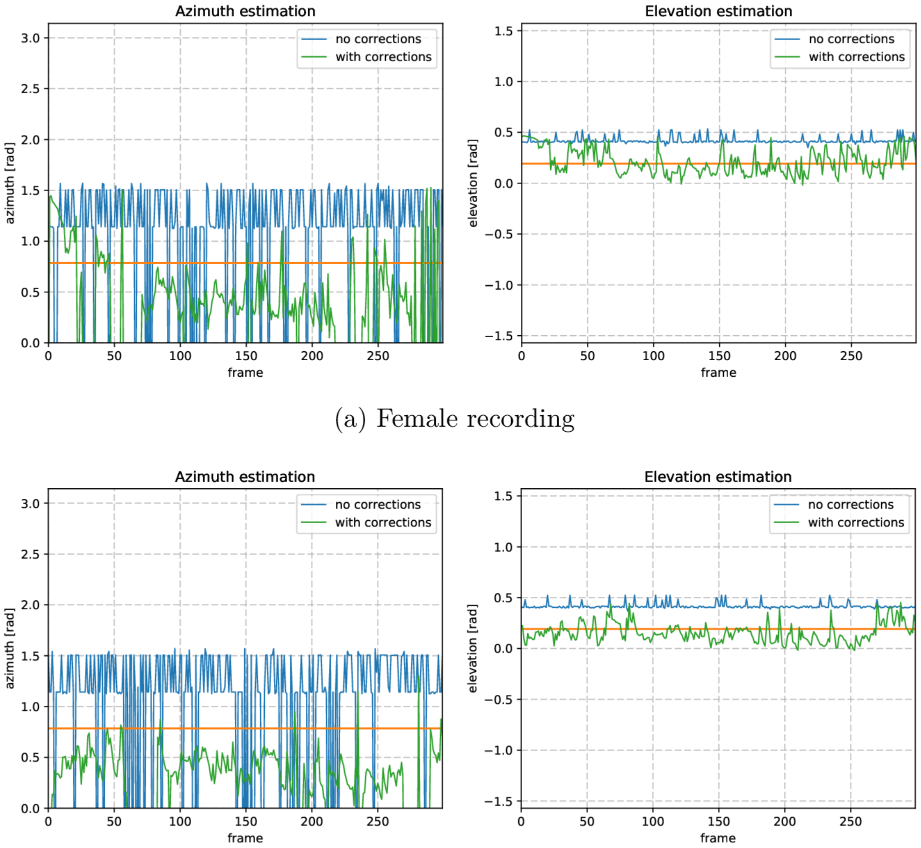
joint estimation In the second approach, the position and identity estimation should be performed jointly. It is inspired by position estimation algorithms that find the maximum of steered response power function. As it is meant to also perform speaker recognition, no beamformer output power will be computed in our case. Instead, the function whose maximum will be sought is the SRE score.

In the subsequent sections, a description of experiments that employ both presented methods will be given.

7.1 Separate estimation

In order to obtain the source position using the GCC-PHAT estimates of TDOAs, we wrote a MATLAB script, that implements Brandstein's linear intersection algorithm [3]. In the location estimation experiments, we will focus only on cases in which microphones are placed in far field, hence we will consider only "the hall". We have a planar microphone array consisting of two ULAs at our disposal, but the algorithm expects quartets of microphones to form so-called bearing lines. Therefore the array will be divided into three quartets that share microphones.

In the first experiment, we used the linear intersection algorithm as is, taking the best GCC-PHAT estimation into account. However, reverberation proved to cause serious problems when estimating TDOAs and subsequently the position (which we use to compute the direction of a sound wave arrival). To describe inaccuracy, we show the estimation of the elevation and azimuth in Figure 7.1 (blue curves). Since the algorithm works on a frame-by-frame basis (being a part of beamforming), the values in the graphs correspond to frames of one randomly chosen female and one randomly chosen male recordings. The values seem to oscillate between few values. It may be a result of wrong correlation function peak that was produced when strong reflected signals got aligned.



(a) Female recording

(b) Male recording

Figure 7.1: Speaker position azimuth and elevation estimated by linear intersection algorithm. Orange lines denote correct angles.

Subsequently, we decided not to consider only the delays between microphone pairs that correspond to the maximum of the cross-correlation function but also next three delays. We then came up with an experimental objective function $L(d_1, d_2)$ for pairs of delays d_1, d_2. Those two delays are meant to be between two pairs of microphones whose connecting lines are parallel. We set two assumptions: First, in a far field, a wavefront can be approximated by plane, thus d_1 and d_2 should be close to each other. Second, the direct sound arrives first at the microphone (before early reflections), so the absolute value of the delay should be rather small. Bearing this in mind, the empirical objective function is given by

$$L(d_1, d_2) = 0.5|d_1 - d_2| + 0.2\frac{|d_1| + |d_2|}{2} + 0.15|d_1 - d_1^{old}| + 0.15|d_2 - d_2^{old}|, \tag{7.1}$$

where the first and the second terms express the first and the second assumption, respectively. We also take the previous values d_1^{old}, d_2^{old} into account. For every pair of delays we find $\hat{d}_1$ and $\hat{d}_2$ that minimize (7.1) given delay alternatives (multiple peaks of a correlation function). After we incorporated aforementioned corrections, we obtained new estimations that are shown in Figure 7.1 (green curves). There is a significant improvement of the elevation estimation, but the azimuth we computed does not seem to be satisfactory (at least with selected recordings).

In Table 7.1, we present how direction estimation affects the subsequent speaker recognition when the microphone array outputs are used as the input to the SRE system. We used the original system without any modifications. For the sake of comparison, EERs for the far-field single-microphone recognition are displayed. Results obtained when delay-and-sum that relies entirely on GCC-PHAT TDOA estimation (DS) is incorporated are shown as well. The accuracy of direction estimation is crucial because correct estimation can lead to recognition improvements. Naturally, the overall accuracy may decrease as a result of wrong expected position.

Table 7.1: Comparison of the SRE system accuracy in terms of EER when delay-and-sum relies on TDOA estimation or the position estimation.

	one microphone	**DS**	**no corrections**	**with corrections**
females	19.741	7.703	9.550	9.114
males	10.925	4.237	4.604	3.943

A position estimation using linear intersection seems to be a difficult problem. However, there are possibilities that may lead to improvements of the estimation accuracy. We used the microphone array whose elements lay on the same plane and are relatively close to each other in comparison to the dimensions of “the hall”. Then TDOA estimation errors in terms of samples lead to a significant variation of bearing lines whose intersections can change a lot. In the original paper [3], Brandstein used perpendicular microphone array that is more suitable. Placing quartets of microphones around the whole interior would be even better. Certainly, more sophisticated methods could be employed – linear intersection is rather simple and is dated to 1997. We expect that for instance SRP-PHAT [8] would work better, but then beamforming would have to be performed twice – to locate the speaker and to obtain the output signal.

7.2 Joint estimation

Steering the beam over a room similar with steered response power (SRP) algorithms. They make no assumptions about speaker positions. Scanning (or steering) is performed by adjusting delays associated with elements of the microphone array. It means that the microphone array successively focuses on all the points in 3D space from a predefined discrete grid. When the microphones are in a far field region, it is advantageous to change azimuth and elevation to steer the beam because sound waves are well approximated by planes. Discrete values of angles are considered in this case as well because continuous sweeping is not tractable. SRP algorithms evaluate power of the beamformer output for each point (or direction) as follows [8]

$$P(\Delta_1, \ldots, \Delta_M) = \int_{-\infty}^{\infty} Y(f, \Delta_1, \ldots, \Delta_M) Y^*(f, \Delta_1, \ldots, \Delta_M) df, \tag{7.2}$$

where $\Delta_1, \ldots, \Delta_M$ are delays that correspond to actual look direction (M microphones in array is considered), $Y(f, \Delta_1, \ldots, \Delta_M)$ is the Fourier transform of the microphone array output. Then the speaker position is computed from $\hat{\Delta}_1, \ldots, \hat{\Delta}_M$ that maximize (7.2).

In our approach, we extract an i-vector from the beamformer output for every predefined direction. Given the enrollment i-vector, a score of similarity based on PLDA is expressed. We assume that when the speaker of interest is present in the room, the beamformer output should contain enhanced speaker voice when the microphone array is steered correctly. Then the extracted i-vector should correspond to the enrollment one more than those that were obtained when the beam was steered to different directions. When the speaker is not in the room, scores should be ideally lower than zero for all parts of the 3D space .

Because we aimed at the case in which the microphone array is located in a far field, we steered the beam over the room with respect to two angles – azimuth, elevation. The origin of the spherical coordinate system is located in the middle of the microphone array. We consider the following interpretation of the angles: when the array beam is steered to the right, azimuth is 0° and elevation is 0°; when it is steered to the left, azimuth is 180° and elevation remains 0°; when the array looks up, azimuth is set to 90° and elevation to 90° as well; by fixing azimuth and setting elevation to $-90°$ we make the microphones look down.

When performing joint estimation, it is crucial to explore whether the function that assigns each direction a score value reaches its peak for the pair of angles that correspond to the source position. Thus, for convenience, we will assume that there is only one person in the room and we have only one enrollment i-vector. The enrolment i-vector was extracted from the same utterance of the same person that is uttered in the room. All subsequent actions just extend this concept. It means that when we perform identification, we have more i-vectors to compare for every azimuth and elevation and the maximum is found over multiple responses.

The first experiment was aimed at the question about the peak of the function. The source and microphones' positions correspond to the layout of "the hall". However, we omitted walls and their effect. It resulted in an open space free of reverberation. The angles used for steering linearly sampled the aforementioned intervals – 7 values for the azimuth and 7 values for the elevation in this case. After sweeping, we interpolated the response to artificially increase resolution and found the maximum. One of the obtained responses is shown in Figure 7.2. A real position of the source in terms of azimuth and elevation is displayed as the rectangle in the plot. The circle represents estimate based on the maximum score. We can see that they do not have the same coordinates, but are very close to each other. It is rather result of the sampling frequency. The recordings we work with are sampled at 8 kHz. Therefore, shifts of signals that corresponds to sound propagation delays was performed in 0.125 ms steps. It naturally introduces errors. The more important finding is that score tends to rise when the microphone array is focused on the person we have an enrollment i-vector from. The peak in Figure 7.2 is obvious and its coordinates nearly match real azimuth and elevation.

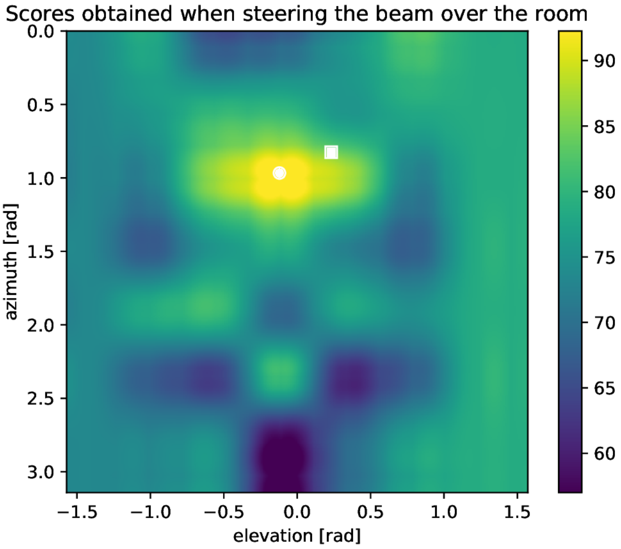
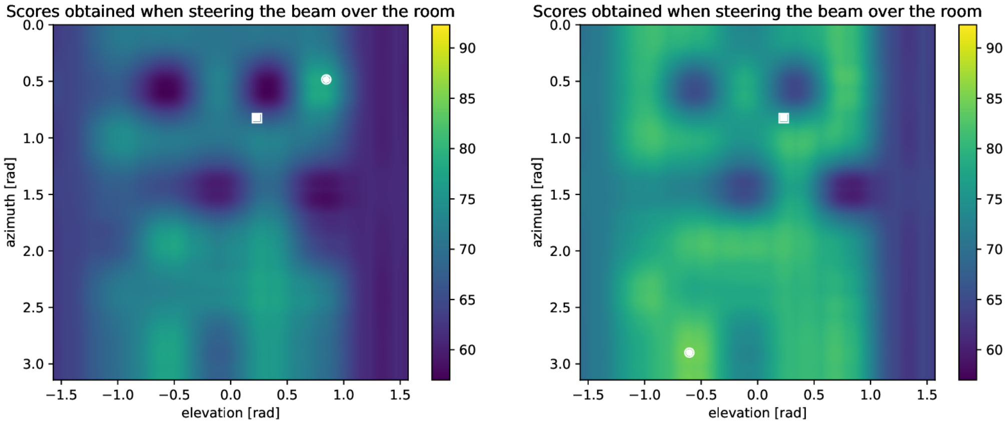


Figure 7.2: Response of the beamformer in terms of scores for the speaker recognition while steering over certain azimuth and elevation angles in a free space. The circle represents the maximum. Square denotes the ground truth.

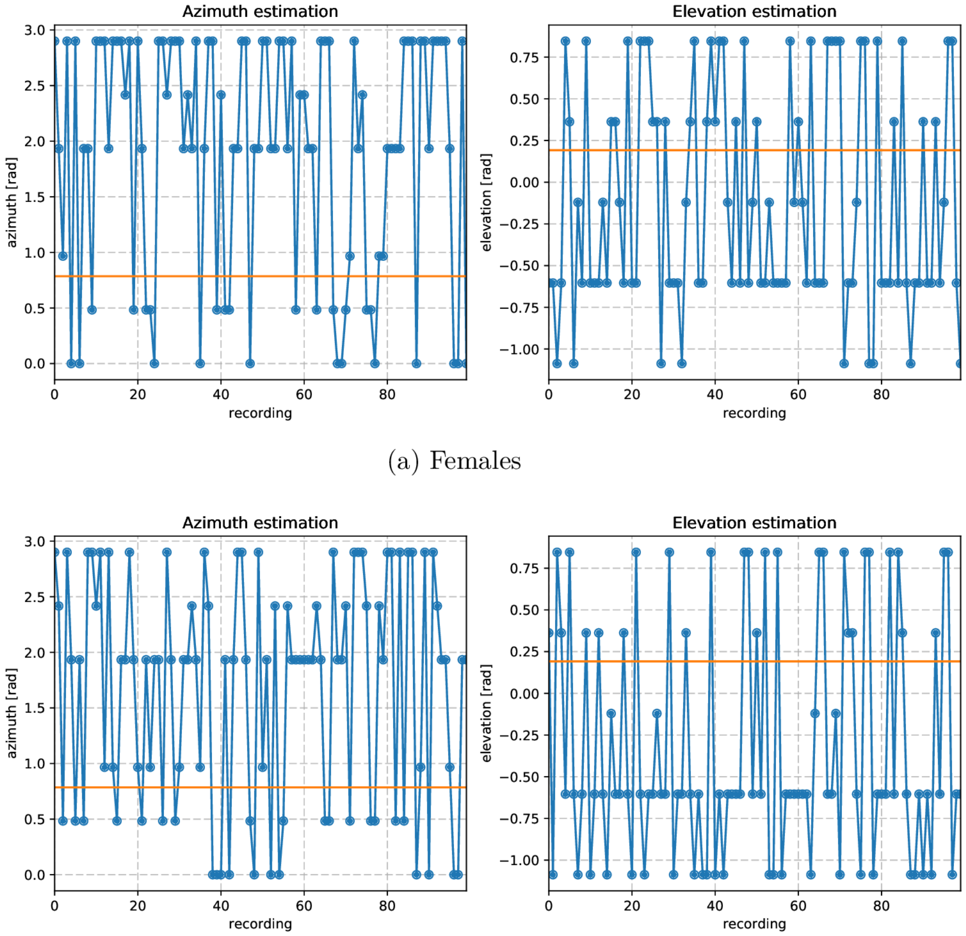


(a) Female

(b) Male

Figure 7.3: Response of the beamformer in terms of scores for the speaker recognition while steering over certain azimuth and elevation angles in the hall. The circle represents the maximum. Square denotes the ground truth.

Next, we wanted to move closer to a real scenario, hence we took walls that cause reverberation into account again. We used the male and female data simulated in “the hall”. All other settings remained the same as in the previous experiment. In Figure 7.3, examples of obtained responses are shown. The female recording that was used equaled the one from the previous experiment. We can see that scores do not reach as high values as in the case without reverberation. Moreover, room acoustics caused such problems that the responses seem not to be usable for correct direction estimation anymore. Unfortunately, the drawback represented by the figure is noticeable in every test we attempted. In Figure 7.4, estimations of azimuth and elevation based on responses are shown for 100 test recordings. The values vary a lot, but there are certain angles for which the response reaches its maximum repeatedly.



(a) Females

(b) Males

Figure 7.4: Estimation of the speaker position azimuth and elevation that was obtained by steering the beam over a closed interior (blue circles). Orange lines denote correct angles.

Obviously, reflections of the sound on walls that were not present in a free space caused significant deterioration. Probably when the array steers its beam to a direction, in which no sound source is placed, it can still capture a strong reflection resulting in a high score. Another aspect that relates to reflections is a beam pattern. As it was presented in chapter 3, directivity pattern comprises the main lobe, but also side lobes. Therefore, even when the microphone array focuses on the source, it does not attenuate sounds from other directions completely. Then the reflected signal is combined with the direct one even though they are not aligned. This summation can disturb the correct sound which in turn results in deterioration of the score. It seems that the beam pattern is of importance. In order to obtain an idea about the microphone array we used, Figure 7.5 shows the directivity

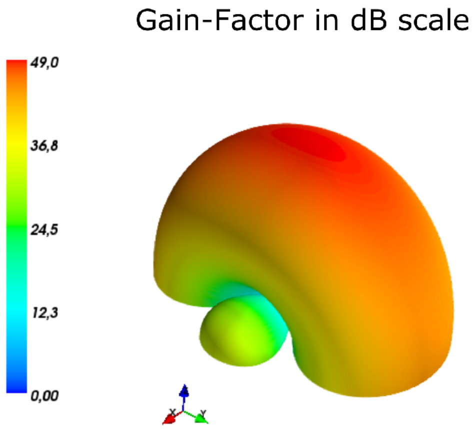


Figure 7.5: Beam pattern of the used microphone array when steered ahead for frequency 150 Hz.

pattern[1] when the array “looks ahead”. In the axis (Y in the figure) with only two microphones, the selectivity is bad and can also worsen the results. Also, placing multiple sound sources into the room would be worth trying. We expect that stronger direct sounds could mask reflections that make score rise when the beam is not steered to the source of interest. In this experiment, we used the simulated data again. The question is, whether the same trend as was mentioned would be present in a real-world scenario.

[1] The beam pattern was created by Arraytool: https://zinka.wordpress.com/arraytool/.

Chapter 8

Comparison of the simulated data with the real data

In all the presented experiments, the simulated test data were used. McCowan warns [19] that more realistic methods generating simulated data should be used with far-field speaker recognition. This caution motivated our next work. We, therefore, focused on two questions. Is the image method simulation that insufficient and how can we describe inappropriateness? Can we expect the same or similar behavior we observed in the experiments also in a real world? In the next two sections, we will cope with those issues in more details.

8.1 Comparison of the RIR Generator and the real room impulse responses

In order to quantitatively express the correspondence of the impulse responses that were recorded in a real room with those that RIR Generator outputs, we will use standard acoustic parameters [15, 24]. Two out of three metrics, which will be used, are associated with the characteristics of perceived speech. A very brief description of them will follow.

Clarity, C$_{80}$

Criterium C$_{80}$ assesses the level of audible details. The definition takes a human hearing into account: in case that reflected copies of the original signal arrive within 80 ms, they are perceived as the amplification of the original sound. The following equation reflects this feature:

$$C_{80} = 10 \log_{10} \frac{\int_0^{80ms} h^2(t)dt}{\int_{80ms}^{\infty} h^2(t)dt}, \tag{8.1}$$

where $h(t)$ is the room impulse response.

Definition, D$_{50}$

The definition was the first objective parameter to describe the speech intelligibility [15]. Using the room impulse response $h(t)$, it is given as

$$D_{50} = \frac{\int_0^{50ms} h^2(t)dt}{\int_0^{\infty} h^2(t)dt}. \tag{8.2}$$

Rapid Speech Transmission Index, RASTI

Speech Transmission Index (STI) is a common objective parameter to assess the speech intelligibility computed more sophistically than D_{50} [24]. RASTI is a fast method for measurement of STI. It ranges from 0 to 1, while 1 denotes excellent intelligibility. The interval is divided into labeled subintervals, which are shown in Table 8.1.

Table 8.1: Verbal description of RASTI subintervals.

RASTI value	< 0.30	0.30–0.45	0.45–0.60	0.60–0.75	> 0.75
Intelligibility	bad	poor	fair	good	excellent

To compare real and simulated room impulse responses in terms of acoustics parameters defined above, real ones must have been acquired. In this thesis, we used the database of room impulse responses recorded with omnidirectional microphones at Queen Mary, University of London [28]. They were captured at 130 different places in the room of the approximate size 7.5 × 9 × 3.5 m as it is show in Figure 8.1. Using RIR Generator, the same room has been modeled[1] and the impulse responses were acquired by placing virtual microphones to known positions. For the computation of acoustic room parameters, Dirac[2] tool was used.

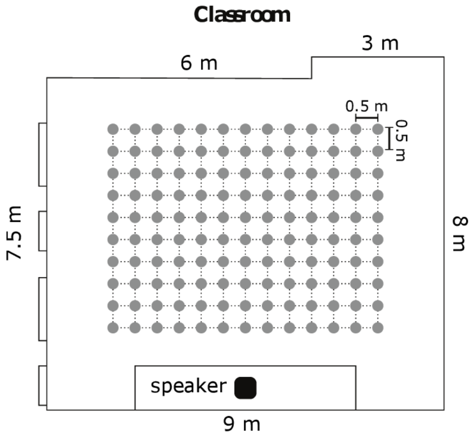


Figure 8.1: Scheme of the room, where real impulse responses were captured. Gray circles represent positions of microphones.

The result assessing C_{80} criterion is displayed in Figure 8.2. Regarding the real room, the clarity reaches its maximum around the sound source. With increasing distance, the clarity decreases. In the case of simulated room, there is no single significant peak, but rather local maxima at different positions. In the case of definition parameter, the situation remains approximately the same (see Figure 8.3). As far as RASTI index is considered, the values for the simulated room fall within the intervals corresponding to verbal descriptions “poor” and “fair”, whereas in the real room, values around the sound source approach “good” category as seen in Figure 8.4.

[1] Wall reflection – the parameter for RIR Generator – was derived from RT_{60} [10] parameter of the real room at 250 Hz.

[2] http://www.acoustics-engineering.com/html/dirac.html

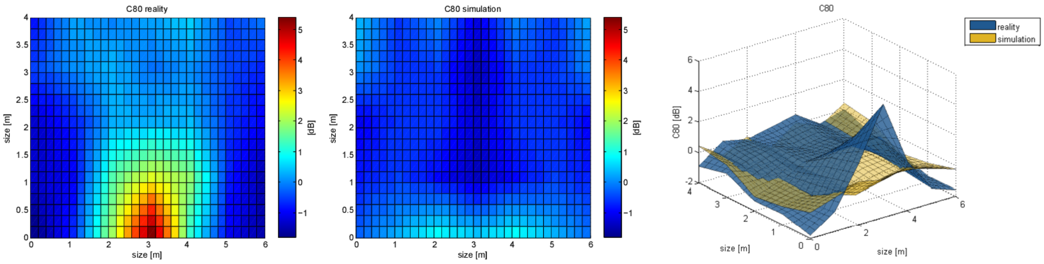


Figure 8.2: Comparison of impulse responses in terms of clarity.

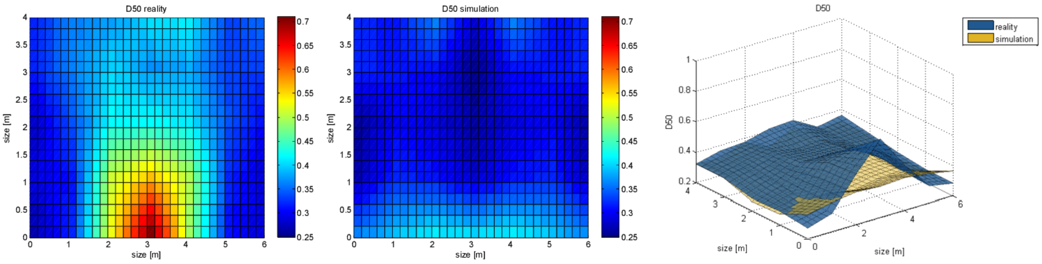


Figure 8.3: Comparison of impulse responses in terms of definition.

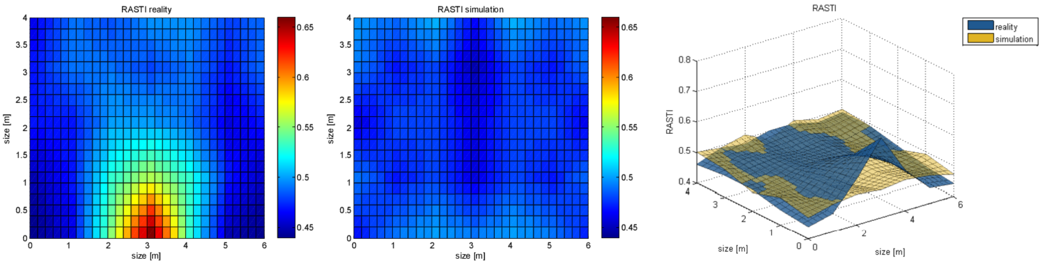


Figure 8.4: Comparison of impulse responses in terms of Rapid Speech Transmission Index.

8.2 Retransmitted data versus simulated data

Experiments performed with RIR Generator have shown that in terms of parameters assessing room acoustic properties, simulated impulse responses barely correspond to reality. The question is how this fact affects the performance of speaker recognition.

BUT Speech@FIT equipment allows to perform retransmission of recordings. It means that audio signals that we possess are played aloud and captured again by a microphone or multiple microphones. Due to replaying in a room, the output signals are affected by reverberation and noise. Such signals are close to recordings of real voices in real conditions. The only simplification is that there is a loudspeaker as a source instead of a person. On the other hand, it makes the process of recording more convenient. We will take the opportunity and make use of such audio.

To shorten the time needed for retransmission, we created a new test dataset. It is a subset of the previously used one (NIST Year 2010 Speaker Recognition, condition 1). To enlarge diversity, only short recordings (3 min) were included. Overall, it comprises approximately 5 hours of male utterances and 5 hours of female utterances.

We could perform all the types of aforementioned experiments. However, as it is time demanding, we will restrict them to basic ones. The original SRE system without any changes (PLDA, i-vector extractor adaptation) will be considered. The test datasets will be as follows:

clean the subset of the NIST 2010 condition 1 recordings (10 hours),

1mic_simu recordings from *clean* dataset captured in the simulated room by one microphone,

ds_simu multichannel simulated recordings processed by delay-and-sum,

1mic_real the same as *1mic_simu* but recorded in the real room,

ds_real the same as *ds_simu* but recorded in the real room.

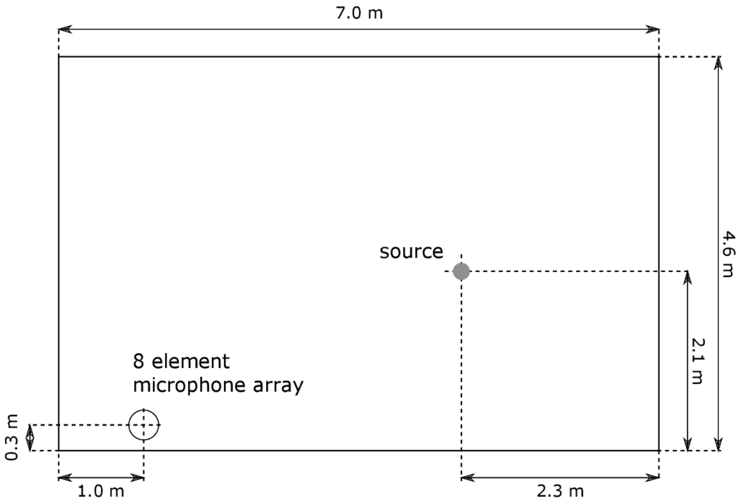


Figure 8.5: Scheme of the room, in which retransmission was performed.

The real room, in which the retransmission was performed (Figure 8.5), differs from "the room" and "the hall" that appeared in the previous experiments. Nor the microphone array geometry is the same. It implies a need for a new simulation as we wanted the real and artificial rooms to be rather similar. Thus, we measured the dimensions of the genuine interior along with positions of the microphone array and the loudspeaker. The obtained sizes were used for specification of the simulation settings[3]. As BUT Speech@FIT possesses a circular microphone array, we also changed the one we used until now to match the real one.

The results we obtained in the experiment are shown in Table 8.2. It can be seen that recording with only one microphone is insufficient in real conditions as well. EER is even higher for both the male and female test recordings. In genuine room conditions, a noise is an important factor. In experiments with simulated data, it was not considered as we focused more on the reverberation. A greater deterioration in comparison with simulated data is observable when the female data are evaluated. Delay-and-sum beamforming helps

[3] Even though the simulated room resembles the real one, there are still inaccuracies. The furniture that affects the acoustic conditions is omitted in the simulation. Also, effect of the walls will differ as we do not know their absorption.

Table 8.2: Comparison of the SRE system accuracy in terms of EER for simulated and retransmitted data.

	clean	**1mic_simu**	**ds_simu**	**1mic_real**	**ds_real**
females	1.798	5.846	4.572	24.996	20.336
males	0.032	6.498	1.246	9.557	1.9424

to improve results in both conditions. Regarding the male test data, the accuracy obtained for a real room data approached the simulation. Overall, when the male test data were employed, we obtained comparable results. It does not hold for the female data. It seems that physically correct environment is even more harmful to them. Therefore, a more effort should be put into the investigation of this behavior in order to prove trends that were shown in this experiment.

Chapter 9

Conclusion

In this thesis, we have dealt with the topic of far-field speaker recognition. We emphasized that it is insufficient to use only one microphone that records in an interior. Instead, microphone array seems to be a good choice. It allows performing beamforming that combines signals from the sensor into an output in a way, that it enhances sounds that come from a particular direction. Incorporation of delay-and-sum beamforming led to the improvement of the accuracy. The following work aimed at system adaptation. We, therefore, augmented training dataset for different parts of the SRE system. We conclude that the more variable data we prepared the better results we obtained. Combination of multiple techniques – beamforming, i-vector extractor and PLDA retraining – proved to be advantageous. In terms of recovery from error (RFE), we achieved up to 75 % recovery of the performance gap between close-talk microphone data and single far-field microphone data.

Next, we attempted to simultaneously estimate speaker location and identity. We introduced two methods. The first of them initially estimated a position. Knowledge of approximate location was then used during beamforming. We discovered that correct position estimation is crucial for subsequent recognition. Linear intersection method proved to be prone to erroneous GCC-PHAT estimates of TDOAs. The second approach combined localization and recognition. It was based on steering a beam over a room and for every pair of azimuth and elevation, it computed the score that is an output of PLDA. This algorithm was inspired by SRP method. In a free space, we achieved satisfactory behavior. However, reverberation caused serious problems and our method failed in an interior.

In our experiments, we used simulated data due to a lack of real recordings. Therefore, we tried to quantitatively express the difference between both types of audio. Regarding acoustic parameters, we observed quite different behavior in real and simulated rooms. Our next aim was to compare the SRE accuracy when the same test data are both artificially distorted in a simulated room and retransmitted in real conditions. We obtained comparable results for the male test recordings. However, the evaluation of the female data led to significantly different results.

We conclude that reverberation is a great problem that worsens the accuracy of multiple methods. In our experiments, we had problems to estimate the position of the speaker when reflections occurred in the interior. On the other hand, we were able to improve the recognition accuracy quite significantly. In the future, we would like to focus more on elaborate beamforming methods such as minimum variance distortionless response (MVDR) or generalized sidelobe canceller (GSC). To employ them, the character of experiments must be changed. So far, we considered one source in a room. We would like to place a directional source of noise, ambient noise, and/or more speakers to the room. Also, more attention should be paid to a location estimation inaccuracies. After reaching satisfactory improvements, next step would be speaker tracking.

Bibliography

[1] The NIST Year 2010 Speaker Recognition Evaluation Plan.
Retrieved from: https://www.nist.gov/sites/default/files/documents/itl/iad/mig/NIST_SRE10_evalplan-r6.pdf

[2] Allen, J.; Berkley, D.: Image method for efficiently simulating small-room acoustics. *Journal of the Acoustical Society of America*. vol. 65, no. 4. 1979: pp. 943–950. ISSN 0001-4966. doi:10.1121/1.382599.

[3] Brandstein, M.; Adcock, J.; Silverman, H.: A closed-form location estimator for use with room environment microphone arrays. *IEEE Transactions on Speech and Audio Processing*. vol. vol. 5, no. issue 1: pp. 45–50. ISSN 10636676. doi:10.1109/89.554268.
Retrieved from: http://ieeexplore.ieee.org/document/554268/

[4] Brandstein, M. S.: *A Framework for Speech Source Localization Using Sensor Arrays*. PhD. Thesis. Providence, RI, USA. 1995. aAI9540732.

[5] Brümmer, N.: EM for Probabilistic LDA. 2010.

[6] Chen, J.; Benesty, J.; Huang, Y. A.: Time Delay Estimation in Room Acoustic Environments. *EURASIP Journal on Advances in Signal Processing*. vol. vol. 2006. 2006: pp. 1–20. ISSN 16876172. doi:10.1155/ASP/2006/26503.
Retrieved from: http://asp.eurasipjournals.com/content/2006/1/026503

[7] Dehak, N.; Kenny, P. J.; Dehak, R.; et al.: Front-End Factor Analysis for Speaker Verification. *IEEE Transactions on Audio, Speech, and Language Processing*. vol. vol. 19, no. issue 4. 2011: pp. 788–798. ISSN 15587916. doi:10.1109/TASL.2010.2064307.
Retrieved from: http://ieeexplore.ieee.org/document/5545402/

[8] DiBiase, J. H.: *A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays*. Ann Arbor, Mich.: Proquest Information and Learning. 2000. ISBN 0599939079.

[9] Glembek, O.: *Optimization of Gaussian Mixture Subspace Models and Related Scoring Algorithms in Speaker Verification*. PhD. Thesis. Brno University of Technology, Faculty of Information Technology. 2012.
Retrieved from: http://www.fit.vutbr.cz/study/DP/PD.php?id=209

[10] Habets, E. A.: Room Impulse Response Generator. September 2010.
https://github.com/ehabets/RIR-Generator/blob/master/rir_generator.pdf.

[11] Habets, E. A. P.; Benesty, J.; Cohen, I.; et al.: New Insights Into the MVDR Beamformer in Room Acoustics. *IEEE Transactions on Audio, Speech, and Language*

Processing. vol. 18, no. 1. Jan 2010: pp. 158–170. ISSN 1558-7916. doi:10.1109/TASL.2009.2024731.

[12] Kenny, P.: Joint factor analysis of speaker and session variability: Theory and algorithms. Technical report. 2005.

[13] Kenny, P.: Bayesian Speaker Verification with Heavy-Tailed Priors. In *Odyssey 2010: The Speaker and Language Recognition Workshop*. 2010.

[14] Kumatani, K.; McDonough, J.; Raj, B.: Microphone Array Processing for Distant Speech Recognition. *IEEE Signal Processing Magazine*. vol. vol. 29, no. issue 6. 2012: pp. 127–140. ISSN 10535888. doi:10.1109/MSP.2012.2205285.
Retrieved from: http://ieeexplore.ieee.org/document/6296525/

[15] Kuttruff, H.: *Room acoustics*. London & New York: Spon Press/Taylor & Francis. fifth edition. 2009. ISBN 9780203876374.

[16] Lathoud, G.; McCowan, I. A.: A Sector-Based Approach for Localization of Multiple Speakers with Microphone Arrays. Technical report. Martigny, Switzerland. 2004. published in “Proceedings of the 2004 SAPA Workshop”.
Retrieved from:
http://publications.idiap.ch/downloads/reports/2004/rr-04-15.pdf

[17] Lutter, M.: Feature Extraction. January 2015. [Online; cit. 03.01.2017].
Retrieved from: http://recognize-speech.com/feature-extraction

[18] McCowan, I.: Microphone Arrays: A Tutorial. 2001.
Retrieved from: http://www.aplu.ch/home/download/microphone_array.pdf

[19] McCowan, I.; Pelecanos, J.; Sridharan, S.: Robust Speaker Recognition using Microphone Arrays. In *IN PROCEEDINGS OF 2001: A SPEAKER ODYSSEY*. 2001.

[20] Prince, S. J.: *Computer Vision: Models Learning and Inference*. Cambridge University Press. 2012.

[21] Prince, S. J.; Elder, J. H.: Probabilistic Linear Discriminant Analysis for Inferences About Identity. In *2007 IEEE 11th International Conference on Computer Vision*. IEEE. 2007. ISBN 9781424416301. pp. 1–8. doi:10.1109/ICCV.2007.4409052.
Retrieved from: http://ieeexplore.ieee.org/document/4409052/

[22] Rahmani, M.; Akbari, A.; Ayad, B.; et al.: Noise cross PSD estimation using phase information in diffuse noise field. *Signal Processing*. vol. vol. 89, no. issue 5. 2009: pp. 703–709. ISSN 01651684. doi:10.1016/j.sigpro.2008.10.020.
Retrieved from:
http://linkinghub.elsevier.com/retrieve/pii/S0165168408003356

[23] Reynolds, D. A.; Quatieri, T. F.; Dunn, R. B.: Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*. vol. vol. 10, no. 1-3. 2000: pp. 19–41. ISSN 10512004. doi:10.1006/dspr.1999.0361.
Retrieved from:
http://linkinghub.elsevier.com/retrieve/pii/S1051200499903615

[24] Rossing, T. D.: *Springer handbook of acoustics*. New York, N.Y.: Springer. c2007. ISBN 0387336338.

[25] Silovský, J.: *Generativní a diskriminativní klasifikátory v úlohách textově nezávislého rozpoznávání a diarizace mluvčího*. PhD. Thesis. Technická univerzita v Liberci, Fakulta mechatroniky, informatiky a mezioborových studií. 2011.

[26] Sizov, A.; Lee, K. A.; Kinnunen, T.: Unifying Probabilistic Linear Discriminant Analysis Variants in Biometric Authentication. page 464. doi:10.1007/978-3-662-44415-3_47.
Retrieved from: http://link.springer.com/10.1007/978-3-662-44415-3_47

[27] Souden, M.; Benesty, J.; Affes, S.: On Optimal Frequency-Domain Multichannel Linear Filtering for Noise Reduction. *IEEE Transactions on Audio, Speech, and Language Processing*. vol. vol. 18, no. issue 2. 2010: pp. 260–276. ISSN 15587916. doi:10.1109/TASL.2009.2025790.
Retrieved from: http://ieeexplore.ieee.org/document/5089420/

[28] Stewart, R.; Sandler, M.: Database of omnidirectional and B-format room impulse responses. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2010. ISBN 9781424442959. pp. 165–168. doi:10.1109/ICASSP.2010.5496083.

[29] Zhang, Y.: Useful Derivations for i-Vector Based Approach to Data Clustering in Speech Recognition. 2011.
Retrieved from: http://people.csail.mit.edu/yzhang87/tech/Ivector.pdf

[30] Zheng, F.; Zhang, G.; Song, Z.: Comparison of different implementations of MFCC. *Journal of Computer Science and Technology*. vol. vol. 16, no. issue 6. 2001: pp. 582–589. ISSN 10009000. doi:10.1007/BF02943243.
Retrieved from: http://link.springer.com/10.1007/BF02943243

Appendices

Appendix A

Contents of the CD

```
/
├── poster
│   └── mic_arrays_for_sre_poster.pdf ....Poster summarizing the thesis.
│                                          It was exhibited at the student
│                                          conference Excel@FIT 2017[1].
├── src
│   ├── beamforming ......................Folder containing MATLAB scripts
│   │                                      that perform beamforming.
│   ├── location_estimation ..............Folder containing python scripts
│   │                                      that perform location estimation
│   │                                      based on score response.
│   ├── simulation .......................Folder containing python scripts
│   │                                      that simulate room conditions.
│   │                                      The simulation.py script depends
│   │                                      on RIR Generator located in
│   │                                      rir-generator-master.  It must
│   │                                      be installed beforehand[2].
│   └── training .........................Folder containing MATLAB scripts
│                                          for the i-vector extractor and
│                                          PLDA training.
└── thesis
    ├── mic_arrays_for_sre_xmosne01.pdf .Thesis document itself.
    └── tex ..............................Folder containing LaTeX source
                                           codes and images.
```

[2] http://excel.fit.vutbr.cz/

[2] Installation instructions and more information are available at https://github.com/Marvin182/rir-generator.

Appendix B

Results of beamforming and system adaptation experiments.

In table B.1, we summarize full results of beamforming and system adaptation experiments in terms of EER, DCF_{08}^{min} and DCF_{10}^{min}. The naming convention is the same as in chapter 6. Minimum variance distortionless response beamforming assuming diffuse noise field is referred to by MVDR.

Table B.1: Results of all the experiments for mismatching enrollment conditions. Each row separated by lines contains three rows with EER, DCF_{08}^{min} and DCF_{10}^{min} in this order.

	4 × 4.5 × 3 m		**8 × 10 × 5 m**	
	females	**males**	**females**	**males**
original, original	10.824 0.513 0.939	6.933 0.312 0.734	10.622 0.511 0.937	6.471 0.294 0.702
original, 2_rooms_2	5.800 0.256 0.654	3.893 0.152 0.422	6.105 0.258 0.643	3.999 0.155 0.385
original, 2_rooms_1	5.718 0.252 0.646	3.691 0.149 0.399	5.841 0.252 0.648	3.842 0.149 0.402
original, rand_rooms_1	4.609 0.194 0.527	3.190 0.118 0.346	5.021 0.206 0.560	3.033 0.120 0.324
DS, original, original	8.429 0.412 0.876	5.662 0.246 0.625	8.297 0.392 0.836	5.511 0.235 0.620
DS, original, 2_rooms_2	4.299 0.187 0.522	2.983 0.115 0.310	4.344 0.189 0.554	3.387 0.118 0.316

DS, **original**, **2_rooms_1**	4.256 0.181 0.512	2.831 0.110 0.309	4.299 0.180 0.539	3.387 0.111 0.317
DS, **original**, **rand_rooms_1**	4.178 0.171 0.505	2.629 0.106 0.304	4.170 0.168 0.495	2.932 0.105 0.326
DS_known_pos, **original**, **original**	8.513 0.412 0.877	5.561 0.246 0.622	6.578 0.321 0.788	4.500 0.172 0.505
DS_known_pos, **original**, **2_rooms_2**	4.342 0.186 0.523	2.942 0.115 0.311	4.305 0.188 0.561	3.539 0.123 0.303
DS_known_pos, **original**, **2_rooms_1**	4.250 0.181 0.514	2.831 0.110 0.309	4.170 0.180 0.549	3.337 0.117 0.303
DS_known_pos, **original**, **rand_rooms_1**	4.127 0.171 0.503	2.668 0.107 0.305	3.826 0.161 0.478	2.831 0.099 0.283
MVDR, **original**, **original**	12.554 0.613 0.950	8.392 0.378 0.822	9.419 0.446 0.890	6.269 0.277 0.673
rand_rooms, **original**	10.576 0.489 0.925	5.612 0.237 0.649	9.716 0.451 0.913	4.702 0.216 0.615
rand_rooms, **2_rooms_2**	6.707 0.295 0.714	5.410 0.193 0.536	6.492 0.304 0.705	4.853 0.199 0.496
rand_rooms, **2_rooms_1**	6.578 0.288 0.721	5.157 0.195 0.519	6.320 0.290 0.703	4.708 0.191 0.491
rand_rooms, **rand_rooms_1**	4.600 0.192 0.511	3.488 0.125 0.343	4.639 0.195 0.532	2.882 0.122 0.337
DS, **rand_rooms**, **original**	8.427 0.415 0.875	4.759 0.197 0.547	7.782 0.371 0.843	4.550 0.187 0.554
DS, **rand_rooms**, **2_rooms_2**	5.116 0.232	4.196 0.161	5.589 0.248	4.452 0.164

	0.598	0.398	0.625	0.413
DS, rand_rooms, 2_rooms_1	5.073	4.095	5.245	4.247
	0.222	0.162	0.235	0.162
	0.587	0.403	0.606	0.427
DS, rand_rooms, rand_rooms_1	4.009	2.932	3.956	2.828
	0.164	0.110	0.166	0.112
	0.478	0.300	0.479	0.326
DS_known_pos, rand_rooms, original	8.392	4.752	6.148	3.994
	0.413	0.197	0.285	0.147
	0.874	0.545	0.774	0.428
DS_known_pos, rand_rooms, 2_rooms_2	5.116	4.045	5.933	4.651
	0.231	0.160	0.257	0.174
	0.599	0.399	0.605	0.404
DS_known_pos, rand_rooms, 2_rooms_1	5.034	4.039	5.804	4.500
	0.222	0.162	0.244	0.172
	0.588	0.401	0.587	0.403
DS_known_pos, rand_rooms, rand_rooms_1	3.955	2.882	3.783	2.781
	0.164	0.110	0.158	0.106
	0.478	0.299	0.446	0.298