From Algorithmic to Neural Beamforming

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Dipl.-Phys. Jonathan David Ziegler

aus Herbolzheim

Tübingen

2021

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	28.01.2022
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Andreas Schilling
2. Berichterstatter:	Prof. Dr. Andreas Koch
3. Berichterstatter:	Prof. Dr. Bernhard Eberhardt

In loving memory of Lisa Ann Klevit-Ziegler

1957-2020

My hero, role model and inspiration.

Abstract

Human interaction increasingly relies on telecommunication as an addition to or replacement for immediate contact. The direct interaction with smart devices, beyond the use of classical input devices such as the keyboard, has become common practice. Remote participation in conferences, sporting events, or concerts is more common than ever, and with current global restrictions on in-person contact, this has become an inevitable part of many people's reality. The work presented here aims at improving these encounters by enhancing the auditory experience. Augmenting fidelity and intelligibility can increase the perceived quality and enjoyability of such actions and potentially raise acceptance for modern forms of remote experiences. Two approaches to automatic source localization and multichannel signal enhancement are investigated for applications ranging from small conferences to large arenas.

Three first-order microphones of fixed relative position and orientation are used to create a compact, reactive tracking and beamforming algorithm, capable of producing pristine audio signals in small and mid-sized acoustic environments. With inaudible beam steering and a highly linear frequency response, this system aims at providing an alternative to manually operated shotgun microphones or sets of individual spot microphones, applicable in broadcast, live events, and teleconferencing or for human-computer interaction. The array design and choice of capsules are discussed, as well as the challenges of preventing coloration for moving signals. The developed algorithm, based on Energy-Based Source Localization, is discussed and the performance is analyzed. Objective results on synthesized audio, as well as on real recordings, are presented. Results of multiple listening tests are presented and real-time considerations are highlighted.

Multiple microphones with unknown spatial distribution are combined to create a large-aperture array using an end-to-end Deep Learning approach. This method combines state-of-the-art single-channel signal separation networks with adaptive, domain-specific channel alignment. The Neural Beamformer is capable of learning to extract detailed spatial relations of channels with respect to a learned signal type, such as speech, and to apply appropriate corrections in order to align the signals. This creates an adaptive beamformer for microphones spaced on the order of up to 100 m. The developed modules are analyzed in detail and multiple configurations are considered for different use cases. Signal processing inside the Neural Network is interpreted and objective results are presented on simulated and semi-simulated datasets.

Zusammenfassung

Zwischenmenschliche Interaktion stützt sich zunehmend auf die Telekommunikation als Ergänzung oder Ersatz des unmittelbaren Kontaktes. Die Teilnahme an Konferenzen, Sportveranstaltungen oder Konzerten aus der Ferne ist angesichts der derzeitigen globalen Einschränkungen ein unvermeidlicher Bestandteil der Lebensrealität vieler Menschen geworden. Diese Arbeit zielt darauf ab, derartige Erfahrungen aufzuwerten. Die Verbesserung von Klangqualität und Verständlichkeit kann die wahrgenommene Qualität und den Spaß an solchen Events steigern und möglicherweise die Akzeptanz für moderne Formen von Fernerlebnissen erhöhen. Zwei Ansätze zur automatischen Signalquellenortung und die daraus resultierende Signalverarbeitung werden untersucht. Der Anwendungsbereich reicht von kleinen Konferenzen bis zu Großveranstaltungen.

Drei Mikrofone mit fester relativer Ausrichtung werden zu einem kompakten Array kombiniert, welches in kleinen und mittleren Räumen Audiosignale höchster Qualität erzeugen kann. Mit der artefaktfreien Steuerung eines virtuellen Richtmikrofons und einem hochlinearen Frequenzgang soll dieses System eine Alternative zu manuell betriebenen Richtrohrmikrofonen oder Einzelmikrofonie bieten, welche für Rundfunk, Live-Events, Telefonkonferenzen oder für moderne Mensch-Maschine-Interaktionen geeignet sind. Das Array-Design und die Auswahl der Kapseln werden ebenso diskutiert wie die Herausforderungen eines richtungsunabhängigen Frequenzgangs. Der entwickelte Algorithmus, welcher auf einer energiebasierten Schallquellenortung basiert, wird diskutiert und die Leistung analysiert. Es werden objektive Ergebnisse zu synthetischen Audioszenen, sowie zu realen Aufnahmen vorgestellt. Die Ergebnisse detaillierter Hörtests werden vorgestellt und Überlegungen zur Echtzeitfähigkeit ausgeführt.

Mehrere Mikrofone unbekannter räumlicher Verteilung werden zu einem Array mit enorm großer Apertur kombiniert und mit neuronalen Netzen in end-to-end-Konfiguration betrieben. Dieses Verfahren kombiniert moderne Einkanal-Signaltrennungsnetze mit adaptiver, domänenspezifischer Kanalsynchronisation. Der Neural Beamformer kann lernen, räumliche Beziehungen von Kanälen, bezogen auf einen gelernten Signaltyp wie Sprache, zu extrahieren und geeignete Korrekturen anzuwenden. Hieraus entsteht ein adaptiver Beamformer für Mikrofone mit einem Abstand in der Größenordnung von 100 m. Die entwickelten Module werden detailliert analysiert und mehrere Konfigurationen für verschiedene Anwendungsfälle werden betrachtet. Die Signalverarbeitung innerhalb des neuronalen Netzes wird analysiert und objektive Ergebnisse werden für simulierte und halbsimulierte Datensätze präsentiert.

Acknowledgements

I wish to thank my advisors Professor Andreas Koch and Professor Andreas Schilling for their untiring support and enthusiasm for my research interests. Additionally, I would like to express my gratitude to Professor Bernhard Eberhardt for his assistance and ready willingness to review this dissertation.

I wish to thank the Institute for Applied Artificial Intelligence at the Stuttgart Media University under Professor Johannes Maucher. The collaborative spirit and collective drive is inspirational, and being able to help found and shape such a wonderful endeavor was an honor and pleasure. In particular, I thank my colleagues Hendrik Paukert and Leon Schröder for their enthusiasm, support, and friendship.

I would like to express my sincere regards to my collaborators and co-authors, who include my advisors (Professor Andreas Koch and Professor Andreas Schilling), my colleagues Hendrik Paukert and Leon Schröder, Professor Oliver Curdt (Stuttgart Media University), Bernfried Runow (University of Tübingen), and Mark Rau (Stanford University).

Special gratitude is owed to Maximus Mutschler for his commitment to building and maintaining the Training Center for Machine Learning (TCML) at the Eberhard Karls University of Tübingen.

Many thanks to the entire team at the LAWO AG for their collaboration and their interest in exploring new possibilities.

And many thanks to everyone at Schoeps Mikrofone for supporting my research with their ideas and hardware.

I am deeply grateful to my family for their unstinting love, help, and support, even across continents and to my wife, who is always there for me. I feel exceptionally fortunate to have my sister's love and support. I am grateful to my father for his guidance and life experience.

Finally, I am forever indebted to my mother for giving me the opportunities, experiences, and encouragement that have made me who I am.

Contents

Acronyms **xiii**

1 List of Publications **1**
1.1 Accepted Manuscripts 1
1.2 Submitted Patents 2
1.3 Preprints 2
1.4 Supervised Student Theses 2

2 Personal Contribution **3**
2.1 A Neural Beamforming Frontend for Distributed Microphone Arrays (2021) 3
2.2 Spatially Informed Neural Beamforming with Distributed Microphone Arrays (2020) 4
2.3 Extraktion eines Audioobjektes (2020) 5
2.4 ASL Using Coincident Microphone Arrays (2020) 6
2.5 Speech Classification for ASL using Convolutional Neural Networks (2018) 7
2.6 Hörversuche zur Entwicklung eines neuartigen Mehrkapsel-Mikrofons (2018) 8
2.7 The Fundamental Problem of the Spectral Subtraction (2018) 9
2.8 Interpolation of Directivity Measurements using Spherical Harmonics (2017) 10
2.9 Interactive Display of Polarity Patterns with non-fixed Frequency Point (2017) 11
2.10 Listening Tests in the Process of Microphone Development (2016) . . . 12

3 Introduction **13**
3.1 Differential Microphone Arrays 13
3.1.1 Gradient Synthesis 15
3.1.2 Energy-Based Acoustic Source Localization 16
3.2 Additive Beamforming with Distributed Arrays 17
3.2.1 Signal Mixture Model 17
3.2.2 Beamforming with Known Positions 18
3.2.3 Acoustic Source Localization for Ad Hoc Arrays 21

3.3 End-to-End Neural Networks for Multichannel Audio Processing 22
3.3.1 Time-Domain Neural Networks 22
3.3.2 Mask Estimation Networks 23
3.3.3 Adaptive Front-Ends 24
3.3.4 Multichannel Networks 25

4 Research Objective **27**

5 Results and Discussion **29**
5.1 Algorithmic Beamforming using Coincident Microphone Arrays 29
5.1.1 Choice of Array Configuration 29
5.1.2 Algorithm Design 31
5.1.3 Tracking Accuracy and Signal Separation 37
5.1.4 Influence of Hop Size on Tracker and Beamformer Performance 39
5.2 Neural Beamforming using Large-Aperture Microphone Arrays 42
5.2.1 Model Design 42
5.2.2 Signal Separation 47
5.2.3 Analysis of Multichannel Processing 51

6 Conclusion **53**

Bibliography **55**

Appendices **59**
A Proof of Spherical Harmonic Base Equivalence 60
B Publications 62

Acronyms

AOI	Angle of Incidence
ASL	Acoustic Source Localization
Bi-LSTM	Bidirectional Long Short-Term Memory
CI	Confidence Index
CNN	Convolutional Neural Network
DI	Directivity Index
DL	Deep Learning
DMA	Differential Microphone Array
DMS	Double-M/S
DNN	Deep Neural Network
DOA	Direction of Arrival
DPRNN	Dual-Path Recurrent Neural Network
DRR	Direct-to-Reverberant Ratio
DS	Delay-and-Sum
DWT	Discrete Wavelet Transform
EBSL	Energy-Based Source Localization
FIR	Finite Impulse Response
FOA	first-order Ambisonics
FS	Filter-and-Sum
GCC	Generalized Cross Correlation
GCC-PHAT	Generalized Cross Correlation with Phase Transform
IR	Impulse Response
MSE	Mean Square Error
NBF	Neural Beamformer

NN	Neural Network
RIR	Room Impulse Response
RMS	Root Mean Square
RNN	Recurrent Neural Network
SAR	Signal-to-Artifact Ratio
SDR	Signal-to-Distortion Ratio
SDRi	Signal-to-Distortion Ratio Improvement
SH	Spherical Harmonics
SHT	Spherical Harmonics Transform
SIR	Signal-to-Interference Ratio
SNR	Signal-to-Noise Ratio
SRP	Steered Response Power
STFT	Short Time Fourier Transform
STN	Spatial Transformer Networks
TCN	Temporal Convolutional Network
TDOA	Time Difference of Arrival
TF	Time-Frequency
VAD	Voice Activity Detection

Chapter 1

List of Publications

1.1 Accepted Manuscripts

- Ziegler, J. D., Schröder, L., Koch, A., and Schilling, A. (2021). A Neural Beamforming Frontend for Distributed Microphone Arrays. In *Proceedings of the 151st Audio Engineering Society International Convention*, Online
- Ziegler, J. D., Paukert, H., Koch, A., and Schilling, A. (2020a). Acoustic Source Localization and High Quality Beamforming Using Coincident Microphone Arrays. In *Proceedings of the 148th AES International Convention*, Vienna, Austria
- Ziegler, J. D., Koch, A., and Schilling, A. (2018). Speech Classification for Acoustic Source Localization and Tracking Applications using Convolutional Neural Networks. In *Proceedings of the 145th Audio Engineering Society International Convention*, New York City, USA
- Ziegler, J. D., Rau, M., Schilling, A., and Koch, A. (2017b). Interpolation and Display of Microphone Directivity Measurements using higher-order Spherical Harmonics. In *Proceedings of the 143rd Audio Engineering Society International Convention*, New York City, USA
- Ziegler, J. D., Paukert, H., and Runow, B. (2017a). Interactive Display of Microphone Polarity Patterns with non-fixed Frequency Point. In *Proceedings of the 142nd Audio Engineering Society International Convention*, Berlin, Germany
- Runow, B., Ziegler, J. D., Paukert, H., Schilling, A., and Curdt, O. (2018). The Fundamental Problem of the Spectral Subtraction. In *30th Tonmeistertagung - VDT International Convention*, Cologne, Germany
- Paukert, H., Ziegler, J., and Koch, A. (2018). Hörversuche zur Entwicklung eines neuartigen Mehrkapsel-Mikrofons. In *30th Tonmeistertagung VdT International Convention*, Cologne, Germany
- Paukert, H. and Ziegler, J. (2016). Listening Tests in the Process of Microphone

Development. In *Proceedings of the 29. Tonmeistertagung VdT International Convention*, Cologne, Germany

1.2 Submitted Patents

- Ziegler, J. D. and Schröder, L. (2020). Extraktion eines Audioobjektes. *Deutsches Patent- und Markenamt (DPMA)*

1.3 Preprints

- Ziegler, J. D., Schröder, L., Koch, A., and Schilling, A. (2020b). Spatially Informed Neural Beamforming with Distributed Microphone Arrays

1.4 Supervised Student Theses

- Hirt, R. (2017). *Development of a virtual conference with focus on optimal reproduction of pre recorded speech.* Bachelor's Thesis, Stuttgart Media University
- Simbürger, C. (2020). *Anwendungen für KI in Digitalmischpulten bei Fußball-Bundesliga-Übertragungen und die damit verbundene Automatisierung von Arbeitsprozessen.* Bachelor's Thesis, Stuttgart Media University

Chapter 2

Personal Contribution

2.1 A Neural Beamforming Frontend for Distributed Microphone Arrays (2021)

In this paper, the adaptive beamformer described in section 2.2 is used as a front-end for state-of-the-art time-domain signal separation networks. The resulting end-to-end network is capable of processing large-aperture microphone arrays and outperforms all baseline methods by a significant amount.

Author	Author position	Scientific ideas %	Data generation %	Analysis & interpretation	Paper writing %
Jonathan Ziegler	1	50	70	60	90
Leon Schröder	2	45	30	40	5
Andreas Koch	3	0	0	0	5
Andreas Schilling	4	5	0	0	0
status	published				

Table 2.1: Author contribution for: *A Neural Beamforming Frontend for Distributed Microphone Arrays*

The personal contribution to this publication covers:

- research idea
- development of algorithms and neural network architectures (60 %)
- implementation of baseline algorithms and networks
- generation of training data (70 %)
- generation of analysis- and results data

Contribution of co-authors:

- Leon Schröder
  - development of algorithms and neural network architectures (40 %)
  - generation of training data (30 %)
- Andreas Koch - project management
- Andreas Schilling - guidance on research direction

2.2 Spatially Informed Neural Beamforming with Distributed Microphone Arrays (2020)

This manuscript presents a physics-informed approach to end-to-end multichannel signal separation. Using fully differentiable frequency-domain Generalized Cross Correlation with Phase Transform (GCC-PHAT) within a deep neural network, domain-specific beamforming filters are generated and applied to massively distributed microphone array signals.

Author	Author position	Scientific ideas %	Data generation %	Analysis & interpretation	Paper writing %
Jonathan Ziegler	1	45	70	50	85
Leon Schröder	2	45	30	50	10
Andreas Koch	3	5	0	0	5
Andreas Schilling	4	5	0	0	0
status	preprint				

Table 2.2: Author contribution for: *Spatially Informed Neural Beamforming with Distributed Microphone Arrays*

The personal contribution to this publication covers:

- research idea
- development of algorithms and neural network architecture (50 %)
- generation of training data (70 %)
- generation of analysis and results data

Contribution of co-authors:

- Leon Schröder
  - development of algorithms and neural network architecture (50 %)

  - generation of training data (30 %)
- Andreas Koch - project management and technical guidance
- Andreas Schilling - guidance on research direction

2.3 Extraktion eines Audioobjektes (2020)

This patent application focuses on one innovative component of the methods described in sections 2.1 and 2.2. Using the multitude of audio channels available within a mixing/broadcasting environment, additional spectral and spatial context information can be obtained for high-quality signal separation. The proposed method combines multiple channels within an end-to-end neural network, incorporating both deep-learning methods and differentiable analytical operations to perform domain-specific beamforming.

Author	Author position	Scientific ideas %	Data generation %	Analysis & interpretation	Paper writing %
Jonathan Ziegler	1	50	-	50	50
Leon Schröder	2	50	-	50	10
Marc Kannengießer	3	0	-	0	40
status	pending approval				

Table 2.3: Author contribution for: *Extraktion eines Audioobjektes*

The personal contribution to this publication covers:

- research idea
- development of methodology (50 %)
- formulation of methodology and visualizations

Contribution by Leon Schröder:

- development of methodology (50 %)

Contribution by Marc Kannengießer:

- technical formulation of patent application
- technical adaptation of visualizations

2.4 Acoustic Source Localization and High Quality Beamforming Using Coincident Microphone Arrays (2020)

This publication presents the second of the two main research ideas of this dissertation. Using three high-quality microphone capsules and a real-time algorithmic signal processing chain, speech signals are tracked and a first-order beam is synthesized. Compared to other approaches, this produces an audio signal with no audible coloration or processing artifacts, making uses within pro-audio applications feasible.

Author	Author position	Scientific ideas %	Data generation %	Analysis & interpretation	Paper writing %
Jonathan Ziegler	1	85	75	80	95
Hendrik Paukert	2	0	25	20	0
Andreas Koch	3	5	0	0	5
Andreas Schilling	4	10	0	0	0
status	published				

Table 2.4: Author contribution for: *Acoustic Source Localization and High Quality Beamforming Using Coincident Microphone Arrays*

The personal contribution to this publication covers:

- research idea, building on the proposition of coincident beamformers by Helmut Wittek (Schoeps Mikrofone)
- planning of the development process, including research assistants and student theses
- preparation of custom hardware components for microphone measurements (70 %)
- measurement of microphone characteristics (50 %)
- preparation of virtual conference based on supervised student thesis [15] and recording of evaluation data (50 %)
- generation of synthetic audio material
- development and implementation of all signal processing components
- development of application prototype with custom microphone hardware by Schoeps
- objective performance evaluation

- methodology for subjective listening tests (10 %)

Contribution of co-authors:

- Hendrik Paukert
  - listening tests on subjective noise disturbance characterization for development prioritization (see 2.10)
  - preparation of custom hardware components for microphone measurements (30 %)
  - measurement of microphone characteristics (50 %)
  - preparation of virtual conference and recording of evaluation data (50 %)
  - listening tests for subjective performance evaluation and comparison with adaptive filtering approaches (see 2.6)
- Andreas Koch - project management
- Andreas Schilling - guidance on research direction

2.5 Speech Classification for Acoustic Source Localization and Tracking Applications using Convolutional Neural Networks (2018)

Based on the broader research topic described in 2.4, this work focuses on a subsection of the processing chain and presents an initial investigation of the use of Convolutional Neural Networks (CNNs) for Voice Activity Detection (VAD) within an Acoustic Source Localization (ASL) algorithm. A multitude of audio buffers is stored and VAD is performed on a Time-Frequency (TF) representation of the signal.

Author	Author position	Scientific ideas %	Data generation %	Analysis & interpretation	Paper writing %
Jonathan Ziegler	1	85	100	100	95
Andreas Koch	2	5	0	0	5
Andreas Schilling	3	10	0	0	0
status	published				

Table 2.5: Author contribution for: *Speech Classification for Acoustic Source Localization and Tracking Applications using Convolutional Neural Networks*

Personal contribution:

- research idea
- implementation of pre-processing algorithms
- Deep Neural Network (DNN) architecture design and training
- incorporation of module into ASL algorithm
- performance evaluation

Contribution of co-authors:

- Andreas Koch - project management
- Andreas Schilling - guidance on research direction

2.6 Hörversuche zur Entwicklung eines neuartigen Mehrkapsel-Mikrofons (2018)

Based on 2.10, a larger set of listening tests was performed using the results of the developed algorithms presented in 2.4 and 2.5. The goal was to determine subjective preferences between different methods of speech enhancement and beamforming. Next to anchor and reference signals required for the selected test method, the supercardioid beam created by the developed algorithm was compared with a static, omnidirectional signal and various adaptive filters, some commercially available, some currently under development [32, 34].

Author	Author position	Scientific ideas %	Data generation %	Analysis & interpretation	Paper writing %
Hendrik Paukert	1	50	60	80	85
Jonathan Ziegler	2	45	40	20	10
Andreas Koch	3	5	0	0	5
status	published				

Table 2.6: Author contribution for: *Hörversuche zur Entwicklung eines neuartigen Mehrkapsel-Mikrofons*

Contribution of Hendrik Paukert (main author):

- conception, implementation, execution and evaluation of listening tests
- execution of virtual conference scenario (50 %)
- recording of test signals for and generated by the virtual conference (25 %)

Personal contributions are:

- research idea
- conception and realization of reproducible loudspeaker scenario (virtual conference) as test system based on [15]
- execution of virtual conference scenario 50 %
- recording of test signals for and generated by the virtual conference 75 %

Contribution of co-authors:

- Andreas Koch - project management

2.7 The Fundamental Problem of the Spectral Subtraction (2018)

During the development of adaptive filtering methods used in one of the compared beamforming algorithms in the work described in section 2.6, basic questions of frequency-domain audio filtering arose. In this publication, spectral subtraction is analyzed on a theoretical basis in order to assess potential shortcomings of the approach, compared to time-domain Finite Impulse Response (FIR)-filtering.

Author	Author position	Scientific ideas %	Data generation %	Analysis & interpretation	Paper writing %
Bernfried Runow	1	85	100	80	85
Jonathan Ziegler	2	0	0	10	10
Hendrik Paukert	3	0	0	0	5
Andreas Schilling	4	10	0	5	0
Oliver Curdt	5	5	0	5	0
status	published				

Table 2.7: Author contribution for: *The Fundamental Problem of the Spectral Subtraction*

Contribution of Bernfried Runow (main author):

- research idea
- theoretical base
- conception, implementation and execution of experiments

Personal contributions to this publication are:

- assistance with theoretical work
- partial translation and editorial work

Contribution of co-authors:

- Hendrik Paukert - writing assistance
- Andreas Schilling - guidance on research direction and theoretical base
- Oliver Curdt: - guidance on research direction and theoretical base

2.8 Interpolation and Display of Microphone Directivity Measurements using higher-order Spherical Harmonics (2017)

While developing the application presented in 2.9, questions arose about different interpolation and smoothing methods for microphone polar plots. This work explores the advantages of using Spherical Harmonics (SH) as a set of base functions that match the behaviour of Differential Microphone Arrays (DMAs) and expands the work presented in 2.9 into three dimensions.

Author	Author position	Scientific ideas %	Data generation %	Analysis & interpretation	Paper writing %
Jonathan Ziegler	1	85	100	90	75
Mark Rau	2	0	0	10	20
Andreas Koch	3	5	0	0	5
Andreas Schilling	4	10	0	0	0
status	published				

Table 2.8: Author contribution for: *Interpolation and Display of Microphone Directivity Measurements using higher-order Spherical Harmonics*

The personal contribution spans:

- research idea
- conception of experiment
- theoretical basis (90 %)
- implementation of Spherical Harmonics Transform (SHT) algorithms
- complete development of the application

Contribution of co-authors:

- Mark Rau - assistance in theoretical foundation of spherical harmonics (10 %)
- Andreas Koch - project management
- Andreas Schilling - guidance on research direction

2.9 Interactive Display of Microphone Polarity Patterns with non-fixed Frequency Point (2017)

During initial experiments with different methods of ASL, exact information about the frequency-dependent directivity of the individual microphone capsules and synthesized beams was essential. Manufacturers generally supply smoothed graphs, either as stacked frequency-response plots with several Angles of Incidence (AOIs) or as stacked polar plots. Even in close collaboration with a microphone manufacturer, no detailed information was attainable. For this reason, frequency measurements were performed in an ISO 3745 Precision Class 1 anechoic chamber and processed to accurately display directivity patterns of individual microphones or entire arrays at any desired frequency set by the user [16].

Author	Author position	Scientific ideas %	Data generation %	Analysis & interpretation	Paper writing %
Jonathan Ziegler	1	90	60	80	90
Hendrik Paukert	2	10	40	0	10
Bernfried Runow	3	0	0	20	0
status	published				

Table 2.9: Author contribution for: *Interactive Display of Microphone Polarity Patterns with non-fixed Frequency Point*

The personal contribution within this work covers:

- research idea
- conception of the experiment
- design of custom microphone mounting hardware
- construction of custom microphone hardware (30 %)
- recording of the test signals in cooperation with Hendrik Paukert and the research assistants at the Fraunhofer IDMT in Ilmenau (45 %)
- post processing and smoothing to extract the Room Impulse Responses (RIRs)

- complete development of the final user application

Contribution of co-authors:

- Hendrik Paukert
  - construction of custom microphone hardware (70 %)
  - recording of the test signals (45 %)
- Bernfried Runow - theoretical base for microphone directivity plots

2.10 Listening Tests in the Process of Microphone Development (2016)

This publication focuses on listening tests prior to the development of the application and algorithms described in publications 2.4 and 2.5. The tests were performed to assess subjective reactions to various types of noise in order to prioritize during the development process.

Author	Author position	Scientific ideas %	Data generation %	Analysis & interpretation	Paper writing %
Hendrik Paukert	1	60	100	80	70
Jonathan Ziegler	2	40	0	20	30
status	published				

Table 2.10: Author contribution for: *Listening Tests in the Process of Microphone Development*

Contribution of Hendrik Paukert (main author):

- research idea
- conception, implementation, execution, and evaluation of listening tests

Personal contributions to this publication are:

- programming assistance for listening test
- assistance with selection of test methods
- translation and editorial work

Chapter 3

Introduction

Beamforming is the process of manipulating and combining the output signals of multiple sensors within an array. The objective of this process is generating an enhanced array output signal with respect to the spatial relations of the sensors and the desired signal components [40]. In audio signal processing, both additive and subtractive methods are applied. The specific approaches chosen within the scope of this dissertation are outlined in the following sections. The discrete time and frequency indices $[t]$ and $[f]$ are omitted for improved clarity of the equations, except for cases in which the variable is relevant to the performed operations.

3.1 Differential Microphone Arrays

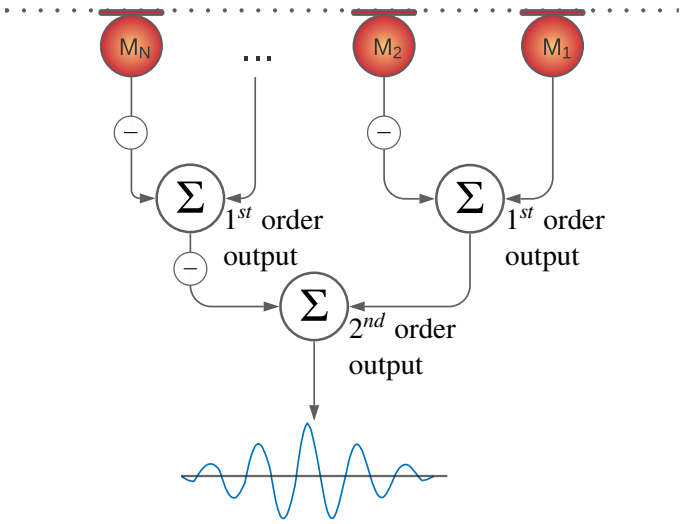


Figure 3.1: Signal processing of a second-order Differential Microphone Array.

Differential Microphone Arrays are sensitive to the sound pressure gradient, or the spatial derivative of the sound pressure field. This response can be implemented by subtracting

the signals of two closely spaced omnidirectional, sound pressure-sensitive microphones, resulting in a first-order DMA. Higher orders N are achieved by subtracting two differential arrays of order N-1. Figure 3.1 shows a basic linear microphone array, with the output signals processed to create a second-order DMA.

For a first-order DMA, the angular response magnitude can be approximated to [2]:

$$P(\theta) = \cos\theta, \tag{3.1}$$

with phase inversions at $\pm 90°$. This response is shown in Figure 3.2. Sound pickup of pressure gradient sensors is inherently frequency dependent. With the pressure gradient expressed as

$$\nabla e^{i\omega t} = i\omega e^{i\omega t}, \tag{3.2}$$

a linear dependency with the angular velocity ω becomes apparent [10]. Equation (3.2) translates into an attenuation of $6\,\mathrm{dB/octave}$ and a phase shift of $90°$ with respect to the sound pressure. In practice, the frequency and phase response are compensated by electrical equalization circuits to linearize the microphone output. Figure 3.3 shows the initial and corrected frequency responses of a figure-of-eight pressure gradient microphone. Besides the linear slope, nulls at high frequencies can be observed. These nulls correspond to frequencies where the distance d matches the wavelength λ. In order to push the first null to higher frequencies, small distances d are required. This restriction poses the challenge of diminishing pressure gradients, requiring significant amplification. Diffraction and resonance effects can be utilized to increase the spacing d slightly, thus increasing the Signal-to-Noise Ratio (SNR) of the output [10].

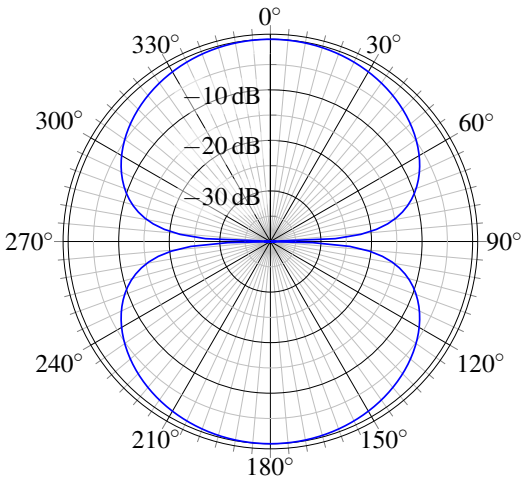


Figure 3.2: Polar response of microphone or microphone array with a dipole, or figure-of-eight pickup pattern.

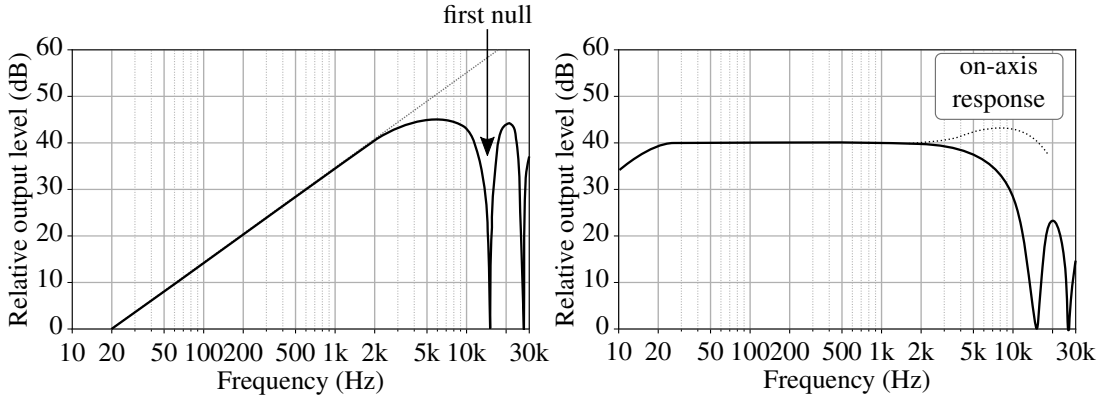


Figure 3.3: Frequency response of figure-of-eight microphone, before and after ideal correction filters are applied. The dotted line indicates the actual frequency for on-axis incidence due to diffraction and resonance effects, reproduced from [10].

3.1.1 Gradient Synthesis

The most basic sound pressure gradient sensor can be realized without multiple transducers. By openly mounting a single diaphragm without a sealed backing chamber, the excitation of the diaphragm becomes pressure gradient-sensitive. As current transducer technology relies on two-dimensional diaphragms, for instance as one side of a capacitor or in a moving coil or ribbon configuration, sound pressure gradients can be efficiently detected only in one dimension, resulting in the anisotropic sensitivity shown in Figure 3.2 and described in Eq. (3.1) [10]. The signals of closely spaced omnidirectional and bidirectional microphones W and Y can be combined to create variable pickup patterns, parameterized by the pickup parameter p [10, 33]:

$$M_p = pW + (1-p)Y. \tag{3.3}$$

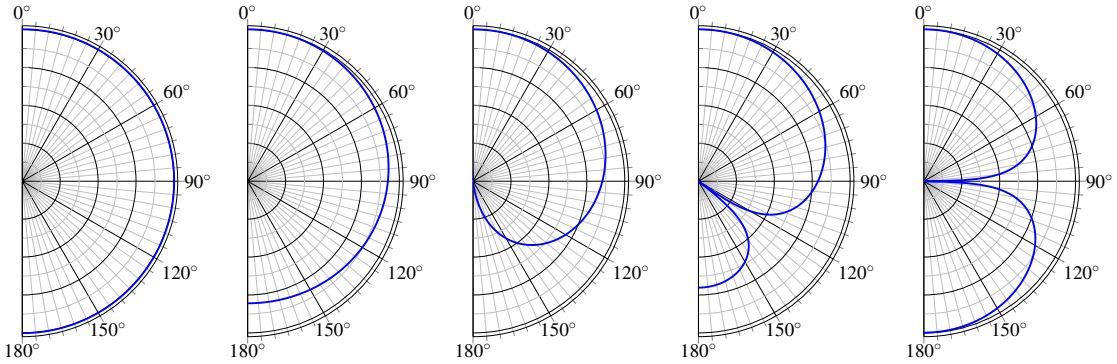
Reconstructions of the most common microphone pickup patterns are shown in Figure 3.4. By mixing the signals X and Y of two stacked, orthogonal figure-of-eight microphones, the signal $\hat{Y}$ of a virtual figure-of-eight capsule can be synthesized. The orientation θ of this virtual capsule can be variably set by combining the signals using

$$\hat{Y}_\theta = X\cos\theta + Y\sin\theta. \tag{3.4}$$

By combining (3.3) and (3.4), any first-order virtual pickup pattern can be pointed at any angle on the plane constructed by the axes of X and Y using

$$M_{\theta,p} = pW + (1-p)(X\cos\theta + Y\sin\theta). \tag{3.5}$$

Equation (3.5) represents the basic approach to beamforming used in [51, 52] to create both the first-order output signal and the steered beam described in the following section.



$p = 1$	$p = 0.7$	$p = 0.5$	$p = 0.37$	$p = 0$
Omnidirectional	Subcardioid	Cardioid	Supercardioid	Figure-of-Eight

Figure 3.4: Variable polar pickup pattern when mixing omnidirectional and bidirectional microphone signals according to Equation (3.3) under variation of the mixing parameter p. Different values for p represent known pickup patterns.

3.1.2 Energy-Based Acoustic Source Localization

In order to make better use of the array described in Section 3.1.1, an ASL algorithm must be implemented. The localization algorithm outputs the desired direction of accentuation. For moving sources and arrays, this algorithm must be adaptive and, optimally, provide sufficiently short processing times to be used in a real-time environment. A popular approach to ASL is the Steered Response Power (SRP) algorithm [9, 24]. In the chosen time-domain approach, a virtual microphone beam is generated using Eq. (3.5) and steered across the entire range of possible AOIs before returning the direction with the highest sound pressure level. When using a synthesized set of cardioid signals $M_{\theta,p=0.5}$, spanning the entire space of possible angles θ,

$$\mathcal{M} = \{M_\theta \ \forall \ \theta = 0, 1, \ldots, 359\}, \tag{3.6}$$

the energy of the discrete signals of length L corresponding to the directions can be computed via Root Mean Square (RMS):

$$\overline{\mathcal{M}} = \left\{ \sqrt{\frac{1}{L}\sum_{t}^{L} {M_\theta}^2} \ \forall \ \theta = 0, 1, \ldots, 359 \right\}. \tag{3.7}$$

The signal corresponding to the angle with the highest signal energy then represents the approximated Direction of Arrival (DOA) of the current audio frame:

$$\theta_{DOA} = \max_{\theta} \left(\overline{\mathcal{M}} \right). \tag{3.8}$$

Within the scope of this dissertation, multiple enhancements to the basic ASL algorithm are introduced to generate a stable and reactive beam from a coincident microphone array [51, 52]. The basis of these stabilization algorithms is the process of exponential smoothing, in which the output of the smoothing algorithm is composed partly of the output of the previous time step and partly of the current estimation for θ_{DOA}:

$$\theta_s^0 = \theta_{DOA}^0 \tag{3.9}$$

$$\theta_s^t = \alpha\, \theta_{DOA}^t + (1-\alpha)\, \theta_s^{t-1}. \tag{3.10}$$

By changing α for each audio buffer dynamically, depending on a set of quality metrics discussed in Section 5.1.2, the filtered DOA output θ_s greatly improves upon the simple maximization described above and provides sufficiently stable directional information for first-order beamforming as described in Equation (3.5).

3.2 Additive Beamforming with Distributed Arrays

3.2.1 Signal Mixture Model

The mixtures of I signals s_i and J noises n_j at multiple positions in an acoustic environment recorded by microphones m_ν can be expressed as:

$$m_\nu = \sum_{i=1}^{I} \tilde{s}_i^\nu + \sum_{j=1}^{J} \tilde{n}_j^\nu. \tag{3.11}$$

Every signal $\tilde{s}_i^\nu$ and noise $\tilde{n}_j^\nu$ consist of the corresponding signal and noise sources s_i and n_j, propagated from their respective source positions to the transducer m_ν. The propagation transformations are expressed as convolutions with corresponding impulse responses h [4, 53]:

$$\tilde{s}_i^\nu = s_i * h_{s_i}^\nu, \quad \tilde{n}_j^\nu = n_j * h_{n_j}^\nu, \tag{3.12}$$

resulting in

$$m_\nu = \sum_{i=1}^{I} \left(s_i * h_{s_i}^\nu\right) + \sum_{j=1}^{J} \left(n_j * h_{n_j}^\nu\right). \tag{3.13}$$

Equation (3.13) is visualized in Figure 3.5. The arrows represent propagation paths, expressed with the corresponding impulse responses h_x^ν.

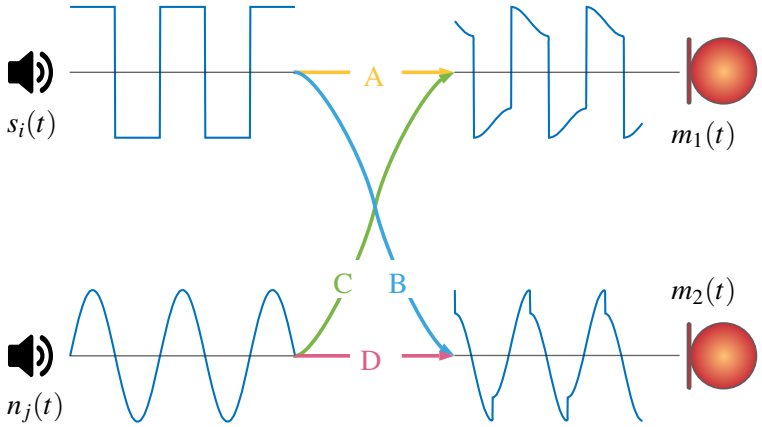


Figure 3.5: Visualization of signal and noise propagation to multiple transducers.

3.2.2 Beamforming with Known Positions

Delay-and-Sum beamformers utilize spaced arrays with transducer spacings starting at a few centimeters. As displayed in Figure 3.6, the Time Difference of Arrival (TDOA) between individual microphone pairs can be directly computed from the AOI θ, the microphone spacing d, and the speed of sound c.

By appropriately delaying the individual microphones by the calculated TDOA prior to summation, sound arriving from the angle θ is favored in the output signal. The array output m_ν of microphone ν with respect to the desired signal s and the diffuse noise n at time step t can be expressed similarly to (3.11) as:

$$m_\nu(t) = s(t-\tau_\nu) + n(t), \quad \nu = 1,\ldots,M, \tag{3.14}$$

with τ_ν describing the relative time delay between microphones 1 and ν [2]. For the linear array shown in Figure 3.6, τ can be computed using the speed of sound c and the AOI θ as

$$\tau_\nu = \frac{(\nu-1)\,d\cos\theta}{c}. \tag{3.15}$$

Following McCowan [23], and assuming an odd number N, equally spaced transducers with identical frequency responses, the horizontal, frequency-dependent directivity pattern D for the array shown in Figure 3.6 can be expressed using the complex frequency-dependent weights w_n as

$$D\left(f,\theta,\phi=\frac{\pi}{2}\right) = \sum_{-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f)\,e^{i\frac{2\pi f}{c}nd\cos\theta}. \tag{3.16}$$

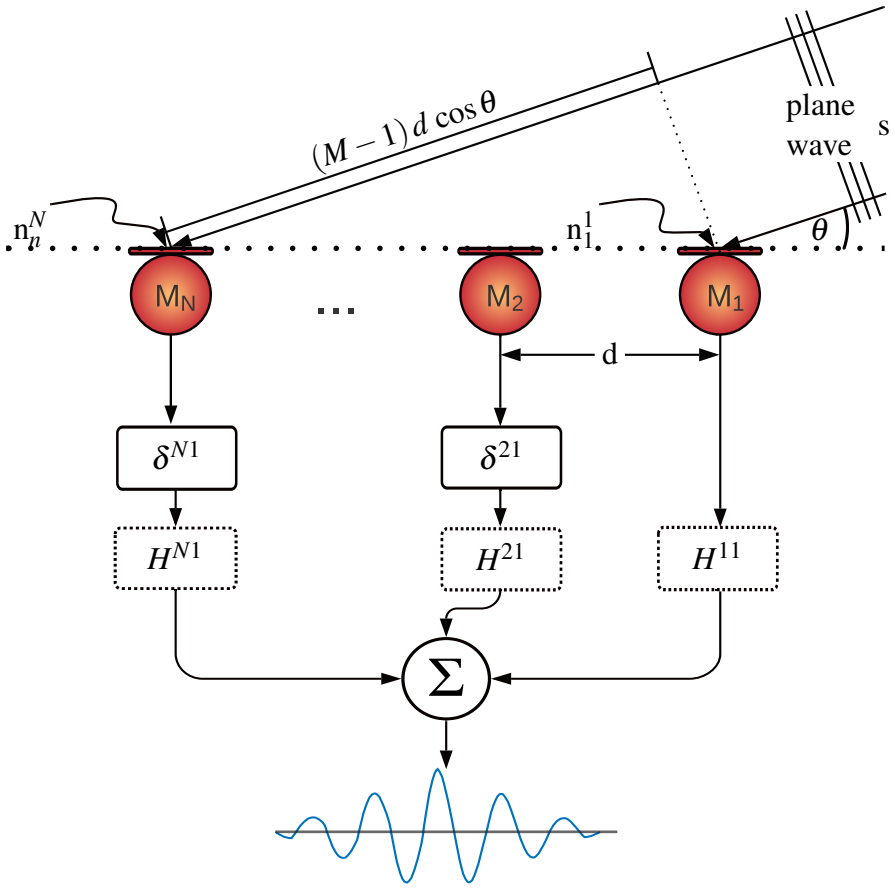
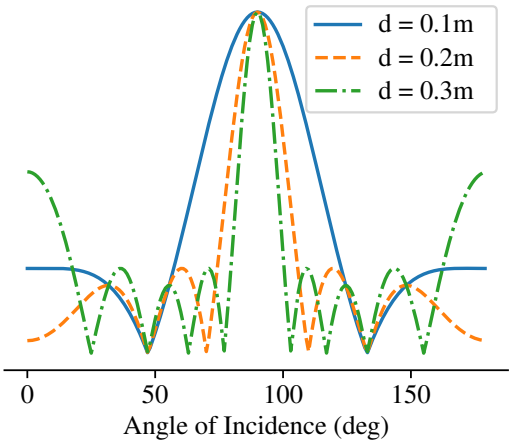


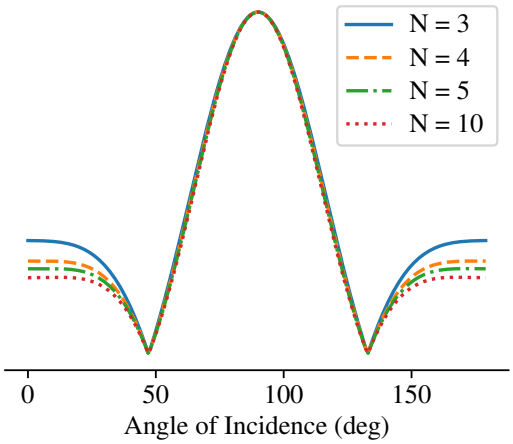
Figure 3.6: Signal processing for a Delay-and-Sum beamformer with optional Finite Impulse Response filters H for Filter-and-Sum beamforming, reconstructed from [2].

Figure 3.7 a shows the directivity pattern for such arrays under variation of the microphone spacing d. While the main lobe is sharpened for larger inter-transducer distances, the number and relative sensitivity of the side lobes increases. Figure 3.7 b compares the directivity patterns of multiple arrays of the same geometric proportions under the variation of the number of transducers. Additional transducers attenuate the side lobe without affecting the position of nulls.

As the described method exhibits fundamental restrictions, such as frequency-dependent directivity patterns shown in Figure 3.8, sub-band and Filter-and-Sum (FS) beamformers were proposed [3, 11]. Within the scope of this dissertation, a data-driven approach was chosen to implicitly estimate time shifts δ and filters H for FS beamforming [48, 53, 54]. This method is detailed in Section 3.3.



(a) Fixed number of microphones with different inter-microphone spacing d.



(b) Fixed array length with different number of transducers N.

Figure 3.7: Relative directivity pattern for linear microphone arrays under variation of spatial sampling. Linear scale used.

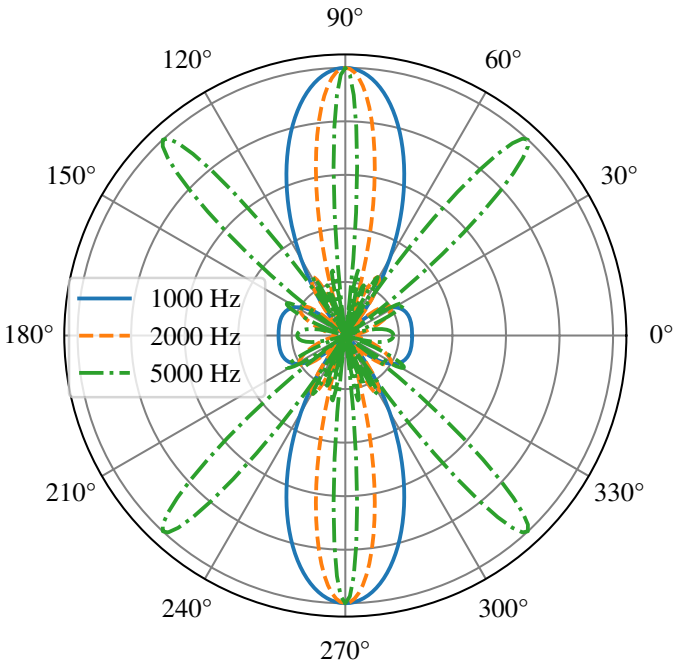


Figure 3.8: Linear microphone array: polar visualization of relative directivity for multiple frequencies. Linear scale used.

3.2.3 Acoustic Source Localization for Ad Hoc Arrays

For situations with known transducer and source positions, solving for τ_m is a geometrical problem. Unknown transducer distributions or source positions require alternative approaches to find the correct TDOA. One such approach uses the cross-correlation between signal pairs as an indicator for the correct delay between the two microphones.

Considering two real-valued input vectors $m_x[t]$, the discrete correlation, represented with the $\star$ operator, can be expressed as

$$CC_{m_1,m_2}[t] := (m_1 \star m_2)[t] = \sum_{\tau=-\infty}^{\infty} m_1[\tau] \cdot m_2[\tau + t]. \tag{3.17}$$

In the absence of interference and reverberation and for a single source x, the main peak of the CC vector indicates the TDOA of the source with respect to the two input channels:

$$\delta^{1,2} = argmax\left[\mathrm{CC}_{m_1,m_2}[t]\right]. \tag{3.18}$$

Unfortunately, this method becomes unstable for reverberant or noisy environments. One attempt to improve the stability of such algorithms is to apply weighting filters to the individual signals prior to the computation of the cross-correlation vector. Using the convolution theorem, (3.17) can be expressed as

$$(m_1 \star m_2) = \mathcal{F}^{-1}\left\{\overline{\mathcal{F}\{m_1\}} \cdot \mathcal{F}\{m_2\}\right\}, \tag{3.19}$$

with $\mathcal{F}\{\quad\}$ representing the Fourier transform and $\overline{[\quad]}$ the complex conjugation. Applying Phase Transform weighting leads to the computation of the GCC-PHAT:

$$\text{GCC-PHAT}_{m_1,m_2} = \mathcal{F}^{-1}\left\{\frac{\overline{\mathcal{F}\{m_1\}} \cdot \mathcal{F}\{m_2\}}{\left|\overline{\mathcal{F}\{m_1\}} \cdot \mathcal{F}\{m_2\}\right|}\right\}. \tag{3.20}$$

This Generalized Cross Correlation (GCC) provides an increased robustness to noise but still becomes quite unstable in reverberant environments.

3.3 End-to-End Neural Networks for Multichannel Audio Processing

With machine learning helping to advance the state of the art in many areas of audio signal processing, Neural Beamforming remains a challenging research subject. The approach pursued in this dissertation aims at combining knowledge of the physical systems with data-driven, adaptive models for filtering and masking of time-domain audio signals.

3.3.1 Time-Domain Neural Networks

Time-domain processing with DNNs is a relatively new approach, with dilated Temporal Convolutional Networks (TCNs) currently producing among the best results. This approach was proposed for audio as the WaveNet by Oord *et al.* [27]. The architecture utilizes dilated causal convolutional layers with exponentially increased dilation rates. Dilated convolutions insert empty positions into the convolution kernels, hence increasing the receptive field without increasing the number of computations. WaveNet uses causal convolutions, making the model capable of operating at per-sample resolution. Alternatively, block processing with sufficiently large blocks can be applied [21]. Additionally, gated activation units are implemented as proposed in [26]. This method combines the output of two dilated convolutions $W_f * x$ and $W_g * x$ at layer k, applied to the same input x, using the tanh and the sigmoid activation functions and an element-wise multiplication $\bigotimes$, so that

$$z = \tanh\left(W_{f,k} * x\right) \bigotimes \sigma\left(W_{g,k} * x\right). \tag{3.21}$$

A pointwise convolution and a residual connection with the input complete the gated activation block. The entire module is shown in Figure 3.9.

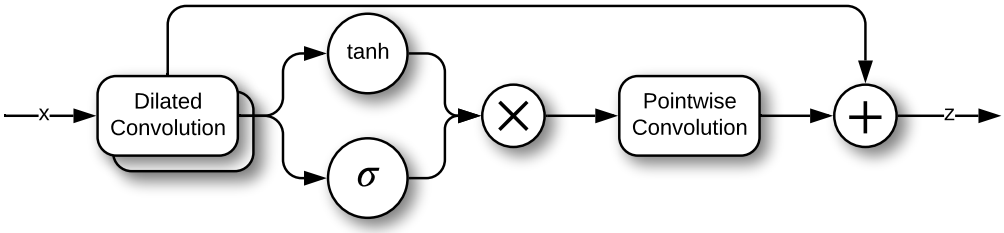


Figure 3.9: Gated Activation Unit with skip connections, reconstructed from [27].

Oord *et al.* additionally introduced Conditional WaveNets in which a latent global context vector h or context time series y can be used to condition the generative output by

expanding (3.21) with the linear projections $V_{*,k}$ to

$$z = \tanh\left(W_{f,k} * x + V_{f,k}^T h\right) \bigotimes \sigma\left(W_{g,k} * x + V_{g,k}^T h\right) \tag{3.22}$$

and

$$z = \tanh\left(W_{f,k} * x + V_{f,k} * y\right) \bigotimes \sigma\left(W_{g,k} * x + V_{g,k} * y\right). \tag{3.23}$$

3.3.2 Mask Estimation Networks

The most effective and popular method of Deep Learning (DL)-based time-domain audio signal separation currently is mask estimation. Comparable to a mask for an image, per-sample amplitude masks are generated by the DNN. This mask is then multiplied element-wise with the audio signal to remove unwanted components. Luo and Mesgarani use stacked convolutional blocks as shown in Figure 3.10 for mask estimation in the TasNet architecture [21]. Analogous to the previously described WaveNet architecture, these convolutional blocks are stacked M times with dilation rates increasing exponentially from 2^0 to 2^{M-1}. This stack is then repeated R times to create the full model.

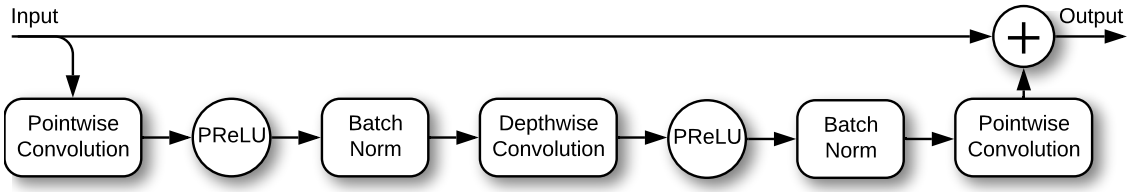


Figure 3.10: Convolutional Block in TasNet architecture, reconstructed from [21].

An alternative approach uses Recurrent Neural Networks (RNNs) for time-domain mask estimation. The Dual-Path Recurrent Neural Network (DPRNN) architecture segments the input buffers into volumes of smaller, overlapping sections, or chunks, which are then processed individually by Bidirectional Long Short-Term Memory (Bi-LSTM) layers. During intra-chunk processing, visualized as the first block in Figure 3.11, the recurrent units, a dense layer, and LayerNorm normalization are applied to the time axis of the individual chunks. Subsequent inter-chunk processing, shown in the second block, applies the same steps along the chunk axis of the input tensor.

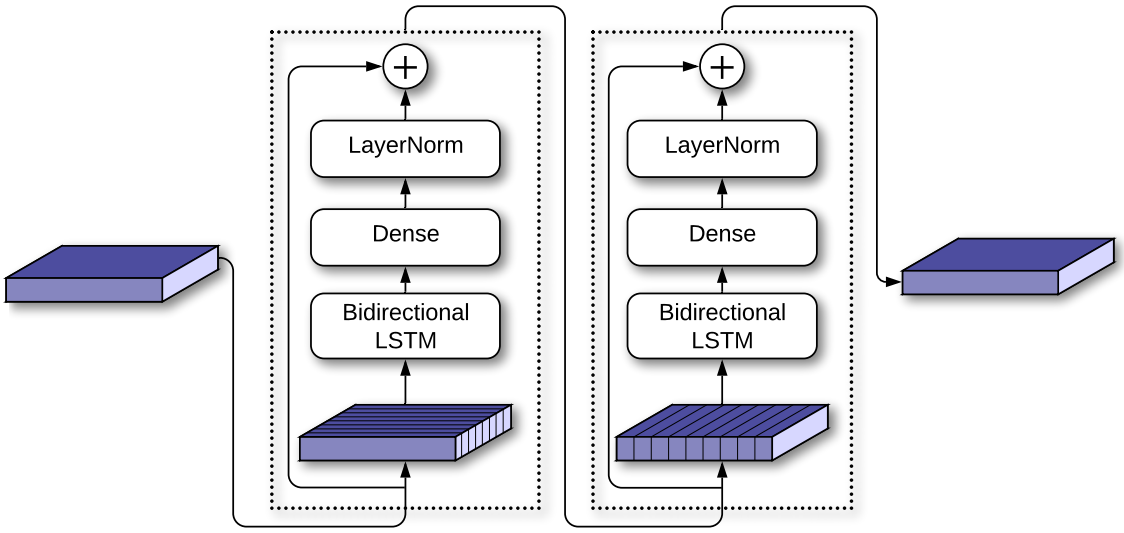


Figure 3.11: Basic configuration of a Dual-Path RNN block. The input volume of segmented, overlapping time series is processed along the time axis and the chunk axis independently, using Bidirectional RNN units [22].

3.3.3 Adaptive Front-Ends

Mask estimation is often combined with an adaptive front-end. This module learns decompositions into a higher-dimensional feature space and the corresponding recombinations of the signal, facilitating signal separation [43]. Examples of algorithmic analoga are TF representations such as the Discrete Wavelet Transform (DWT) and the Short Time Fourier Transform (STFT) [1, 5, 37]. This decomposition can easily be implemented using linear, one-dimensional convolutional layers, with the recombination consisting of one-dimensional, transposed convolutional layers. The large advantage of incorporating the encoding into the network architecture is the ability to learn domain-specific representations. While the algorithmic approaches are theoretically capable of accounting for domain-specific features such as restricted frequency ranges or nonlinear frequency bin spacing, fine-tuning this manually requires precise knowledge of the entire space of possible input signals. As the parameter space of a linear encoder architecture is comparatively small, the time consumed by this task of manual fine-tuning is in no relation to the negligible gain in processing time. A hybrid approach has been proposed by Ravanelli and Bengio in which the kernels of the convolutional layer are fixed to discrete sinc functions, functioning as band-pass filters [31]. The only learnable parameters are the lower and upper cutoff frequency for each band. This approach functions as expected; an actual advantage over learning the entire filters can only be shown for extremely small datasets. Limited data are not a restriction for the pursued application.

3.3.4 Multichannel Networks

In order to apply the described methods to microphone array signals, some form of cross-channel dependency is required within the network. For small TDOA and fixed array geometries, a CNN is capable of learning to implicitly compensate time delays using one-dimensional convolutional kernels. For ad hoc, large-aperture arrays, this is not possible, as the convolutional kernels are orders-of-magnitude smaller than the required time shift. The use of dilated convolutions can mitigate this issue, although the reduced resolution of such operations makes accurate, large-scale shifts impossible. Multiple approaches to solving the task of Time Difference of Arrival estimation using DNNs are possible. A relatively small Neural Network (NN) can be used to estimate time delays from the cross-correlation matrix of the entire array or individual microphone pairs. This delay can then be applied algorithmically prior to channel summation, resulting in a neural Delay-and-Sum (DS) beamformer with increased tracking accuracy.

Building on this approach, the estimated delays can be applied to the signals by way of Spatial Transformer Networks (STN) [17]. This presents the advantage of differentiability, hence enabling further processing of the signals post-alignment within an end-to-end system. To implement a STN module, a localization network component $f_{loc}(U)$ is dedicated to outputting the correct transformation parameters θ for the transformation $\mathcal{T}_\theta$ from the cross-correlation feature map U. Depending on the desired transformation $\mathcal{T}$ and the dimensionality of U, the size of θ can vary. The set of two-dimensional affine transformations from $U \in \mathbb{R}^{H\times W\times C}$, for example, requires $\theta \in \mathbb{R}^6$. The less complex problem of extracting a subsection L' of a one-dimensional time-series of length L, expanded to C feature channels, expressed with the feature map $U \in \mathbb{R}^{L\times C}$, requires only a single value for θ. A sampler is used to obtain the output feature map V. This process requires the transformation $\mathcal{T}(G)$, applied to a regular grid G, and the input feature map U. Any differentiable sampling can be utilized. With the parameters Φ_t of a generic sampling kernel $k()$, this can be expressed for the source time steps x_i^s as

$$V_i^c = \sum_t^L U_t^c k(x_i^s - t; \Phi_t) \quad \forall i \in [1 \dots L'] \quad \forall c \in [1 \dots C]. \tag{3.24}$$

The common linear interpolation can be expressed as

$$V_i^c = \sum_t^L U_t^c \max(0, 1 - |x_i^s - t|). \tag{3.25}$$

A more powerful approach is that of impulse response estimation, proposed and presented as the Neural Beamformer within this dissertation. Creating a DNN that not only estimates the TDOA, but the entire associated impulse response, enables the network to perform more sophisticated alignment of the individual array channels. Frequency cor-

rections, as well as echo cancellation and the suppression of reverberation, can be learned and applied through the convolution with a single, albeit relatively long, kernel. Starting with the recorded signals m_ν from (3.13), additional impulse responses h^ν can be found that maximize the signal components s_i in the sum over ν:

$$\varsigma_i = max_{s_i} \left\{ \sum_\nu m_\nu * h^{\nu i} \right\}. \tag{3.26}$$

Additional beams can be synthesized for subtractive noise reduction, resulting in

$$\varsigma_i = max_{s_i} \left\{ \sum_{\nu,j} m_\nu * h^{\nu i j} \right\}. \tag{3.27}$$

Using the distributive properties of convolutions

$$f * g + f * h = f * (g + h), \tag{3.28}$$

Equation (3.27) can be modified in a way that only one filter is required per target i and microphone ν. The noise components are implicitly combined into a single filter $\tilde{h}$, resulting in

$$\varsigma_i = max_{s_i} \left\{ \sum_\nu m_\nu * \tilde{h}^{\nu i} \right\}. \tag{3.29}$$

Lastly, this fully differentiable Neural Beamformer can be combined with sophisticated single-channel mask estimation networks to produce state-of-the-art results.

Chapter 4

Research Objective

The goal of the research presented in this dissertation is high quality beamforming in challenging acoustic environments that is accessible to pro-audio use. In order to successfully apply signal enhancement in a pro-audio setting, two main constraints have to be considered, namely processing time and fidelity.

For most uses, real-time processing is essential. The definition of real-time can vary between the lowest possible processing latency of one sample to the threshold of human disturbance. Multiple studies outlined in [14] show that, for speakers without hearing loss, time delays of over 10 ms to 30 ms are experienced as bothersome. In broadcasting and telecommunication, real-time capability faces additional technical restrictions. For broadcasting, a time delay of less than one video frame, equivalent to 25 ms, can be considered acceptable. For telecommunication, network delays of well over 100 ms mask potential audio delays [46]. To achieve such low processing times, the audio must be processed in blocks shorter than the upper bound on sample delays. Additionally, processing cannot utilize backward passes through the signal. Lastly, computation cost is important, as processing times must be shorter than the block lengths. This requires efficient processing methods optimized for the deployment hardware.

While many audio signal separation tasks are performed on reduced-bandwidth signals to meet the requirements of telecommunication, pro-audio applications require full bandwidth signals ranging at least from 20 Hz to 20 000 Hz. Accurate high-frequency processing can be challenging, especially for DL-based approaches. The sampling rate f_s of the signal processing chain must be chosen according to the sampling theorem to be at least twice the highest frequency present in the signal [44].

Apart from bandwidth, high fidelity signifies reducing distortion to a minimum, or preventing it completely. Distortion can affect the frequency response, causing coloration of the sound. Dynamic distortion can occur when introducing nonlinear processing. This can lead to effects ranging from amplitude modulation to the introduction of unwanted frequencies into the output signal. When using generative approaches such as the presented DNN, unwanted signal components can be generated and introduced into

the signal. Lastly, discontinuities at the borders of the processing blocks can cause periodic artifacts such as clicking or ringing, depending on the block size. Throughout the dissertation, the Signal-to-Distortion Ratio (SDR) is chosen as a quality metric for signal separation and fidelity retention, combining characteristics of the Signal-to-Interference Ratio (SIR) and Signal-to-Distortion Ratio (SDR) metrics [18].

To match the defined requirements, two approaches were researched. Algorithmic processing of coincident microphone array signals was developed for applications in which a predefined array can be used in comparatively simple acoustic environments. Simple acoustic environments are defined by small room dimensions, microphone-to-source distances in the order of 10^{-1} m to 10^{1} m, and a high Signal-to-Noise Ratio (SNR). One main challenge for this method lies in accurately localizing and tracking the target source in order to synthesize a microphone beam in the correct direction. Rapid, erroneous jumps can cause two forms of distortion. Strong discontinuities between the processing blocks are not sufficiently masked by the window function used prior to recombination, causing clicks and pops. Jumps at the block level can additionally cause amplitude modulation effects. Hence, the second requirement is stabilizing the tracking algorithm sufficiently to prevent audible artifacts caused by rapid jumps in the synthesized beam direction.

Algorithmic processing becomes increasingly unstable for large, randomly distributed microphone arrays with large room dimensions and microphone-to-source distances in the order of 10^{1} m to 10^{2} m. Due to this instability, the output signal fidelity does not meet the requirements of pro-audio applications. For this reason, an approach using DL-based signal processing was chosen. The main constraints of full bandwidth and real-time processing present more significant challenges for such models. Many existing models and datasets for signal separation use sampling rates of 16 kHz, setting the upper bound of the frequency range for the audio material at 8 kHz. This frequency range is sufficient for telecommunication and speech-to-text applications but not for professional audio uses. Curating datasets of the desired fidelity can be challenging and time consuming. The higher sampling rate creates an additional strain on computation as the time complexity scales with $\mathcal{O}(L \log L)$ with respect to the number of samples L per input buffer [7, 41]. Complexity-optimized model architectures and advanced optimization techniques, such as model quantization, are required to provide real-time capabilities. Additionally, generative models are far more likely to introduce new, unwanted content into the audio stream, producing disturbing artifacts. For these reasons, a physics-informed Neural Beamformer approach was investigated, utilizing both the content and the spatial relation of multiple audio signals. This approach requires precise attention to the desired range of applications and the resulting requirements set for the training dataset. A high variance in spatial and spectral information is required for the model to generalize well over a large range of signals and noises.

Chapter 5

Results and Discussion

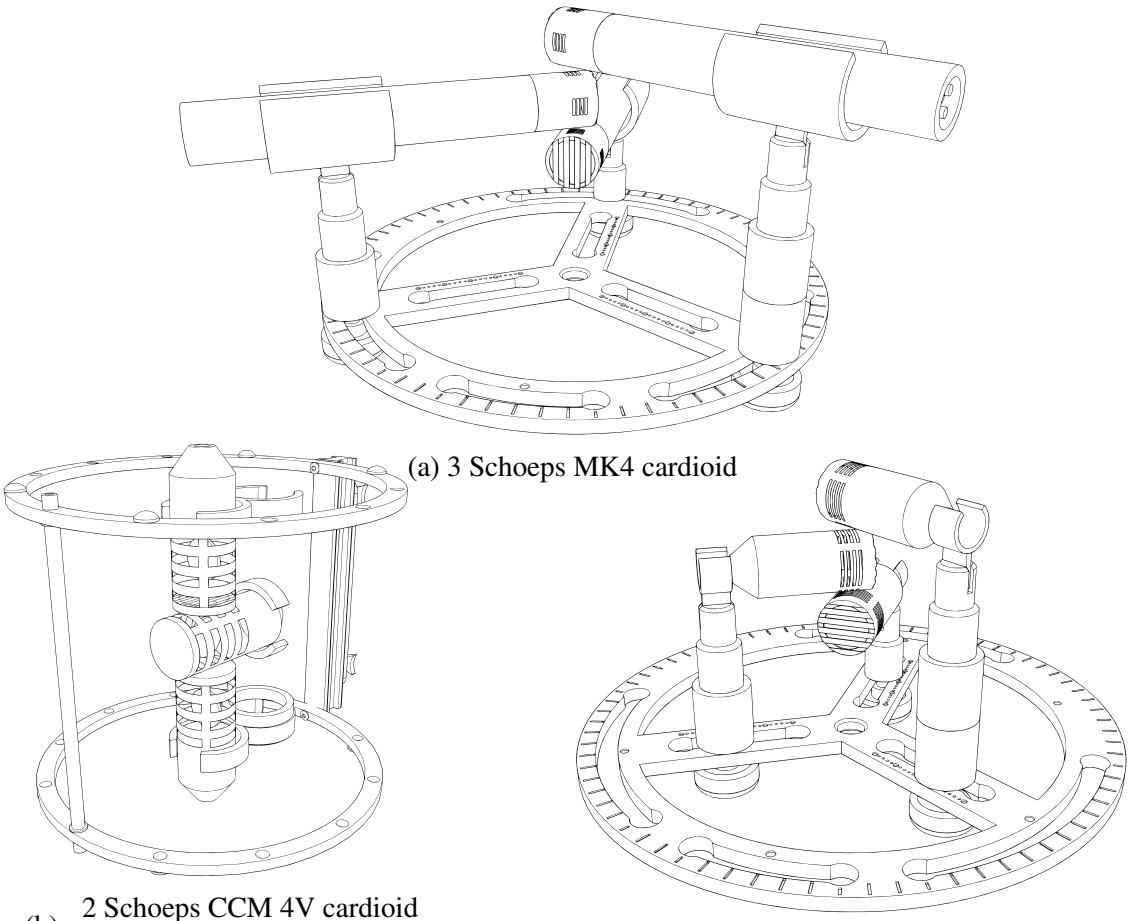
5.1 Algorithmic Beamforming using Coincident Microphone Arrays

Within the scope of this dissertation, a coincident, 3-capsule differential microphone array was used to develop an accurate and stable tracking algorithm. Using the estimated DOA, first-order beams can be directly synthesized using gradient synthesis, as described in section 3.1.1. More elaborate adaptive filtering algorithms, as proposed by Runow *et al.*, can be combined with the presented algorithm to create adaptive beamformers with higher Directivity Indices (DIs), capable of tracking a tight beam across 360° [32, 34]. This chapter aims to cover the entire research project, while providing an aggregation of results presented in the corresponding publications, and, in some cases, more detailed and current results. Section 5.1.1 details the design process of the microphone arrays chosen for the application, Section 5.1.2 focuses on the detailed algorithm design, and Section 5.1.3 presents an analysis of the ASL and beamforming performance, including novel and improved results that differ from the corresponding publication. In Section 5.1.4, an analysis of the effect of overlapping windows is presented.

5.1.1 Choice of Array Configuration

One consideration during the initial development process was the array configuration. As this microphone array is intended for teleconferencing and live speech amplification, a one-dimensional tracker, as described in Section 3.1.1, is sufficient. Equation (3.5) describes a method of gradient synthesis using two figure-of-eight and one omnidirectional microphone. There are multiple alternative configurations that can be transformed onto this base of SH, also known as the planar Ambisonics B-Format [12, 13]. Multiple configurations were examined (publications referenced in 2.4 [52] and 2.5 [51]), as shown in Figure 5.1. Configuration a consists of three cardioid capsules equally spaced at 120°, configuration c uses the same microphone distribution with three supercardioid capsules. Configuration b, also known as the Schoeps Double-M/S (DMS), uses two opposing

cardioid capsules and a figure-of-eight capsule rotated at 90° [45]. Additionally, an Ambisonics A-Format tetrahedral configuration, consisting of four cardioid capsules inside a 3D-printed mount was tested [20]. In order to maintain accuracy and reproducibility during the testing of microphone setups, a custom mounting ring as seen in Figure 5.1 a and c was designed and fabricated. A commercially available shockmount was chosen for the DMS setup. Test recordings of actual acoustic scenes, as well as high-resolution anechoic recordings of the capsules' Impulse Responses (IRs), were performed.



(a) 3 Schoeps MK4 cardioid

(b) 2 Schoeps CCM 4V cardioid
1 Schoeps CCM 8 figure-of-eight

(c) 3 Schoeps CCM 41 supercardioid

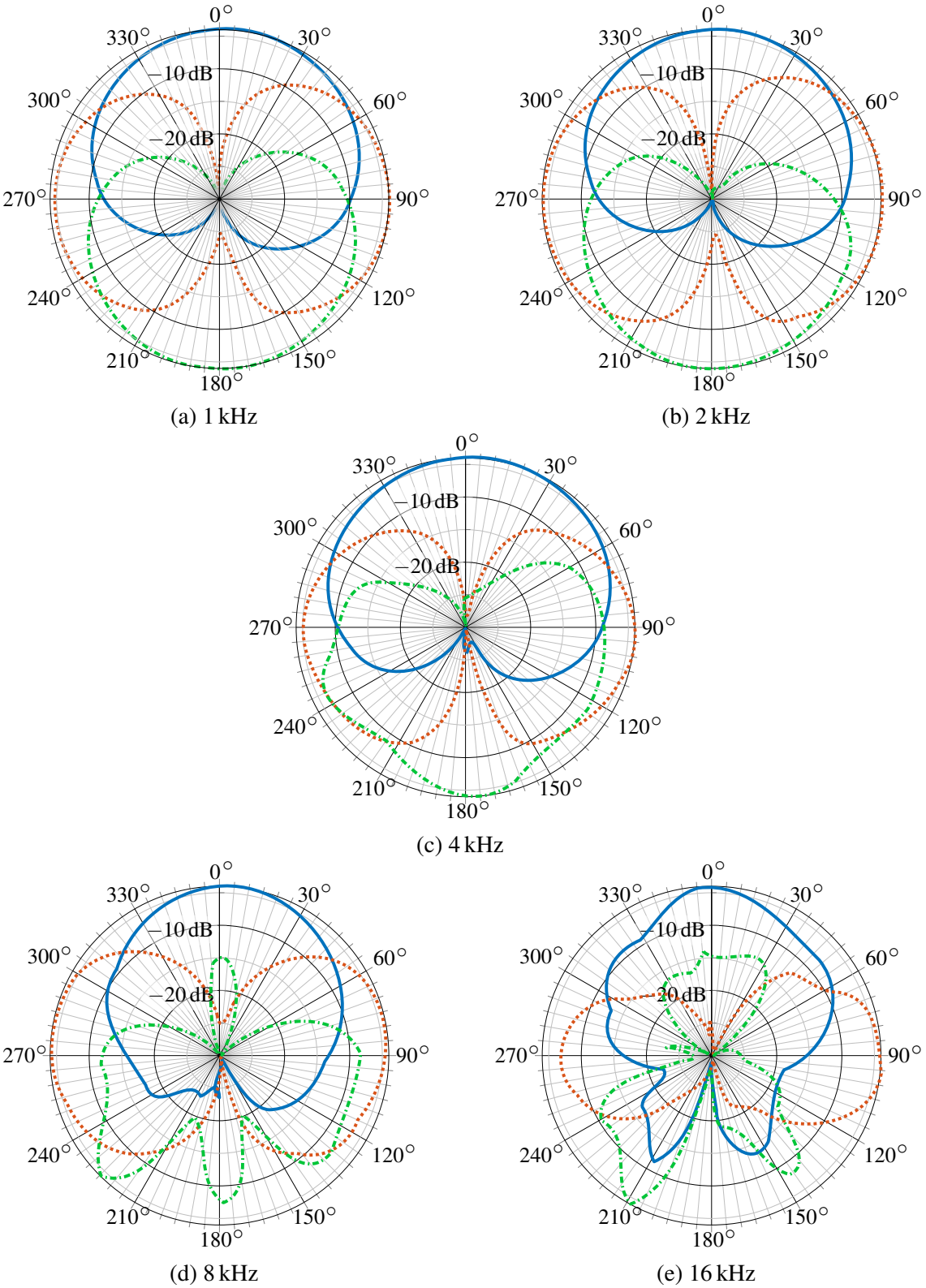
Figure 5.1: Test setup for different capsule configurations.

In order to accurately monitor the frequency-dependent directivity of the tested capsules, the application detailed in publication 2.9 [49] was developed. This program is used to process the recorded IRs dynamically in order to gauge individual capsule responses and full array performance. Using the interface as shown in Figure 5.3, array configurations can be inspected with respect to the individual capsule's frequency-dependent directivity and the quality of the resulting synthesized beams. Using three identical microphones as

in the configurations shown in Figures 5.1 a and c offers the advantage of identical frequency responses of the individual capsules, eliminating the need for individual capsule correction filters. If the individual capsules differ in their frequency response, noticeable coloration of the output signal while tracking moving targets could occur. On the other hand, the larger custom mounting hardware created significant distortion of the polar patterns at frequencies as low as 6 kHz over the entire range of possible DOA. Considering the well documented and precisely measured frequency responses of the individual capsules and the exceptionally high linearity and levels of constant directivity of the chosen microphones, the ease of use with the DMS setup was favored during algorithm development. Additionally, the asymmetric mounting hardware of the Double-M/S array provides relatively low levels of distortion over a wide range of DOA. Figure 5.2 shows the measured polar sensitivity plots of the three microphone capsules, mounted in the DMS configuration. As the front-facing cardioid and the figure-of-eight capsules reject sound coming from the direction of the mount, their sensitivity remains relatively stable over a much larger frequency range than the rear-facing cardioid. Although the development process was performed using the DMS array, the developed algorithm can be transferred to any other first-order Ambisonics (FOA)-compatible configuration with a linear transformation. The dependency between the arrays and the linear transformation are detailed in Appendix A. For a production prototype with custom hardware, a configuration consisting of three cardioid capsules in a shared housing is most suitable. The MK4 capsule can be produced with remarkably low tolerances and provides excellent constant directivity. A common housing could be tuned to prevent unwanted diffraction and reflections, making the resulting beam more stable over a wider range of frequencies and angles.

5.1.2 Algorithm Design

Building on the principle of Energy-Based Source Localization (EBSL) described in Section 3.1.2, a multi-stage algorithm was developed to create a more stable ASL algorithm, while maintaining a high level of reactivity. As detailed in publication 2.4 [52], the approach combines multiple processing blocks, each producing a confidence score used to score the directional information of the currently processed time frame. Equation (3.8) is used to determine the angle θ_{DOA} with the highest energy for the current audio buffer. As reverberation, noise, and interference can influence the estimated angle, smoothing must be applied, using Equation (3.10). When using a fixed smoothing factor α, the resulting output would either be too reactive (α too large) or too slow to perform jumps between different sources (α too small). By assigning α dynamically for each buffer based on the audio content, the level of directivity in the sound field and previously detected positions, stable and reactive tracking can be performed. The Confidence Indices (CIs) used to determine the smoothing factor are directivity-weighting, long-term weighting, level-weighting and, optionally, speech-weighting.



(a) 1 kHz

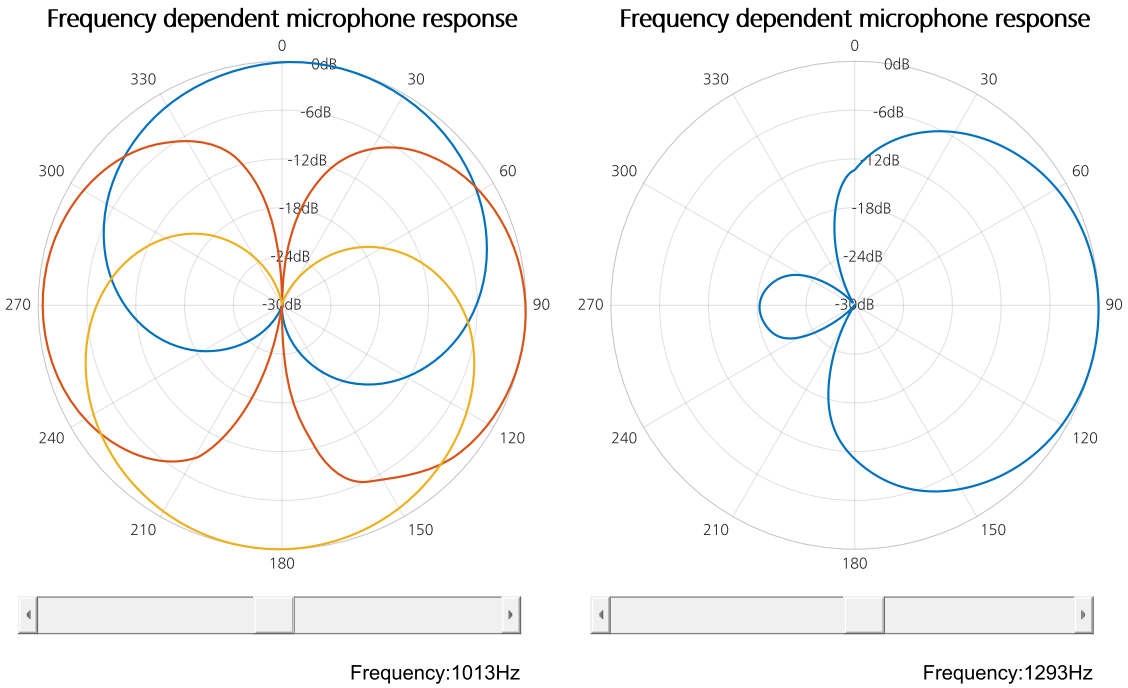
(b) 2 kHz

(c) 4 kHz

(d) 8 kHz

(e) 16 kHz

Figure 5.2: Angular sensitivity for Schoeps Double-M/S microphone array consisting of two CCM 4V cardioid microphones and one CCM 8 figure-of-eight microphone.



(a) All responses in DMS configuration

(b) Response of sythesized Supercardioid

Figure 5.3: Developed application for the evaluation of capsule configurations with respect to the polar response of individual capsules and synthesized beams.

Directivity-weighting provides an initial angle estimation θ_{fast} of relatively high precision, without requiring any information on previous audio buffers. This method compares the energy of the detected sound field with the ideal energy pickup of a single source.

In Figure 5.4, examples of buffers with different levels of directivity are shown. The corresponding CI is obtained through the mean distance between the detected, normalized energy distribution and the unidirectional level distribution U:

$$U_i = 0.5 + 0.5\cos\left(\theta_i - \theta_{DOA}\right) \tag{5.1}$$

$$C = \frac{1}{n_M}\sum_{i=1}^{n_M}\left(U_i - \overline{M}_i\left(W,X,Y,\theta_i\right)\right). \tag{5.2}$$

Scaling to the interval (0,1] is performed with

$$C_d = 10^{(vC)}, \tag{5.3}$$

with $\nu > 0$ representing a parameter controlling the reactivity of the tracker. For the results presented in Chapter 5.1.3, ν was set to 6.8. Figure 5.5 b shows the effect of directivity-weighting compared to the raw DOA data shown in Figure 5.5 a.

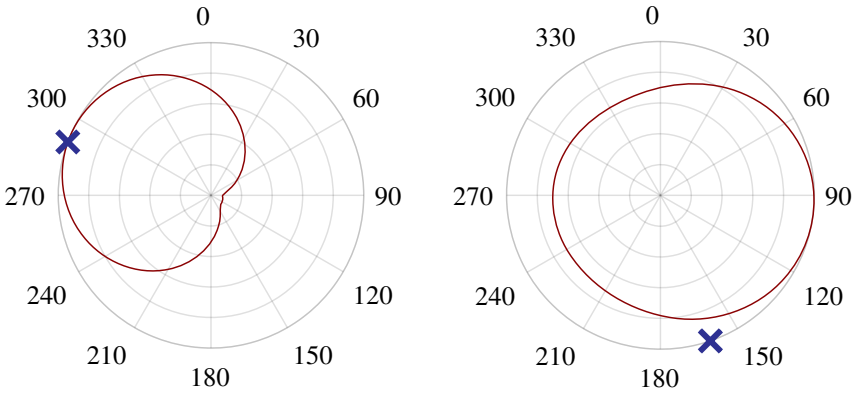


Figure 5.4: The energy of virtual cardioid signals, synthesized over 2π, represents the detected, one-dimensional sound field. The closer this distribution is to the optimal, unidirectional distribution U, the higher the confidence index C_d. The marking indicates the smoothed DOA output θ_{fast} for the displayed buffer. As C_d is large for the buffer on the left, $\theta_{fast} \approx \theta_{DOA}$. For the buffer on the right, a large portion of θ_{fast} is contributed by θ_{fast} of the previous buffer, and not by θ_{DOA}.

Level-weighting, associated with the Confidence Index C_l defines a hard threshold L_{min} for the minimum energy of an audio buffer needed in order to be considered for ASL. This prevents prolonged silence from deteriorating the processing performed by later weighting methods and stabilizes the tracker output during relative silence, reducing the number of artifacts occurring due to erratic tracker movement. C_l is computed as

$$C_l = \begin{cases} 1 & \text{for} \quad \overline{W} \geq L_{min} \\ 0 & \text{for} \quad \overline{W} < L_{min} \end{cases} . \tag{5.4}$$

Long-term weighting makes use of the semi-static nature of sound sources. When tracking speech, sources may move gradually, or the beam must jump between multiple sources. Both scenarios adhere to a fairly well-defined statistical distribution of positions. Motivated by this fact, the smoothed output directions θ_{fast} of previous buffers are stored under the condition that $C_l = 1$. An average position over multiple buffers is is quantized to 5° and stored as a point score. The total number of points is limited to 72, resulting in a uniform distribution over all bins as the initial state. When awarding a point to the most recent position, a point is deducted from the least recent angle. This method results in a form of long-term memory for the algorithm. With the parameters presented in [52], the system can adapt to a new static source in 1.5 s to 3 s and forgets

an audio event after 19.2 s. The associated confidence index C_{lt} corresponds directly to the relative score of the quantized angle and its effect can be seen in Figure 5.5 c.

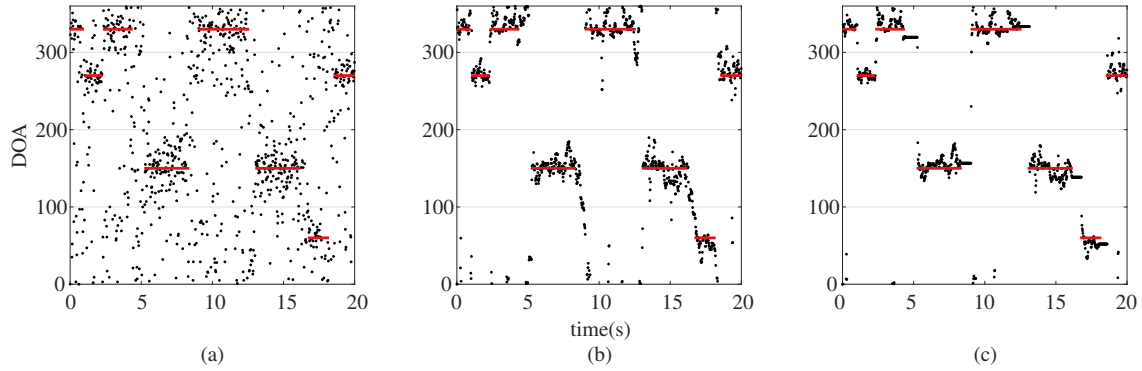


Figure 5.5: Performance analysis of confidence-weighting components. (a): Direct DOA estimate θ_{DOA}. (b): Smoothed DOA with added directivity-weighting θ_{fast}. (c): Additional level-dependent weighting reduces jumps during pauses. Long-term weighting further improves the accuracy and stability of the tracker output θ. Solid lines represent labeled reference positions. The data were down-sampled by a factor of 4 for increased clarity.

In [51], a small CNN was presented for additional confidence-weighting based on Voice Activity Detection (VAD). The network uses mel-scale spectrograms of 24 concatenated audio buffers as an input and functions as a soft classifier for the presence of speech in the 128 ms time frame. The output probability of the classifier directly corresponds to the respective Confidence Index C_s. Figure 5.6 shows the implemented model architecture.

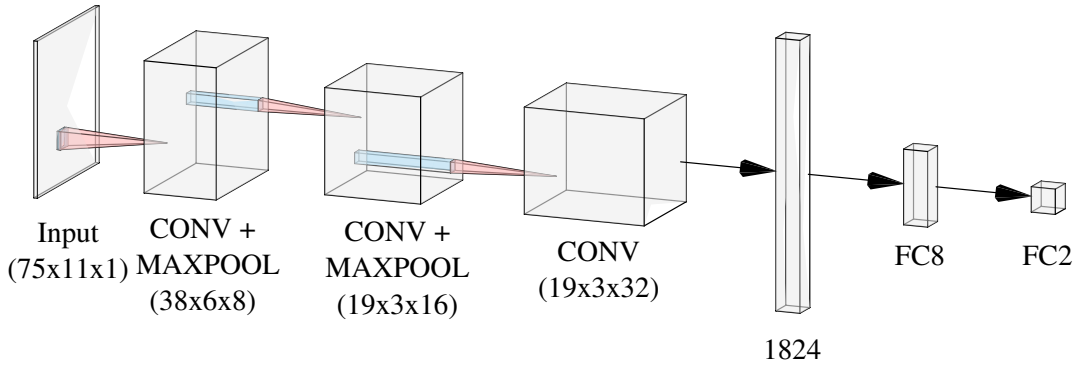


Figure 5.6: Model architecture used for speech weighting. Upon spectral analysis, a spectrogram of the last 24 audio buffers is passed to the CNN as a gray-scale image of 75x11 pixels. The final fully-connected layer with softmax activation differentiates between audio buffers that contain speech and no speech.

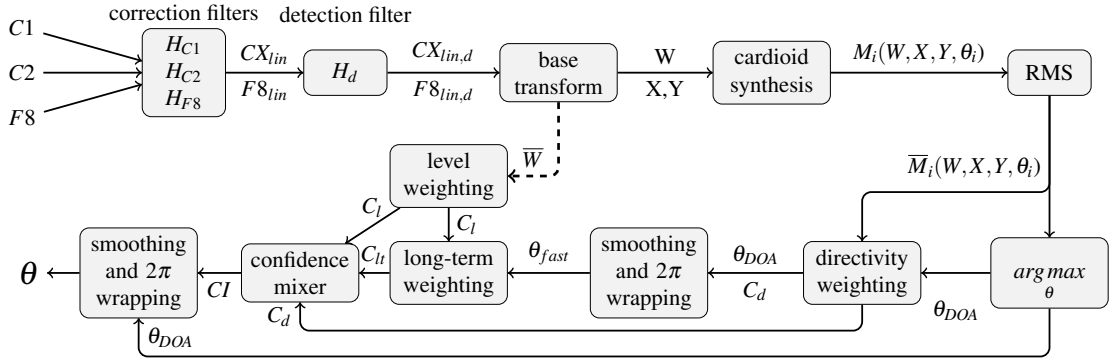


Figure 5.7: Signal flow through the tracking algorithm. After compensating for different frequency responses of the microphones and filtering the incoming signals, the synthesis of virtual cardioids over 2π and RMS-maximization of the signals results in initial DOA estimations. Various weighting algorithms in combination with a variable exponential smoothing process create a stable and reactive tracker.

Figure 5.7 shows the final configuration of the processing blocks as used in [52]. The input signals of the DMS microphone array are passed through individual capsule correction filters to further linearize the frequency response of the capsules. An additional detection filter is applied to attenuate signals not in the frequency range relevant for the target signal. In the case of speech, a fourth-order band-pass filter between 200 Hz and 2 kHz produced the best results. The filtered microphone signals, namely the two opposing cardioids (C1 and C2) and the figure-of-eight (F8), are then transformed according to the relations detailed in Appendix A into one omnidirectional (W) and two orthogonal figure-of-eight (X,Y) signals:

$$W = C1 + C2 \tag{5.5}$$
$$X = F8 \tag{5.6}$$
$$Y = C1 - C2. \tag{5.7}$$

From this new set of signals, 360 virtual cardioids $\mathcal{M} = \{M_i \ \forall i = 1,\ldots,360\}$ are synthesized over 2π, resulting in a 1° resolution over the entire horizontal plane. From these synthesized signals, the direction containing the highest signal energy, computed by means of RMS, is determined. This initial estimation θ_{DOA}, combined with the signal energy vector $\overline{\mathcal{M}}$, is used to compute C_d with Equations (5.2) and (5.3). Applying Eq. (3.10) with $\alpha = C_d$ produces the smoothed position estimation θ_{fast}. The level-dependent C_l is computed using the RMS of the virtual omnidirectional signal $\overline{W}$. This CI, combined with θ_{fast}, is passed to the long-term weighting module, which produces C_{lt}. All computed weighting scores are combined to the final CI using the mixing pa-

rameter κ and

$$CI = (\kappa C_d + (1-\kappa) C_d C_{lt})\, C_l\,. \tag{5.8}$$

This score is passed to Eq. (3.10) together with the initial θ_{doa} to create the final directional estimate θ.

Strict performance criteria must be met, as the developed algorithm is intended to work in live as well as in teleconferencing environments. Restrictions for live applications are tighter and more specific; thus, meeting the live standards fulfills the requirements for broadcasting and teleconferencing.

In order to operate in a live environment, extremely low processing latency is critical. The chosen buffer size of 256 samples at 48 kHz sampling rate introduces a lower bound of 5.3 ms on processing delay, which meets the defined specification. The algorithm itself performs very few cost-intensive calculations, so that the actual processing time needed falls significantly below the bound defined by the buffer size. Current non-optimized development scripts output the DOA and beamformed signal after approximately 412 µs on a modern laptop computer.

All intended applications require highest quality audio. In addition to a full frequency response spanning the entire auditory spectrum, processing artifacts must be strictly prevented, while suppressing interference and reverberation to the highest possible degree. In order to quantify the results, the performance of the ASL algorithm was monitored separately from the Signal-to-Distortion Ratio Improvement (SDRi) of a first-order beam compared to a virtual omnidirectional microphone that was synthesized from the array signals.

5.1.3 Tracking Accuracy and Signal Separation

As detailed in publication 2.4 [52] and [15], both synthetic and real data were used for evaluation of the developed algorithm. Synthetic data were generated using [8]. Virtual rooms were uniformly sampled from sizes spanning 3 m to 8 m with heights between 2.5 m and 4 m. Array, source and noise positions were randomly sampled for each room. Acoustic scenes were synthesized with RT_{60} reverberation times that spanned from 50 ms to 1500 ms in three categories:

- anechoic: $RT_{60} \leq 50$ ms
- mild reverb: RT_{60} = 400 ms to 600 ms
- strong reverb: RT_{60} = 600 ms to 1500 ms

For each category, a speech-only signal and a mixture of speech and noise were generated. Each synthetic recording consists of fifteen 4 s samples with changing array and

source positions. The audio material for the synthesis was randomly chosen from the VCTK and the ESC50 corpora [30, 42].

Speech-weighting provides additional stability at the cost of reduced accuracy, especially in noisy environments. As shown in Table 5.1, this addition provided only marginal improvements and significantly increased the processing complexity. For this reason, all results detailed in this section were generated without speech-weighting.

	$S1$	$S1_{noise}$	$S2$	$S2_{noise}$
accuracy gain	−1.2 %	−1.8 %	−0.4 %	−6.6 %
stability gain	1.2 %	5.4 %	2.4 %	4.2 %

Table 5.1: Performance gain using speech-weighting as part of the stabilization process, based on two recorded scenarios.

Compared to the published results referenced in [52], minor changes of the mixing values of the individual CI improved the tracker accuracy by up to 29.7 % at the cost of a slight reduction in the reactivity. Table 5.2 also shows the SDRi of the beamformer, using the tracker output, compared with the upper-bound based on the known oracle source positions. Although the SDRi are not as large as the results presented for the system described in Section 5.2, it is worth noting that the average difference between using the developed tracking algorithm and a first-order oracle beamformer is only 0.78 dB. As a main goal of this algorithm is minimal processing artifacts, using first-order gradient synthesis is the best choice.

	RT_{60}	SNR	DRR	θ_{err}	$\Delta\theta_{err}$	SDRi	SDRi oracle
clean	<0.05 s		10.98 dB	2.80°	0.59°	−0.18 dB	−0.11 dB
noisy	<0.05 s	6.06 dB		28.49°	1.68°	0.51 dB	1.80 dB
clean	0.4 s to 0.6 s		−7.20 dB	21.53°	0.79°	3.57 dB	3.98 dB
noisy	0.4 s to 0.6 s	5.96 dB		26.57°	0.67°	3.52 dB	4.02 dB
clean	0.6 s to 1.5 s		−9.37 dB	35.86°	0.69°	3.15 dB	4.27 dB
noisy	0.6 s to 1.5 s	6.05 dB		37.76°	0.59°	3.16 dB	4.46 dB

Table 5.2: ASL and beamforming performance analysis on synthetic data. The Direct-to-Reverberant Ratio (DRR) is computed using the virtual omnidirectional signal and the target speech in the clean scenarios. For noisy scenes, the mixture SNR is given, comparing the target and noise components in the virtual omnidirectional microphone signal.

The accurate tracking achieved by the ASL algorithm makes more elaborate beamforming and adaptive filtering methods possible as well. Runow *et al.* present an adaptive-

filtering method based on a DMS microphone array [34]. Impressive signal separation with high audio quality can be achieved by combining ASL with this adaptive beamformer.

	virtual omni	beamformers
Speech Intelligibility	0.23 ± 0.12	0.61 ± 0.16
Noise Suppression	0.16 ± 0.12	0.52 ± 0.19
Subjective Quality	0.19 ± 0.12	0.65 ± 0.22

Table 5.3: Compressed listening test results. In all categories the beamformed signal is preferred over the omnidirectional baseline. Relative scores between zero and one.

The published listening tests performed on 59 test subjects and referenced in 2.6 [29] show that both amateurs and test subjects active in audio engineering prefer the results of the beamformed signal over the virtual omnidirectional microphone signal. As seen in Table 5.3, noise suppression, speech intelligibility, and overall subjective quality all scored considerably higher. In Figure 5.8, the mean results of listening tests based on German and English dialog scenarios are visualized. The best results were obtained using the tracker in combination with an adaptive filtering algorithm developed by Runow *et al.* [34]. Multiple single-channel and multichannel adaptive filtering algorithms were compared. Additionally, the signal of a commercially available high-end spaced array with proprietary signal processing was included in the comparison [6].

5.1.4 Influence of Hop Size on Tracker and Beamformer Performance

Many components in the developed ASL algorithm require a buffer length $L \gg 1$. The energy of a sound buffer, computed via RMS, is essential for the SRP method. Even longer buffers are stored for the long-term smoothing [52] and the CNN-based speech detection [51]. With the hop size H and a pre-roll of $L - H$ samples, combined with higher processing frequencies, the algorithm is capable of running at significantly lower processing delays, down to 41.6 µs. Table 5.4 and the corresponding Figure 5.9 show the results of an investigation into different hop sizes for the algorithm, averaged over a total of twenty 4 s excerpts of *mild reverb* and *strong reverb* categories, as detailed in Table 5.2. Performance changes were investigated with hop sizes $H = 2^h$, with $h = 1, 2, \ldots, 8$. For all hop sizes < 256, a Hann-windowed overlap of $0.5H$ was implemented to prevent artifacts due to periodic discontinuities [39]. Additionally to the absolute error (abs error), the angular correction error (Δ error) and the SDRi for algorithm output and oracle positions, the individual standard deviations μ are supplied.

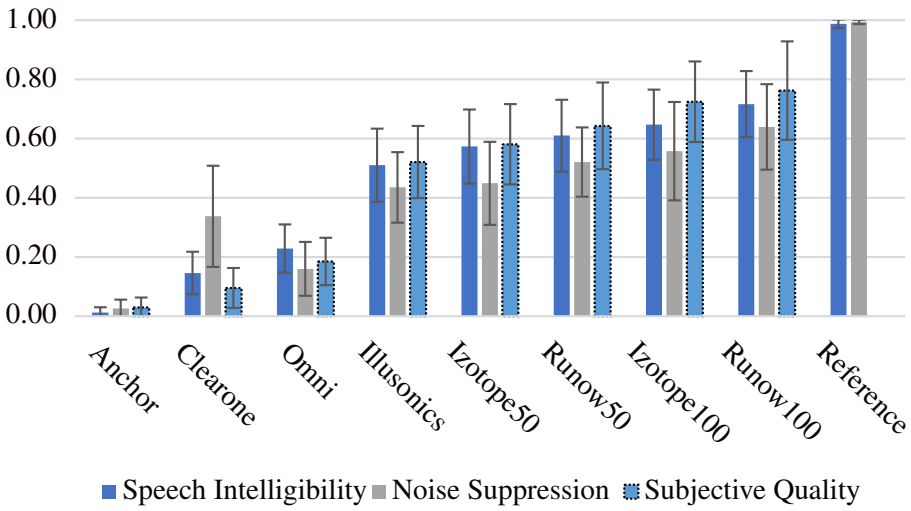


Figure 5.8: Results of MUSHRA listening test, comparing multiple signals generated using the tracked source position, together with the DMS array output and commercial or experimental adaptive filtering algorithms. Additionally, a commercial teleconferencing microphone array with proprietary signal processing was evaluated.

hop	abs error in deg	μ	Δ error in deg	μ	SDRi in dB	μ	SDRi oracle in dB	μ
256	57.78	5.62	**0.21**	$7.78 \cdot 10^{-3}$	2.90	0.47	4.47	0.67
128	53.80	6.59	0.28	$6.67 \cdot 10^{-3}$	3.11	0.42		
64	47.44	6.35	0.35	$1.31 \cdot 10^{-2}$	3.42	0.43		
32	40.86	6.51	0.43	$3.01 \cdot 10^{-2}$	3.67	0.39		
16	32.57	3.49	0.51	$5.03 \cdot 10^{-2}$	**3.95**	0.37		
8	31.52	7.18	0.63	$6.99 \cdot 10^{-2}$	3.80	0.26		
4	**30.43**	9.20	0.70	$9.26 \cdot 10^{-2}$	3.46	0.59		
2	39.45	15.52	0.66	$9.36 \cdot 10^{-2}$	3.09	0.82		

Table 5.4: Results of modifying the hop-size H for the DOA computation and beam synthesis.

While the long-term weighting can easily be adapted for more frequent processing steps, the exponential average filtering involved in the smoothing process is not directly proportional to the run count, thus resulting in less stable directional outputs for shorter hop sizes. This behavior can be clearly observed in the excerpt displayed in Figure 5.10.

Nevertheless, smaller hop sizes produce more consistent results of higher SDRi.

As the increase of processing frequency increases the computational cost significantly, the amount of overlap is a question of quality requirements and processing capabilities and can be varied depending on the desired application. For less complex scenarios with less noise and reverberation, larger hop sizes of 128 samples produce satisfactory results that provide SDRi reaching 91 % of the oracle beamformer.

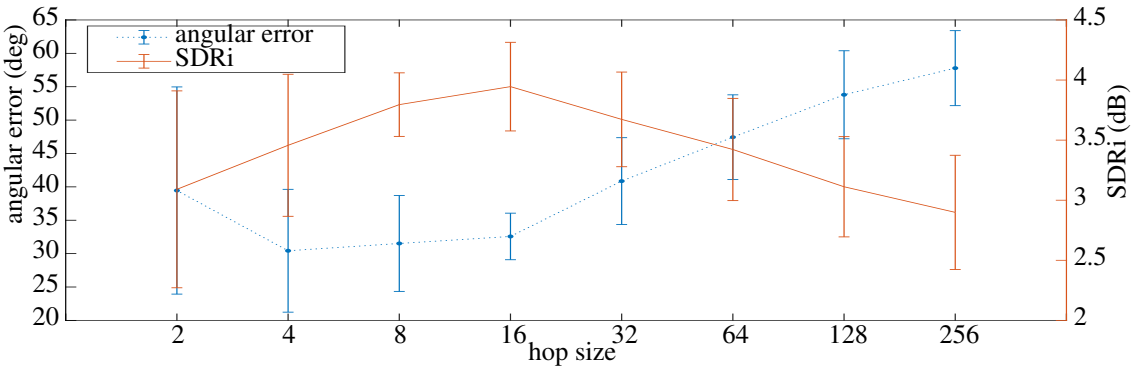


Figure 5.9: Analysis of tracking and signal-enhancement performance with respect to the chosen hop size. A hop size of 16 samples produces the best results.

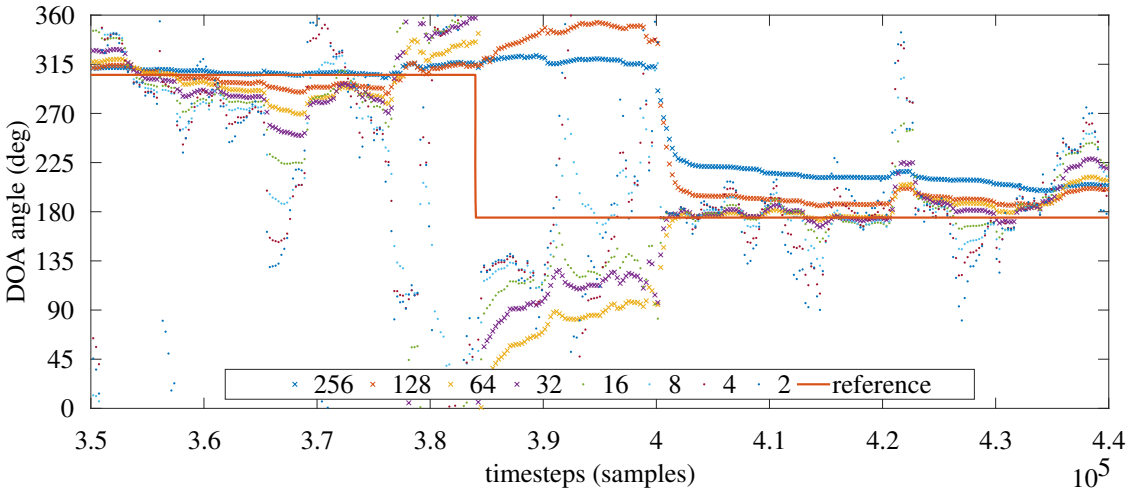


Figure 5.10: Computed DOA using the same algorithm operating with different hop sizes. Results computed at higher processing frequencies were downsampled for readability of the plot.

5.2 Neural Beamforming using Large-Aperture Microphone Arrays

This chapter covers the development of advanced beamforming and signal enhancement methods for ad hoc microphone arrays of unknown spatial distribution. Using Deep Learning models and knowledge of the physical properties of the system, a hybrid end-to-end model architecture is developed, capable of producing state-of-the-art signal separation results in challenging environments. Section 5.2.1 presents the details of the model architecture, Section 5.2.2 presents a detailed analysis of the signal separation capabilities of multiple model configurations and 5.2.3 details the multichannel aspects of the models.

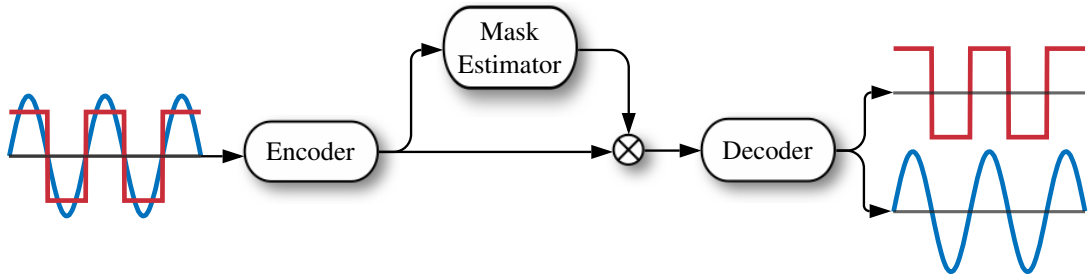
5.2.1 Model Design

The principal approach to single-channel DL-based signal separation most commonly used is that of mask estimation. As described in Section 3.3, a DNN architecture is used to estimate amplitude masks for the audio signal to attenuate noise components in the signal. In certain professional applications, such as live broadcasting and post-processing, a large number of audio channels are available, many containing valuable context information for signal enhancement. The approach pursued in this dissertation uses this context information to expand state-of-the-art methods with the capability of spatial beamforming and generating context vectors for the network to use during signal enhancement.

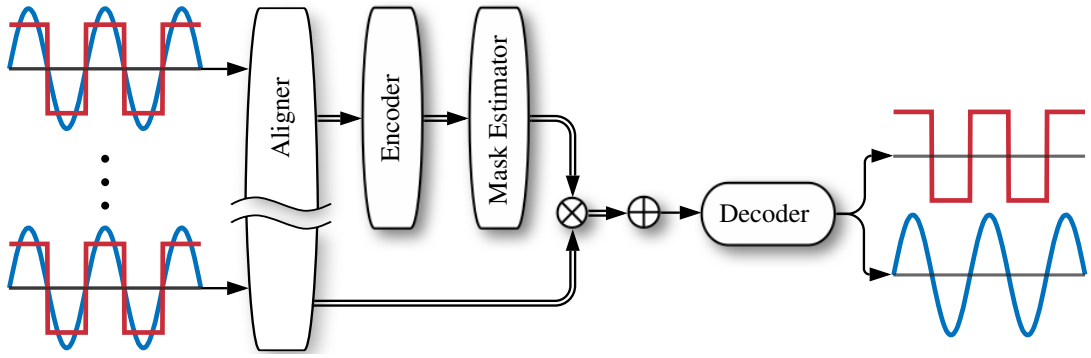
The end-to-end Neural Network can be decomposed into Aligner, Encoder, Mask Estimator, and Decoder, as shown in Figure 5.11. These components will be discussed in detail in the following paragraphs.

Aligner

A large part of the research efforts during this project were applied to the development of a network architecture capable of aligning multiple channels of audio with respect to a desired target source. This process consists of accurate ASL and the subsequent alignment of the individual channels with respect to a reference channel. Equations (3.18) and (3.20) can be used to determine the TDOA between two microphone signals. As desribed in Section 3.2.3, accurate and dependable ASL can be difficult in challenging acoustic environments. This network architecture introduces learnable preprocessing filters prior to the computation of the GCC-PHAT in order to accentuate the energy of the desired sound source and more consistently extract the correct time delays.



(a) Single-channel approach

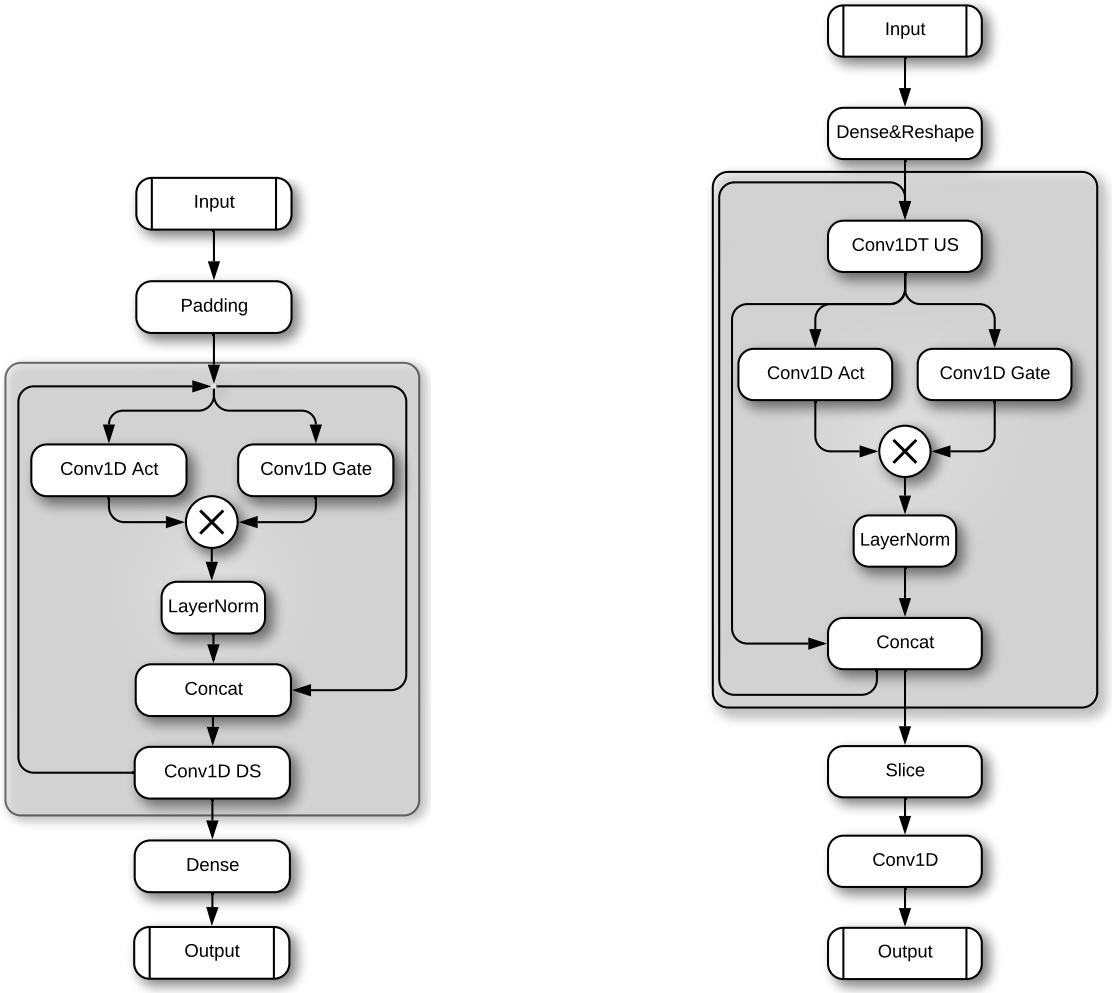


(b) Multichannel approach

Figure 5.11: Transition from single-channel signal enhancement to multichannel beamforming and mask estimation.

These filters use three main components to adaptively generate FIR filters and amplitude masks. First the audio signal is encoded into a latent representation using a configuration named EncodeWaveform, shown in Figure 5.12 a. This approach uses two one-dimensional convolution blocks called activation and gate. Analogous to the WaveNet processing described in Equation (3.21), the activation block is passed through a tanh nonlinearity and the gate block is passed through a sigmoid nonlinearity. The outputs are multiplied and normalized before being downsampled by a strided, linear one-dimensional convolution. Residual connections are in place to stabilize the gradient during training. This encoding process, as displayed in the gray block of Figure 5.12 a, is repeated multiple times, until the desired latent size is reached.

Second, this latent representation is used by the GenerateWaveform block, seen in Figure 5.12 b, to generate spatial and spectral filters, as well as amplitude masks. By prepending a transposed one-dimensional convolution prior to the activation and gate layers and removing the strided convolution, this block is capable of upsampling latent vectors to time-domain sequences. As with the EncodeWaveform, residual connections and multiple processing loops are employed to reach the desired signal length and model stability.



(a) EncodeWaveform block (b) GenerateWaveform block

Figure 5.12: Downsampling and upsampling systems, using two convolutional layers for nonlinear transformation and down-/upsampling.

Third, multiple GenerateWaveform architectures are combined to a FilterBlock, which accepts a latent vector describing both input channels and a signal. Within the FilterBlock, both FIR filtering and per-sample masking is performed. This block is shown in Figure 5.13.

Once the domain-specific spatial information is extracted via cross- and auto-correlation, additional EncodeWaveform blocks are used to generate a combined latent vector, that consists of spectral and spatial information. From this latent vector, an impulse response according to Equation (3.29) is estimated by means of dedicated GenerateWaveform blocks and applied via correlation. This process is encapsulated in additional FilterBlocks. The full Aligner can be seen in Figure 5.14.

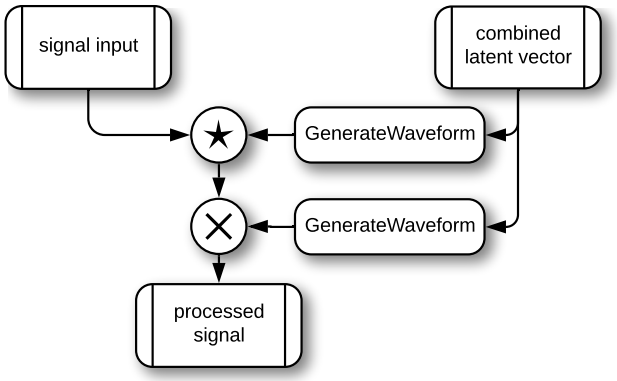


Figure 5.13: FilterBlock used to apply both FIR filtering and per-sample masking to a time-domain signal using a supplied latent vector for context information.

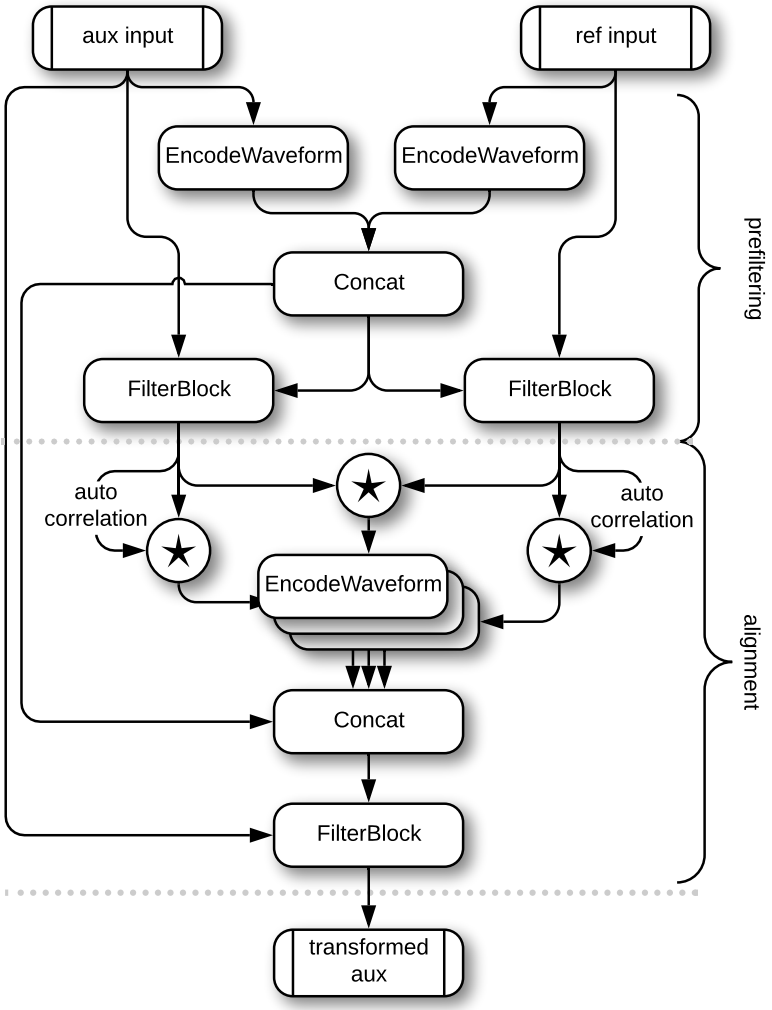
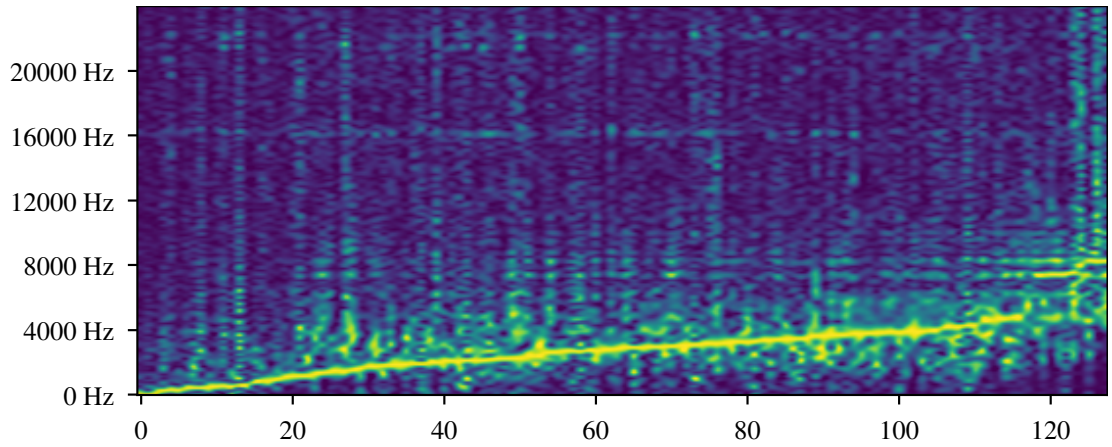
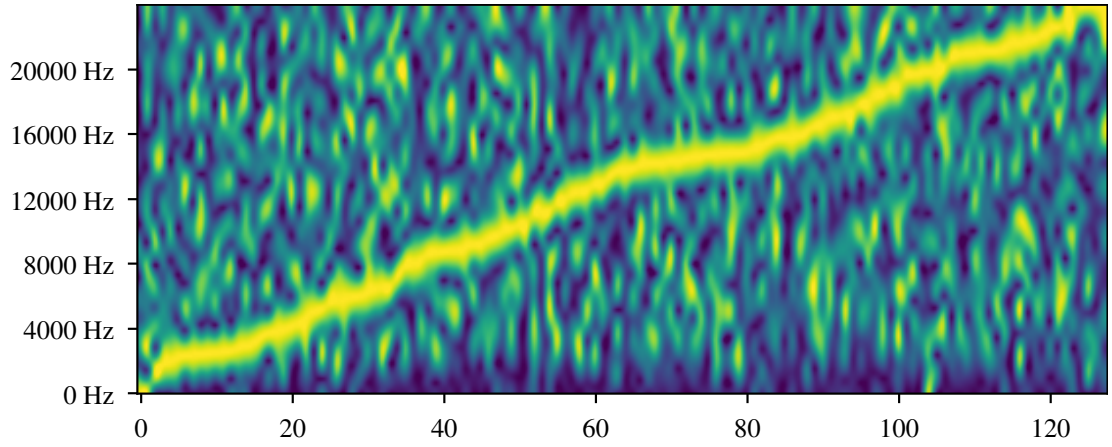


Figure 5.14: Aligner model architecture used to align individual input channels with respect to a desired target source.



(a) Subtractive model with a filter length of 127 samples, trained on speech and crowd noise.



(b) Additive model with a filter length of 20 samples, trained as a multipurpose speech enhancer.

Figure 5.15: Sorted Power Spectral Density distributions of trained Encoder layers. Z-axis represents relative magnitude (dark to light).

Encoder

The Encoder performs the operation of transforming time-domain audio onto a new, higher-dimensional base. The proposed architecture contains a linear encoder consisting of a single one-dimensional convolutional layer, transforming each input channel equally onto a base of 128 feature channels with the same sample resolution as the input signals. This can be understood as 128 channels of the same input audio, each processed with individual, learned FIR filters. Figure 5.15 shows a set of encoder filters extracted from a model trained for speech separation. Preliminary experiments using fixed DWT front-ends as described in Section 3.3.3 proved to be possible; however, the final signal separation quality did not reach the same level as when using the trainable Encoder.

Mask Estimator

Many high-performing network architectures have been developed to generate Time-Frequency or amplitude masks over the last few years. Initially, the architecture was planned to be implemented as described in [53]. This requires the final GenerateWaveform in the final FilterBlock to perform the entire mask estimation for the system, resulting in a strong imbalance of network capability. To perform acceptably, large latent sizes of 1024 samples were required, resulting in large, inefficient networks. For this reason, manuscript 2.2 [53] was not submitted for publication and the approach was expanded by splitting the alignment and the mask estimation, resulting in publication 2.1 [54].

This modification allows for more elaborate mask estimation architectures and much higher SDRi. The DPRNN, a currently popular architecture by Luo *et al.*, was chosen as the mask estimation architecture [22]. This DNN splits the input buffer into smaller chunks. These chunks are then individually processed by one Bi-LSTM layer along the sample axis and one Bi-LSTM along the chunk axis.

As the last component of the Mask Estimator, all processed channels are combined by means of summation.

Decoder

The single-channel Decoder is constructed to match the Encoder architecture and is composed from a single, linear 1D transposed convolution layer. This layer reconstructs the channel from the abstract feature-space and produces time-domain audio signals.

5.2.2 Signal Separation

The Neural Beamformer (NBF) can be used directly as an end-to-end signal separation network. In this case, the multiplicative GenerateWaveform (shown in Figure 5.13) of the last FilterBlock in the Aligner (shown in Figure 5.14) is responsible for generating the amplitude mask. This presents two challenges. First, the mask estimation takes a compressed representation, namely the combined latent vector, as an input. Second, the allocated complexity for the module does not differ from similar components in the aligner, even though the task requires a considerably higher level of modeling complexity. The advantages of this approach are a comparatively light-weight model and more subtle processing, resulting in potentially fewer artifacts. The performance of this model was analyzed in the manuscript detailed in 2.1 [54]. Training and evaluation data were synthesized from the same datasets as in Section 5.1.3, namely the VCTK and ESC50 corpora [30, 42]. Synthesis was performed using the tools developed by Scheibler *et al.* [36]. The model was trained using the Mean Square Error (MSE) loss $\mathcal{L}$ comparing the

target signal recorded by the reference channel and the model output ς:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} \left(\tilde{s}^0[t] - \varsigma[t] \right). \tag{5.9}$$

Optimization was performed using the Ranger algorithm [19, 47]. The performance of the system was compared with algorithmic beamformers using oracle and GCC-PHAT TDOAs. An evaluation of the SDRi of the proposed method using five channels under varying conditions is detailed in Table 5.5.

max mic dist (m)	SNR (dB)	GCC BF SDRi (dB)	oracle BF SDRi (dB)	NBF SDRi (dB)	DPRNN SDRi (dB)
24.9	-6	-0.85 ± 2.17	4.13 ± 0.82	6.26 ± 2.18	**11.14 ± 1.95**
	0	0.09 ± 1.78	4.15 ± 0.82	5.05 ± 1.65	**9.64 ± 1.38**
	6	0.09 ± 1.62	4.15 ± 0.8	1.72 ± 1.75	**6.45 ± 1.35**
55.8	-6	-0.95 ± 2.28	4.19 ± 0.86	6.53 ± 1.87	**11.24 ± 1.8**
	0	0.07 ± 1.83	4.16 ± 0.89	4.78 ± 1.93	**9.63 ± 1.25**
	6	-0.04 ± 1.43	4.03 ± 1.01	0.81 ± 2.23	**6.47 ± 1.18**
97.0	-6	-1.11 ± 1.82	3.68 ± 0.9	6.42 ± 1.82	**11.31 ± 1.67**
	0	-0.36 ± 1.4	3.55 ± 1.0	4.98 ± 1.56	**9.57 ± 1.18**
	6	-0.69 ± 1.06	3.09 ± 1.38	1.42 ± 1.89	**6.26 ± 1.17**
139.4	-6	-0.81 ± 1.68	3.05 ± 0.99	6.39 ± 1.82	**11.16 ± 1.61**
	0	-0.29 ± 1.25	2.79 ± 1.16	4.51 ± 1.88	**9.28 ± 1.22**
	6	-1.07 ± 1.12	2.0 ± 1.72	0.52 ± 2.25	**5.84 ± 1.25**

Table 5.5: Performance comparison of the Neural Beamformer with oracle and GCC-PHAT beamformers and a single-channel DPRNN.

The methods were compared on ten 40 s room simulations per configuration. Audio material, room dimensions, and both microphone and source positions were randomly sampled for each iteration. Each excerpt was converted into blocks of 4096 + 12288 samples with 50 % overlap and processed individually. The resulting model output was Hann-windowed and recombined for evaluation [39]. The advantages of the prefiltering prior to TDOA estimation are especially apparent in scenes with low or negative SNR. Additionally, the adaptive masking and filtering capabilities provide results superior to classic DS beamforming, even when using oracle positions. This is especially apparent

in situations with larger amounts of reverberation. Most examined scenes prove to be overly complex for GCC-PHAT-based beamforming, resulting in low to negative SDRi. Overall, the NBF cannot match the signal separation capabilities of a single-channel mask estimation network of similar complexity. A DPRNN trained on the same dataset produces considerably better results with less variance over the entire space of examined signal configurations. Although the more subtle processing of the NBF results in fewer artifacts, this advantage becomes negligible when mixing the input signal with the processed DPRNN signal. By mixing the input with the output, a variable level of signal separation can be achieved, thus gaining control over the artifacts versus SDRi trade off.

To make better use of the NBF's capability of producing alignment filters, an additional mask estimation network is incorporated into the end-to-end system. This prevents the GenerateWaveform blocks from producing aggressive amplitude masks and allocates sufficient model complexity in the DPRNN part of the network for this task. Training was performed using the Ranger optimizer and the MSE loss comparing the mask approximator output $\hat{y}$ with the target signal recorded by the reference channel.

A shortcoming of this approach remains the real-time capability for large microphone distances. Although all models process relatively short buffers with preceding context windows, geometrical restraints prevent optimal operation. In order for a channel to provide useful information for the summed result, the desired signal component must have arrived at the microphone before it arrived at the reference microphone. Only then can the signal be shifted to match the reference microphone. In reality, the microphone chosen as the reference generally provides the best starting point for a desired target signal and is often closest to the desired source. A simple solution to this challenge is to reverse the processing. The beamformer then aims to accurately model the noise in the reference channel and subtract the sum of processed channels from the input. By changing the processing chain in this very simple way, the auxiliary channels can contribute significantly more information to the resulting output. Table 5.6 shows a comparison of different algorithmic and DL based approaches. Next to oracle and GCC-based beamforming algorithms, a single-channel DPRNN, a DPRNN fed with the outputs of oracle and GCC-PHAT beamformes, and an end-to-end combination of NBF and DPRNN were examined. The NBF + DPRNN network outperforms the baseline for every configuration. The upper bound of the combination of Oracle beamforming and DPRNN masking architecture shows that further improved signal synchronization can provide a significant increase in overall model performance and thus validates the intuition of the proposed multichannel architecture.

An additional proprietary dataset of approximately 15 hours of real multitrack recordings of large-venue crowds was mixed with VCTK data and studio recordings of sports commentators. Sporting arenas provided high quality recordings of massively distributed microphone arrays and presented a fitting application for the developed model. In addition

	SNR	SDRi in dB			
	in dB	28 m	63 m	110 m	159 m
	-6	-3.11 ± 2.31	-3.03 ± 2.40	-2.79 ± 2.81	-2.28 ± 2.84
GCC	0	-0.95 ± 1.51	-1.38 ± 1.98	-1.94 ± 2.37	-2.04 ± 2.48
	6	-1.3 ± 1.10	-0.56 ± 1.62	-0.86 ± 2.12	-1.41 ± 1.98
	-6	6.56 ± 1.14	6.25 ± 0.91	6.1 ± 1.41	5.67 ± 1.61
Oracle	0	6.65 ± 1.13	6.34 ± 1.18	5.25 ± 1.87	4.15 ± 2.41
	6	6.57 ± 1.06	6.04 ± 1.77	4.75 ± 2.81	3.36 ± 3.56
GCC	-6	4.24 ± 3.26	4.19 ± 3.46	4.14 ± 3.92	4.69 ± 3.55
—	0	4.84 ± 1.85	4.15 ± 2.17	3.8 ± 2.26	3.96 ± 2.51
DPRNN	6	0.28 ± 1.82	1.06 ± 1.70	1.32 ± 1.94	1.66 ± 1.57
SC	-6	9.75 ± 1.00	9.72 ± 0.82	9.44 ± 0.80	9.48 ± 0.93
—	0	11.05 ± 1.88	11.08 ± 1.88	10.94 ± 1.86	10.86 ± 1.82
DPRNN	6	6.50 ± 1.32	6.46 ± 1.29	6.39 ± 1.26	6.13 ± 1.29
NBF	-6	10.19 ± 1.15	10.19 ± 1.13	9.96 ± 1.24	9.87 ± 1.18
—	0	11.58 ± 2.16	11.70 ± 2.00	11.59 ± 1.99	11.48 ± 2.12
DPRNN	6	7.12 ± 1.39	7.21 ± 1.27	7.14 ± 1.22	6.91 ± 1.35
Oracle	-6	14.66 ± 1.97	14.64 ± 1.95	14.27 ± 1.74	13.92 ± 2.07
—	0	14.78 ± 1.90	14.34 ± 1.57	13.42 ± 1.83	12.73 ± 2.07
DPRNN	6	8.77 ± 1.42	8.37 ± 1.56	7.75 ± 1.70	7.18 ± 1.72

Table 5.6: Performance comparison of beamforming algorithms and mask estimation networks with and without the Neural Beamforming front-end. The distance shown represents the maximum microphone distance of the respective room configuration.

to a dedicated evaluation portion of the used datasets, the subtractive model was tested on real recordings of sports commentators of unseen soccer games. As no ground truth is available for this portion of the data, only qualitative evaluation is possible. Table 5.7 shows the signal separation in dB SDRi for ten 30 s mixtures with random mixture levels and SNR at the reference receiver. The combination of NBF and DPRNN is compared to two single-channel DPRNN baselines, trained on the same dataset. DPRNN+CONV differs from DPRNN in the way that two convolutional layers of kernel size 5, stride

one, and activated with tanh nonlinearities are added to the encoded signal prior to mask estimation and to the mask prior to its application.

SNR in dB	NBF-DPRNN SDRi in dB	DPRNN SDRi in dB	DPRNN+CONV SDRi in dB
0.37	9.6	2.07	3.23
0.75	8.56	1.01	2.16
0.81	10.55	2.34	3.46
4.97	11.45	1.37	2.45
5.32	9.35	1.33	2.38
6.53	9.54	2.49	4.02
7.17	9.48	1.53	2.25
7.84	8.99	1.52	2.84
9.71	9.39	1.53	2.81
16.47	7.1	1.75	2.94
mean	9.4	1.69	2.85
std	1.08	0.45	0.56

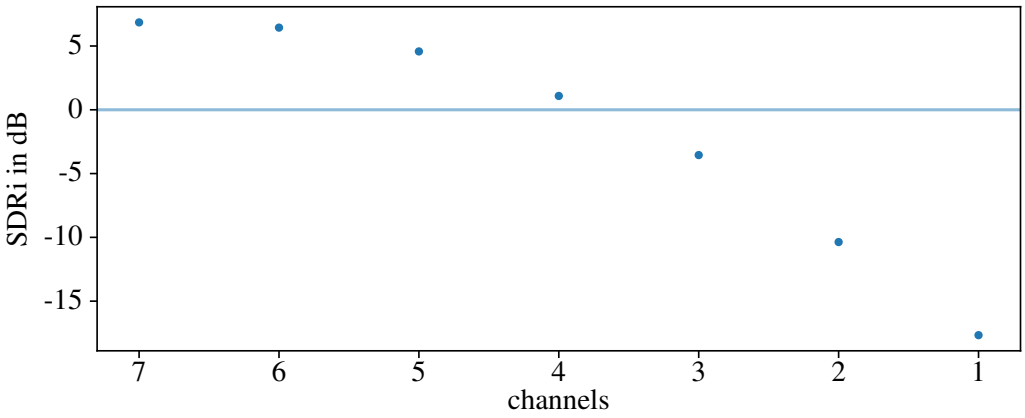
Table 5.7: Signal separation in dB SDRi for subtractive DPRNN architectures.

5.2.3 Analysis of Multichannel Processing

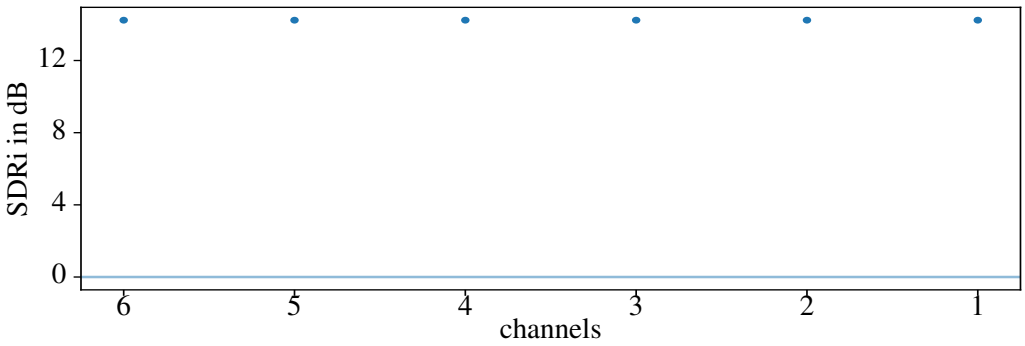
Further analysis of the model described in publication 2.1 provides insights into the approach applied by the converged model. Modifying a pretrained model is possible, as all weights are shared over the channel axis. When comparing the signal separation capabilities of the network over varying channel counts, some requirements for the source and microphone distributions are made clear. The causal (real-time capable) configuration described in the corresponding publications provides additional context from past samples only. This restricts the capability of the network to shift signals forward in time. In other words, signals picked up by transducers that are further away from the source than the reference channel can often not be included in the beamforming. When using an additive configuration analogous to FS beamforming for large-aperture arrays, the channel with the latest time of arrival should be chosen for the reference channel.

The subtractive approach provides exceptional results, especially for low and negative SNR and makes better use of the multichannel information available. The performance gain for mixtures with low SNR can be partly explained by the fact that the model is trained to estimate the noise. Smaller Signal-to-Noise Ratios inversely signify a higher ratio of noise to signal. Figures 5.16 a and 5.16 b show the performance of the described

architecture over a variety of input channels in two different configurations. The model outputting the results shown in Figure 5.16 b uses its entire processing capability to modify the reference channel. No additional performance gain can be observed with higher channel counts. Figure 5.16 a shows the channel-dependent performance of a model applied in a subtractive configuration. A clear degradation of performance can be observed when reducing the number of auxiliary channels. The absolute values of SDRi in the two cases cannot be compared, as the datasets and source configurations differ significantly.



(a) Subtractive model performance depending on number of auxiliary channels.



(b) Additive model performance depending on number of auxiliary channels.

Figure 5.16: Comparison of different implementations of the developed Neural Beamformer. As the reference channel was chosen to be closest to the source, additive beamforming is not capable of compensating the TDOA and the model reverts to single-channel processing. As the subtractive model utilizes the information in every channel, a drop in performance can be observed when reducing the available channel count. Absolute SDRi of the models do not compare, as different data and source positioning are used.

Chapter 6

Conclusion

Within the scope of this work, two approaches to multichannel signal enhancement were examined. Building on the principles of Differential Microphone Arrays and Delay-and-Sum Beamforming, two fundamentally different approaches were investigated.

Using three coincident capsules with first-order directivity and fixed relative orientation, an Acoustic Source Localization algorithm was designed, aimed at providing high quality audio in relatively simple acoustic environments. The automatic orientation of a first-order beam, combined with the possibility of advanced, adaptive beamforming, create a versatile tool for professional audio applications and high-end teleconferencing. Quantitative analysis was performed on simulated data and real recordings to verify the tracker accuracy and stability, as well as the audio quality of a first-order beam. In simple environments, the tracker operates at approximately 90 % accuracy, with correction errors of less than one degree. The signal enhancement of the resulting first-order beam is 0.45 dB below oracle results. A listening test with 59 test subjects displayed a clear preference of amateurs and professional audio engineers towards the beamformed signals. Specifically, a combination of the developed tracker with an adaptive filter developed for the Double-M/S array was rated highest in all examined categories, namely speech intelligibility, noise suppression, and subjective quality.

An ad hoc, spaced microphone array was used as the input for an end-to-end, time-domain neural network for signal enhancement. The developed model architecture combines the state-of-the-art signal separation capabilities of single-channel mask estimation networks with a novel Neural Beamformer architecture capable of extracting domain-specific spatial relations of microphones with distances of up to 110 m. The combined network outperforms algorithmic and DL baselines and provides audio fidelity fit for professional audio applications, such as live amplification and broadcasting. For beamforming on synthesized general-purpose speech extraction scenes, the combination of Neural Beamformer and Dual-Path Recurrent Neural Network outperforms the single-channel counterpart by 0.60 dB; for beamforming on semi-synthetic data of large scale sporting events, the combined beamformer outperformed the single-channel equivalent by 6.55 dB.

Bibliography

[1] Allen, J. (1977). Short term spectral analysis, synthesis, and modification by discrete Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **25**(3), 235–238.

[2] Benesty, J. and Chen, J. (2013). *Study and Design of Differential Microphone Arrays*, volume 6 of *Springer Topics in Signal Processing*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[3] Benesty, J., Jingdong Chen, Yiteng Huang, and Dmochowski, J. (2007). On Microphone-Array Beamforming From a MIMO Acoustic Signal Processing Perspective. *IEEE Transactions on Audio, Speech, and Language Processing*, **15**(3), 1053–1065.

[4] Benesty, J., Jingdong, C., and Huang, Y. (2008). *Microphone Array Signal Processing*. Springer Berlin Heidelberg.

[5] Bömers, F. (2000). *Wavelets in real time digital audio processing: Analysis and sample implementations*. Ph.D. thesis, University of Mannheim, Mannheim.

[6] ClearOne (2021). Beamforming Microphone Array 2. Technical report.

[7] Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, **19**(90), 297–301. Publisher: JSTOR.

[8] Diaz-Guerra, D., Miguel, A., and Beltran, J. R. (2020). gpuRIR: A python library for room impulse response simulation with GPU acceleration. *Multimedia Tools and Applications*.

[9] Dmochowski, J. P., Benesty, J., and Affes, S. (2007). A Generalized Steered Response Power Method for Computationally Viable Source Localization. *IEEE Transactions on Audio, Speech, and Language Processing*, **15**(8), 2510–2526.

[10] Eargle, J. (2004). *The Microphone Book - From Mono to Stereo to Surround, A Guide to Microphone Design and Application*. Focal Press, Oxford. OCLC: 851974436.

[11] Frost, O. L. (1972). An algorithm for linearly constrained adaptive array processing. *Proceedings of the IEEE*, **60**(8), 926–935. Publisher: IEEE.

[12] Gerzon, M. A. (1973). Periphony: With-height sound reproduction. *J. Audio Eng. Soc*, **21**(1), 2–10.

[13] Gerzon, M. A. (1975). The Design of Precisely Coincident Microphone Arrays for Stereo and Surround Sound. In *Audio Engineering Society Convention 50*.

[14] Goehring, T., Chapman, J. L., Bleeck, S., and Monaghan, J. J. M. (2018). Tolerable delay for speech production and perception: effects of hearing ability and experience with hearing aids. *International Journal of Audiology*, **57**(1), 61–68.

[15] Hirt, R. (2017). *Development of a virtual conference with focus on optimal reproduction of pre recorded speech.* Bachelor's Thesis, Stuttgart Media University.

[16] ISO3745 (2012). *Acoustics – Determination of sound power levels and sound energy levels of noise sources using sound pressure – Precision methods for anechoic rooms and hemi-anechoic rooms*. Number 3745 in Request for Comments. RFC Editor.

[17] Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial Transformer Networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2017–2025. Curran Associates, Inc.

[18] Kornycky, J., Gunel, B., and Kondoz, A. (2008). Comparison of Subjective and Objective Evaluation Methods for Audio Source Separation. page 050001, Paris, France.

[19] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. (2020). On the variance of the adaptive learning rate and beyond. In *8th international conference on learning representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[20] Lopez-Lezcano, F. (2016). The* SpHEAR project, a family of parametric 3D printed soundfield microphone arrays. In *Audio engineering society conference on soundfield control*.

[21] Luo, Y. and Mesgarani, N. (2019). Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, **27**(8), 1256–1266.

[22] Luo, Y., Chen, Z., and Yoshioka, T. (2020). Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation. In *ICASSP 2020-2020*

IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 46–50.

[23] McCowan, I. (2001). Microphone arrays: A tutorial. Publisher: Citeseer.

[24] Merimaa, J. and Pulkki, V. (2005). Spatial impulse response rendering I: Analysis and synthesis. *J. Audio Eng. Soc*, **53**(12), 1115–1127.

[25] Merziger, G. and Wirth, T. (1995). *Repetitorium der höheren mathematik*. Feldmann, Hannover, fifth edition.

[26] Oord, A. v. d., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., and Kavukcuoglu, K. (2016a). Conditional Image Generation with PixelCNN Decoders. *CoRR*. eprint: 1606.05328.

[27] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016b). WaveNet: a generative model for raw audio. In *arXiv:1609.03499 [cs]*.

[28] Paukert, H. and Ziegler, J. (2016). Listening Tests in the Process of Microphone Development. In *Proceedings of the 29. Tonmeistertagung VdT International Convention*, Cologne, Germany.

[29] Paukert, H., Ziegler, J., and Koch, A. (2018). Hörversuche zur Entwicklung eines neuartigen Mehrkapsel-Mikrofons. In *30th Tonmeistertagung VdT International Convention*, Cologne, Germany.

[30] Piczak, K. J. (2015). ESC: dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pages 1015–1018, New York, NY, USA. ACM.

[31] Ravanelli, M. and Bengio, Y. (2018). Speaker recognition from raw waveform with sincnet. In *2018 IEEE spoken language technology workshop (SLT)*, pages 1021–1028.

[32] Runow, B. (2016). Störgeräuschreduktion mit einer Mel-Filterbank in Verbindung mit koinzidenten Mikrofonarrays. In *29th Tonmeistertagung - VDT International Convention*.

[33] Runow, B. and Curdt, O. (2014). Microphone Arrays for professional audio production. In *28th Tonmeistertagung*. VdT.

[34] Runow, B., Curdt, O., and Schilling, A. (2016). Shotgun Microphones versus Microphone Arrays. In *29th Tonmeistertagung - VDT International Convention*.

[35] Runow, B., Ziegler, J. D., Paukert, H., Schilling, A., and Curdt, O. (2018). The Fundamental Problem of the Spectral Subtraction. In *30th Tonmeistertagung - VDT International Convention*, Cologne, Germany.

[36] Scheibler, R., Bezzam, E., and Dokmanić, I. (2018). Pyroomacoustics: A Python package for audio room simulations and array processing algorithms.

[37] Sejdić, E., Djurović, I., and Jiang, J. (2009). Time–frequency feature representation using energy concentration: An overview of recent advances. *Digital Signal Processing*, **19**(1), 153–183.

[38] Simbürger, C. (2020). *Anwendungen für KI in Digitalmischpulten bei Fußball-Bundesliga-Übertragungen und die damit verbundene Automatisierung von Arbeitsprozessen.* Bachelor's Thesis, Stuttgart Media University.

[39] Smith III, J. O. (2011). *Spectral audio signal processing*. W3K publishing.

[40] Van Veen, B. D. and Buckley, K. M. (1988). Beamforming: A versatile approach to spatial filtering. *IEEE assp magazine*, **5**(2), 4–24. Publisher: IEEE.

[41] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, , and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

[42] Veaux, C., Yamagishi, J., and MacDonald, K. (2017). CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit. *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*.

[43] Venkataramani, S., Casebeer, J., and Smaragdis, P. (2018). End-to-end source separation with adaptive front-ends. In *2018 52nd asilomar conference on signals, systems, and computers*, pages 684–688.

[44] Whittaker, E. T. (1915). XVIII.—On the Functions which are represented by the Expansions of the Interpolation-Theory. *Proceedings of the Royal Society of Edinburgh*, **35**, 181–194. Publisher: Royal Society of Edinburgh Scotland Foundation.

[45] Wittek, H. (2014). Schoeps Double M/S - Surround Sound Manual. Technical report, Schoeps GmbH.

[46] Yu, C., Xu, Y., Liu, B., and Liu, Y. (2014). "Can you SEE me now?" A measurement study of mobile video calls. In *IEEE INFOCOM 2014-IEEE conference on computer communications*, pages 1456–1464. tex.ids: yu_can_nodate tex.organization: IEEE.

[47] Zhang, M., Lucas, J., Ba, J., and Hinton, G. E. (2019). Lookahead optimizer: k steps forward, 1 step back. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9597–9608. Curran Associates, Inc.

[48] Ziegler, J. D. and Schröder, L. (2020). Extraktion eines Audioobjektes. *Deutsches Patent- und Markenamt (DPMA)*.

[49] Ziegler, J. D., Paukert, H., and Runow, B. (2017a). Interactive Display of Microphone Polarity Patterns with non-fixed Frequency Point. In *Proceedings of the 142nd Audio Engineering Society International Convention*, Berlin, Germany.

[50] Ziegler, J. D., Rau, M., Schilling, A., and Koch, A. (2017b). Interpolation and Display of Microphone Directivity Measurements using higher-order Spherical Harmonics. In *Proceedings of the 143rd Audio Engineering Society International Convention*, New York City, USA.

[51] Ziegler, J. D., Koch, A., and Schilling, A. (2018). Speech Classification for Acoustic Source Localization and Tracking Applications using Convolutional Neural Networks. In *Proceedings of the 145th Audio Engineering Society International Convention*, New York City, USA.

[52] Ziegler, J. D., Paukert, H., Koch, A., and Schilling, A. (2020a). Acoustic Source Localization and High Quality Beamforming Using Coincident Microphone Arrays. In *Proceedings of the 148th AES International Convention*, Vienna, Austria.

[53] Ziegler, J. D., Schröder, L., Koch, A., and Schilling, A. (2020b). Spatially Informed Neural Beamforming with Distributed Microphone Arrays.

[54] Ziegler, J. D., Schröder, L., Koch, A., and Schilling, A. (2021). A Neural Beamforming Frontend for Distributed Microphone Arrays. In *Proceedings of the 151st Audio Engineering Society International Convention*, Online.

A Proof of Spherical Harmonic Base Equivalence

Base functions are:

$$W(\theta) = 1 \tag{1}$$
$$X(\theta) = \sin(\theta) \tag{2}$$
$$Y(\theta) = \cos(\theta). \tag{3}$$

A DMS configuration consisting of two opposing cardioids and one orthogonal figure-of-eight:

$$C1 = 0.5 + 0.5\cos(\theta) \tag{4}$$
$$C2 = 0.5 + 0.5\cos(\theta - \pi) \tag{5}$$
$$F8 = sin(\theta) \tag{6}$$

can be transformed using the angle addition theorem [25]

$$\cos(\alpha \pm \beta) = \cos(\alpha)\cos(\beta) \mp \sin(\alpha)\sin(\beta) \tag{7}$$

to W, X and Y:

$$C1 + C2 = 0.5 + 0.5\cos(\theta) + 0.5 + 0.5\cos(\theta - \pi) \tag{8}$$
$$= 1 + 0.5\cos(\theta) + \underbrace{0.5\cos(\theta)\cos(-\pi)}_{=-0.5\cos(\theta)} - \underbrace{0.5\sin(\theta)\sin(-\pi)}_{=0} \tag{9}$$
$$= 1 = \mathbf{W} \tag{10}$$
$$C1 - C2 = 0.5 + 0.5\cos(\theta) - 0.5 - 0.5\cos(\theta - \pi) \tag{11}$$
$$= 0.5\cos(\theta) - 0.5\cos(\theta - \pi) \tag{12}$$
$$= 0.5\cos(\theta) + 0.5\cos(\theta) \tag{13}$$
$$= \mathbf{Y} \tag{14}$$
$$F8 = \sin(\theta) = \mathbf{X}. \tag{15}$$

Three identical capsules rotated by 120° can be generalized using Eq. (3.3) to:

$$T1 = p + (1-p)cos(\theta) \tag{16}$$
$$T2 = p + (1-p)cos\left(\theta - \frac{2\pi}{3}\right) \tag{17}$$
$$T3 = p + (1-p)cos\left(\theta + \frac{2\pi}{3}\right). \tag{18}$$

Using Eq. (7) and $\gamma = (1-p)cos(\theta)$, we can easily solve

$$T1+T2+T3 = 3p+\gamma+(1-p)cos\left(\theta-\frac{2\pi}{3}\right)+(1-p)cos\left(\theta+\frac{2\pi}{3}\right) \tag{19}$$

$$= 3p+\gamma+\underbrace{\gamma\cos\left(\frac{2\pi}{3}\right)}_{=-0.5}+(1-p)\sin(\theta)\sin\left(\frac{2\pi}{3}\right)+\gamma\cos\left(\frac{2\pi}{3}\right) \tag{20}$$

$$-(1-p)\sin(\theta)\sin\left(\frac{2\pi}{3}\right)$$

$$= 3p+\gamma-0.5\gamma-0.5\gamma \tag{21}$$

$$= 3p \propto \mathbf{W} \tag{22}$$

$$W = \frac{T1+T2+T3}{3} \tag{23}$$

$$T2-T3 = p+(1-p)cos\left(\theta-\frac{2\pi}{3}\right)-p-(1-p)cos\left(\theta+\frac{2\pi}{3}\right) \tag{24}$$

$$= \gamma\cos\left(\frac{2\pi}{3}\right)+(1-p)\sin(\theta)\sin\left(\frac{2\pi}{3}\right)-\gamma\cos\left(\frac{2\pi}{3}\right) \tag{25}$$

$$+(1-p)\sin(\theta)\sin\left(\frac{2\pi}{3}\right)$$

$$= 2(1-p)\sin(\theta)\sin\left(\frac{2\pi}{3}\right) = \sqrt{3}(1-p)\sin(\theta) \propto \mathbf{X} \tag{26}$$

$$X = \frac{T2-T3}{\sqrt{3}(1-p)} \tag{27}$$

$$2T1-T2-T3 = 2p+2\gamma-p-(1-p)cos\left(\theta-\frac{2\pi}{3}\right)-p \tag{28}$$

$$-(1-p)cos\left(\theta+\frac{2\pi}{3}\right)$$

$$= 2\gamma-\gamma\cos\left(\frac{2\pi}{3}\right)-(1-p)\sin(\theta)\sin\left(\frac{2\pi}{3}\right)-\gamma\cos\left(\frac{2\pi}{3}\right) \tag{29}$$

$$+(1-p)\sin\left(\frac{2\pi}{3}\right)$$

$$= 3\gamma = 3(1-p)\cos(\theta) \propto \mathbf{Y} \tag{30}$$

$$Y = \frac{2T1-T2-T3}{3(1-p)} \tag{31}$$

B Publications

Published Work

Audio Engineering Society

Convention Paper 10508

Presented at the 151st Convention
2021 October, Online

A Neural Beamforming Front-End for Distributed Microphone Arrays

Jonathan D. Ziegler[1, 2], Leon Schröder[1], Andreas Koch[1], and Andreas Schilling[2]

[1]*Stuttgart Media University, Institute for Applied Artificial Intelligence, Nobelstr. 10, 70569 Stuttgart, Germany*
[2]*Eberhard Karls Univerity, Institute for Visual Computing, Sand 14, 72076 Tübingen, Germany*

Correspondence should be addressed to Jonathan D. Ziegler (zieglerj@hdm-stuttgart.de)

ABSTRACT

Robust real-time audio signal enhancement increasingly relies on multichannel microphone arrays for signal acquisition. Sophisticated beamforming algorithms have been developed to maximize the benefit of multiple microphones. With the recent success of deep learning models created for audio signal processing, the task of Neural Beamforming remains an open research topic. This paper presents a Neural Beamformer architecture capable of performing spatial beamforming with microphones randomly distributed over very large areas, even in negative signal-to-noise ratio environments with multiple noise sources and reverberation. The proposed method combines adaptive, nonlinear filtering and the computation of spatial relations with state-of-the-art mask estimation networks. The resulting End-to-End network architecture is fully differentiable and provides excellent signal separation performance. Combining a small number of principal building blocks, the method is capable of low-latency, domain-specific signal enhancement even in challenging environments.

1 Introduction

High-quality, noise-free audio has become of ever greater importance with increases in human-computer interaction, telecommunication, web conferencing, and modern pro-audio applications. Ease of communication without personal contact has become a vital part of modern society. The enhancement of signals with respect to specific signal components such as speech can be performed with a wide variety of methods [1]. By time-aligning and filtering a set of microphone signals with respect to a defined signal source, surrounding noise and reverberation can be attenuated in the output signal. One of the largest challenges for such beamformers is the determination of the correct Time Difference of Arrival (TDOA), especially for spontaneous, large aperture microphone arrays of unknown configuration [2]. In recent years, machine learning has had a large impact on audio signal processing, redefining the state of the art in many topics. The task of source separation can be approached in numerous ways, with adaptive, nonlinear filtering on individual audio channels being the most prominent and easily available to date. Recently, multichannel, deep-learning based array processing has become increasingly relevant. The ability for Neural Beamforming networks to detect correlations of domain-specific signal components robustly and to generate the appropriate spatial filters is of great value. Current systems can generally be categorized into two approaches, in which the spatial information is either precomputed analytically and fed into the beamforming network [3, 4], or multiple neural networks

are used independently to achieve source separation [5, 6, 7]. Although investigations into End-to-End approaches have been remarkably successful [8, 9, 10], some basic limitations remain. Both referenced approaches directly use convolutions as beamforming filters. While Sainath et al. use dedicated convolutional layers to generate static spatial filters resulting in learnable "look-directions", Luo et al. use Temporal Convolutional Networks [11] or Dual-Path RNN [12] for the adaptive estimation of filters based on a preprocessed reference channel, raw auxiliary channels and the cosine similarity. While other approaches seem promising [13, 14], no performance data on arbitrary or large-aperture arrays are available for reference. The method introduced in the following sections circumvents the aforementioned limitations by using iterative downsampling and upsampling elements to adaptively produce long beamforming filters. The architecture presents an efficient End-to-End Neural Beamformer for large-aperture arrays and is capable of processing array signals for microphone distances of over 110 m in real time, outperforming the examined baseline approaches.

2 Neural Beamforming

The approach to Neural Beamforming described in the following sections tackles the task of generating appropriate spatial beamforming filters via the encoding of audio signals and spatial information into a shared latent space from which the beamforming filters are generated. Additionally, adaptive filtering of the audio channels prior to the computation of spatial relations is incorporated to enable the system to filter the input signals specifically for the task of spatial analysis. The use of learnable filter elements prior to the computation of pairwise cross correlations creates the ability to compute domain-specific spatial filters, which can greatly increase the beamformer's robustness to noise and reverberation. In section 2.1, an appropriate signal model is presented, sections 2.2 and 3 describe the method and implemented network architectures in detail.

2.1 Problem Definition

Spatial beamforming can be achieved using M audio channels m_ν recorded by $M \geq 2$ transducers. The discrete, time-domain channels consist of I signal and J noise components:

$$m_\nu[t] = \sum_{i=1}^{I} \tilde{s}_i^\nu[t] + \sum_{j=1}^{J} \tilde{n}_j^\nu[t]. \tag{1}$$

Every signal $\tilde{s}_i^\nu[t]$ and noise $\tilde{n}_j^\nu[t]$ consist of the corresponding signal and noise sources $s_i[t-\delta_i^\nu]$ and $n_j[t-\delta_j^\nu]$, propagated from their respective source position i or j to the transducer ν. The propagation transformations introduce time-delays δ and are expressed as convolutions with corresponding impulse responses $h_{s_i}^\nu$ and $h_{n_j}^\nu$ [15]:

$$\tilde{s}_i^\nu[t] = \left(s_i * h_{s_i}^\nu\right)[t], \quad \tilde{n}_j^\nu[t] = \left(n_j * h_{n_j}^\nu\right)[t], \tag{2}$$

resulting in

$$m_\nu[t] = \sum_{i=1}^{I} \left(s_i * h_{s_i}^\nu\right)[t] + \sum_{j=1}^{J} \left(n_j * h_{n_j}^\nu\right)[t]. \tag{3}$$

In this case, the goal is to find additional h_i^ν that maximize s_i in the summed combination ς_i of all m_ν:

$$\varsigma_i[t] = \sum_{\nu=1}^{M} \left(m_\nu * h_i^\nu\right)[t]. \tag{4}$$

Finding optimal h_i^ν is difficult and computationally expensive. For a simplified approach, h_i^ν can be approximated by a time shift δ_i^ν and a Finite Impulse Response (FIR) filter $\tilde{h}_i^\nu$ of relatively short length.

2.2 Differentiable Adaptive Generalized Cross Correlation

For microphone arrays of known spatial distribution, finding the correct time shifts δ_i^ν can be solved either geometrically, if the desired beam direction is known, or by means of a localization method, such as Steered-Response Power Phase Transform (SRP-PHAT) [16]. For arrays of unknown spatial distribution, pairwise time delay estimation can be performed by means of the cross correlation ϕ, expressed with the $\star$ operator [17]:

$$\phi_{m_1,m_2}[\tau] = (m_1 \star m_2)[\tau] \tag{5}$$

$$= \sum_{t=-\infty}^{\infty} \overline{m_1[t]} m_2[t+\tau], \tag{6}$$

with $\overline{[\quad]}$ describing the complex conjugation operation. Using the convolution theorem, (6) can be expressed as

$$\phi_{m_1,m_2}[\tau] = \mathscr{F}^{-1}\left\{\overline{\mathscr{F}\{m_1\}} \cdot \mathscr{F}\{m_2\}\right\}[\tau], \tag{7}$$

with $\mathscr{F}\{\quad\}$ representing the Fourier transform and $\mathscr{F}^{-1}\{\quad\}$ representing the inverse Fourier transform. Applying Phase Transform weighting leads to the computation of the Generalized Cross Correlation with Phase Transform weights (GCC-PHAT) [18]:

$$\phi^g_{m_1,m_2}[\tau] = \mathscr{F}^{-1}\left\{\frac{\overline{\mathscr{F}\{m_1\}} \cdot \mathscr{F}\{m_2\}}{\left|\overline{\mathscr{F}\{m_1\}} \cdot \mathscr{F}\{m_2\}\right|}\right\}[\tau]. \tag{8}$$

In the absence of interference and reverberation and for a single source s, the main peak of ϕ^g indicates the TDOA of the source with respect to the two input channels:

$$\delta^v = \underset{\tau}{\mathtt{arg\,max}}\left\{\phi^g_{m_0,m_v}[\tau]\right\}. \tag{9}$$

In real-world scenarios with multiple target and noise sources, TDOA estimation becomes unreliable, especially when using short audio buffers required for real-time applications. Addressing this problem from a data-driven perspective can improve the performance with domain knowledge.

Performing time alignment of the individual microphone signals within a neural network is a nontrivial task. As (9) is not differentiable, it cannot be incorporated into an End-to-End network architecture. Instead, spatial filters h^v_i are internally generated by the model, using latent representations of the input signals and the respective ϕ^g vectors. The generated filters are then applied to the microphone signals. While the process of Delay-and-Sum (DS) beamforming can be implemented in a strictly analytical way, adaptive pre-filtering and nonlinear pattern enhancement greatly improve performance over conventional methods. Additionally, the spatial relations for domain-specific signal classes can be computed, while (9) strictly extracts the correlation of the signal source with the highest sound pressure level found in the recorded signals.

3 Network Architecture

The main principle of the proposed architecture is to filter and synchronize multiple channels prior to multichannel mask estimation. GCC on adaptively filtered input signals is internally used by the network to extract feature-dependent, domain-specific spatial relations between channel pairs. Figure 1 shows the top-level model architecture, developed using the TensorFlow and Keras frameworks. A total of M microphones, with one defined reference channel m_0 are processed. Prior to multichannel mask estimation, every pair of signals m_0 and m_v is passed to an Aligner block which is discussed in detail in section 3.1 and can be seen in Figure 2. In the following sections, the individual components and the motivation behind the design choices are discussed.

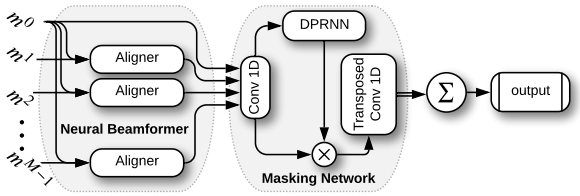


Fig. 1: Multiple audio channels are passed through the Neural Beamformer for synchronization. After spatial filtering, a linear encoder transforms the time-domain audio into a higher-dimensional feature vector. Then, a DPRNN mask approximation network generates channel-specific, per-sample feature-space amplitude masks, which are applied to the respective channels of encoded audio. After decoding the output to time-domain audio signals and summing the result, the model outputs a single-channel time-domain audio signal with the same buffer length as the reference input channel.

3.1 Channel Synchronization - Aligner

One important requirement for the proposed method is the ability to extract the spatial relations of domain-specific signal components. Signal classes, for example *speech*, are to be enhanced in the signal prior to the computation of spatial relations. This enables the model to better make use of beamforming capabilities in environments that contain high levels of interference and noise. To achieve this, adaptive, nonlinear filtering of the input signals is implemented. The signals are encoded using EncodeWaveform blocks (subsection 3.3) and passed to FilterBlocks (subsection 3.2) for masking and filtering. The filtered signals are correlated to extract the spatial relation of the channels with respect to the desired signal components. The correlation

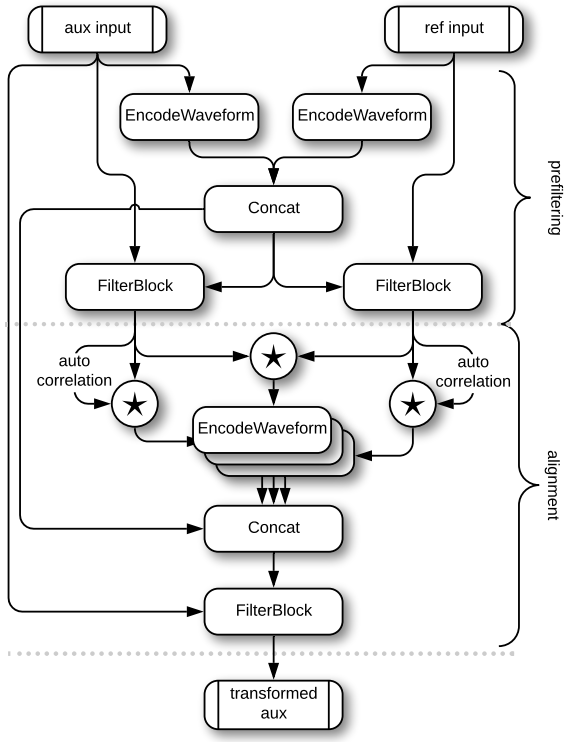


Fig. 2: Aligner architecture: Individual auxiliary and reference microphone pairs are processed with respect to the reference microphone. Adaptive FilterBlocks learn domain-specific signal components which are enhanced before spatial relations are computed for the filtered signals via cross- and autocorrelations. The encoded spatial vector is then used to generate multiplicative and convolutional filters in the final FilterBlock.

vectors are then encoded and concatenated with the latent vectors of both input channels. This expanded latent vector is then passed to an additional FilterBlock in combination with the original auxiliary input. In this block, channel synchronization and masking are performed to generate the transformed auxiliary signal.

3.2 Filtering and Masking - FilterBlock

The FilterBlock, shown in Figure 3, combines two main filtering approaches, namely convolutive filters and per-sample filter masks. Two individual GenerateWaveform blocks are used to generate filter vectors

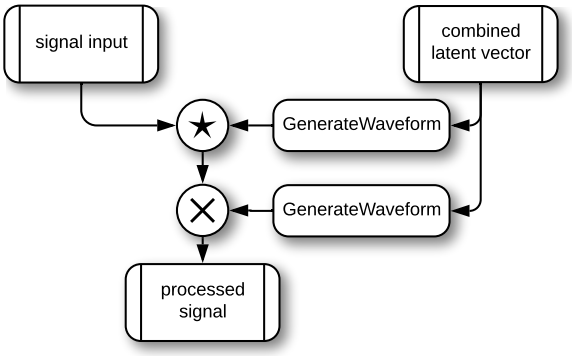


Fig. 3: The FilterBlock uses two GenerateWaveform blocks to generate FIR filters and amplitude masks and applies them to an input signal.

from the latent vector input. The filters are then applied via convolution and element-wise multiplication to the input signal. This flexible architecture provides capabilities for masking, FIR-filtering, scaling, and any desired combination of the aforementioned methods. For the configuration used to generate the results presented in this paper, the prefiltering blocks within the Aligner only use the masking components. The alignment filter generation exclusively relies on convolutive filtering to compensate spatial propagation and enhance the signal's spectral content.

3.3 Encoder - EncodeWaveform

To accommodate variable signal lengths with the same general architecture, the EncodeWaveform blocks, shown in Figure 4, are constructed using an iterative core, highlighted in gray. After normalization to zero mean and unit variance, activations (abbreviated with Act in the corresponding visualizations) and a multiplicative gate are created using convolutional layers feature space[19]. After passing through Layer Normalization, the output and the previous input are concatenated and passed to a convolutional downsampling layer, which serves as the input of the next iteration [20]. Once the required number of iterations has been performed, a final dense layer transforms the activations into the latent vector, which is then concatenated with the standard deviation and the mean of the input signal.

3.4 Decoder - GenerateWaveform

The GenerateWaveform blocks, shown in Figure 5, present the inverse operation of the EncodeWaveform.

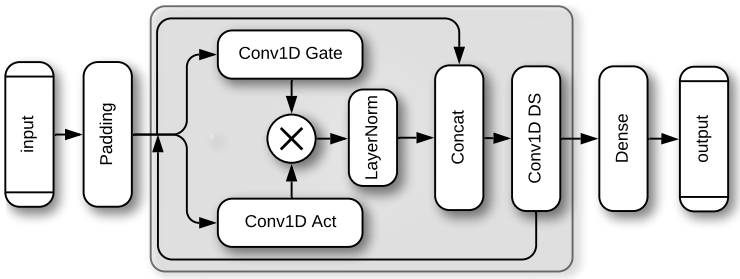


Fig. 4: The EncodeWaveform block uses gated and strided convolutions to perform encoding of signals to a defined latent size. Downsampling is performed by the strided convolutions and indicated by DS.

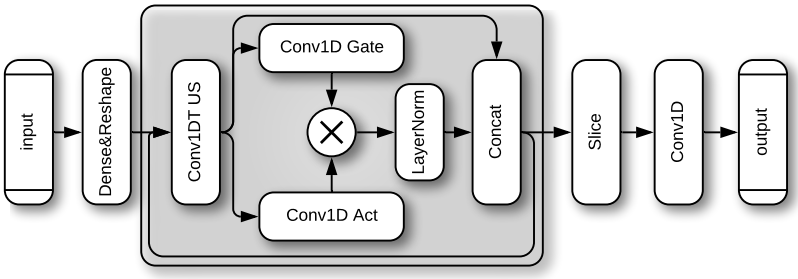


Fig. 5: The GenerateWaveform block uses gated and transposed convolutions to generate filters and amplitude masks from latent signal representations. Upsampling is performed by the transposed convolutions as indicated with US.

Such blocks can be used to create filter masks and spatial filters. Although the general architecture is identical for all applications, the iterative core enables a variable definition of the desired output size.

3.5 Mask Estimator - Dual-Path RNN

The Dual-Path RNN (DPRNN) architecture by Luo et al. generates state-of-the-art signal separation results for single-channel applications [12]. The approach splits the time-domain input buffers into smaller chunks, which are stacked to create input volumes. These volumes are then processed with separate bidirectional LSTM layers operating on the time and chunk axes. Additional linear convolution and transposed convolution layers encase the model and provide a learnable, time-domain base for the signal separation process. As shown in Figure 1, the synchronized array channels are individually processed by a DPRNN subnetwork and the resulting amplitude masks are applied to the signals prior to summation.

4 Experimental Setup

4.1 Training Data

Test and training data were created by simulating virtual acoustic environments [21, 22]. Room geometries were synthesized ranging from 8 m to 115 m per dimension, focusing on direct propagation and early reflections by restricting room simulation to third-order processing. One *signal*, a random number from three to seven *noise* sources, and a fixed number of microphones depending on the network configuration were randomly placed in the synthetic virtual environment. For training, 550000 multichannel buffers were randomly sampled from 550000 virtual recordings. Each buffer contains a reference channel, spanning 4096 samples and M-1 additional microphone signals of the same length, preceded by a context window of 12288 samples. The validation set contains 55000 buffers sampled correspondingly, using previously unseen source data. For the *signal* class, speech recordings from the VCTK corpus were used. *Noise* samples were extracted from the ESC50 corpus [23, 24]. All data are sampled at 48 kHz.

Conv Chanels	Conv Kernel Size	LSTM Features	Chunk Size	DPRNN Blocks	Buffer Size (Samples)
128	20	128	128	6	4096

Table 1: Parameters of the DPRNN Sub-Network.

Latent Size	US & DS Stride	US & DS Kernel Size	Gated Conv Kernel Size	Aligner Filter Length
32	8	8	5	16384

Table 2: Parameters of the Neural Beamformer Sub-Network.

4.2 Training Process

The model was trained using a MSE loss comparing the target signal recorded by the reference channel and the model output. Optimization was performed using the Ranger optimization algorithm [25, 26]. Standard Adam and Lookahead parameters were used, combined with a learning rate of 10^{-3}, a warm-up period of 500 steps and a learning rate decay of 4500 steps to reach the final rate of 10^{-4}. The training process was enhanced with channel dropout in which individual input channels were randomly set to 0 or filled with uncorrelated audio of the same signal statistics. The motivation of this dropout was to force the network to learn to completely reject individual input channels if necessary.

4.3 Model Configuration

Detailed information on the configuration chosen for the individual elements of the network can be referenced in Tables 1 and 2. Using this setup, the single-channel DPRNN contains approximately 1.4M parameters. The conversion to a multi-channel separation network using the Neural Beamformer front-end adds 347000 parameters, thus introducing an increase in model size of 25 % to 1.75M. Although the increase in model complexity is fairly modest, processing multiple channels with this shared architecture significantly increases the model runtime. When comparing the five-channel combination of Neural Beamformer and DPRNN with a single-channel DPRNN, the inference time of the model for a 85 ms buffer[1] increased from 32 ms to 80 ms on a single GPU.

5 Results

As a baseline, single- and multichannel versions of the same DPRNN architecture as used in the Neural Beamformer were trained (SC-DPRNN and MC-DPRNN). Additionally, Oracle and GCC-PHAT-based Delay-and-Sum beamforming, and combinations of GCC-PHAT and Oracle beamforming with the single channel DPRNN are referenced (Oracle, GCC-PHAT, GCC-DPRNN, Oracle-DPRNN). Overall separation performance is monitored by means of SDRi, the improvement of the Signal-to-Distortion Ratio, compared to the reference receiver, using [27]. In Table 3, an evaluation of the SDRi of the proposed method using five channels is presented under varying conditions, compared to the baseline approaches. The methods were compared on ten 20 s room simulations per configuration, resulting in a total of 40 min of audio used for evaluation. Audio material, room dimensions, and both microphone and source positions were randomly sampled for each iteration. During inference, each 20 s example is converted to blocks of 4096 + 12288 samples with 50 % overlap and passed to the individual processors. The resulting signals are (time-domain) Hann-windowed and recombined for evaluation. Even though the single-channel version of the DPRNN performs exceptionally well, applying the approach to multiple channels without spatial alignment results in negative SDRi (MC-DPRNN not shown in Table 3). As the Beamformer is trained in an End-to-End fashion, extracting this model component and evaluating the signal separation performance without the masking network would be misleading and thus has been omitted. Although the NBF-DPRNN model provides a relatively modest improvement in SDRi over the single-channel approach, the subjective reduction of artifacts and low-frequency residual noise is quite noticeable. The combination of Oracle beamforming with the DPRNN masking architecture shows that improved signal synchronization can provide a significant increase in overall model performance and future work will investigate improvements in the ability of the Beamformer components. Audio examples can be found at `https://zieglerj.home.hdm-stuttgart.de/nbf.html`.

[1] This time refers to the 4096 samples at 48 kHz sampling rate without the 12288 samples context provided to the network in the case of the Neural Beamformer.

	SNR	SDRi in dB			
	in dB	28 m	63 m	110 m	159 m
	-6	-3.11 ± 2.31	-3.03 ± 2.40	-2.79 ± 2.81	-2.28 ± 2.84
GCC	0	-0.95 ± 1.51	-1.38 ± 1.98	-1.94 ± 2.37	-2.04 ± 2.48
	6	-1.3 ± 1.10	-0.56 ± 1.62	-0.86 ± 2.12	-1.41 ± 1.98
	-6	6.56 ± 1.14	6.25 ± 0.91	6.1 ± 1.41	5.67 ± 1.61
Oracle	0	6.65 ± 1.13	6.34 ± 1.18	5.25 ± 1.87	4.15 ± 2.41
	6	6.57 ± 1.06	6.04 ± 1.77	4.75 ± 2.81	3.36 ± 3.56
GCC	-6	4.24 ± 3.26	4.19 ± 3.46	4.14 ± 3.92	4.69 ± 3.55
–	0	4.84 ± 1.85	4.15 ± 2.17	3.8 ± 2.26	3.96 ± 2.51
DPRNN	6	0.28 ± 1.82	1.06 ± 1.70	1.32 ± 1.94	1.66 ± 1.57
SC	-6	9.75 ± 1.00	9.72 ± 0.82	9.44 ± 0.80	9.48 ± 0.93
–	0	11.05 ± 1.88	11.08 ± 1.88	10.94 ± 1.86	10.86 ± 1.82
DPRNN	6	6.50 ± 1.32	6.46 ± 1.29	6.39 ± 1.26	6.13 ± 1.29
NBF	-6	10.19 ± 1.15	10.19 ± 1.13	9.96 ± 1.24	9.87 ± 1.18
–	0	11.58 ± 2.16	11.70 ± 2.00	11.59 ± 1.99	11.48 ± 2.12
DPRNN	6	7.12 ± 1.39	7.21 ± 1.27	7.14 ± 1.22	6.91 ± 1.35
Oracle	-6	14.66 ± 1.97	14.64 ± 1.95	14.27 ± 1.74	13.92 ± 2.07
–	0	14.78 ± 1.90	14.34 ± 1.57	13.42 ± 1.83	12.73 ± 2.07
DPRNN	6	8.77 ± 1.42	8.37 ± 1.56	7.75 ± 1.70	7.18 ± 1.72

Table 3: Performance Comparison of the Neural Beamformer in dB SDRi under variation of maximum microphone distance and mixture SNR at the reference channel.

6 Discussion

6.1 Processing of Large Time Delays

In order to operate in real time, buffers of 4096 samples were chosen. This presents a fundamental challenge for time-delays of over 85 ms, or distances of over 29 m. By inverting the application of the described model and focusing on noise modeling at the reference microphone instead of signal enhancement, the context vector of 12288 samples can be used more effectively. Beamforming is performed on the noise, which is more probable to have been recorded by other microphones before reaching the reference receiver and thus is captured in the context buffers. Signal enhancement is then performed by subtracting the modelled noise from the reference microphone signal. In this configuration, the possible delay compensation is only restricted by the microphone configuration and the chosen length of the context vector, which in this case contains a total of 16384 samples, resulting in a possible delay compensation of 341 ms, or 116 m. Even though the largest maximum microphone distance in Table 3 is 159 m, most examples are within the required distance as room dimensions, as well as microphone and source positions are uniformly sampled.

6.2 Training with Simplified Simulated Data

As mentioned in sections 4.1, data simulation was performed without diffuse reverberation. As previously stated, the exact inference of optimal impulse responses h_i^v is extremely complex, preventing reliable training convergence of the proposed model. Concentrating on the main contributing factors during training and excluding the task of explicit dereverberation presents an option for efficient and reliable training of Neural Beamformers. Inference experiments were performed showing that models trained with reduced simulation complexity are capable of performing well on data that contains a wide range of reverberation levels.

7 Conclusion

This paper presents a physics-informed and fully differentiable front-end for multichannel array processing, aimed at extracting domain-specific signals from a noisy mixture. Combined with mask estimation models, the high accuracy and short inference times enable the system to be used in real time applications. The proposed method outperforms the examined baseline approaches and provides a fully End-to-End neural beamforming network architecture, capable of processing microphone array signals with microphone distances of over 110 m.

Acknowledgments

This research was in part funded by the *Zentrales Innovationsprogramm Mittelstand*, a grant from the *Bundesministerium für Wirtschaft und Energie*.

References

[1] Benesty, J., Cohen, I., and Chen, J., *Fundamentals of Signal Enhancement and Array Signal Processing*, John Wiley & Sons Singapore Pte. Ltd, Singapore, 2017, ISBN 978-1-119-29313-2 978-1-119-29312-5, doi:10.1002/9781119293132.

[2] Ying Yu and Silverman, H. F., "An improved TDOA-based location estimation algorithm for large aperture microphone arrays," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, 2004.

[3] Gu, R., Chen, L., Zhang, S.-X., Zheng, J., Xu, Y., Yu, M., Su, D., Zou, Y., and Yu, D., "Neural spatial filter: target speaker speech separation assisted with directional information," in *Interspeech 2019*, pp. 4290–4294, ISCA, 2019, doi:10.21437/Interspeech.2019-2266.

[4] Qian, K., Zhang, Y., Chang, S., Yang, X., Florencio, D., and Hasegawa-Johnson, M., "Deep learning based speech beamforming," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5389–5393, 2018, doi:10.1109/ICASSP.2018.8462430.

[5] Wang, Z.-Q. and Wang, D., "All-neural multichannel speech enhancement," in *Interspeech 2018*, pp. 3234–3238, ISCA, 2018, doi:10.21437/Interspeech.2018-1664.

[6] Koyama, Y. and Raj, B., "W-Net BF: DNN-based Beamformer Using Joint Training Approach," in *arXiv:1910.14262 [cs, eess]*, 2019.

[7] Yoshioka, T., Chen, Z., Liu, C., Xiao, X., Erdogan, H., and Dimitriadis, D., "Low-latency speaker-independent continuous speech separation," in *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 6980–6984, 2019.

[8] Sainath, T. N., Weiss, R. J., Wilson, K. W., Narayanan, A., Bacchiani, M., and Senior, Andrew, "Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 30–36, IEEE, Scottsdale, AZ, USA, 2015, ISBN 978-1-4799-7291-3, doi:10.1109/ASRU.2015.7404770.

[9] Sainath, T. N., Weiss, R. J., Wilson, K. W., Li, B., Narayanan, A., Variani, E., Bacchiani, M., Shafran, I., Senior, A., Chin, K., Misra, A., and Kim, C., "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(5), pp. 965–979, 2017, ISSN 2329-9290, 2329-9304, doi:10.1109/TASLP.2017.2672401.

[10] Luo, Y., Han, C., Mesgarani, N., Ceolini, E., and Liu, S.-C., "FaSNet: Low-latency adaptive beamforming for multi-microphone audio processing," in *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*, pp. 260–267, 2019.

[11] Luo, Y. and Mesgarani, N., "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, 27(8), pp. 1256–1266, 2019.

[12] Luo, Y., Chen, Z., and Yoshioka, T., "Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 46–50, 2020.

[13] Gu, R., Wu, J., Zhang, S.-X., Chen, L., Xu, Y., Yu, M., Su, D., Zou, Y., and Yu, D., "End-to-End Multi-Channel Speech Separation," in *arXiv:1905.06286 [cs, eess]*, 2019.

[14] Wu, J., Chen, Z., Li, J., Yoshioka, T., Tan, Z., Lin, E., Luo, Y., and Xie, L., "An End-to-End Architecture of Online Multi-Channel Speech Separation," in *Interspeech 2020*, pp. 81–85, ISCA, 2020, doi:10.21437/Interspeech.2020-1981.

[15] Benesty, J., Jingdong, C., and Huang, Y., *Microphone Array Signal Processing*, Springer Berlin Heidelberg, 2008, ISBN 978-3-540-78612-2.

[16] DiBiase, J. H., *A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays*, Ph.D. thesis, Brown University, 2000.

[17] Azaria, M. and Hertz, D., "Time delay estimation by generalized cross correlation methods," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2), pp. 280–285, 1984, doi:10.1109/TASSP.1984.1164314.

[18] Knapp, C. and Carter, G., "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4), pp. 320–327, 1976, doi:10.1109/TASSP.1976.1162830.

[19] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K., "WaveNet: a generative model for raw audio," in *arXiv:1609.03499 [cs]*, 2016.

[20] Ba, J. L., Kiros, J. R., and Hinton, G. E., "Layer normalization," *arXiv:1607.06450 [cs, stat]*, 2016.

[21] Allen, J. B. and Berkley, D. A., "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, 65(4), pp. 943–950, 1979.

[22] Scheibler, R., Bezzam, E., and Dokmanić, I., "Pyroomacoustics: A Python package for audio room simulations and array processing algorithms," 2018.

[23] Veaux, C., Yamagishi, J., and MacDonald, K., "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2017.

[24] Piczak, K. J., "ESC: dataset for environmental sound classification," in *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pp. 1015–1018, ACM, New York, NY, USA, 2015, ISBN 978-1-4503-3459-4, doi:10.1145/2733373.2806390, event-place: Brisbane, Australia.

[25] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J., "On the variance of the adaptive learning rate and beyond," in *8th international conference on learning representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020.

[26] Zhang, M., Lucas, J., Ba, J., and Hinton, G. E., "Lookahead optimizer: k steps forward, 1 step back," in H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pp. 9597–9608, Curran Associates, Inc., 2019.

[27] Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., and Ellis, D. P. W., "MIR_EVAL: a transparent implementation of common MIR metrics." in H.-M. Wang, Y.-H. Yang, and J. H. Lee, editors, *ISMIR*, pp. 367–372, 2014.

Audio Engineering Society

Convention Paper 10321

Presented at the 148th Convention, 2020 June 2-5, Online

Acoustic Source Localization and High Quality Beamforming Using Coincident Microphone Arrays

Jonathan D. Ziegler[1,2], Hendrik Paukert[1], Andreas Koch[1], and Andreas Schilling[2]

[1]*Stuttgart Media University, Nobelstr. 10, 70569 Stuttgart*
[2]*Eberhard Karls Univerity, Sand 14, 72076 Tübingen*

Correspondence should be addressed to Jonathan D. Ziegler (zieglerj@hdm-stuttgart.de)

ABSTRACT

This paper presents an application-oriented approach to Acoustic Source Localization using a coincident microphone array. Multiple processing blocks are presented to generate a reactive, yet stable Direction of Arrival estimation tuned toward speaker tracking. Building on an energy based scanning method, individual characteristics, such as sound field directivity and static sound source positions are used for adaptive smoothing of the detected angle. The methods and resulting performance gain are discussed for the individual components of the algorithm. Objective performance is evaluated using simulated and recorded data. Audio quality is assessed using listening tests, which show a significant increase in subjective sound quality, noise suppression, and speech intelligibility when combining the tracker with a beamforming algorithm for coincident microphone arrays.

1 Introduction

With beamformers in mobile and smart-home devices gaining relevance, many applications focus on low-cost linear and circular arrays for Acoustic Source Localization (ASL) and tracking [1]. Advances in spherical array beamforming have enabled the creation of versatile, robust beamformers in three dimensions, often using spherical harmonics as an orthonormal base for beamforming [2–7]. Beamforming for professional high quality audio is still uncommon, as the large number of transducers needed for higher-order beams prevents the use of professional quality microphones [8]. Combinations of shotgun microphones and adaptive spectral beamformers have proven effective and can generate high quality audio [9]. The downside of such microphones is the need for manual, mechanical source tracking. Producing an audio signal of consistently high quality is difficult and requires skilled personnel. Recent research has shown that a first-order beamformer using a coincident microphone array can produce beam patterns similar to shotgun microphones - with a more linear frequency response for off angle sound incidence [10–12]. The combination of such beamformers with an effective algorithm for ASL can produce high quality audio signals of moving sources with Directivity Indices beyond the possibilities of classical first-order microphones [13]. This paper presents an application-oriented algorithm for real-time ASL using a coincident microphone array consisting of three high-end microphone capsules. The array configuration represents a simple setup that can be transformed onto a spherical harmonic base in two dimensions and create any beampattern expressible with spherical harmonics of $\mathcal{O}(1)$ [14]. The configuration of the capsules allows for first-order beamforming on the horizontal plane,

presenting an acceptable and well researched solution for many applications [15, 16]. A Steered Response Power (SRP) approach is chosen for initial Direction of Arrival (DOA) estimation, using virtual cardioid microphones as a scanning beam [17].

Details about the chosen microphone configuration, the resulting virtual microphone synthesis, and the ASL algorithm can be found in sections 2.1 and 2.2. A variable exponential smoothing algorithm increases the algorithm's angular stability, while maintaining high sensitivity for directional changes. The basic concept and the individual weighting factors are discussed in section 3. Performance is evaluated using objective error analysis and listening tests based on a set of subjective quality metrics. The experimental set up is discussed in section 4 and results are presented in section 5.

2 Acoustic Source Localization

2.1 Microphone Configuration

The described system uses a microphone configuration consisting of three high-end professional microphones. This guarantees a known and consistent frequency response of the individual capsules for on-axis as well as for off-axis pick up of audio events. Uniform frequency response for all angles is a critical requirement for high quality broadband beamforming [18]. To optimize coincidence relative to the horizontal plane, the capsules are stacked vertically, with a spacing of $\leq 30\,\text{mm}$. The capsules are mounted in a double-M/S configuration, with one cardioid capsule F facing 0°, a second cardioid R facing 180°, and a bidirectional figure-of-eight capsule B facing $\pm 90^\circ$. Capsule correction filters H_x are applied for further linearization of the signals. This step improves tracking and the beamformer's isotropic frequency response.

2.2 Steered Response Power ASL

Prior to ASL processing, a detection filter H_d is applied to the linearized microphone signals. This filter is designed to reject non-speech signals. The corrected and filtered microphone signals can be transformed onto the base of horizontal b-format Ambisonics [14]:

$$W = F_{lin,d} + R_{lin,d} \tag{1}$$

$$X = F_{lin,d} - R_{lin,d} \tag{2}$$

$$Y = B_{lin,d} \tag{3}$$

As W, X and Y represent a two-dimensional, orthonormal basis, any arbitrary first-order microphone pattern $M(\theta, p)$ can be synthesized on the horizontal plane, using the WXY-decoded signals and

$$M(W,X,Y,\theta,p) = pW + (1-p)\left(X\cos\theta + Y\sin\theta\right), \tag{4}$$

with p representing the polar pattern shape and θ the orientation on the horizontal plane [4, 19, 20]. The factor p can be statically set or dynamically manipulated in a range between 0, which results in the polar pattern of a dipole, and 1, which results in an omnidirectional polar pattern. The most commonly used values for p in this paper are $p = 0.5$, resulting in the unidirectional polar pattern of a virtual cardioid and $p = \frac{1}{3}$, creating a virtual supercardioid.

Using (4) with $p = 0.5$, any number n_M of virtual cardioid microphone signals can be synthesized. The virtual microphone with the highest relative Root Mean Square (RMS) value indicates the Direction of Arrival of the sound source:

$$\theta_{DOA} = \underset{\theta_i}{arg\,max}\left(\overline{M}(W,X,Y,\theta_i)\right), \tag{5}$$

with $\overline{M}$ representing Root Mean Square of M.

3 Tracker Stabilization

The described real-time setup uses audio buffers of 256 samples, sampled at 48 kHz. Figure 5a shows the raw directional information θ_{DOA}. The large amount of noise in the angle detection requires additional filtering, since beamformers using θ_{DOA} as beam orientation perform poorly and produce strong audible artifacts. Filtering is performed using exponential smoothing [21]:

$$\theta_s^t = \alpha\,\theta_{DOA}^t + (1-\alpha)\,\theta_s^{t-1}, \tag{6}$$

with θ_{DOA}^t and θ_s^t representing the input and smoothed output angle for time frame t and $\alpha \in (0,1]$ as the reactivity factor. Circular continuity of the angle is ensured within a separate function. If α is set dynamically, a smoothing effect can be achieved that is directly connected to a set of signal characteristics. In the following paragraphs, these factors will be called Confidence Indices (CI) and the smoothing process will be defined as Confidence Weighting. Figure 1 shows three characteristics contributing to two stages of smoothing. The

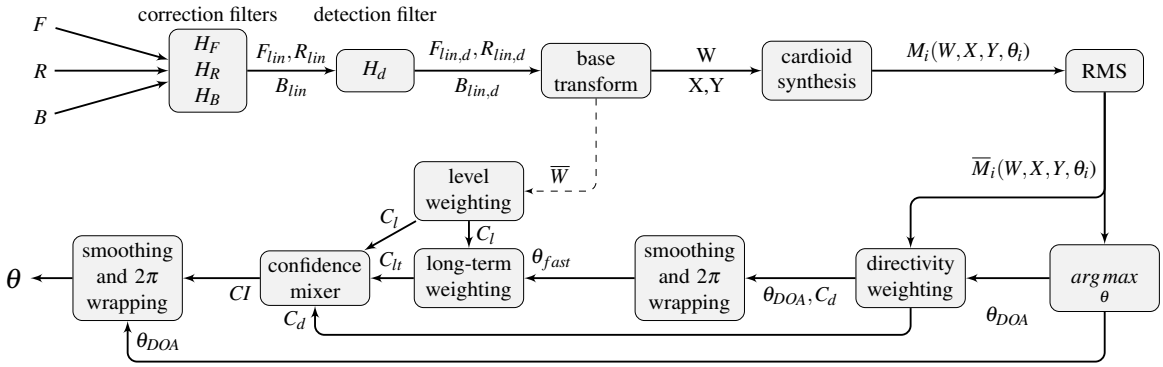


Fig. 1: Signal flow through the tracking algorithm. After compensating for nonlinear frequency responses of the microphones and filtering the incoming signals to a range of 200 Hz to 2000 Hz, the synthesis of virtual cardioids over 2π and RMS-maximization of the signals results in initial DOA estimations. Various weighting algorithms in combination with a variable exponential smoothing process create a more stable, yet reactive tracker.

initial filtering is performed using directivity weighting, a process that analyzes the level of directivity in the detected one-dimensional sound field. The output angle of this process θ_{fast} is passed on to the two following Confidence Weighting algorithms. A buffer is filled with multiple cycles of θ_{fast} to compare the detection angle with known source positions, which are dynamically learned and forgotten. Aditionally, a level weighting algorithm compares the omnidirectional level of the current buffer with the average level during speech. The Confidence Weighting processes create a combined CI, which is in turn used for a second filtering operation to compute the final tracker output θ. The algorithms are described in detail in the following sections.

3.1 Directivity Weighting

Directivity weighting uses the level of directivity within the recorded sound field as an indicator as to whether a given buffer contains an actual audio event. Figure 2 shows examples of buffers with high directivity (*left*) and low directivity (*right*). The Confidence Index C_d is obtained using the mean distance between the detected sound field, which is normalized, so that $\max(\overline{M}_i) = 1$, and the unidirectional level distribution U:

$$U_i = (0.5 + 0.5\cos(\theta_i - \theta_{DOA})), \tag{7}$$

$$C = \frac{1}{n_M}\sum_{i=1}^{n_M}\left(U_i - \overline{M}_i(W,X,Y,\theta_i)\right). \tag{8}$$

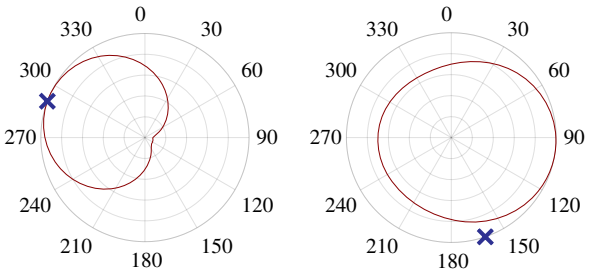


Fig. 2: Directivity of two audio buffers. The levels of 360 virtual cardioid microphones arranged with 1° spacing represent the detected, one-dimensional sound field. The closer the level distribution is to the optimal unidirectional distribution U, the higher the confidence index C_d. *Left*: Buffer with high level of directivity, *Right*: Buffer with low directivity. The marking indicates θ_{fast} for the displayed buffer. As C_d is large for the buffer on the left, $\theta_{fast} \approx \theta_{DOA}$. For the buffer on the right, a large portion of θ_{fast} is contributed by θ_{fast} of the previous buffer, and not by θ_{DOA}.

Scaling to the interval (0,1] is performed with

$$C_d = 10^{(\nu C)}, \tag{9}$$

with $\nu > 0$ representing a parameter controlling the reactivity of the tracker. Considering that for most cases

$$\overline{M}_i \geq U_i, \tag{10}$$

it is clear, that

$$C \leq 0 \text{ and } 0 < C_d \leq 1. \tag{11}$$

Using (6) and setting $\alpha = C_d$, an initial direction of arrival θ_{fast} can be computed. Figure 5b shows the effect of directivity weighting compared to the raw DOA data shown in Figure 5a.

3.2 Level Weighting

Level weighting analyzes the level of the current audio buffer and compares it to a threshold L. The signal used for level weighting is $\overline{W}$. The confidence index associated with level weighting C_l interacts directly with long-term weighting, as shown in Figure 1. C_l is computed as

$$C_l = \begin{cases} 1 & \text{for} \quad \overline{W} \geq L \\ 0 & \text{for} \quad \overline{W} < L \end{cases}. \tag{12}$$

3.3 Long-Term Weighting

In many acoustic scenarios the speaker positions remain quasi-static. Participants of a meeting mostly stay seated, a driver will remain in the driver's seat, etc. Long-term weighting makes use of this fact by assessing the sound field over a longer period of time. The initial DOA estimation θ_{fast} is stored in a buffer under the condition that the level confidence index C_l is set to 1. If $C_l \neq 1$, θ_{DOA} of the previous buffer is used. An average over 50 buffers is passed to the long-term weighting algorithm. The directional information is then classified using a point system. Every incoming angle is quantized with a resolution of 5° and results in a point for the associated bin. The total number of points is limited to 72, resulting in one point per 5° bin in the initial state. For a point to be awarded to the most recent position, a point must be deducted from the least recent DOA. This procedure creates a type of long-term memory for the algorithm. With the parameters presented in this paper, the system "forgets" an audio event after 19.2 s and can adapt to a new static source in 1.5 s to 3 s. Figure 3 shows the point score after processing an excerpt of Scenario I, described in section 4. All five speaker positions listed in Table 1 are clearly discernible. For the determination of the associated confidence index C_{lt}, θ_{DOA} is quantized to 5° and the point distribution is normalized to 1. The relative point value at the quantized angle corresponds directly to C_{lt}. This comparison is performed every cycle of the algorithm. If θ_{DOA} is within $\pm 1°$ of a peak in the long-term angle distribution, an additional confidence bonus is awarded (*snap-to* process).

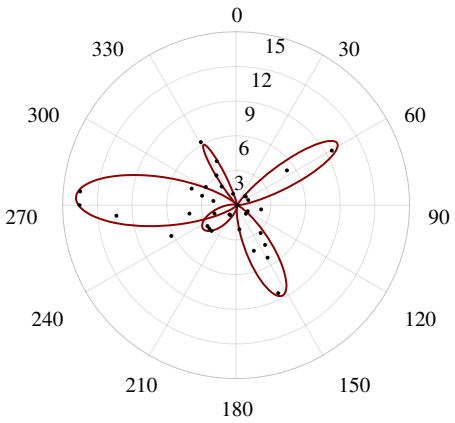


Fig. 3: Visualization of long-term confidence. The average detected angle θ_{fast} over the most current 267 ms, rounded to 5°, results in a point for the associated segment. As the total point count is limited, a point is deducted from the segment with the least recent position detection. With 72 available points, the algorithm learns a static position within 1.5 s to 3 s and forgets an audio event within 19 s. The point score is normalized to 1.

3.4 Confidence Mixing

Confidence mixing describes the process of combining all previously described Confidence Indices in the most effective way. Given (9), the final Confidence Index CI can be computed using C_d, C_l, C_{lt} and the mixing parameter κ:

$$CI = (\kappa C_d + (1-\kappa) C_d C_{lt})\, C_l. \tag{13}$$

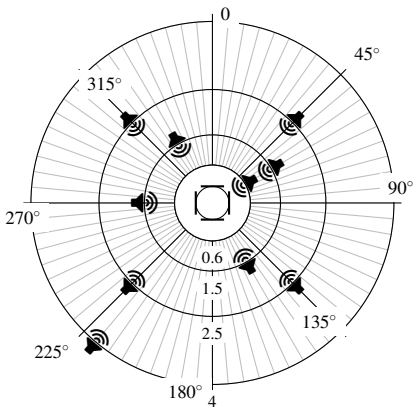


Fig. 4: Loudspeaker distribution for the test setup. Ten studio monitors are placed at four distances and varying angles around the microphone array. The configuration is used to record dialog, noise and ambiance for beamformer evaluation.

4 Experimental Setup

Both synthetic data and real recordings were used during development. The subjective results in section 5.2.1 are exclusively presented using real recordings of the setup described in subsection 4.2. Objective results are presented using synthetic and real data.

4.1 Synthetic Data

Convolving audio data with appropriate room impulse responses (RIR) can generate accurate simulations of auditory scenes [22]. The synthetic data used for the results in section 5 are generated using [23]. The speech data were randomly selected from the VCTK speech corpus, consisting of short passages read by 109 different speakers [24]. Optional noise interference was selected from the ESC50 corpus, consisting of 2000 recordings of environmental sounds [25]. The acoustic environment was randomly sampled with room geometries ranging from 3 m to 8 m and room heights of 2.5 m to 4 m. Absorption coefficients and RT60 reverberation times were uniformly sampled in different ranges, as described along with the results in table 3.

4.2 Virtual Conference

ASL and tracking were evaluated on real recordings using a reproducible multi-channel loudspeaker setup consisting of eight identical Genelec 1029A loudspeakers, arranged on two concentric rings around the microphone array, combined with a far-range and a close-range loudspeaker. The microphone array was constructed using two Schoeps CCM 4V and one Schoeps CCM 8, mounted within a dedicated double-M/S shock mount. The results described in the following sections were gathered using a RME Fireface UFX.

Table 1: Speaker Positions, Scenario I

	1	2	3	4	5
Angle	60	150	220	270	330
Distance	1.5 m	1.5 m	4 m	1.5 m	1.5 m
Sum Duration	7.1 s	9.0 s	4.0 s	16.0 s	8.9 s

Table 2: Speaker Positions, Scenario II

	1	2	3	4
Angle	60	150	270	330
Distance	1.5 m	1.5 m	1.5 m	1.5 m
Sum Duration	24.7 s	1.0 s	22.3 s	9.1 s

The angular positioning of the virtual sources is shown in Figure 4. The audio material played back was based on [26] and consisted of near-anechoic recordings of male and female speech in German and English, recordings of office and household noise sources such as cell phones, moving chairs, doors, etc. and multi-channel recordings of traffic and construction noise with open and closed windows. Three scenarios were recorded, each with and without interference of background and object noise. The room used for the results of this paper was 8.3 m by 8.2 m, with a total height of 3.8 m. No acoustic treatment or furniture was present, which resulted in an RT60 of 2.31 s, averaged over the 500 Hz and 1000 Hz frequency bands.

4.3 Listening Tests

Listening tests were performed to evaluate the impact of different types of signal degradation prior to the tracker design. The results are presented in [27] and were used to prioritize during the development process. In addition, a larger listening test was performed using the tracker output with various beamforming algorithms. A short summary of the listening test can be seen in Table 5, detailed methods and results can

be found in [28]. For the test, 59 test subjects were asked to grade various sound recordings which were recorded using the test setup described in section 4 and processed with the tracking algorithm and a selection of 3-, 2-, and 1-channel beamformers, both commercially available and currently under development.

5 Results

The following sections will present both subjective and objective evaluations of the system's performance. Objective results are presented using synthetic and real audio data. It is important to mention that the results are only partially comparable as the synthetic data contains no speech pauses, which prevents error accumulation due to C_l-driven static positions at unfavorable angles between speech sections. Additionally, the simulated data cannot make use of the long-term Confidence Weighting as every position is randomly sampled on a Cartesian grid.

Objective error analysis is performed using two connected error metrics, θ_{err} and $\Delta\theta_{err}$. The angular error θ_{err} is computed using the circular distance between the reference angles θ_r^t and the detected angles θ^t, averaged over all time bins t:

$$\theta_{err}^t = \begin{cases} |\theta^t - \theta_r^t| & \text{for} \quad |\theta^t - \theta_r^t| \leq 180^\circ \\ 360 - |\theta^t - \theta_r^t| & \text{for} \quad |\theta^t - \theta_r^t| > 180^\circ \end{cases} . \tag{14}$$

The gradient is calculated using (14) and a two-point calculation:

$$d\theta^t = \frac{\theta^{t+1} - \theta^{t-1}}{2} \tag{15}$$

$$d\theta_r^t = \frac{\theta_r^{t+1} - \theta_r^{t-1}}{2} \tag{16}$$

$$\Delta\theta_{err}^t = \begin{cases} |d\theta^t - d\theta_r^t| & \text{for} \quad |\ \cdot\ | \leq 180 \\ 360 - |d\theta^t - d\theta_r^t| & \text{otherwise} \end{cases} \tag{17}$$

Table 4 shows the mean errors over all n_T time bins:

$$\theta_{err} = \frac{1}{n_T} \sum_{t=1}^{n_T} \theta_{err}^t \tag{18}$$

$$\Delta\theta_{err} = \frac{1}{n_T} \sum_{t=1}^{n_T} \Delta\theta_{err}^t . \tag{19}$$

The error calculations shown in Table 4 are performed on reference information which was manually labeled using the session file of the digital audio workstation used for the playback and recording of the test scenarios. The results presented in Table 3 are calculated using the geometric parameters of every individual simulation.

Both θ_{err} and $\Delta\theta_{err}$ represent important quality metrics for the tracking algorithm. While accurate localization of an acoustic source is important, stable tracking of sources while maintaining high reactivity during change of speakers equally influences the system's real-world usefulness.

Subjective quality assessments are presented using listening tests, performed on recorded audio[1]. The test subjects were asked to grade the recordings with respect to speech intelligibility, noise suppression and subjective quality for German and English test scenarios using a MUSHRA test [29]. Speech intelligibility was additionally analyzed using the Short Time Objective Intelligibility Index proposed in [30]. STOI compares clean speech with processed versions of the same audio. In this case, the clean studio recordings of the speech used for the virtual scenarios were compared to the recorded multi-channel playback.

5.1 Simulated Data

	RT_{60}	SNR	DRR	θ_{err}	$\Delta\theta_{err}$
C	<0.05 s		10.98 dB	3.17°	0.53°
N	<0.05 s	6.06 dB		33.65°	1.21°
C	0.4 s to 0.6 s		−7.20 dB	21.15°	0.68°
N	0.4 s to 0.6 s	5.96 dB		31.41°	0.59°
C	0.6 s to 1.5 s		−9.37 dB	37.75°	0.63°
N	0.6 s to 1.5 s	6.05 dB		50.92°	0.56°

Table 3: ASL performance analysis on synthetic data. Reverberation and additional noise both have strong negative effects on ASL.

ASL performance is evaluated on six one-minute sets of synthetic data, each containing 15 scenes of 4 s. The six sets can be categorized into three subsets, each containing a clean (C) and a noisy (N) simulation of the same scenario. Within the clean sets, only speech and the corresponding reverberation are present, the noisy

[1] Audio examples can be found at zieglerj.home.hdm-stuttgart.de/aslt-companion.html.

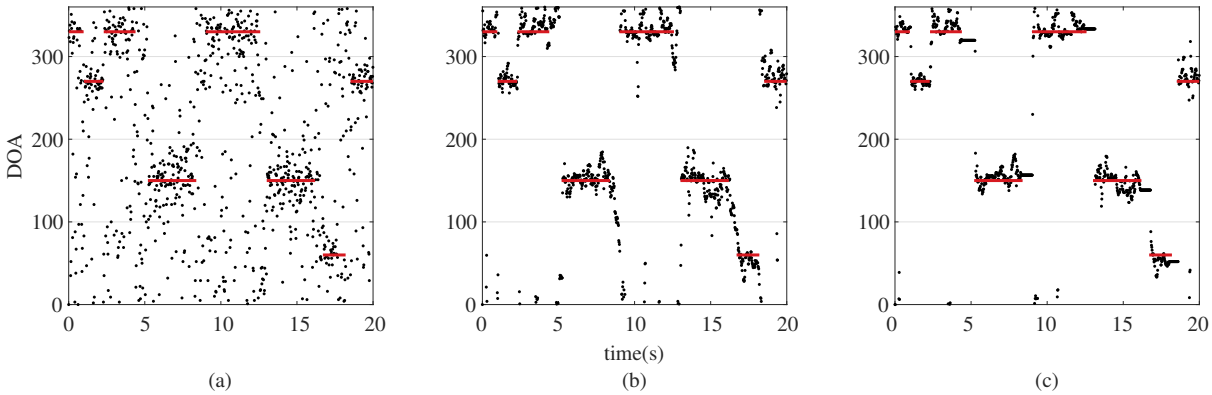


Fig. 5: Performance analysis of Confidence Weighting components. (a): Direct DOA estimate θ_{DOA}. (b): Added directivity weighting results in considerably less noise in the output θ_{fast}. (c): Additional level-dependent weighting reduces jumps during pauses. Long-term weighting further improves the accuracy and stability of the tracker output θ. Solid lines represent labeled reference. The values on display were down-sampled by a factor of 4 for increased clarity.

sets contain three additional noise sources randomly placed within the same area as the speaker. The three sets differ in their level of reverberation, which is given as Direct to Reverberant Ratio (DRR) and span RT60 reverberation times from ≈ 0 ms up to 1500 ms. Within the noisy sets, the signal to noise ratio is given for the virtual omnidirectional microphones.

Table 3 shows the results for the simulated audio data. ASL on clean speech in near-anechoic environments produces a mean error of 3.17° or approximately 1.8 %. An additional noise source within the simulated scenario increases the error by a factor of 10, mild reverberation causes a similar degradation of ASL performance. Furthermore, a clear correlation between the DRR and the localization error can be observed.

5.2 Recorded Data

The results shown in Table 4 were created using two different speech scenarios. Scenario I is a 45 s office scene in German, with one female and three different male speakers, located at five positions around the microphone. Scenario II is a 57 s dialog in English, between a female and a male speaker, with additional comments from two less prominently featured positions by the same speakers. Speaker positions and speech duration can be found in Tables 1 and 2[2]. Pauses and overlaps were intentionally added to simulate more realistic conversations. Two versions of each scene were recorded. The first version contains desired speech only. The second version contains interference consisting of office noises, such as cell phones, ripping paper, coffee cups and shifting chairs, being played back at 0.6 m to 1.5 m, whispered side-conversations being played back at 1.5 m and quadraphonic ambient recordings, such as traffic and construction noise, played back on a quadraphonic playback system, positioned at a radius of 2.5 m.

Figure 5 shows a 15 s extract from Scenario I at various steps of the signal processing chain, compared to the reference position. Figure 5a shows θ_{DOA}, the raw output of the virtual cardioid maximization process. Figure 5b shows θ_{fast}, the fast position estimation obtained using the variable exponential smoothing and only one Confidence Index, C_d, associated with directivity weighting. It can be seen that this step greatly reduces θ_{err} and $\Delta\theta_{err}$. For this reason, θ_{fast} is used throughout the processing chain as a good initial guess for θ. Figure 5c shows the additional improvement realized through the use of the algorithms described in section 3. While the values for θ_{err} are large, it is worth noting that the calculation is performed over the entire recording. Pauses between words and phrases were not removed during evaluation. This operation would require a subjective threshold of pauses and seemed

[2]Some sources shown in figure 4 only contained interference and ambiance, hence they are not listed in tables 1 and 2.

	German		German (noisy)		English		English (noisy)	
	θ_{err}	$\Delta\theta_{err}$	θ_{err}	$\Delta\theta_{err}$	θ_{err}	$\Delta\theta_{err}$	θ_{err}	$\Delta\theta_{err}$
θ_{DOA}	46.18	24.67	52.59	23.86	46.24	23.19	45.44	22.73
$+C_d$	13.53	1.96	22.25	1.88	14.97	2.07	15.02	1.95
$+C_{lt}$	**11.20**	0.97	**17.84**	0.96	11.55	1.24	**12.27**	1.20
$+C_l$	12.08	**0.84**	17.90	**0.93**	**11.54**	**1.23**	12.30	**1.19**
$+snap$	12.41	0.85	18.00	**0.93**	11.58	**1.23**	12.29	**1.19**
Δ_{SC}	−0.08 dB		−0.16 dB		−0.07 dB		−0.08 dB	

Table 4: Tracker Performance Analysis on recorded audio. Adding Confidence Weighting components improves the performance. While C_d globally improves stability and accuracy, other Confidence Indices show a more situation-dependent behavior.

	omni	beamformers
Speech Intelligibility	0.23 ± 0.12	0.61 ± 0.16
Noise Suppression	0.16 ± 0.12	0.52 ± 0.19
Subjective Quality	0.19 ± 0.12	0.65 ± 0.22

Table 5: Listening Test Results. In all categories the beamformed signal is preferred over the omnidirectional baseline. The mean result and pooled standard deviation over all tested beamformers are presented.

arbitrary and situation-dependent[3]. For the calculation of $\Delta\theta_{err}$, the pauses between labeled clips were additionally filled with the last available reference position of the preceding audio clip. This reflects the fact that a passive behavior of the tracker is desired during speech pauses.

5.2.1 Listening Tests

Regardless which beamformer is used, the signal outperforms that of a virtual omnidirectional microphone $F_{lin} + R_{lin}$. Even a simple gradient synthesis beamformer creating a virtual supercardioid facing the tracked direction θ provides improved intelligibility, noise reduction and subjective quality, compared to the omnidirectional signal. Once confidence weighting is applied, the tracked supercardioid performs without any audible artifacts.

Table 5 shows the summarized results of a listening test performed with 59 test subjects. Possible scores in the categories *Speech Intelligibility*, *Noise Suppression*, and *Subjective Quality* range from zero to one. On average, the use of the tracking algorithm in combination with a beamformer improves *Speech Intelligibility* by 170 %, *Noise Suppression* by 225 % and *Subjective Quality* by 256 %, compared to the signal of a static omnidirectional microphone of equal quality. Detailed results can be found in [28].

5.2.2 Speech Intelligibility

The Short Term Objective Intelligibility was calculated by comparing the dry voice recordings from the test scenarios with the signal of a virtual omnidirectional microphone and of a virtual supercardioid microphone synthesized towards the tracked angle θ, combined with various beamforming algorithms. The use of a virtual omnidirectional microphone results in an average STOI of 0.593, while a virtual, tracked supercardioid produces a STOI of 0.744 and all beamformers used in the test produce a mean STOI of 0.745, a 26 % improvement to the virtual omnidirectional signal.

6 Discussion

The overall effect of the various Confidence Indices depends on the application scenario. While directivity weighting universally improves ASL performance, long term smoothing and position snapping improve performance in static environments such as meetings. The results in Table 4 reflect the mean errors in the four described scenarios. The last row of data provides insight into the performance of a first-order supercardioid

[3] Ex: calculating the error for $\theta_{DOA} + C_d$ in Scenario I (German) using only buffers with an rms larger than 20 % of the mean rms of the recording results in an error θ_{err} of 10.06°. This equals a performance increase of 24.8 % when only examining frames subjectively deemed relevant.

beamformer driven with the tracker output. On average, the level of the target signal deviates by 0.1 dB from the reference value. This is well below the threshold of just-noticeable amplitude difference measured by Zwicker and Fastl for common SPL [31]. The STOI measurements presented in section 5.2.2 show that the objective difference between a static omnidirectional signal and a simple first-order beamformer is significantly larger than the improvement gained by the introduction of more complex beamforming algorithms. This is, in part, due to the focus of STOI. For further investigations, a testing algorithm with stronger focus on high quality audio will be selected.

7 Conclusion

The described system for acoustic source localization and tracking provides real-time Direction of Arrival information for coincident beamforming. A set of processing blocks is introduced to provide application-specific improvement over the direct output of energy based scanning methods, resulting in a more accurate and stable DOA-detection. Listening tests show a strong increase in speech intelligibility, noise suppression and subjective quality, when comparing the combination of tracker and beamformer with static microphone signals. When using a simple synthesized supercardioid driven by the tracker, the resulting signal is not subjectively discernible from a signal based on the reference position as input. The algorithm generates artifact-free audio and makes the system suitable for professional audio production applications as well as high-end conferencing and on-set recording.

Acknowledgments

This research was in part funded by the *Zentrales Innovationsprogramm Mittelstand*, a grant from the *Bundesministerium für Wirtschaft und Energie*.

The authors would like to thank Bernfried Runow for his contribution of beamformer test data.

References

[1] Reinette, A., Cornejo, M., Rouchon, C., and Fester, M., "Benchmarking Microphone Arrays: ReSpeaker, Conexant, MicroSemi AcuEdge, Matrix Creator, MiniDSP, PlayStation Eye," *Snips Labs*, 2017.

[2] Abhayapala, T. D. and Ward, D. B., "Theory and design of high order sound field microphones using spherical microphone array," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pp. II–1949–II–1952, 2002, doi:10.1109/ICASSP.2002.5745011.

[3] Meyer, J. and Agnello, T., "Spherical microphone array for spatial sound recording," *NEW YORK*, p. 9, 2003.

[4] Meyer, J. and Elko, G. W., "Spherical Microphone Arrays for 3D Sound Recording," in Y. Huang and J. Benesty, editors, *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, pp. 67–89, Springer US, Boston, MA, 2004, ISBN 978-1-4020-7769-2, doi:10.1007/1-4020-7769-6_3.

[5] Li, Z. and Duraiswami, R., "Flexible and Optimal Design of Spherical Microphone Arrays for Beamforming," *IEEE Transactions on Audio, Speech, and Language Processing*, 15(2), pp. 702–714, 2007, ISSN 1558-7924, doi:10.1109/TASL.2006.876764.

[6] Rafaely, B., Koretz, A., Winik, R., and Agmon, M., "Spherical microphone array beampattern design for improved room acoustics analysis," 2008.

[7] Yan, S., Sun, H., Svensson, U., Ma, X., and Hovem, J., "Optimal Modal Beamforming for Spherical Microphone Arrays," *Audio, Speech, and Language Processing, IEEE Transactions on*, 19, pp. 361 – 371, 2011, doi:10.1109/TASL.2010.2047815.

[8] *mh acoustics - product catalog*, 2018.

[9] Wittek, H., Faller, C., Favrot, A., Langen, C., and Tournery, C., "Digitally Enhanced Shotgun Microphone with Increased Directivity," in *Audio Engineering Society Convention 129*, 2010.

[10] Benesty, J., Jingdong, C., and Huang, Y., *Microphone Array Signal Processing*, Springer Berlin Heidelberg, 2008, ISBN 978-3-540-78612-2.

[11] Runow, B. and Curdt, O., "Microphone Arrays for professional audio production," in *28th Tonmeistertagung - VDT International Convention*, p. 7, 2014.

[12] Runow, B., Curdt, O., and Schilling, A., "Shotgun Microphones versus Microphone Arrays," in *29th Tonmeistertagung - VDT International Convention*, 2016.

[13] Eargle, J., *From Mono to Stereo to Surround, A Guide to Microphone Design and Application*, Focal Press : [distributor] Elsevier Books Customer Services, Oxford, 2004, ISBN 978-0-240-51961-6, oCLC: 851974436.

[14] Wittek, H., Haut, C., and Keinath, D., "Double M/S – a Surround recording technique put to test," in *Tonmeistertagung*, Verband Deutscher Tonmeister eV, 2006.

[15] Benjamin, E. and Chen, T., "The Native B-Format Microphone," in *Audio Engineering Society Convention 119*, 2005.

[16] Benjamin, E. and Chen, T., "The Native B-Format Microphone: Part II," in *Audio Engineering Society Convention 120*, 2006.

[17] Jarrett, D. P., Habets, E. A. P., and Naylor, P. A., "3D source localization in the spherical harmonic domain using a pseudointensity vector," in *2010 18th European Signal Processing Conference*, pp. 442–446, 2010.

[18] Veen, B. D. V. and Buckley, K. M., "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Magazine*, 5(2), pp. 4–24, 1988, ISSN 0740-7467, doi:10.1109/53.665.

[19] Gerzon, M. A., "The Design of Precisely Coincident Microphone Arrays for Stereo and Surround Sound," in *Audio Engineering Society Convention 50*, 1975.

[20] Freiberger, K., *Development and Evaluation of Source Localization Algorithms for Coincident Microphone Arrays*, Ph.D. thesis, Institute of Electronic Music and Acoustics (IEM), University of Music and Performing Arts, Graz, Austria, 2010.

[21] Brown, R. G., *Smoothing, forecasting and prediction of discrete time series*, Prentice-Hall Englewood Cliffs, N.J, 1963.

[22] Allen, J. B. and Berkley, D. A., "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, 65(4), pp. 943–950, 1979.

[23] Diaz-Guerra, D., Miguel, A., and Beltran, J. R., "gpuRIR: A Python Library for Room Impulse Response Simulation with GPU Acceleration," *arXiv e-prints*, p. arXiv:1810.11359, 2018.

[24] Veaux, C., Yamagishi, J., MacDonald, K., and others, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2017.

[25] Piczak, K. J., "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pp. 1015–1018, ACM, New York, NY, USA, 2015, ISBN 978-1-4503-3459-4, doi:10.1145/2733373.2806390, event-place: Brisbane, Australia.

[26] Hirt, R., *Entwicklung einer virtuellen Konferenz unter besonderer Berücksichtigung der Reproduktion von zuvor aufgenommener Sprache - Development of a virtual conference with focus on optimal reproduction of pre recorded speech.*, Bachelor's Thesis, Stuttgart Media University, 2017.

[27] Paukert, H. and Ziegler, J., "Listening Tests in the Process of Microphone Development," in *29. Tonmeistertagung VdT International Convention*, p. 8, 2016.

[28] Paukert, H., Ziegler, J., and Koch, A., "Hörversuche zur Entwicklung eines neuartigen Mehrkapsel-Mikrofons," in *30th Tonmeistertagung VdT International Convention*, p. 8, 2018.

[29] "MUSHRA : Bs. 1534-1. method for the subjective assessment of intermediate sound quality," 2001.

[30] Taal, C. H., Hendriks, R. C., Heusdens, R., and Jensen, J., "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4214–4217, 2010, doi:10.1109/ICASSP.2010.5495701.

[31] Zwicker, E. and Fastl, H., *Psychoacoustics: Facts and Models*, Springer Series in Information Sciences, Springer Berlin Heidelberg, 2013, ISBN 978-3-662-09562-1.

Audio Engineering Society

Convention Paper 10101

Presented at the 145th Convention
2018 October 17 – 20, New York, NY, USA

Speech Classification for Acoustic Source Localization and Tracking Applications using Convolutional Neural Networks

Jonathan D. Ziegler[1,2], Andreas Koch[1], and Andreas Schilling[2]

[1]*Stuttgart Media University, Institute for Electronic Media, Stuttgart, Germany*
[2]*Eberhard Karls University Tübingen, Visual Computing, Tübingen, Germany*

Correspondence should be addressed to Jonathan D. Ziegler (zieglerj@hdm-stuttgart.de)

ABSTRACT

Acoustic Source Localization and Speaker Tracking are continuously gaining importance in fields such as human computer interaction, hands-free operation of smart home devices and telecommunication. A set-up using a Steered Response Power approach in combination with high-end professional microphone capsules is described, and the initial processing stages for detection angle stabilization are outlined. The resulting localization and tracking can be improved in terms of reactivity and angular stability by introducing a Convolutional Neural Network for signal/noise discrimination tuned to speech detection. Training data augmentation and network architecture are discussed, classification accuracy and the resulting performance boost of the entire system are analyzed.

1 Introduction

For the scenario discussed in this paper, a Steered Response Power (SRP) algorithm, combined with a coincident microphone array is used to track speakers in a conference environment. As SRP is an energy-based detection algorithm, no distinction between a desired signal (i.e. human speech) and undesired interference (i.e. office or traffic noise) can be made. Some improvement in direction of arrival (DOA) estimation can be achieved by applying detection filters[1] prior to the SRP processing, thus only registering energy in a frequency range relevant to human speech. This approach is relatively limited, as many types of noise show a wide frequency range, often overlapping that of speech signals. A more sophisticated sound source discrimination is described using a Convolutional Neural Network (CNN) for sound source classification. Spectral and temporal information is processed by the CNN, using spectrograms of buffers spanning 128 ms and 75 frequency bands, in a frequency range of 200 Hz to 8000 Hz.

[1]The results presented in this paper were obtained using a band-pass detection filter in the range of 200 Hz to 4000 Hz.

2 Methods

2.1 Microphone Array Configuration

The task of Acoustic Source Localization and tracking of a moving acoustical source can be approached in many different ways, the use of linear or circular spaced

arrays being favored in many consumer-grade applications [1]. For audio capturing, the disadvantage of conventional spaced arrays, compared to coincident microphone configurations, is the inferior audio quality of the created beam. Spaced arrays are prone to distorted frequency responses, due to the fact that the created beam patterns are frequency-dependent [2]. Some recent advances have been made, although satisfactory results require an upper frequency limit of 8 kHz [3]. The audio quality of beams created by coincident microphone arrays solely depends on the quality of the microphone capsules used, thus resulting in a more linear frequency response, even with respect to moving beams required for source tracking. However, higher-order beams can not be achieved using first-order coincident arrays [4]. For Machine Listening applications, the requirements regarding audio quality are often relatively low and are defined by the algorithms used. Often a frequency range of 100 Hz to 8000 Hz is chosen. In other cases the bandwidth of telephone conversations (5 Hz to 3700 Hz) is sufficient [5]. Because the array described in this paper is used for audio capturing in conference environments, optimal sound quality is required. Therefore, a configuration consisting of three high-end microphone capsules is chosen. Due to hardware considerations, a Double-M/S configuration is used, consisting of two Schoeps CCM-4 cardioid capsules and a Schoeps CCM-8 figure-of-eight capsule. One cardioid c_f faces 0°, while the other cardioid c_r faces 180° and the figure-of-eight f_8 is positioned facing ±90°.

2.2 Acoustic Source Localization

From the Double-M/S configuration, a horizontal Ambisonics B-format can be decoded [6]:

$$W = c_f + c_r \tag{1}$$
$$X = c_f - c_r \tag{2}$$
$$Y = f_8 \tag{3}$$

Using the WXY-decoded signals, any arbitrary first-order microphone pattern $M(\theta, p)$ can be synthesized on the horizontal plane [7, 8]:

$$M(\theta, p) = pW + (1-p)\left(X\cos\theta + Y\sin\theta\right), \tag{4}$$

with p representing the polar pattern shape between $p = 0$ (figure-of-eight) and $p = 1$ (omnidirectional), and θ describing the orientation on the horizontal plane.

Using (4), n_M virtual cardioid microphone signals[2] can be synthesized. The virtual microphone with the highest relative RMS level indicates the Direction of Arrival of the sound source θ_{DOA}:

$$\theta_{DOA} = \underset{\theta_i}{arg\,max}\left(\overline{M}(\theta_i, p = 0.5)\right), i = 1, ..., n_M. \tag{5}$$

Under certain conditions reflected sound can surpass the original source in sonic energy. Currently no scenarios have been recorded in which a significant performance decrease could be attributed to false DOA detection due to reflections.

2.3 Confidence Weighting

Building on the SRP maximization described in section 2.2, additional angular stabilization is applied. This is achieved using exponential smoothing [9]:

$$s_t = \alpha\, x_t + (1-\alpha)\, s_{t-1}, \tag{6}$$

with x_t and s_t representing the input and smoothed output angle for time frame t. 2π-wrapping of the angle is addressed in a separate function.

Using the smoothing factor $\alpha \in [0, 1]$ creates a static smoothing effect which does not reflect any characteristics of the processed signal buffer. To achieve variable smoothing, the coefficient α is dynamically assigned, depending on a set of signal quality metrics. In the following paragraphs, this will be called confidence weighting, which consists of four types of confidence indeces C:

- Directivity weighting C_d – the level of anisotropy of the detected sound indicates whether an actual sound event is detected.
- Level weighting C_l – if a buffer contains a low relative sound level, no relevant sound events are expected.
- Long-term weighting C_{lt} – if sound events have frequently been detected from a direction, a quasi-static sound source such as a speaker at a table can be assumed.

[2] $p = 0.5$

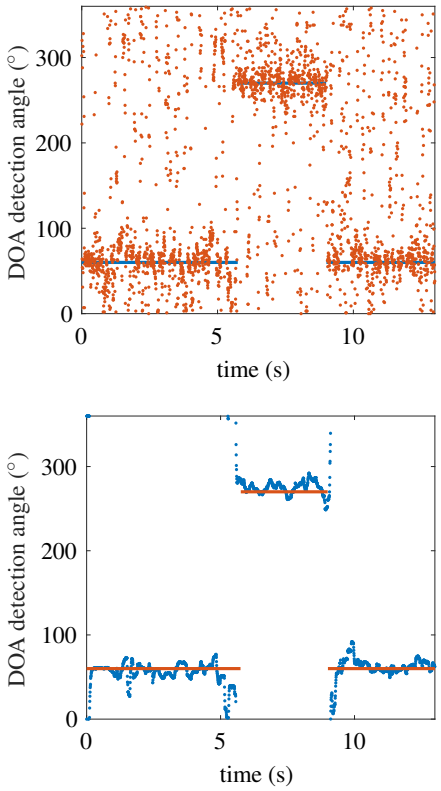


Fig. 1: Comparison of tracker output with and without angular stabilization. The average error can be reduced from 25.68 % to 6.46 % when using a dynamic smoothing coefficient α. Solid lines represent reference position.

- Speech detection C_s– if a buffer is not classified as speech, no relevant sound events are expected.

The last contribution to the confidence index is determined using a Convolutional Neural Network (CNN), trained to discriminate between speech and non-speech.

The confidence indeces are combined to create the dynamic weighting factor α:

$$\alpha = \left(\kappa C_d + (1-\kappa) C_d C_{lt} C_s^2\right) C_l\,, \qquad (7)$$

using the empirically determined mixing factor κ.

Angular stabilization is essential for this type of acoustic source localization. Figure 1 shows a comparison of the tracker output with and without stabilization.

2.4 Mel-Scale Spectral Analysis

Convolutional Neural Networks provide excellent processing capabilities on two-dimensional arrays, such as images. For training and classification, the audio stream is processed via Fourier Transform to Log-Mel-scale spectrograms, using the feature extraction toolbox provided by the University of Oldenburg [10]. The Mel scale is used, since it closely resembles human perception of sound and has proven effective in combination with Neural Networks for audio classification and speech detection [11, 12]. Buffers of 2048 samples are analyzed at $Fs = 16\,\text{kHz}$ sampling rate[3], resulting in 128 ms of audio per buffer. The spectral transform is performed using a window size of 28 ms, which is successively shifted by 10 ms. The processed frequency range is between 200 Hz and 8 kHz, divided into 75 Mel-bands. Examples of extracted spectrograms can be seen in Figure 3.

2.5 Neural Network Architecture

As the entire signal processing chain was created in MATLAB, the use of MATLAB's Neural Network Toolbox for the speech detector ensures a seamless integration and easy fine-tuning of the processing.

The processing steps presented in section 2.3 and 2.4 output Log-Mel-scale spectrograms of the dimension 75x11x1. These define the dimensions of the input layer of the CNN. Two-dimensional convolution is applied, using 8 5x5 filter matrices and zero-padding to maintain the input layer dimension ("same" padding) [13, 14]. The convolution output is then shrunk by choosing the maximum value of every 2x2 subset. This operation is called Max-pooling with a pool size of 2 and a stride of 2, and is used to transform the matrix to a dimension of 38x6x8[4]. The next convolution operation uses 16 3x3 filters and *same* padding. Combined with a max-pooling operation with a pooling size and stride of 2, the dimensions are transformed to 19x3x16. The last convolution uses 32 3x3 filters and *same* padding, resulting in 1824 inputs for the first fully connected layer, which outputs

[3] The entire tracking algorithm runs at 48 kHz. The decision to down-sample by a factor of 3 is the result of the data set used for augmentation, as described in section 2.6, and the chosen frequency range with an upper limit of 8 kHz.

[4] To achieve the desired output dimensions, max-pooling is padded with $p_{bottom} = 1$ and $p_{right} = 1$. Details are discussed in section 2.7.

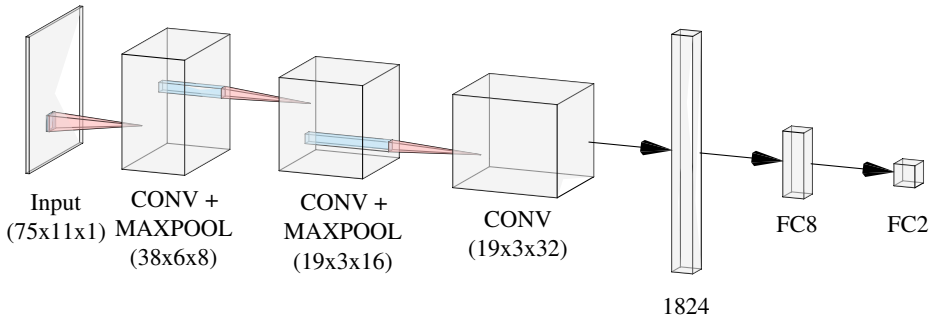


Fig. 2: Architecture of the neural network used as speech detector. The input Mel spectrogram measures 75x11x1 pixels. Using three convolution layers, two max-pool layers and two fully connected layers, validation accuracy is 91.23 %.

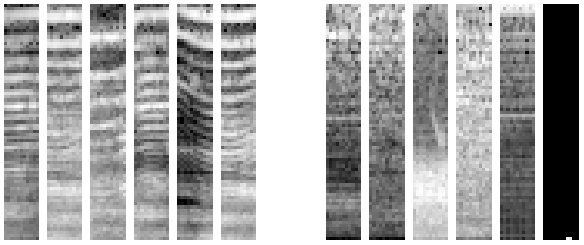


Fig. 3: Log-Mel-scale spectra created for training and classification using a CNN. 128 ms of audio are processed using 75 Mel-bands, spanning 200 Hz to 8000 Hz. The resulting spectrograms are displayed as gray-scale images of 75x11 pixels. *Left:* spectrograms of audio buffers containing speech. *Right:* spectrograms of buffers without speech.

8 activations for the final layer. This layer, using a softmax activation function, discriminates between *speech* and *non-speech*. An Adam optimization algorithm was used to train the network [15].

2.6 Training Data and Augmentation

The network was trained using 30866 speech spectra and 29360 noise spectra. The validation set consisted of 2×2822 labeled samples. Lacking sufficient training data, the dataset was augmented using the Musan dataset [16]. Within the dataset, the Librivox speech files and the Free-Sound noise samples were used. To create training data more similar to the test data, the relatively direct recordings of the dataset needed to be placed in virtual rooms. Room impulse responses (RIR) were created using the image method described by Allen and Berkley [17], implemented in the RIR-generator, provided by the International Audio Laboratories Erlangen [18]. To prevent the Neural Network from overfitting to a specific room dimension, random room dimensions were chosen to create impulse responses of virtual rooms similar in size to a potential application environment. For every audio file of the dataset, room dimensions were varied from 2 m to 7 m. Within these randomly chosen room dimensions, the sound-source and sound-detector were randomly positioned. Once the RIR was created, a convolution with the audio file from the dataset created a reverberant version of the file. This reverberant audio was then divided into frames of 2048 samples and transformed into the Log-Mel-spectrograms described in section 2.4. The validation set consisted of 2822 spectrograms for *speech* and *non-speech*, respectively. The spectrograms labeled *speech* in the validation set were obtained from recordings of the virtual conference described in section 3, using speech-only scenarios. To obtain the maximum possible number of spectrograms from the recordings, all individual microphone streams, as well as the combined omnidirectional and virtual cardioid signals, were analyzed individually and used for training and cross validation.

2.7 Real-Time Classification

The spectral analysis described in section 2.4 requires 128 ms of audio per spectrogram. With the main tracking algorithm running at 48 kHz, this is equivalent to 6144 samples. To maintain the low-latency operation of the tracking algorithm, which runs at 256 samples, classification is performed on the current audio buffer, in combination with the 23 previous buffers. To give the

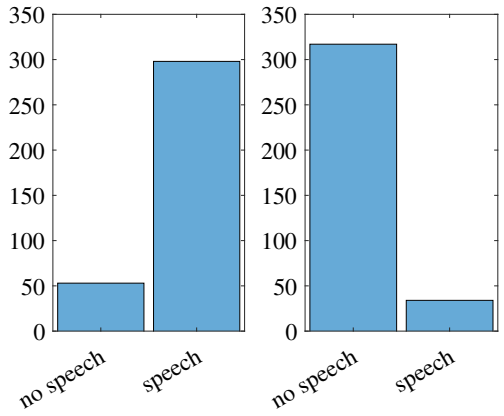


Fig. 4: Results of the decomposed test files. *Left:* Only analyzing the clean speech file results in 298 *speech* classifications and 53 *non-speech* classifications. *Right*: Analyzing the noise components returns 317 *non-speech* classifications and 34 *speech* classifications. The overall accuracy in this test case is 87.61 %.

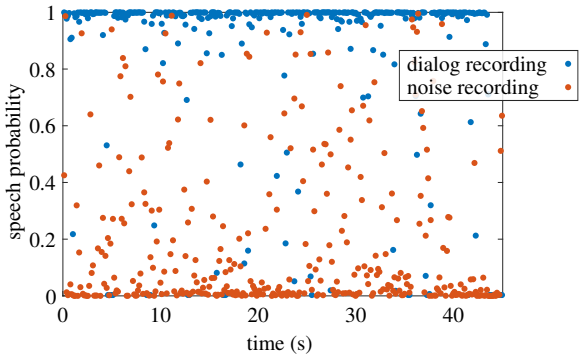


Fig. 5: Probability of a buffer containing speech, analyzed over the entire test file. As in Figure 4, the file was decomposed in *speech* and *non-speech* components which were analyzed individually.

current buffer a higher weight, the first max-pooling layer is padded only on the right, thus reducing the importance of the left-most (oldest) part of the spectrogram. The choice of audio stream for the classification operation is still under investigation. The most promising choices are a virtual omnidirectional microphone

$$S_o = W \tag{8}$$

and a virtual supercardioid microphone signal S_ρ ($p = 0.34$), aimed at the detected direction of the previous buffer θ_{DOA}:

$$S_\rho = 0.34 \cdot W + 0.66 \cdot (X \cos\theta_{DOA} + Y \sin\theta_{DOA}). \tag{9}$$

The results presented in section 3 were gathered using S_o. Current measurements show no performance gain[5] when using the computationally more expensive S_ρ.

3 Results

The tracker performance was evaluated using a multichannel playback system, reproducing a virtual conference scenario. The set-up was placed within a large, acoustically untreated room[6] with 8 loudspeakers arranged in two concentric rings of 1.5 m and 2.5 m around the microphone array. One additional loudspeaker was placed at 0.5 m distance from the array, another at 4 m. Microphone and loudspeaker hight were chosen to realistically match a real-world scenario. While the microphone and close-range loudspeaker were placed at the hight of a table-top (800 mm and 700 mm, respectively), the loudspeakers placed at 1.5 m and 2.5 m distance were set to the height of the mouth of a seated person (1240 mm and 1390 mm, respectively). The distant loudspeaker was placed at the approximate height of the mouth of a standing speaker (1700 mm). All heights were measured from the center of the tweeter. A prepared scenario was played back[7], consisting of male and female speech in German and English. Additionally, a variety of non-speech signals were played back, such as cell phone ring-tones, moving chairs, et cetera, combined with recordings of construction sites and office noise. The recording and playback format of the multichannel noise recordings were chosen to be identical. Two scenarios were played back, once exclusively using speech signals, once containing additional noise. The introduction of speech

[5] Measured performance gain was < 0.1 %.
[6] 8.3 m × 8.2 m × 3.8 m, $RT60 \approx 2.3$ s.
[7] Recording and playback format: 48 kHz, 24 Bit.

	$S1$	$S1_{noise}$	$S2$	$S2_{noise}$
accuracy gain	−1.2 %	−1.8 %	−0.4 %	−6.6 %
stability gain	1.2 %	5.4 %	2.4 %	4.2 %

Table 1: Measured performance gain when using speech classification as part of the angular stabilization process.

detection decreased the average accuracy by 2.5 % and increased the angular stability by 3.3 %. Table 1 shows increased smoothing especially in noisy environments. To further evaluate the classifier performance, the multichannel scenario was split into speech and non-speech components and rendered to mono-files. The classifications throughout the speech and non-speech files can be seen in Figures 4 and 5. The sum test accuracy in this case is 84.61 %. Because the split was performed on the near-anechoic scenario without being played back in the virtual conference environment, the comparison is skewed, with both real-time application and training being performed on reverberant signals. Additional testing is described in section 4. Within the tracking scenario, the addition of the CNN classifier results in a performance boost. Increased angular stability during speech, combined with less erratic movement in periods without speech, improve the audio quality of beamforming algorithms being driven by the tracked position data. A beamformer tuned to the signals of the array in use is described by Runow et al. [19]. Figure 6 shows the classifier-induced performance boost for a virtual conference recording. Close sources with a high signal to noise ratio can be tracked well without the need of speech classification. During the second half of the recording, the sources are played back on the mid-range[8] and far-range[9] loudspeakers, with a larger amount of ambient noise. Here, the discrimination between *speech* and *non-speech* (desired signal and noise) increases tracking stability. Since this test was designed for general tracking performance evaluation, and not for speech classification evaluation, additional testing will be required to better assess the added confidence factor. In real-time tests, a clear improvement of speaker tracking can be observed, with office and traffic noise, as well as structural vibration being rejected well beyond the level achieved when using only the detection filter described in section 1.

[8] $r = 2.5\,\mathrm{m}$
[9] $r = 4\,\mathrm{m}$

4 Discussion

Because the available test data were not recorded for the specific purpose of evaluating speech detection, additional testing is needed to assess the full benefit of the added confidence weighting. Initial real-time tests indicate a considerable performance boost; quantitative measurements are the next step. With the small amount of training data requiring additional synthetic data, the training and validation sets do not come from the same data distribution. This is not ideal, but could not be prevented without recording and labeling large amounts of additional data. To ensure satisfactory generalization of the trained net within the intended application, most of the recorded data was used for cross validation. Initial training of the CNN indicated overfitting, which has been countered with the use of stronger L2-regularization [20]. This suggests that test accuracy will profit from additional training data recorded in environments more similar to the final application. The test environment used for evaluation was considerably larger than the virtual rooms used for data augmentation, which were chosen to be closer to the final application environment. A performance gain is expected when using more realistic surroundings for further testing. If the desired increase in performance is not observed, additional training rounds will contain a larger variety of virtual spaces.

5 Summary

A system for Acoustic Source Localization and Tracking is described, which is capable of locating and tracking speech sources in real-time. The main system is set up using an algorithmic approach, with Steered Response Power maximization as the direction-of-arrival estimator and a series of weighting factors for variable exponential smoothing of the detected angle. Additionally, a Convolutional Neural Network is used for speech detection. Discrimination between *speech* and *non-speech* events enables the system to effectively reject sound sources which are not of relevance for the the application of speaker tracking, increasing the performance beyond that of the purely algorithmic approach. Initial tests show high classification accuracy within the final application, and additional data promise still higher accuracy.

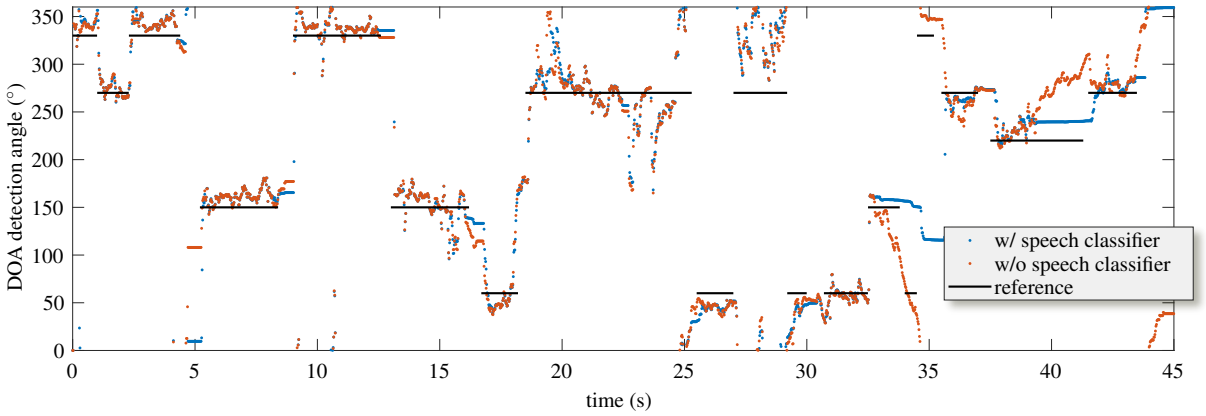


Fig. 6: Output of tracking algorithm with and without CNN speech detection. The first half of the test file is played back at a distance of $r = 1.5\,\text{m}$ around the microphone array. At this distance, confidence weighting works well and the speech detection has a negligible impact on performance. Towards the end of the sample, playback distance is increased to a distance of 2.5 m to 4 m around the microphone array and the SNR is reduced. The increased levels of non-directional reverberation and noise components considerably reduce the tracker's performance. Using the speech detector, a higher level of stability can be maintained.

Acknowledgments

This research was in part funded by the *Zentrales Innovationsprogramm Mittelstand*, a grant of the *Bundesministerium für Wirtschaft und Energie*, Germany.

This work was supported by *Kooperatives Promotionskolleg Digital Media* at Stuttgart Media University and the University of Tübingen.

References

[1] Reinette, A., Cornejo, M., Rouchon, C., and Fester, M., "Benchmarking Microphone Arrays: ReSpeaker, Conexant, MicroSemi AcuEdge, Matrix Creator, MiniDSP, PlayStation Eye," *Snips Labs*, 2017.

[2] Benesty, J., Jingdong, C., and Huang, Y., *Microphone Array Signal Processing*, Springer Berlin Heidelberg, 2008, ISBN 978-3-540-78612-2.

[3] Delikaris-Manias, S., Valagiannopoulos, C. A., and Pulkki, V., "Optimal directional pattern design utilizing arbitrary microphone arrays: A continuous-wave approach," in *Audio Engineering Society Convention 134*, Audio Engineering Society, 2013.

[4] Benesty, J. and Jingdong, C., *Study and Design of Differential Microphone Arrays (Springer Topics in Signal Processing)*, Springer, 2012, ISBN 364233752X.

[5] Gruhn, R. E., Minker, W., and Nakamura, S., *Automatic Speech Recognition*, pp. 5–17, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, ISBN 978-3-642-19586-0, doi:10.1007/978-3-642-19586-0_2.

[6] Wittek, H., Haut, C., and Keinath, D., "Double M/S – a Surround recording technique put to test," in *Tonmeistertagung*, Verband Deutscher Tonmeister eV, 2006.

[7] Gerzon, M. A., "The Design of Precisely Coincident Microphone Arrays for Stereo and Surround Sound," in *Audio Engineering Society Convention 50*, 1975.

[8] Freiberger, K., *Development and Evaluation of Source Localization Algorithms for Coincident Microphone Arrays*, diploma thesis, 2010.

[9] Brown, R. G., *Smoothing, forecasting and prediction of discrete time series*, Prentice-Hall Englewood Cliffs, N.J, 1963.

[10] Schädler, M. R., "Reference Matlab/Octave implementations of feature extraction algorithms," 2015, Carl von Ossietzky Universität Oldenburg, Department für Medizinische Physik und Akustik.

[11] Piczak, K. J., "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2015, ISSN 1551-2541, doi:10.1109/MLSP.2015.7324337.

[12] Hinton, G., Deng, L., Yu, D., Dahl, G. E., r. Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B., "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, 29(6), pp. 82–97, 2012, ISSN 1053-5888, doi:10.1109/MSP.2012.2205597.

[13] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016, `http://www.deeplearningbook.org`.

[14] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86(11), pp. 2278–2324, 1998.

[15] Kingma, D. P. and Ba, J., "Adam: A Method for Stochastic Optimization," *CoRR*, abs/1412.6980, 2014.

[16] Snyder, D., Chen, G., and Povey, D., "MUSAN: A Music, Speech, and Noise Corpus," *CoRR*, abs/1510.08484, 2015.

[17] Allen, J. B. and Berkley, D. A., "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, 65(4), pp. 943–950, 1979, doi:10.1121/1.382599.

[18] Habets, E., "RIR Generator," 2018, International Audio Laboratories Erlangen.

[19] Runow, B., Schilling, A., and Curdt, O., "Störgeräuschreduktion mit einer Mel-Filterbank in Verbindung mit koinzidenten Mikrofonarrays," *29. Tonmeistertagung des Verbandes Deutscher Tonmeister*, 2016.

[20] Schmidhuber, J., "Deep Learning in Neural Networks: An Overview," *CoRR*, abs/1404.7828, 2014.

Audio Engineering Society

Convention Paper 9877

Presented at the 143rd Convention
2017 October 18–21, New York, NY, USA

Interpolation and Display of Microphone Directivity Measurements using higher-order Spherical Harmonics

Jonathan D. Ziegler[1,3], Mark Rau[2], Andreas Schilling[3], and Andreas Koch[1]

[1]*Stuttgart Media University, Institute for Electronic Media, Stuttgart, Germany*
[2]*Stanford University, CCRMA, Department of Music, Stanford, CA, USA*
[3]*Eberhard Karls University Tübingen, Visual Computing, Tübingen, Germany*

Correspondence should be addressed to Jonathan D. Ziegler (zieglerj@hdm-stuttgart.de)

ABSTRACT

The accurate display of frequency dependent polar response data of microphones has largely relied on the use of a defined set of test frequencies and a simple overlay of two-dimensional plots. In recent work, a novel approach to digital displays without fixed frequency points was introduced. Building on this, an enhanced interpolation algorithm is presented, using higher-order spherical harmonics for angular interpolation. The presented approach is compared to conventional interpolation methods in terms of computational cost and accuracy. In addition, a three-dimensional data processing prototype for the creation of interactive, frequency-dependent, three-dimensional microphone directivity plots is presented.

1 Introduction

Traditional displays of directional microphone sensitivity provide a limited insight into the frequency-dependent directivity characteristics. The use of defined test frequencies, multiple measurement overlays, and the restriction to two dimensions reduces the amount of information that can be obtained from such figures. As an improvement, the authors suggested a software-based display with a non-fixed frequency point. Using this, an interactive display of the directivity properties of microphones and coincident arrays can be created [1]. One crucial element of data processing for this application is the angular interpolation. This paper focuses on the use of spherical harmonic interpolation (SHI) for this task. Both speed and accuracy are compared to the performance of traditional 3^{rd}- order spline interpolation. In an evaluation using measured data, depending on the order of SHI, the interpolation speed and accuracy outperformed traditional spline interpolation. In addition, the simplicity of adaptation to three-dimensional measurements is shown on simulated measurement data.

2 Methods

2.1 Cubic Spline Interpolation

The angular resolution of measurement data can be increased by creating virtual measurement points. This

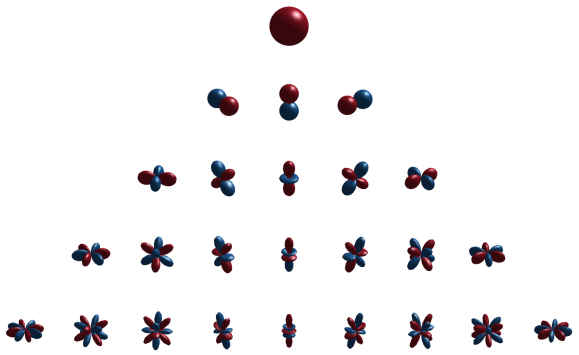


Fig. 1: Spherical harmonics with $n \leq 4$. Only components relevant to the xy plane are used for two-dimensional interpolation.

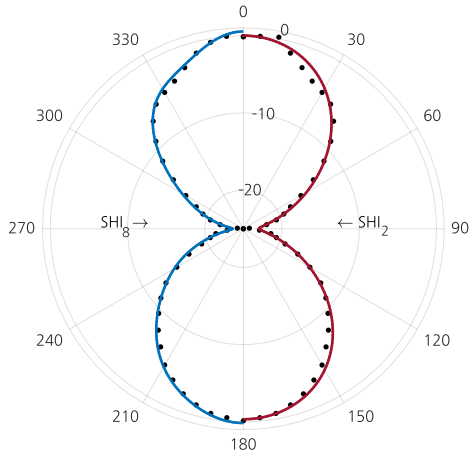


Fig. 2: Comparison of spherical harmonic interpolation computed with order limits of 2 and 8. While $n \leq 2$ provides a smoother angular response, $n \leq 8$ retains a higher level of detail. Measurement data: Schoeps MK8 at 10 kHz

was formerly achieved using cubic spline interpolation, which uses third order polynomials within every interval between measurement points $[S_i(\nu), S_{i+1}(\nu)]$ with $i = 0, 1 \ldots$ [2, 3].

Considering the i^{th} spline interval S_i, the interpolation function takes the form:

$$S_i(\tau) = a_i + b_i\,\tau + c_i\,\tau^2 + d_i\,\tau^3 \tag{1}$$

with $0 \leq \tau \leq 1$. By defining a set of boundary conditions appropriate to the system's physical behavior, it is possible to solve for all variables a_i, b_i, c_i and d_i in every interval i and at all frequencies ν.

2.2 Spherical Harmonic Interpolation (SHI)

A more elegant approach uses spherical harmonics for this task. This set of orthogonal base functions defined on the surface of a sphere can be expressed as

$$Y_n^m(\theta,\phi) = \sqrt{\frac{2n+1}{4\pi}\frac{(n-m)!}{(n+m)!}}P_n^m(\cos\theta)e^{im\phi}, \tag{2}$$

where $P_n^m(\cdot)$ are the associated Legendre functions, m is an integer representing the function degree, and n is a natural number representing the function order [4]. The associated Legendre functions are derived by differentiating the Legendre polynomials and are given as

$$P_n^m(x) = (-1)^m(1-x^2)^{m/2}\frac{d^m}{dx^m}P_n(x),\ x \in [-1,1], \tag{3}$$

with $P_n(x)$ representing the Legendre polynomials which arise when $m = 0$. They are defined as

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}(x^2-1)^n. \tag{4}$$

Spherical harmonics have the useful property that any arbitrary function on a sphere $f(\theta,\phi)$ can be be represented as

$$f(\theta,\phi) = \sum_{n=0}^{\infty}\sum_{m=-n}^{n} f_{nm}Y_n^m(\theta,\phi), \tag{5}$$

with f_{nm} being the function weights defined as

$$f_{nm} = \int_0^{2\pi}\int_0^{\pi} f(\theta,\phi)[Y_n^m(\theta,\phi)]^* \sin\theta\, d\theta\, d\phi. \tag{6}$$

The weights form what is known as the spherical Fourier transform, while equation (5) is the inverse spherical Fourier transform [4, 5].

Using equations 5 and 6, a spherical harmonic data interpolation method can be devised. Measurement data are transformed via spherical Fourier transform

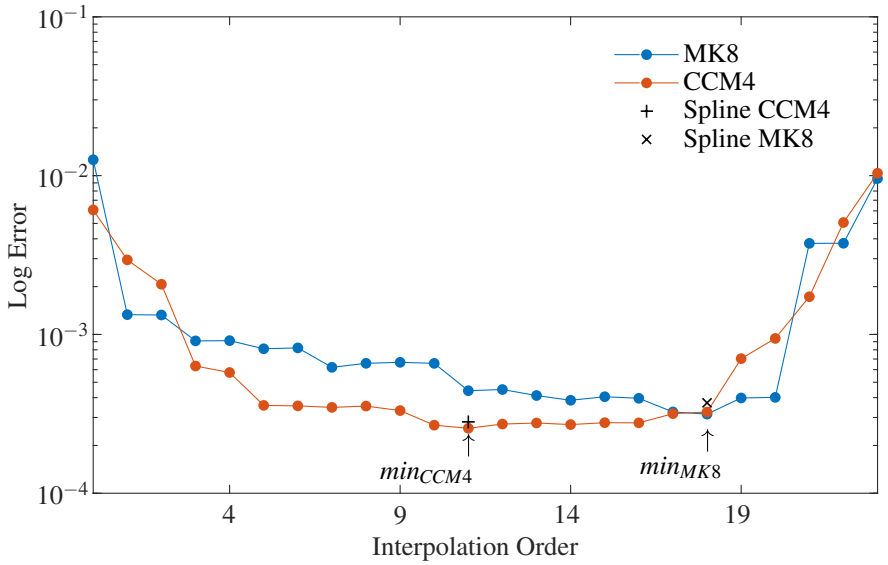


Fig. 3: Comparison of algorithm accuracy. Interpolated measurement points from a down-sampled dataset are compared to high-resolution measurements. The error is computed as the sum of absolute errors over 72 points on 360°. Due to the small change in error over a large range of SHI orders, a logarithmic display is chosen. At the indicated minima, SHI outperforms cubic spline interpolation by approximately 0.25 dB for the Schoeps CCM4 cardioid capsule and by approximately 0.5 dB for the Schoeps MK8 figure-of-eight capsule used for the measurements.

onto a base of spherical harmonic functions $Y_n^m(\theta_j, \phi_k)$, sampled on a grid of dimension $j \times k$, matching the resolution of the measurement data. Later, an inverse spherical Fourier transform onto a grid with a higher spatial resolution results in the desired discrete angular interpolation. Since the spherical harmonic base functions are continuous, the discrete resolution of the interpolated data depends on the grid for the inverse spherical Fourier transform and therefore can be varied.

3 Results

The use of spherical Fourier transforms for data interpolation creates an effective approach to angular smoothing within the application described in section 4. Lower-order transforms provide the capability to retrieve the basic microphone directivity characteristics with computational efficiency, while higher-order transforms outperform the traditional spline methods in terms of accuracy.

All basic microphone polar patterns inherent to pressure sensors and pressure-gradient sensors can be described using an omnidirectional sphere and a bidirectional

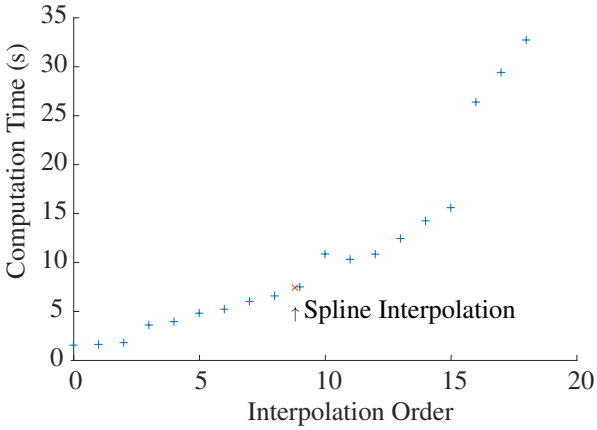


Fig. 4: Computational cost of interpolation algorithms. 24 Measurement points with 20000 frequency bins each were processed. Up to $n = 8$, spherical harmonic interpolation is faster than spline interpolation.

figure-of-eight [6, chapter 5]. Hence, combinations of spherical harmonics with $n \leq 1$ are sufficient. Adding higher order spherical harmonics subsequently adds additional information about the measured microphone response. Figure 1 shows the first 5 orders of spherical harmonics ($0 \leq n \leq 4$). It is apparent that $n = 0$ is analogous to omnidirectional microphone characteristics, while $n = 1$ produces functions in clear relation to figure-of-eight microphone polar patterns with orthogonal spatial orientation.

Figure 2 shows data interpolation using spherical harmonics with $n \leq 2$ and $n \leq 8$. The measurements were performed on a Schoeps MK8 figure-of-eight capsule sampled at 37 points between 0° and 180° along the horizontal plane, resulting in an angular resolution of 5°. For ease of display and assuming rotational symmetry in the MK8's polar pattern, the 180° measurement was expanded to a full circle.

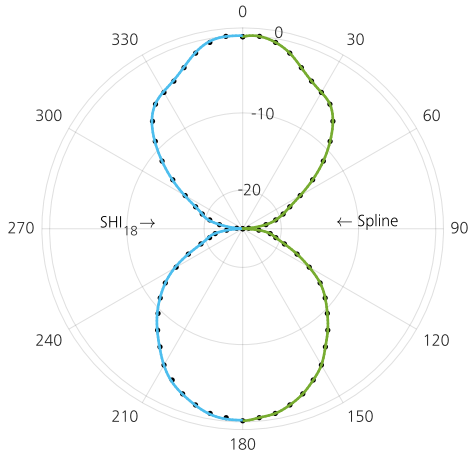


Fig. 5: Comparison of higher-order ($n \leq 18$) spherical harmonic interpolation and cubic spline interpolation.

3.1 Performance

The proposed algorithms are currently computed within Mathworks' Matlab®, using the AKtools toolbox [7]. With this setup, the processing time for data interpolation was inspected on a dataset with 24 measurement points ($\Delta\theta = 15°$) and 20000 frequency bins. Figure 4 shows that for the presented case, spherical harmonic interpolation provides faster results than spline interpolation up to an order of $n = 8$.

3.2 Accuracy

To compare the quality of interpolated data, a set of measurements was down-sampled by a factor of 3, going from $\Delta\theta = 5°$ to $\Delta\theta = 15°$. After data interpolation, the difference between interpolated data points and actual omitted measurement points was calculated. Figure 3 shows the resulting error values for different orders of spherical harmonic interpolation, compared to the error of spline interpolation. Both cardioid and figure-of-eight characteristics can be interpolated to a high level of accuracy with surprisingly low orders of interpolation. Taking the logarithmic nature of Figure 3 into account, acceptable results are achieved with orders as low as 3. This is in part due to the very rough sampling of only 24 points. Figures 2 and 5 show that with higher measurement resolution, higher order interpolation is advisable. Figure 3 also shows that, assuming maximum-order SHI as defined in section 3.3, cubic spline interpolation is outperformed by spherical harmonic interpolation. For the CCM4 capsule, SHI at 10 kHz results in approximately 0.25 dB less total error, for the MK8 capsule, the difference amounts to approximately 0.5 dB.

3.3 Aliasing

There are multiple ways to sample points on a sphere but the choice is often dependent on the measurement apparatus. Two common methods are Equal Angle Sampling which samples a sphere at uniformly-spaced angular positions, and Gaussian Sampling which samples the sphere with evenly spaced angles along the sphere [4]. Equal Angle Sampling requires $4(n+1)^2$ samples, where n is the desired order of spherical harmonics, while Gaussian Sampling only requires $2(n+1)^2$ samples. This study uses equal sampling along the azimuthal angle, so Gaussian sampling is used and $2(n+1)$ equal-angle samples are required along the azimuthal angle. Originally, measurements were taken at 5° along the azimuth. After extrapolation to 360° and the removal of duplicate measurement locations at 0° / 360° and 180°, 72 measurement points remain, resulting in a maximum spherical harmonic order of $n = 35$. When the data are down-sampled to 15° angles for verification, the maximum spherical harmonic order becomes $n = 11$. If interpolation is performed at a higher order than the maximum order defined by the sampling rate, aliasing can occur. An example of possible aliasing is shown in Figure 6.

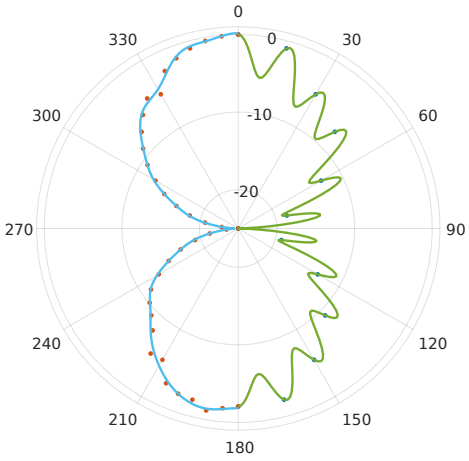


Fig. 6: Aliasing effects due to interpolation above the sampling limit. *Left:* Dataset with 72 measurement points in 360°, interpolated with $n \leq 23$. *Right:* Dataset with 24 measurement points in 360°, interpolated with $n \leq 23$. Choosing interpolation orders above the sampling limit introduces unwanted oscillations in the interpolated data. The higher the order, the more drastic the oscillations.

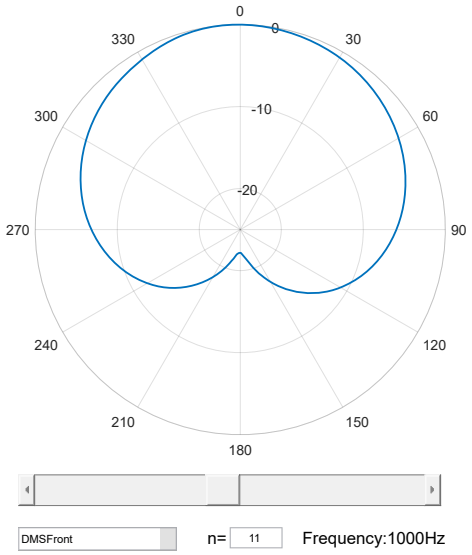


Fig. 7: Example application for SHI methods. Software prototype interactively displaying frequency dependent microphone polar data with variable angular smoothing.

4 Application

Currently the primary use for the developed approach is within a software prototype for the interactive display of frequency dependent microphone polar patterns [1]. Building on this prototype, spherical harmonic interpolation enables the user to adjust the amount of angular smoothing applied to the data. Figure 7 shows measurement data of a Schoeps CCM4 cardioid capsule being displayed at 1000 Hz with an interpolation order of 11. The original measurements were gathered with an angular resolution of 15°, therefore $n \leq 11$ is the highest order of interpolation below the aliasing threshold. Expanding the software to three-dimensional balloon plots is easily achieved by expanding the grids for the spherical Fourier transform and the inverse transform to a 3-D system. This is discussed in the following section. The multidimensional display of transducer measurement data is common practice for loudspeaker measurements and can be achieved using various approaches, with contour and balloon plots being the most prominent [8, 9]. In the context of microphone characterization, this is less common.

5 Outlook

As described by Angus and Evans [10], SHI can be used to interpolate three dimensional measurements of transducer behavior. Lacking sufficient measurement data, the two-dimensional set used for Figures 2, 5, and 6 was extrapolated to a three-dimensional system. Added noise was applied to create a dataset with imperfect rotational symmetry. Figure 8 shows the raw data, alongside interpolations using $n \leq 7$ and $n \leq 17$. Future investigations will be focused on the acquisition and processing of three dimensional microphone characteristics.

6 Summary

In the context of an interactive method for the frequency-dependent display of microphone directivity measurements, spherical harmonic interpolation is introduced. The computational cost of the operation is compared to that of the more traditional and less application-specific approach of cubic spline interpolation. Within the used environment, SHI can be shown to be the faster processing method when interpolating at lower orders. In addition, the accuracy of the mentioned interpolation methods are compared by

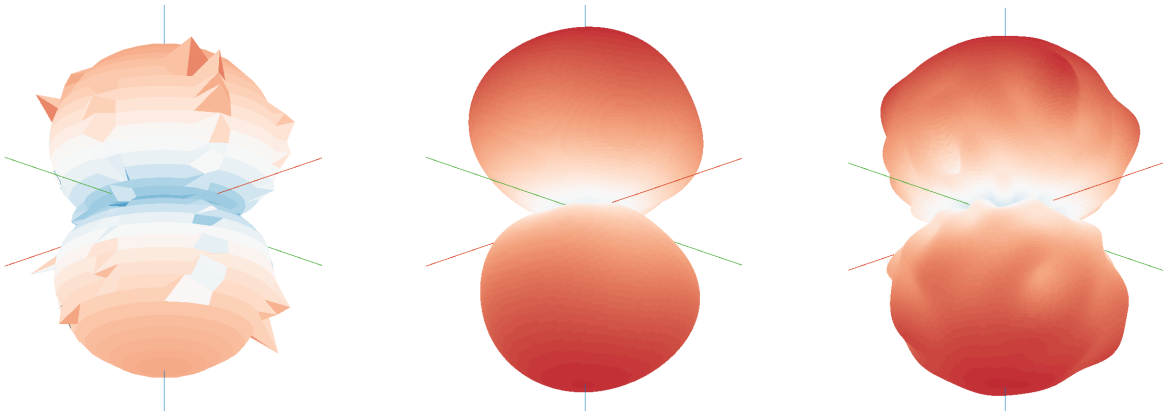


Fig. 8: Three-dimensional SHI demonstrated on simulated measurement data. The planar measurement of a Schoeps MK8 figure-of-eight capsule is expanded, making use of the inherent rotational symmetry of such microphones. Later, this symmetry is partially broken by randomly scaling some of the impulse responses using a normal distribution with $\mu = 1, \sigma = 0.5$. This synthesized dataset is interpolated using SHI_7 and SHI_{17}.

omitting data from a measurement and comparing the algorithmically synthesized data with actual measurements. Based on this comparison, it is possible to show that SHI outperforms cubic spline interpolation when the interpolation order is chosen close to the aliasing limit described in section 3.3. As a proof of principle, three-dimensional SHI for microphone patterns is demonstrated on a semi-synthesized dataset consisting of planar measurement data and noise.

References

[1] Ziegler, J. D., Paukert, H., and Runow, B., "Interactive Display of Microphone Polarity Patterns with Non-Fixed Frequency Point," in *Audio Engineering Society Convention 142*, 2017.

[2] Bartels, R. H., Beatty, J. C., and Barsky, B. A., *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling (The Morgan Kaufmann Series in Computer Graphics)*, Morgan Kaufmann Pub, 1987, ISBN 0934613273.

[3] Weisstein, E. W., "Cubic Spline." *MathWorld – A Wolfram Web Resource*, 2017.

[4] Rafaely, B., *Fundamentals of spherical array processing*, volume 8, Springer, 2015.

[5] Williams, E. G., *Fourier acoustics: sound radiation and nearfield acoustical holography*, Academic press, 1999.

[6] Eargle, J., *Eargle's The Microphone Book: From Mono to Stereo to Surround - A Guide to Microphone Design and Application (Audio Engineering Society Presents)*, Focal Press, 2004, ISBN 0240519612.

[7] Brinkmann, F. and Weinzierl, S., "AKtools - An Open Software Toolbox for Signal Acquisition, Processing, and Inspection in Acoustics," in *Audio Engineering Society Convention 142*, 2017.

[8] Sridhar, R., Tylka, J. G., and Choueiri, E., "Metrics for Constant Directivity," in *Audio Engineering Society Convention 140*, 2016.

[9] Klippel, W. and Bellmann, C., "Holographic Nearfield Measurement of Loudspeaker Directivity," in *Audio Engineering Society Convention 141*, 2016.

[10] Angus, J. A. S. and Evans, M. J., "Polar Pattern Measurement and Representation with Surface Spherical Harmonics," in *Audio Engineering Society Convention 104*, 1998.

Audio Engineering Society

Convention Paper 9793

Presented at the 142nd Convention
2017 May 20–23, Berlin, Germany

Interactive Display of Microphone Polarity Patterns with non-fixed Frequency Point

Jonathan D. Ziegler[1,2], Hendrik Paukert[1], and Bernfried Runow[1,2]

[1] *Stuttgart Media University, Institute for Electronic Media, Nobelstr. 10, 70569 Stuttgart, Germany*
[2] *University of Tübingen, Visual Computing, Sand 14, 72076 Tübingen, Germany*

Correspondence should be addressed to Jonathan D. Ziegler (zieglerj@hdm-stuttgart.de)

ABSTRACT

With the development of bidirectional and unidirectional microphones dating back to the 1930's, the parameter of directivity has been an integral aspect of microphone construction for nearly 100 years [1]. This characteristic is commonly visualized with the microphone's sensitivity displayed as a radius *r* over a 360 - degree span within a polar coordinate system. Measured directivity is generally shown as an overlay of well-defined frequencies [2]. Although this is common practice, in-depth analysis of the actual performance of a microphone is difficult. In this paper, a novel approach to displaying the directional characteristics of a microphone is presented, providing an interactive display of the angular sensitivity at any frequency. Furthermore, the application within microphone array development is discussed.

1 Introduction

The directional sensitivity of a microphone is traditionally displayed as a theoretical plot within a polar coordinate system (polar plot). As shown in Figure 1, more information can be extracted from measurement data, which is generally given at a few select frequencies defined by IEC60268 [3]. This method provides minimal insight into the actual frequency-dependent angular sensitivity of the microphone. Moreover, using a more prominent line type for lower frequencies or showing plots of non-IEC60269 frequencies may cause misperceptions. Frequencies below 2 kHz generally show near optimal angular sensitivity. This paper proposes a method for providing frequency-dependent directivity information. A prototype application is introduced and data interpolation and smoothing for different applications are presented.

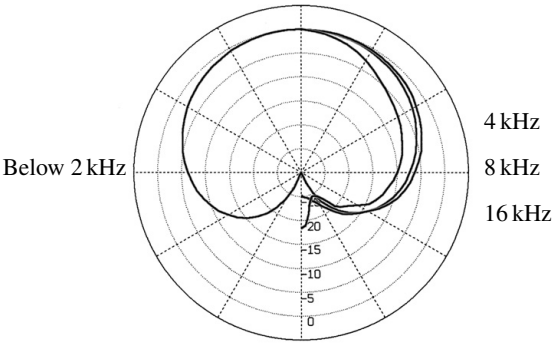


Fig. 1: Traditional display of directive microphone sensitivity at frequencies defined by IEC60268 [2]. The overlapping plots make exact observations about the frequency-dependent angular sensitivity difficult.

2 Methods

2.1 Data Acquisition

To acquire all necessary information to describe the angle- and frequency-dependent microphone sensitivity $S(\theta, \nu)$ impulse responses (IR's) were gathered:
In an anechoic chamber meeting ISO 3745 Precision Class 1 standards [4], sine sweep test tones of 3 s were played back using a high quality studio monitor at approximately 4 m distance from the microphone. The test tone was recorded at $n = 24$ rotation positions of the microphone, set via a motorized rotation device connected to the microphone stand and monitored with the appropriate software. The resulting recordings, representing an angular resolution of 15°, were deconvolved with the sweep signal to aquire the impulse responses. These were later treated to reduce the influence of loudspeaker imperfections by deconvolving the signals with an impulse response of the speaker, recorded with a high quality measurement microphone prior to the IR-recordings. The microphone's power spectrum at the angle $(n-1)\cdot 15°$ can be accessed via Discrete Fourier Transform (DFT) [5].

The resulting power spectra can be seen in Figure 2. The observed irregularities at higher frequencies are largely due to the fact that multiple microphone capsules were combined in a shock-mount during recording of the impulse responses. This causes reflections, leading to interference effects which can drastically increase or decrease sound pressure at arbitrary locations and frequencies.

2.2 Frequency Smoothing

Many applications require a certain degree of data smoothing. For the power spectra shown in Figure 2, $\frac{1}{N}$ - octave smoothing with $N = 12$ was applied. Some marketing brochures show data smoothed with up to $N = 3$. Figure 3 shows an IEC61260-compliant $\frac{1}{N}$ - filter bank with $N = 3$ [6].

2.3 Angle Interpolation

To get from 24 steps, represented by $S_i(\nu)$, to full 360° resolution as shown in Figure 4, described with $S(\theta, \nu)$, cubic spline data interpolation is applied. Cubic spline interpolation is achieved by constructing third-order polynomials within every interval $[S_i(\nu), S_{i+1}(\nu)]$ with $i = 0, \ldots, n-2$.

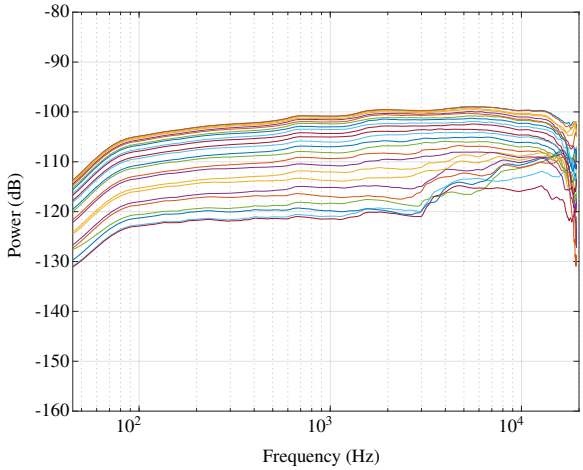


Fig. 2: Power spectra of measured microphone capsule in 15°- steps around a full rotation. Data smoothing with $\frac{1}{12}$ - octave filterbank is applied.

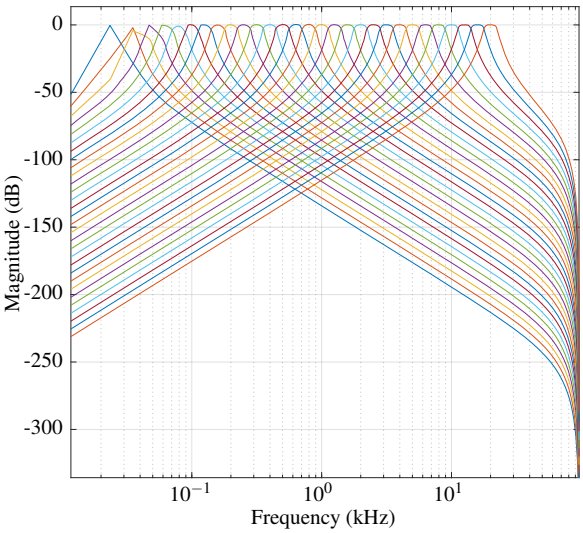
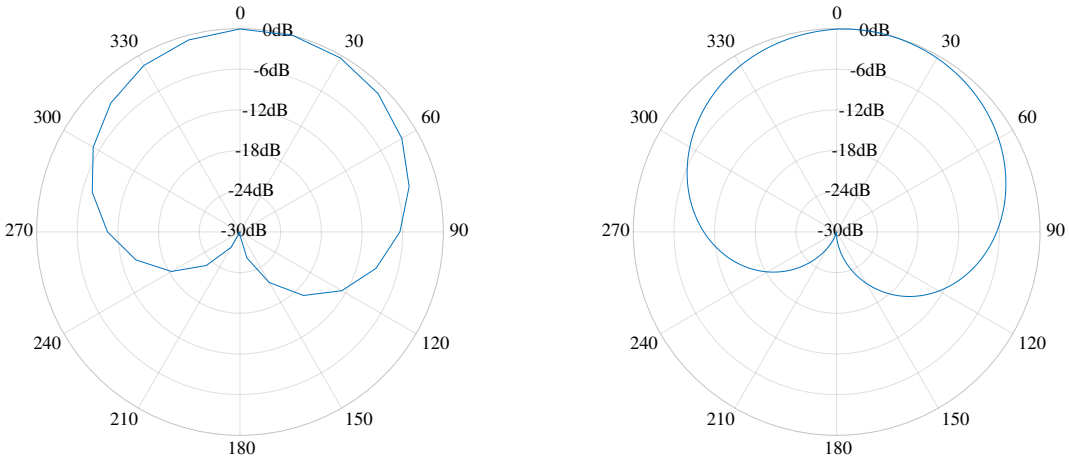


Fig. 3: Magnitude response of IEC61260 - compliant $\frac{1}{3}$- octave filter bank used to smooth frequency responses [6]. The filters shown result in strong smoothing, whereas more accurate data can be retained by applying filter banks with narrower bands.



(a) Before interpolation. The 24 data points for the given frequency bin are simply connected with lines.

(b) After interpolation the polar graph shows great resemblance to the theoretical curve for cardioid directivity.

Fig. 4: Interpolation process of angular sensitivity plot. Figure 4a shows the contours of a polarity plot with measured data simply connected by lines. This creates a rough and improbable graph. Figure 4b shows a version which has been interpolated with an angular resolution of 1° using cubic splines.

Considering the i^{th} part of the spline Y_i we get ([7, 8]):

$$Y_i(\tau) = a_i + b_i\,\tau + c_i\,\tau^2 + d_i\,\tau^3 \tag{1}$$

where τ is a parameter between 0 and 1. This results in

$$Y_i(0) = S_i = a_i \tag{2}$$
$$Y_i(1) = S_{i+1} = a_i + b_i + c_i + d_i \tag{3}$$

The derivatives of Y_i with respect to τ at the points S_i then are

$$Y'_i(0) = b_i \tag{4}$$
$$Y'_i(1) = b_i + 2c_i + 3d_i \tag{5}$$

Required boundary conditions are matching splines in all measurement points, as well as matching first- and second-order derivatives. Therefore we get

$$Y_{i-1}(1) = S_i \tag{6}$$
$$Y_i(0) = S_i \tag{7}$$
$$Y'_{i-1}(1) = Y'_i(0) \tag{8}$$
$$Y''_{i-1}(1) = Y''_i(0) \tag{9}$$

Additionally, to guarantee a sufficient number of boundary conditions to be able to solve the $4(n-1)$ unknowns, the second derivative of the endpoints is set to zero.

$$Y''_0(0) = 0 \tag{10}$$
$$Y''_{n-2}(1) = 0 \tag{11}$$

The conditions for a correct interpolation of the directivity patterns discussed in this paper are the demand for 360°- 0° continuity and a continuous derivative in the mentioned interval. As cubic spline interpolation guarantees both conditions (see equations 2, 3, and 8), simple data wrapping between 360° and 0° suffices for these conditions to be met. Therefore we set

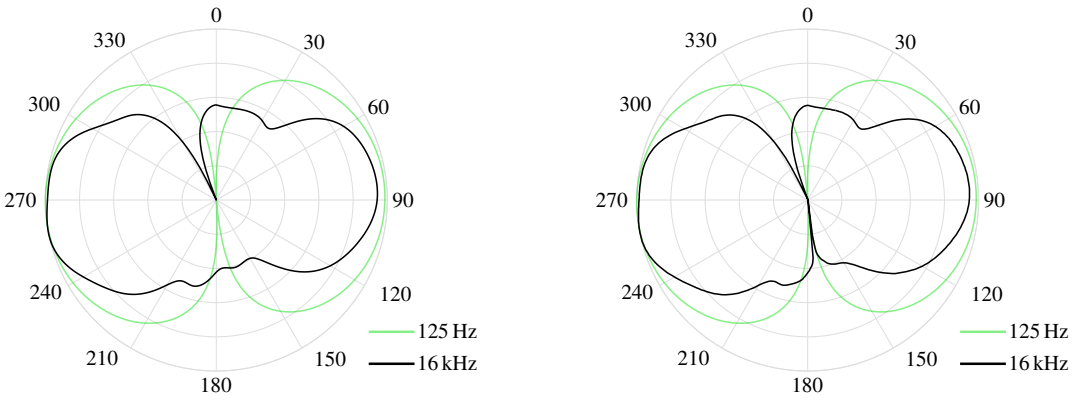
$$S_{24} := S_0 \tag{12}$$

and solve for all 4n variables a_i, b_i, c_i and d_i at all frequencies ν.

With certain directivities, such as hypercardioid and figure-of-eight, transitions into areas of negative sensitivity have to be addressed. Figure 5a shows an example, where the transition between positive and negative sensitivity was not taken into account in the interpolation process. Figure 5b shows the corrected version.

3 Results

The resulting application prototype is capable of converting a dataset of $m \times n$ impulse responses of m microphones, recorded at n angles, sampled at up to 192 kHz, into an interactive polar plot. Current research has focussed on capsules with cardioid and figure-of-eight directivity characteristics, although any arbitrary directivity is possible. As the processing relies on simple



(a) No correction at $\pm$ - transition.

(b) Correct interpolation.

Fig. 5: Interpolation errors occurring in the transition area between positive and negative sensitivities. Not taking this into account can lead to distorted directivity plots, especially at high frequencies, where the measured angular sensitivity shows more deviation from the theoretical curve.

impulse responses, the software is not restricted to single-capsule setups. Monophonic coincident arrays of any type can be evaluated as well (see Figure 6). Smoothing and normalization guarantee a seamless experience when sweeping through the frequencies. Figure 6 shows a screenshot of the prototype, displaying the measurements of a Schoeps double-M/S setup at approximately 1 kHz. The angle of the array towards the loudspeaker during IR-capturing was slightly off-axis, resulting in an offset of θ. As the setup was mounted in a common shock-mount, the same offset applies to all three capsules. For the potential use as a marketing instrument, offset correction can be applied.

4 Discussion

Through the use of interactive frequency-dependent angular sensitivity displays, the performance of a given microphone can be assessed in greater detail than with traditional polar plots. This can be advantageous when used as marketing material for high-performing microphones, or to help engineers find weaknesses in current hardware design. A promising aspect of the presented application is the evaluation of the performance of beamforming arrays. Figure 7 shows the polar response of a synthesized supercardioid, created with the double-M/S configuration shown in Figure 6,

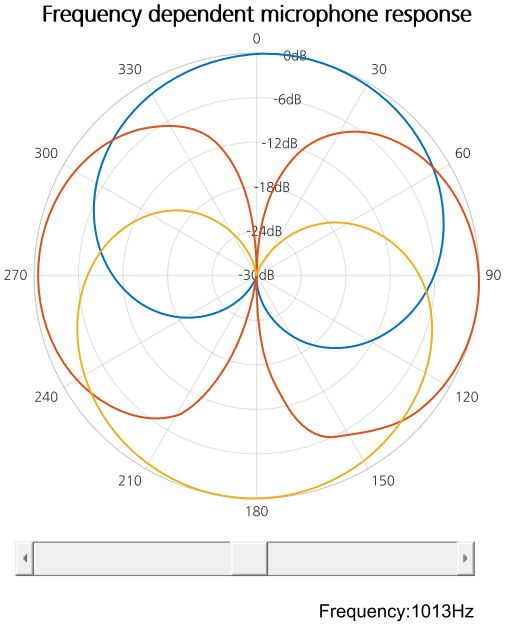


Fig. 6: Application prototype displaying measurements of a Schoeps double-M/S setup consisting of two CCM 4 cardioid capsules and one CCM 8 figure-of-eight capsule at $\approx$ 1 kHz [2].

using the synthesis models shown in equations 13 to 16 [10]. The use of this application for continuous performance monitoring of beamforming algorithms can be a useful aid for DSP development and quality assessment.

Using the microphone configuration shown in Figure 6, an arbitrary 1^{st} order directivity pattern can be synthesized, oriented towards any desired angle.

The signal from pressure sensors $p(\theta,t)$ and pressure gradient sensors $g(\theta,t)$ can be combined using:

$$S(\theta,t) = \alpha\, p(\theta,t) + (1-\alpha)\, g(\theta,t) \tag{13}$$

For this example, using a double-M/S setup, the pressure sensor (omnidirectional signal) is synthesized using the front- and rear-facing cardioids:

$$p(\theta,t) = S_{frontC}(\theta,t) + S_{rearC}(\theta,t) \tag{14}$$

For the desired supercardioid we set ([9]):

$$\alpha = \sqrt{2} - 1 \tag{15}$$

Using equations 13 - 15 we can compute a supercardioid as shown in Figure 7, using:

$$\begin{aligned} S_{SC}(\theta,t) = (\sqrt{2}-1)\left(S_{frontC}(\theta,t) + S_{rearC}(\theta,t)\right) \\ + (2-\sqrt{2}) S_{fig8}(\theta,t) \end{aligned} \tag{16}$$

Although current parameters show promising results and good performance, further research is required to determine optimal frequency-smoothing for different applications. Also, interpolation using n^{th} degree spherical harmonics instead of cubic splines could lead to better results for areas with changing sensitivity polarity as found in figure-of-eight directivity patterns, and for areas with a large angular derivative of the signal $\left(\left|\frac{\mathrm{d}S(\theta,v)}{\mathrm{d}(\theta)}\right| \gg 0\right)$. As the IR recording took place with three microphones simultaneously, an obvious amount of interference can be observed. The rear-facing cardioid microphone shows dramatic distortion of directivity characteristics at frequencies as low as 5 kHz, while the front-facing cardioid capsule of the same model shows relative frequency invariance up to 14 kHz. For more valid results additional IR recordings are needed without the reflective surfaces of elaborate shock-mounts and other capsules.

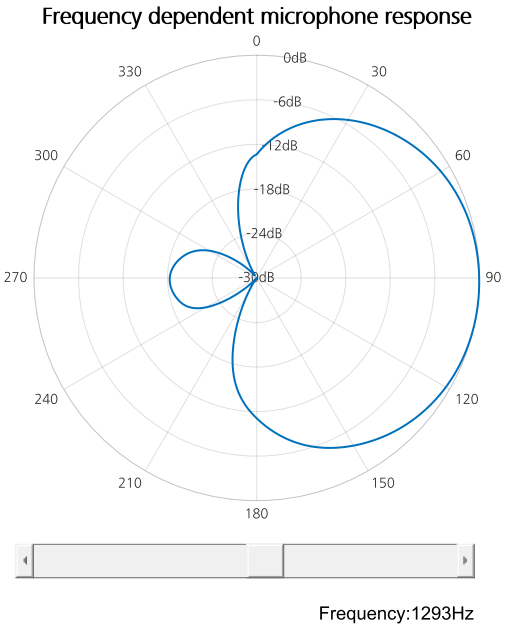


Fig. 7: A synthesized supercardioid response created using signals from the Schoeps double-M/S configuration shown in Figure 6.

5 Summary

A method to capture and interactively display the frequency-dependent angular sensitivity of microphones is introduced. Data capturing via the recording of impulse responses in an anechoic environment is described. Subsequently, signal processing in the form of deconvolution, smoothing, and interpolation are discussed and examples are shown in Figures 4 and 5. A functioning application prototype is introduced and application examples are provided:

- Display of single capsules to assess the quality of frequency invariance of a given directivity pattern (Figure 5b).
- Display of multi-capsule setups to examine the effect of inter-capsule-reflections and microphone mounting (Figure 6).
- Display of synthesized directive signals to assess the quality of beamforming algorithms (Figure 7).

Furthermore, future improvements in data acquisition and processing are discussed.

References

[1] Olson, H. F., "A History of High Quality Studio Microphones," in *Audio Engineering Society Convention 55*, 1976.

[2] *Schoeps Product Catalogue*, Schalltechnik Dr.-Ing. SCHOEPS GmbH, 6 edition, 2016.

[3] *Sound system equipment - Part 4: Microphones*, IEC 60268-4, 2014.

[4] ISO3745, *Acoustics – Determination of sound power levels and sound energy levels of noise sources using sound pressure – Precision methods for anechoic rooms and hemi-anechoic rooms*, 2012, revision 3.

[5] Stein, J. Y., *Digital Signal Processing: A Computer Science Perspective*, Wiley-Interscience, 2000, ISBN 0471295469.

[6] IEC61260, *Electroacoustics - Octave-band and fractional-octave-band filters - Part 1: Specifications*, 2012, revision 3.

[7] Bartels, R. H., Beatty, J. C., and Barsky, B. A., *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling (The Morgan Kaufmann Series in Computer Graphics)*, Morgan Kaufmann Pub, 1987, ISBN 0934613273.

[8] Weisstein, E. W., "Cubic Spline." *MathWorld–A Wolfram Web Resource*, 2017.

[9] Benesty, J. and Jingdong, C., *Study and Design of Differential Microphone Arrays (Springer Topics in Signal Processing)*, Springer, 2012, ISBN 364233752X.

[10] Runow, B. and Curdt, O., "Microphone Arrays for professional audio production," in *28th Tonmeistertagung*, VdT, 2014.

Hörversuche zur Entwicklung eines neuartigen Mehrkapsel-Mikrofons

Hendrik Paukert[1], Jonathan Ziegler[1, 2], Andreas Koch[1]
[1]*Hochschule der Medien Stuttgart, Germany, Email: paukert@hdm-stuttgart.de*
[2]*Eberhard Karls Universität Tübingen, Germany Email: zieglerj@hdm-stuttgart.de*
[1]*Hochschule der Medien Stuttgart, Germany, Email: kocha@hdm-stuttgart.de*

Abstract

Für die Entwicklung eines neuartigen digital prozessierten Mikrofonarrays, haben wir im Vorfeld einen Hörversuch zur Untersuchung verschiedener Störgeräusche vorgestellt [1]. Diese dienten dem besseren Verständnis, welchen Faktoren bei der Entwicklung der Algorithmen die größte Beachtung geschenkt werden muss. In diesem Teil werden die Hörversuche, deren Audiodatengenerierung sowie die Ergebnisse zur Einschätzung und Abgrenzung verschiedener Fremd- und Eigenalgorithmen zur Entwicklung des Mehrkapsel-Mikrofonarrays im Haupteinsatzgebiet der Sprache vorgestellt. Hierzu wird das Open Source Tool "WAET" [2] für die menschliche-, sowie die STOI-Methode [3] für die algorithmische Bewertung genutzt.

1. Vorüberlegungen

Der vorausgegangene Hörversuch zur Untersuchung von Störgeräuschen wurde komplett in Max/MSP [2] programmiert und von den Teilnehmern offline an einem Laptop bearbeitet. Um den aktuellen Test nun schneller aufbauen und auch online einer größeren Anzahl an Teilnehmern anbieten zu können, sollte das frei zugängliche „Web Audio Evaluation Tool" - kurz „WAET" [3] genutzt werden. Dieses Tool bietet u.a. die Möglichkeit, einen den ITU-R BS.1534-Empfehlungen [4] entsprechenden Test erstellen zu können. Im Detail kann damit eine zufällige Prüfdatenausgabe, unterschiedliche Skalenbeschriftungen automatische Lautstärkennormalisierung, die Einbindung von Kommentarfeldern, versteckter Anker- und Referenzdatei und das Prüfen auf versehentlich doppelt eingebundene Dateien umgesetzt werden. Auch können weitere Daten wie z.B. Bearbeitungsdauer, Anzahl der Abspielwiederholungen gespeichert und der Hörversuch den Teilnehmern von einem php-fähigen Server browserbasiert angeboten werden.

Auf Grund der zu erwartenden großen Datenmengen, sollte außerdem eine zeitsparende und weitgehend automatisierte Auswerteroutine möglich sein und, um Überbeanspruchung der Probanden zu vermeiden, die Bearbeitungsdauer von 30 min. im Mittel nicht überschritten werden.

1.1 Vor- und Nachteile von Online-Hörversuchen

Durch online zugängliche Hörversuche kann eine große Anzahl an möglichen Teilnehmern erreicht werden. Diese können den Bearbeitungszeitpunkt außerdem frei wählen und den Test bequem an einem beliebigen Ort bearbeiten. Auch für den Versuchsdurchführenden reduziert sich der Aufwand für Auf- und Abbau der Testumgebungen an einem spezifischen Ort, ebenso reduziert sich der Organisationsaufwand.

Nachteile sind jedoch die indirekten Hilfe- bzw. Assistenzmöglichkeiten bei auftretenden Fragen und Problemen, sowie die schlechte Kontrollmöglichkeit des verwendeten Equipments. Ein Online-Versuch sollte also möglichst übersichtlich und selbsterklärend aufgebaut und die Aufgabenstellung dem entsprechend geeignet sein. Durch die in diesem Zusammenhang teilweise geringen Unterschiede unserer Testdaten in Bezug auf Nachhallzeit, Klangveränderung und Bildung von Artefakten, bat sich generell die Nutzung von Kopfhörern an. Auch erwies sich uns die Nutzung einer eingemessenen Abhöre in akustischen optimierter Umgebung als wenig relevant, da in diesem Hörversuch keine frequenz-vollumfänglichen Musik-bearbeitungen, sondern Sprache abgefragt wird. Weiterhin kann davon ausgegangen werden, dass der jeweilige Proband für die Bearbeitung des Hörversuches nur einen Kopfhörer nutzt und somit die Bewertung der Audiodaten zueinander stimmig ist. Die Typenbezeichnung der Kopfhörer fragten wir aus Interesse zusätzlich ab.

2. Zu untersuchende Algorithmen

Haupteinsatzgebiet des neu entwickelten Mikrofonarrays werden Konferenzen sein. Hierzu wurde an der Hochschule der Medien ein Tracking-Algorithmus zur Detektion und Verfolgung jeweiliger Sprecher entwickelt [5], [6]. Eine aus drei Mikrofonkapseln synthetisierte Nieren- oder Superniercharakteristik, kann in Echtzeit ein aktuelles Sprachereignis erfassen, verfolgen und so von unerwünschten Schallereignissen besser freistellen.

Um diesen Freistellungseffekt weiter erhöhen zu können, soll zusätzlich ein Beamformer- bzw. Dereverb-Algorithmus implementiert werden. Hierzu stand prozessiertes Audiomaterial des Beamformers von Bernfried Runow [7], [8], zwei Dereverb-Plug-Ins und ein Spaced-Array-Konferenz-Komplett-System zur Verfügung. Durch die fixe Architektur der verschiedenen Algorithmen, konnte jedoch nur Runows Beamformer alle drei Mikrofonsignale des neuen Mehrkapselmikrofons nutzen und verfügte somit theoretisch über das größte Potential. Ein Plug-In konnte so nur das Summen-Trackingsignal, das andere nur zwei Signale des Mikrofons nutzen. Eine weiterhin geprüfte Spaced-Array-Konferenzanlage, verfügt über ihr eigenes räumliches Arraymikrofon mit 24 Kapseln und dient dem direkten Vergleich zu einem auf dem Markt bereits erhältlichen System.

ISBN
978-3-9812830-9-9

Nach Möglichkeit wurden die verschiedenen Algorithmen auch in verschiedenen Stärkeeinstellungen abgeprüft. Inklusive der Ankerdatei, dem bewusst verschlechterten Signal, und der Referenzdatei, dem optimalen Signal, umfasste der Hörversuch 9, jeweils in zwei Sprachen zu bewertende Audiofiles. Angesichts der gesetzten Bearbeitungsobergrenze von 30 Minuten, erwies sich diese Anzahl bei mehreren Vorversuchen bereits als ausreichend.

Bezeichnung	Beschreibung
Anker	Ankerdatei mit 3,5kHz-Lowpass und einer der Kategorie entsprechenden Degradierung durch hohe Nebengeräusche, Rauschen oder hohen Hallanteil. Dieses Signal muss vom Probanden als „schlecht“ bewertet werden.
Referenz	Nach Möglichkeit optimales Signal (trockene Studiosprachaufnahme), ohne Hall, Nebengeräusche oder Rauschen. Dieses Signal muss als „gut“ bewertet werden.
Kugel	Kugelsignal des Prototypenmikrofons
Runow 50%	B. Runows-Beamforming-Algorithmus in mittlerer Stärke, gespeist mit den drei einzelnen Kapselsignalen und der vom Tracker ermittelten Richtungsinformation.
Runow 100%	B. Runows-Beamforming-Algorithmus in voller Stärke, gespeist mit den drei einzelnen Kapselsignalen und der vom Tracker ermittelten Richtungsinformation.
Plug-In Nr.1 50%	DAW-Plug-In in mittlerer Stärke, gespeist mit zwei geführten Mono-Signalen.
Spaced-Array	Gesamtsystem mit eigenem räumlichen Mikrofonarray und schwacher Einstellung des produkteigenen Algorithmus. Stärkere Einstellungswerte verschlechterten das Signal enorm und mussten daher vom Test ausgeschlossen werden.
Plug-In Nr.2 50%	DAW-Plug-In in mittlerer Stärke, gespeist mit geführtem Mono-Signal.
Plug-In Nr.2 stark	DAW-Plug-In in starker Einstellung, gespeist mit geführtem Mono-Signal. Noch stärkere bzw. maximale Einstellung verschlechterten in unserem Fall die Signalqualität sehr stark.

Tab. 1: Übersicht der genutzten Signale

3. Generierung der Audiodaten

Zur Generierung der Audiodaten nutzten wir den neu entwickelten Aufbau einer reproduzierbaren Konferenzumgebung: über mehrere Iterationen hinweg konnten wir hier, auch durch die detaillierte BA von Robin Hirt [9], eine „virtuelle Konferenz“ gestalten. Mit diesem Mikrofonteststand kann über ein Lautsprechersetup mit definierten Abständen und Zuspielszenarien die Wirkungsweise der verschiedenen Algorithmen bzw. Algorithmen-Revisionen geprüft werden.

Insgesamt werden 10 Lautsprecher in zwei Ringen bzw. Radienabständen vom zu prüfenden Mikrofon aufgebaut. Über die vier Lautsprecher des inneren Rings werden die nahezu reflexionsfreien Sprachdateien mehrerer Personen getrennt und auch parallel ausgegeben. Die Lautsprecher des äußeren Rings erzeugen ein- und mehrkanalige Atmogeräusche wie z.B. Straßenlärm oder das Öffnen und Schließen einer Tür. Weiterhin befindet sich, nahe am Mikrofon, ein einzelner Lautsprecher, welcher z.B. Papier-, Tassen- und Stiftgeräusche ausgibt, sowie in 4 Metern Abstand ein weiterer Lautsprecher zur Generierung von sehr indirekten und räumlichen Signalen. Zur Simulation von Körperschall, z.B. verursacht durch Vibrationen eines Laptops oder Smartphones, kann zur Simulation optional ein kleiner E-Motor am Tisch befestigt werden. Durch eine elastische Aufhängung des Mikrofons ist hier allerdings schon von einer guten Körperschallentkopplung auszugehen.

Zur Nutzung im Hörversuch wurde letztendlich ein 3s kurzer Zeitabschnitt der Aufnahmen in deutscher und englischer Sprache ausgesucht. Längere Abschnitte machten aus Gründen der Vergleichbarkeitsschwäche der menschlichen Klangwahrnehmung und der dadurch folgenden Erhöhung der Versuchsdauer keinen Sinn.

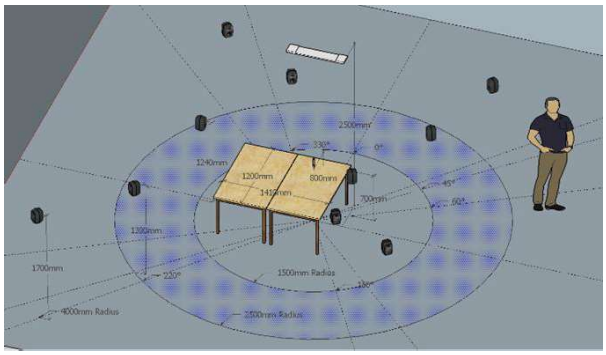


Abb. 1: Aufbauskizze der virtuellen Konferenz [10]: Zu sehen sind die beiden konzentrischen Lautsprecherringe, ergänzende Nah- und Fernlautsprecher, das kleine Prototypenmikrofon in der Mitte sowie das Konferenzsystems in Deckenmontage.

Abb. 2: 360°-Aufnahme des Genelec 1029A [11]-Lausprechersetups der virtuellen Konferenz: Die Raummaßen betragen 8,3x8,2x3,8m, die RT60 2,31s im Bereich von 500-1000Hz 2,31s (leerer und akustisch unbehandelter Raum).

4. Versuchsinterface

Das WAET stellt verschiedene Interface-Arten wie z.B. AB-, ABX- und Checkbox-Verfahren zur Verfügung. Anstatt des klassischen Mehrfachschieberegler-Interface der klassischen

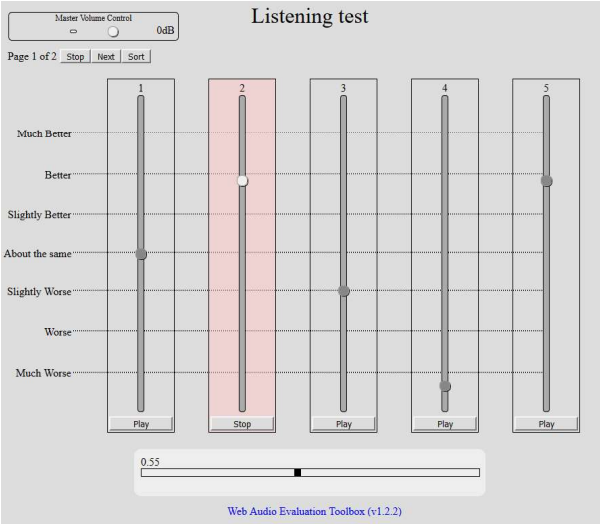


Abb. 3: Mehrfachschieberegler-Interface des Web Audio Evaluation Tool: jedes der fünf angebotenen Signal verfügt über einen eigenen Schieberegler

Abb. 4: einachsiges Hörversuchsinterface mit dem Web Audio Evaluation Tool: Jeder der grünen Balken repräsentiert eine auf der Skala zu positionierende Sprachdatei. Weitere Auffälligkeiten können anhand der Kommentarfelder weitergegeben werden.

MUSHRA-Methode (**Abb. 3**), fiel die Entscheidung zu Gunsten der APE-Variante (Audio Perceptual Evaluation) [12]. Wie in **Abb. 4** ersichtlich, wird hierbei nur eine Achse genutzt, auf der alle Soundfiles als Schiebebalken repräsentiert sortiert werden müssen. Das Interface ist dadurch kompakter und lässt pro Prüfabschnitt mehr Platz für Kommentarfelder oder andere Zusatzoptionen. Außerdem schien uns der einachsige Aufbau die Relation der Soundfiles zueinander mehr in den Vordergrund zu stellen, als dies bei getrennten Achsen der Fall wäre [Vgl. 13]. Durch die Textfelder konnten die Probanden auch möglicherweise gar nicht abgefragte, ihnen aber als wichtig erscheinende Informationen weitergeben. Um ein mögliches Biasing bzw. bewussten oder unterbewussten Manipulationen entgegenzuwirken, wurden die Audiodaten pro Abschnitt außerdem zufallsverteilt ausgeben.

Auf Grund des weiten Wertebereichs des einachsigen Aufbaus jedoch, kann bei nicht vollständiger Nutzung der Skala eine Verzerrung der Standardabweichung σ auftreten: ein Proband könnte seine Daten z.B. nur in der unteren Hälfe der Skala anordnen, ein anderer jedoch in der oberen Hälfte. Durch diesen Offset würde die σ aller Daten nun fälschlicherweise ansteigen. Das WAET bietet hier an, gewünschte Soundfiles in einstellbaren (Toleranz-)Bereichen an den Skalenenden anordnen zu müssen. Das nachträgliche Normalisieren der vielen Ergebniswerte kann so vermeiden werden. In unserem Falle musste also die Ankerdatei, welche vom Probanden definitiv als schlechteste Datei erkannt werden muss, am linken Rand, und die Referenzdatei, am rechten Rand angeordnet werden. Diese Maßnahme stellt auch sicher, dass jeder Teilnehmer den Test aktiv und aufmerksam durchführt und ihn bis zum Erkennen und dem korrekten Einordnen der Dateien nicht fortführen kann. Die restlichen Sprachdateien können nun zwischen beiden Werten verteilt werden. Leider konnte die angezeigte Fehlermeldung bei Nichterkennen nicht zu einem verständlichen Hinweis umformuliert werden.

Weiterhin erlaubt die WAET-Testumgebung eine Überprüfung ob jedes Audiodatei komplett abgespielt und bewegt worden ist. Das ist wichtig, da die Dateien, abgesehen von Anker- und Referenzdatei, sonst an ihren zufallsgenerierten Startpositionen stehen bleiben können.

5. Beschreibung der Prüfabschnitte und Ergebnisse

5.1. Einleitende Fragen

Zu Beginn des Hörversuches wurde eine einleitende Beschreibung angezeigt und die verwendeten Kopfhörer, das Alter der Teilnehmer sowie die audiospezifischen Erfahrungsbereiche abgefragt. Wie in **Abb. 5** erkennbar, gaben die meisten Teilnehmer an, ein oder mehrere Instrumente zu spielen und sich mit dem Aufnehmen und Bearbeiten von Musik, Foleys oder Sprache zu befassen. Ein weiterer großer Teil gab an, Musikliebhaber und Konsument von höherwertiger Audiotechnik zu sein. Da auch Mehrfachnennungen möglich waren, wurden oft zwei oder drei Kategorien parallel genannt, siehe **Abb. 6**.

Bei den genutzten Kopfhörern wurden die Marke Beyerdynamic [14], gefolgt von Sennheiser [15] und AKG [16], am meisten genannt (**Abb. 7**). Die vielen Einzelmeldungen unterschiedlicher Marken wurde unter „sonstige Nennungen“ zusammengefasst. Das Durchschnittsalter aller 59 Teilnehmer betrug 39 Jahre, allerdings mit einer hohen Streuung. Jüngster war 19 und ältester Teilnehmer 63 Jahre jung. Auch die durchschnittliche Bearbeitungsdauer schwankte stark, lag im Mittel allerdings bei knapp 22 Minuten.

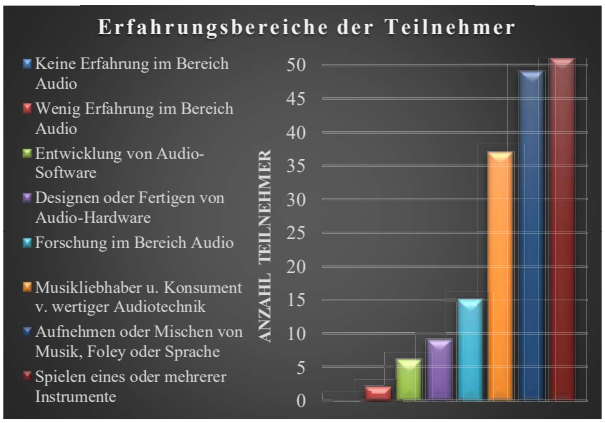


Abb. 5: Erfahrungsbereiche der Teilnehmer: am meisten kennen sich die Versuchsteilnehmer bei Audioproduktionen aus und spielen ein oder mehreren Instrumente.

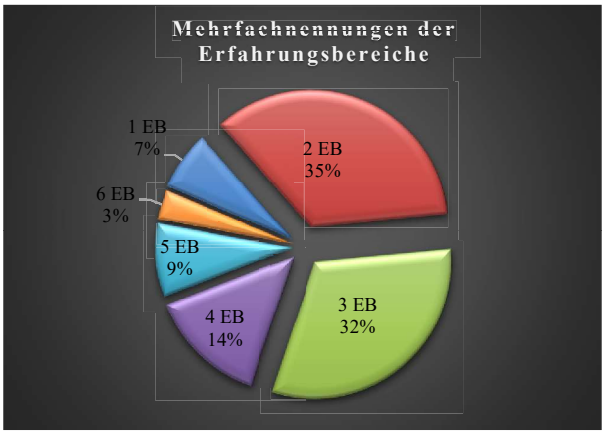


Abb. 6: Mehrfachnennungen der Teilnehmer: 67% nannten max. 3 Erfahrungsbereiche parallel.

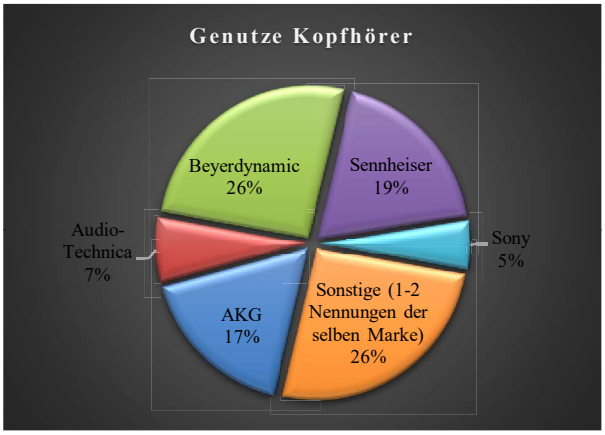


Abb. 7: Genutzte Kopfhörer der Teilnehmer: die größte Blöcke bilden mit insg. 62% Beyerdynamic, Sennheiser und AKG.

5.2 Beschreibung der Balkendiagramme

Die nachfolgenden Diagramme setzen sich aus den Bewertungen der abgeprüften Sprachdateien anhand von blauen Balken, sowie deren Standardabweichung σ anhand von roten Balken zusammen. Der Wert 0,0 steht für eine sehr schlechte und der Wert 1.0 für eine sehr gute Bewertung. Niedrige Werte der roten Balken zeigen eine niedrige σ auf. In diesem Zusammenhang liegen die σ-Werte der Anker- und Referenzdatei generell und trotz des Toleranzbereichs für die Zwangspositionierung an den Skalenenden auf einem sehr niedrigen Niveau. Für den ersten Prüfabschnitt „Training“ steht die 1,0 für eine als sehr hallig empfundenen Bewertung.

5.3 Training

Damit sich die Teilnehmer an die Testumgebung gewöhnen konnten, wurde ein vom Prüfumfang reduzierter Trainingsmodus implementiert. Anstatt 9 wurden nur 6 Soundfiles in einer Sprache, Deutsch oder Englisch, abgefragt. Auch wenn das Training dadurch nicht voll bewertet werden kann, können die Daten auf Grund der Menge der Teilnehmer und der abgefragten Eigenschaft „empfundene Halligkeit“ zumindest für eine erste Abschätzung genutzt werden. Grundsätzlich stellt das Kugelsignal das unbearbeitete und räumlichste Realsignal dar und wird nur von der künstlich verhallten Ankerdatei übertroffen (3s Nachhallzeit). Nachfolgend zur nachhalllosen Referenzdatei, wurde B. Runows Beamformer als sehr trocken bewertet.

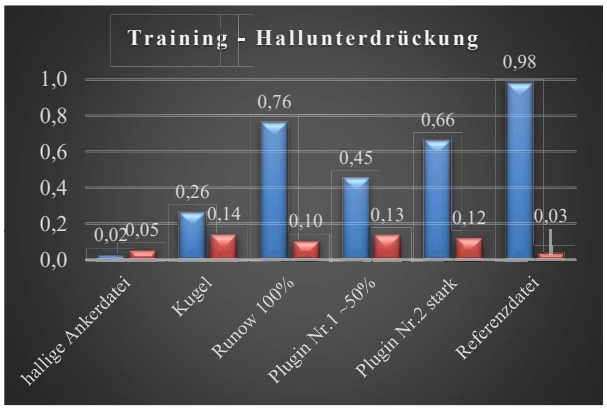


Abb. 8: Ergebnis „empfundene Halligkeit“ im Trainingsmodus: Runows Beamformer wird mit geringster σ und Halligkeit bewertet.

5.4 Sprachverständlichkeit

Eine Haupteigenschaft für (Konferenz-)Mikrofone ist die Verständlichkeit der Sprache. Hierzu wurden nun alle im Versuch vorkommenden Sprachdateien in deutscher und englischer Sprache abgefragt.

Bei beiden Durchläufen wurde B. Runows Beamformer im Maximaleinstellung als bestes bewertet, dicht gefolgt von Plug-In Nr.2. Deutlicher Verlierer ist hier das Spaced-Array-Konferenzsystem, dessen Sprachverständlichkeit deutlich unter einer übermäßigen Signalverschlechterung durch das interne Processing leiden musste.

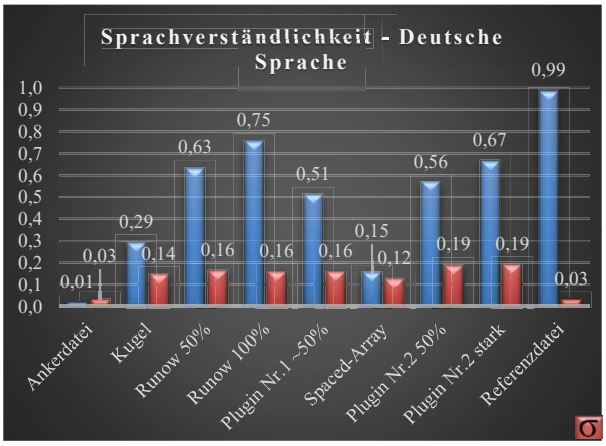


Abb. 9: Ergebnis „Sprachverständlichkeit“ in deutscher Sprache

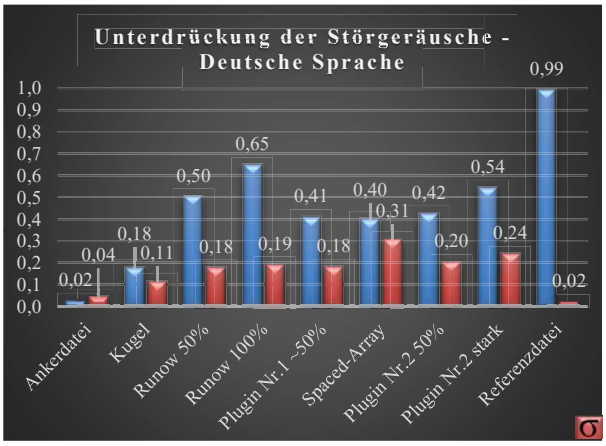


Abb. 11: Ergebnis „Unterdrückung der Störgeräusche“ in deutscher Sprache

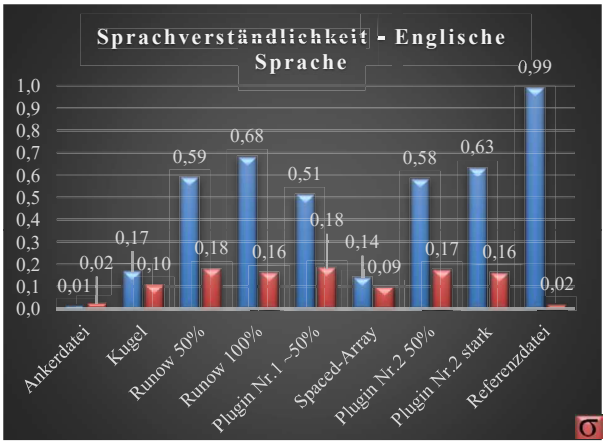


Abb. 10: Ergebnis „Sprachverständlichkeit“ in englischer Sprache

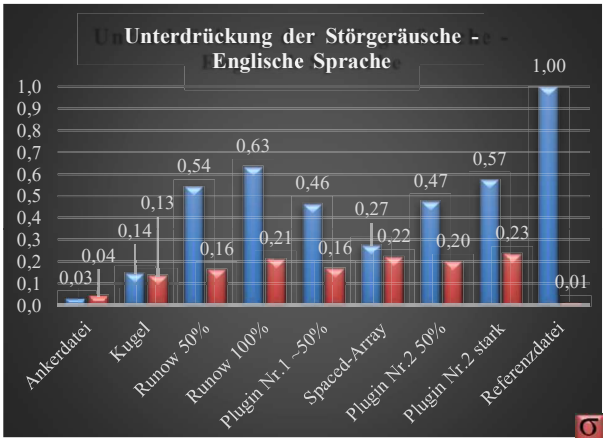


Abb. 12: Ergebnis „Unterdrückung der Störgeräusche“ in englischer Sprache

5.5 Unterdrückung der Störgeräusche

In diesem Abschnitt fragten wir die Sprachdateien mit Fokus auf die Störgeräuschunterdrückung ab. Das vorher weit abgeschlagene Konferenzsystem konnte hier deutlich aufholen. Sein starkes Processing versteht es, Nebengeräusche wirksam zu unterdrücken, im Vergleich zur Konkurrenz litt dadurch aber auch erheblich die Qualität, womit die erhöhte σ zusammenhängen könnte. Abgesehen von der störgeräuschsfreien Referenzdatei, führt B. Runows Algorithmus in Maximaleinstellung erneut die Auswertung an.

5.6 Empfundene Qualität

Das Gehör reagiert sehr empfindlich auf Veränderungen des Klangs der Sprache und schon geringe Nuancen werden wahrgenommen. Als weiterer wichtige abzuprüfende Eigenschaft galt daher die empfundene Qualität der bearbeiteten Sprachdateien. Hierbei liegt das Spaced-Array nur knapp über der bewusst mit Rauschen und Bandpassfilterung degradierten Ankerdatei. Trotz hoher Sprachverständlichkeit und Störgeräuschunterdrückung kann B. Runows Beamformer auch hier den ersten Platz behaupten. Da gute Qualität ein individuelles Maß ist, vermieden das Einbinden einer vermeintlich hochwertigen Referenzdatei. Auch in diesem Prüfabschnitt wichen die Ergebnisse in deutscher und englischer Sprache nur gering voneinander ab.

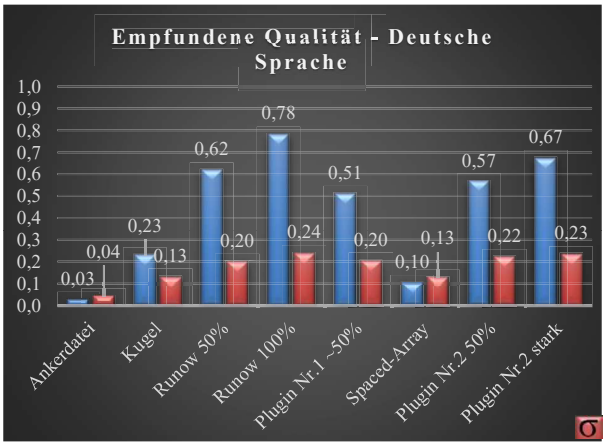


Abb. 13: Ergebnis „empfundene Qualität“ in deutscher Sprache

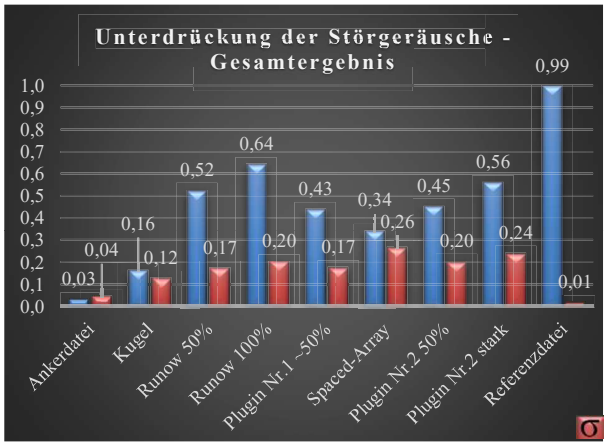


Abb. 16: Gesamtergebnis „Unterdrückung der Störgeräusche“

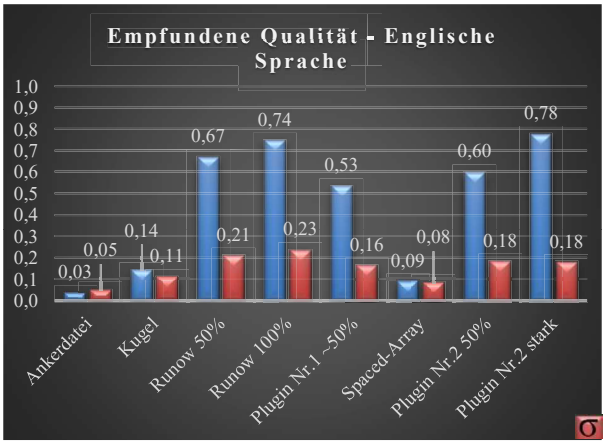


Abb. 14: Ergebnis „empfundene Qualität“ in englischer Sprache

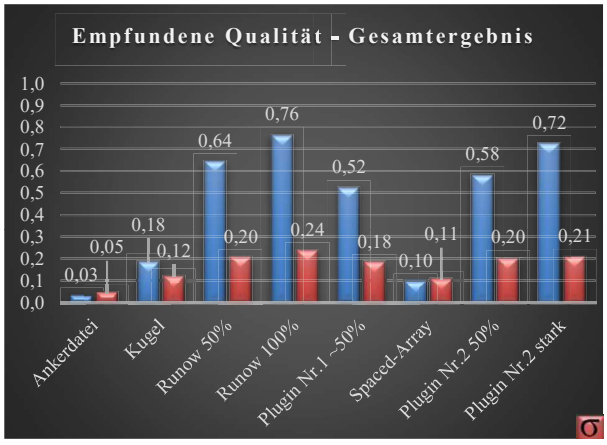


Abb. 17: Gesamtergebnis „empfundene Qualität“

5.7 Zusammenfassende Ergebnisse beider Sprachen

Zu besseren Übersicht werden hier nochmal die zusammenfassenden Ergebnisse beider Sprachen in Summe dargestellt.

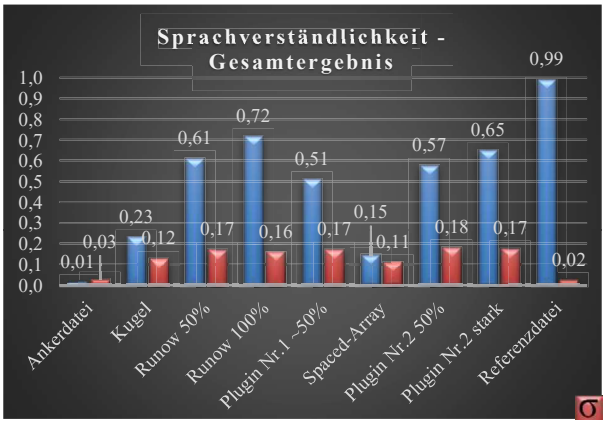


Abb. 15: Gesamtergebnis „Sprachverständlichkeit“

5.8 Sprachverständlichkeitsindex STOI

Zum automatischen Bewerten der Sprachverständlichkeit bietet sich das frei zugängliche Matlabscript „STOI“ [17] an. „STOI“ steht für Short Time Objective Intelligibility Messurement und wurde von C.H. Taal et al an der Delft University of Technology in Holland entwickelt. Das Skript vergleicht in unserem Fall die trockene Studioaufnahme (Referenzdatei) mit denen, über das in der virtuellen Konferenz mit dem Prototypenmikrofon aufgenommen und Dateien. Hohe Ergebniswerte bedeuten eine hohe Sprachverständlichkeit, so dass auch beim Vergleich zweier gleicher Dateien stets eine 1.0 herauskommt. Wir konnten so unsere Sprachdateien erneut abprüfen, gespannt darauf, in wie fern sich die Ergebnisse des Prüfalgorithmus mit unseren bisherigen Ergebnissen decken würden.

Grundsätzlich liegen die Ergebnisse näher beieinander, als bei der durch den Menschen durchgeführten Bewertung. Dadurch sollte den Nachkommastellen mehr Beachtung geschenkt werden. Interessant ist, dass beide Sprachdateien des Spaced-Arrays schlechter als die Ankerdatei bewertet wurde. Wie bei der Ankerdatei sind die Daten des Spaced-Arrays im

Frequenzband stark beschnitten - für den Bewertungs-Algorithmus wahrscheinlich zu stark und in einem als wichtig bewertetem Bereich. Auch das indirekte Kugelsignal wird in beiden Fällen erkannt und in Summe nicht viel besser bewertet als das räumliche Kugelsignal. Interessanterweise schneidet das Tracking Signal bei der STOI-Bewertung auf ähnlich hohem Niveau ab, wie mit kombiniertem Beamformer bzw. Dereverb. Grund dafür könnte die interne Korrelationsmessung bis nur 5000Hz sein, welche a) das Processing der höheren Frequenzen unbewertet lässt und b) die Artefaktbildung korrelationsbedingt höher bewertet als der Mensch, der hier teilweise deutliche Unterschiede hören kann.

Die weiteren Sprachdateien bzw. Algorithmen liegen relativ nahe beieinander, allerdings zeichnet sich auch in diesen geringen Bereich das Führen des Beamformer von B. Runow ab. Deutlich erkennbar ist auch, dass das Plug-In Nr. 1 in dieser Bewertungsrunde stark aufholen kann. Zur Ergänzung wurden zwei weitere Einstellungen des Plug-Ins ausprobiert.

Bezeichnung	Beschreibung
Ankerdatei	Ankerdatei mit 3,5kHz-Lowpass und einer Degradierung durch hohe Nebengeräusche und Rauschen. Dieses Signal muss als „schlecht“ bewertet werden.
Referenz	Sie diente im STOI-Versuch als Vergleichssignal (Wert 1.0). Aus Übersichtsgründen wird dieses Signal daher im Chart nicht dargestellt.
Kugel	Kugelsignal des Prototypenmikrofons
Tracking	Sprachverfolgungs-Algorithmus Hochschule der Medien Stuttgart mit synthetisierter Niere bis Superniere.
Runow 50%	B. Runows-Beamforming-Algorithmus in mittlerer Stärke, gespeist mit den drei einzelnen Kapselsignalen und der vom Tracker ermittelten Richtungsinformation.
Runow 100%	B. Runows-Beamforming-Algorithmus in maximaler Stärke, gespeist mit den drei einzelnen Kapselsignalen und der vom Tracker ermittelten Richtungsinformation.
Plug-In Nr.1 ~25%	DAW-Plug-In in schwacher Einstellung, gespeist mit zwei geführten Mono-Signalen.
Plug-In Nr.1 ~50%	DAW-Plug-In in mittlerer Einstellung, gespeist mit zwei geführten Mono-Signalen.
Plug-In Nr.1 100%	DAW-Plug-In in maximaler Stärke, gespeist mit zwei geführten Mono-Signalen.
Spaced-Array	Gesamtsystem mit eigenem räumlichen Mikrofonarray und schwacher Einstellung des produkteigenen Algorithmus.
Plug-In Nr.1 50%	DAW-Plug-In in mittlerer Stärke, gespeist mit geführtem Mono-Signal.
Plug-In Nr.2 stark	DAW-Plug-In in starker Einstellung, gespeist mit geführtem Mono-Signal.

Tab. 2: Übersicht der ergänzten Signale für die Sprachverständlichkeitsanalyse „STOI“

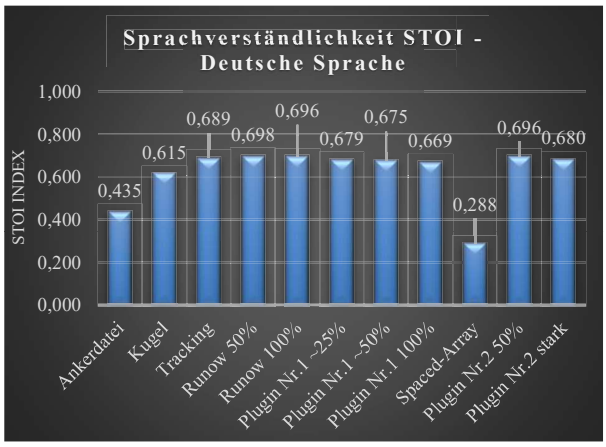


Abb. 18: Ergebnis Sprachverständlichkeitsanalyse „STOI“ in deutscher Sprache

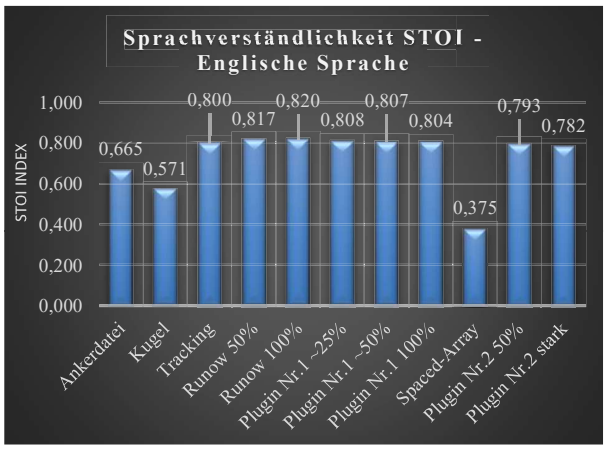


Abb. 19: Ergebnis Sprachverständlichkeitsanalyse „STOI“ in englischer Sprache

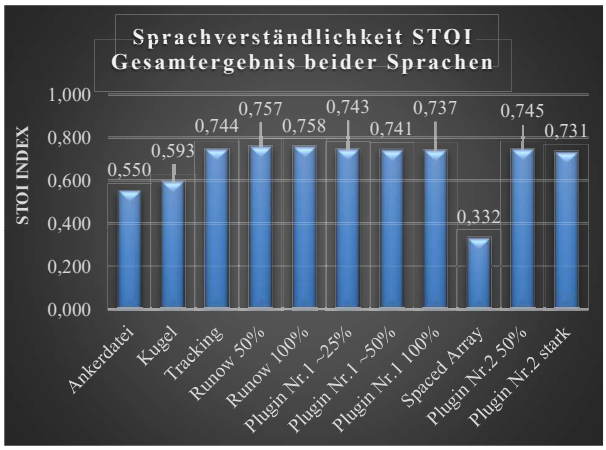


Abb. 20: Gesamtergebnis Sprachverständlichkeitsanalyse „STOI“ beider Sprachen

6. Datenauswertung

Pro Teilnehmer standen über 70, insgesamt über 4000 Werte zur Auswertung bereit. Zur zügigen Verarbeitung der Daten musste beim Erstellen des Hörversuches jedoch beachtet werden, dass WAET die zufällige Reihenfolge der ausgegeben Sprachdateien genauso in der Ergebnisdatei abspeichert. Daher müssen die jeweiligen Sprachdateien bei der Programmierung des Hörversuchs mit Indizes versehen werden um sie nachträglich sortieren zu können. Anschließend konnten die sortierten Datenblöcke in eine Sammelliste übertragen und ausgewertet werden.

7. Zusammenfassung und Ausblick

Durch das Web Audio Evaluation Tool konnte ein individuell programmierbarer und den ITU-Empfehlungen folgender Hörversuch einem großen Teilnehmerkreis online zugänglich gemacht werden.

Insgesamt wurde die Kombination des Trackingalgorithmus mit B. Runows Beamformer in Maximaleinstellung als bestes bewertet. Dieser Beamformer konnte allerdings auch alle drei Kapselsignale nutzen, so dass auch die anderen Algorithmen theoretisch Optimierungspotential besitzen. Plug-In Nr.1 lag mit seinen Ergebnissen oft knapp hinter denen des Plug-Ins Nr.2 und konnte vor allem bei den Ergebnissen der automatischen Sprachverständlichkeitsanalyse STOI deutlich aufholen. Das eigenständige Konferenzsystem wurde von allen prozessierten Dateien in allen Kategorien als schlechtestes bewertet und konnte nur im Bereich der Störgeräuschunterdrückung einigermaßen überzeugen.

Weiterhin können die gesammelten Daten zur fortführenden Mikrofonentwicklung genutzt werden und die Grundlage zur Implementierungen eines Beamformers und dessen Stärkeeistellungen bilden.

8. Quellenangaben

[1] H. Paukert, J. D. Ziegler: “Listening Tests in the Process of Microphone Development”: Tagungsbericht der 29. internationalen Tonmeistertagung des Verbands deutscher Tonmeister e.V., Nov. 2016, Seite 273-280, ISBN 978-3-9812830-7-5, URL: https://www.tonmeister.de/index.php?p=tonmeistertagung/2016/downloads

[2] Cycling ’74, Max 7 perpetual licence, URL: https://www.cycling74.com

[3] N. Jillings, B. De Man, D. Moffat, J. D. Reiss: “Web Audio Evaluation Tool: a browser-based Listening Test Environment”, Centre for Digital Music, Queen Mary University of London, URL: https://github.com/BrechtDeMan/WebAudioEvaluation Tool

[4] “Method for the subjective assessment of intermediate quality level of coding systems”, International Telecommunication Union (ITU) Empfehlung BS.1534-3, 2015, URL: https://www.itu.int/rec/R-REC-BS.1534-3-201510-I/en

[5] J. D. Ziegler, A. Koch, A. Schilling: „Speech classification for acoustic source localization and tracking applications using convolutional neural networks”, Audio Engineering Society convention 145, October 2018

[6] J. D. Ziegler, H. Paukert, A. Koch, A. Schilling: “Speaker Tracking with Coincident Microphone Arrays and Convolutional Neural Networks”, IEEE Journal, aktuell ausstehende Bewilligung

[7] B. Runow, O. Curdt: “Mikrofonarrays in der professionellen Audioproduktion”, Tagungsbericht der 28. Internationalen Tonmeistertagung Verband deutscher Tonmeister e.V., Nov. 2014, Seite 263-269, ISBN 978-3-9812830-5-1 URL: https://www.tonmeister.de/index.php?p=tonmeistertagung/2014/downloads

[8] B. Runow, O. Curdt, A. Schilling: „Richtrohrmikrofone versus Mikrofonarrays“, Tagungsbericht der 29. internationalen Tonmeistertagung des Verbands deutscher Tonmeister e.V., Nov. 2016, Seite 221-228, ISBN 978-3-9812830-7-5, URL: https://www.tonmeister.de/index.php?p=tonmeistertagung/2016/downloads

[9] R. Hirt: „Entwicklung einer virtuellen Konferenz unter besonderer Berücksichtigung der Reproduktion von zuvor aufgenommener Sprache“, Bachelorthesis, Hochschule der Medien Stuttgart, 2017

[10] Timbre Inc., URL: https://www.sketchup.com/

[11] Genelec Inc. URL: https://www.genelec.com/support-technology/previous-models/1029a-studio-monitor

[12] B. D. Man, J. D. Reiss, “APE: Audio Perceptual Evaluation toolbox for Matlab”, 136th AES Convention 2014, Berlin, Germany

[13] B. D. Man, J. D. Reiss, “APE: Audio Perceptual Evaluation toolbox for Matlab”, 136th AES Convention 2014, Berlin, Germany, Seite 1

[14] Beyerdynamic GmbH & Co. KG, URL: http://www.beyerdynamic.de

[15] Sennheiser electronic GmbH & Co. KG, URL: https://de-de.sennheiser.com

[16] Harman Deutschland GmbH, URL: https://de.akg.com/

[17] C.H. Taal, R. C. Hendriks, R. Heusdens, J. Jensen: A short-time objective intelligibility measure for time-frequency weighted noisy speech”, Delft University of Technology, Niederlande

The Fundamental Problem of the Spectral Subtraction

B. Runow[1], J. D. Ziegler[2], H. Paukert[3], A. Schilling[4], O. Curdt[5]

[1] *Wilhelm-Schickard Institut, Eberhard Karls University, Tübingen, Email: bernfried@runow.info*
[2] *Wilhelm-Schickard Institut, Eberhard Karls University, Tübingen, Email: zieglerj@hdm-stuttgart.de*
[3] *Hochschule der Medien Stuttgart, Email: paukert@hdm-stuttgart.de*
[4] *Wilhelm-Schickard Institut, Eberhard Karls University, Tübingen, Email: schilling@uni-tuebingen.de*
[5] *Hochschule der Medien Stuttgart, Email: curdt@hdm-stuttgart.de*

Abstract

Spectral Subtraction is often used for noise reduction and speech enhancement. It is an important tool of digital audio signal processing. Since its introduction in 1979, several problems like Phase Errors, Cross-time Errors and Magnitude Errors cause rather disappointing results. Beyond these errors, there is a fundamental problem within the basic principles of Spectral Subtraction, which is documented in this publication.

1. Introduction

Spectral Subtraction is a widespread method to dynamically process the spectrum of a digital audio signal. It gives you the possibility to edit a signal in a specific spectral range. The basis for this procedure is the discrete Fourier transform (DFT), which converts a time-series signal into the frequency domain and makes frequency analysis possible. In the spectral domain it is possible to edit individual spectral components, the so-called spectral coefficients. This makes it possible to subtract information from a specific frequency component. Finally, the processed signal can be resynthesized by means of an inverse discrete Fourier transform (iDFT). Therefore, the edited signal is available in the time domain once again.

The crucial advantage of the Spectral Subtraction is given by the short-time Fourier transform (STFT). With the STFT, it is possible to decompose a continuous stochastic signal and transform each time segment into the spectral domain. There, the time segments can be edited one after another. After the inverse transformation, the time segments can be recomposed into a continuous signal.

Because of the segmental processing, it is possible to edit each segment individually. This means, we can create an adaptive, real-time signal processing algorithm with a short latency. This is the reason for the importance of the Spectral Subtraction in the last decades. A multitude of applications use this technique, like noise reduction and speech enhancement.

2. Fundamentals

2.1. Windowing of a Signal

The segmentation of a continuous input signal $x(n)$ can be achieved with a window function $w(n)$, as we can see in Fig. 1.

Each segment is multiplied with the window function $w(n)$:

$$x_{win}(n) = x(\eta_{win} + n) \cdot w(n)\,, \qquad (1)$$

where $n = 0,1,2,\ldots,N-1$ is the discrete time index and N the length of the segments. The variable η_{win} defines the first sample of the current segment.

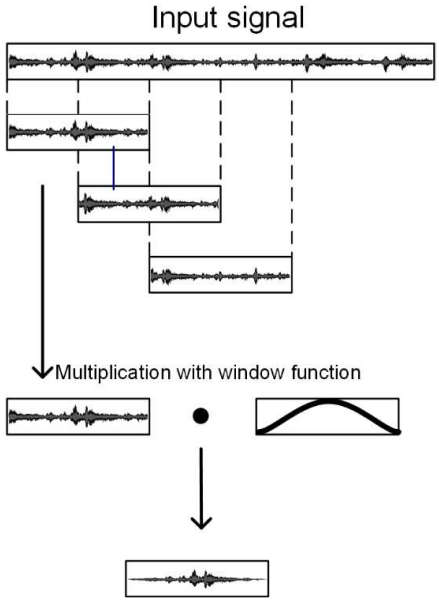


Fig. 1: Windowing of a continuous input signal using the von-Hann window function with an overlap of 50%.

An overlap of the segments is possible. Depending on the length of overlap, a compatible window function has to be chosen. The sum of the successive window functions always has to be one. This restriction is given in order to prevent a distortion of the signal within the resynthesis process, more precisely through the multiplication with the window

ISBN
978-3-9812830-9-9

function. This means that the windowing must result in a constant amplification of 1.

If we don't want an overlap of segments, we can choose the rectangular window:

$$w_{rect}(n) = 1\,, \tag{2}$$

with $n = 0,1,2,\dots,N-1$.

If we want an overlap of 50%, we can, for example, choose the von-Hann window function:

$$w_{hann}(n) = \frac{1}{2} - \frac{1}{2}\cos\left(2\pi\frac{n}{N-1}\right), \tag{3}$$

with $n = 0,1,2,\dots,N-1$.

In Fig. 2 you can see the von-Hann window functions for the segmentation of an input signal. Any window function can be used as long as the constraint of constant amplification is met.

Because of the use of a window function, a segment is also called window.

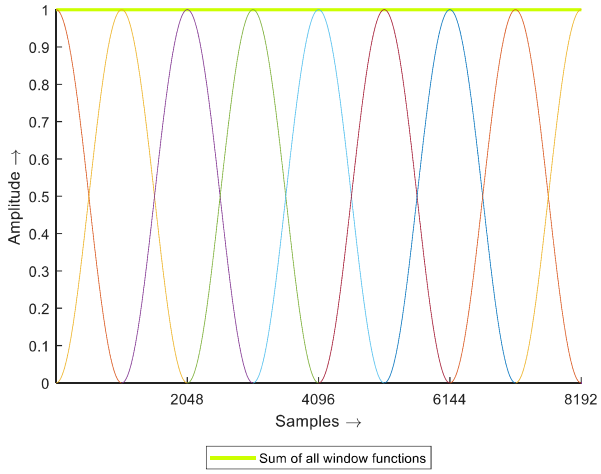


Fig. 2: Von-Hann window functions with a length of 2048 samples and their sum.

2.2. Short-Time Fourier Transform

After the segmentation of the input signal, the short-time Fourier transform (STFT) uses the Discrete Fourier transform to transport each window into the frequency domain. We obtain the DFT-coefficients using [4][8]:

$$X_{win}(k) = \sum_{n=0}^{N-1} x_{win}(n) \cdot e^{-j2\pi k\frac{n}{N}}\,, \tag{4}$$

where $k = 0,1,2,\dots,N-1$ is the discrete frequency index.

Each DFT coefficient represents a constant oscillation with the dedicated frequency f_k:

$$f_k = f_s \cdot \frac{k}{N}\,, \tag{5}$$

where f_s represents the sampling frequency which was used for the sampling during the digitalisation of the input signal. The absolute value of the DFT coefficient is the amplitude $|X_{win}(k)|$ of the oscillation and $\angle X_{win}(k)$ describes the corresponding phase angle.

By means of the inverse discrete Fourier transform we can transport the spectral signal $X_{win}(k)$ back into the time domain [4][8]:

$$x_{win}(n) = \frac{1}{N}\sum_{k=0}^{N-1} X_{win}(k) \cdot e^{j2\pi n\frac{k}{N}}\,, \tag{6}$$

with $n = 0,1,2,\dots,N-1$. Thus, the two signal sequences $x_{win}(n)$ and $X_{win}(k)$ are a transform pair.

Finally, the processed signal segments can be recombined according to the defined overlap.

2.3. Characteristics of the STFT

The Short-time Fourier transform has a number of characteristics which are accurately described in the relevant literature [2][4][8][9]. Two of these characteristics are especially important for Spectral Subtraction: the periodicity and the resolution of time and frequency.

2.3.1. Periodicity

The exponential function $e^{-j2\pi kn/N}$ behaves in a periodic fashion depending on N. This results the periodicity of the DFT and consequently of the STFT [4][8]:

$$X_{win}(k) = X_{win}(k+N) \tag{7}$$

and

$$x_{win}(n) = x_{win}(n+N)\,. \tag{8}$$

2.3.2. Time Resolution and Frequency Resolution

By using a clever analogy to the Heisenberg uncertainty principle, Küpfmüller points out that it is not possible to simultaneously achieve both a high resolution in time and in frequency within the spectral domain [7].

The background of this principle is the identical length N of the transform pair consisting of the time-domain signal $x_{win}(n)$ and the signal in the frequency domain $X_{win}(k)$. To get a high frequency resolution, we need a preferably long signal length. Contrarily we achieve a high time resolution using a short window in the time domain as this enables us to compute an individual spectrum for each short time segment.

Fig. 3-5 make this uncertainty principle clear. We can see several spectra over time. The test signal, which is a sine wave changing its frequency every second, was transformed into the spectral domain by means of STFT. The charts differ in the window lengths which were used for the STFT.

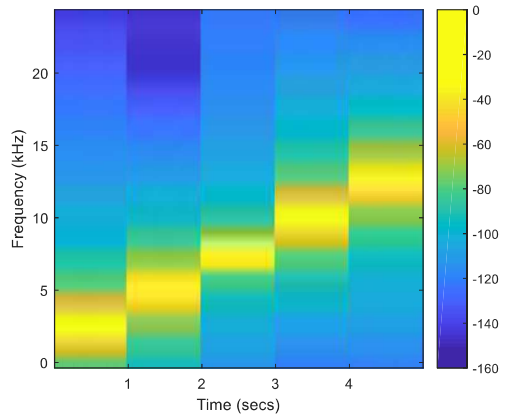


Fig. 3: Spectrogram of a sine wave changing its frequency every second. Analysed using STFT with a window length of 64 samples and a sampling frequency of 48kHz.

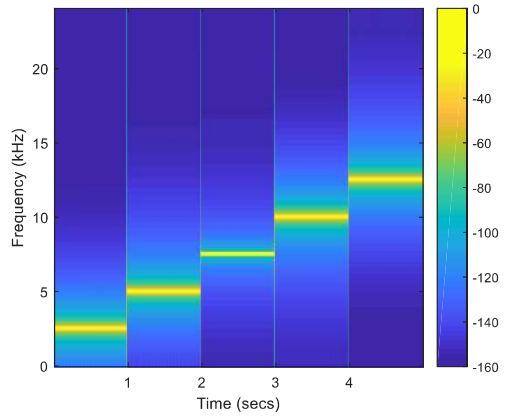


Fig. 4: Spectrogram of a sine wave changing its frequency every second. Analysed using STFT with a window length of 512 samples and a sampling frequency of 48kHz.

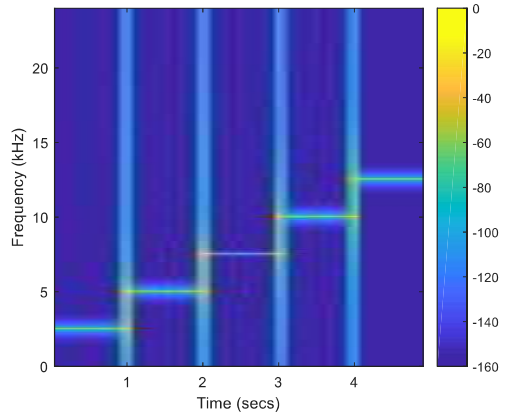


Fig. 5: Spectrogram of a sine wave changing its frequency every second. Analysed using STFT with a window length of 8192 samples and a sampling frequency of 48kHz.

We can solve this conflict with the help of a process called 'Zero Padding'. To get a high frequency resolution for a short time segment we can add a number of zeros at the end of the windowed time signal:

$$\tilde{x}_{win}(n) = \begin{cases} x_{win}(n) & for\ 0 \le n \le N-1 \\ 0 & for\ N \le n \le N+L-1 \end{cases}, \quad (9)$$

where $n = 0, 1, 2, \dots, N+L-1$ and L represents the number of the added zeros. Thus, it is possible to simultaneously achieve a high time resolution and a high frequency resolution within the STFT.

2.4. Spectral Subtraction

During Spectral Subtraction the amplitudes of two spectral signals are subtracted from each other. If $|X_{win}(k)|$ is the minuend and $|U_{win}(k)|$ is the subtrahend, we obtain the difference [3]:

$$|Y_{win}(k)| = |X_{win}(k)| - |U_{win}(k)| \cdot \upsilon(k, p=1), \quad (10)$$

where $\upsilon(k,p)$ is a real weighting factor which regulates the subtrahend $|U_{win}(k)|$, so that $|Y_{win}(k)|$ cannot assume negative values:

$$\upsilon(k,p) = \begin{cases} 1 \cdot \iota & for\ |U_{win}(k)| \le |X_{win}(k)| \\ \dfrac{|X_{win}(k)|^p}{|U_{win}(k)|^p} \cdot \iota & for\ |U_{win}(k)| > |X_{win}(k)| \end{cases}. \quad (11)$$

The real factor $0 \le \iota \le 1$ defines the intensity of the Spectral Subtraction. If $\iota = 0$, there is no subtraction. If $\iota = 1$, the subtraction is maximal. The quotient of $|X_{win}(k)|$ and $|U_{win}(k)|$ prevents that $|Y_{win}(k)|$ can become negative if the absolute value of $U_{win}(k)$ is larger than the absolute value of $X_{win}(k)$.

If we don't want to subtract the amplitudes, but the power, equation (10) is modified to produce $|Y_{win}(k)|$:

$$|Y_{win}(k)| = \sqrt{|X_{win}(k)|^2 - |U_{win}(k)|^2 \cdot \upsilon(k, p=2)}. \quad (12)$$

A more general form can be written as:

$$|Y_{win}(k)| = \left(|X_{win}(k)|^p - |U_{win}(k)|^p \cdot \upsilon(k,p)\right)^{\frac{1}{p}}. \quad (13)$$

This is often named parametric spectral subtraction [5] and sets a variable exponent. With $p = 1$ we obtain the spectral subtraction from (10) and with $p = 2$ we obtain the spectral subtraction of the power from (12).

Combined with the phase $\angle X_{win}(k)$ of the input signal $X_{win}(k)$, the output signal can be computed with:

$$Y_{win}(k) = |Y_{win}(k)| \cdot e^{j\angle X_{win}(k)}. \quad (14)$$

To an extent, this operating sequence is a makeshift method. It is to be expected that after the subtraction, the correct phase of $Y_{win}(k)$ is not identical to the phase of the input signal $X_{win}(k)$. Jens Groh asserts that the correct phase often cannot be derived [6]. Thus, in many cases, the correct phase of the output signal is simply unknown. Studies have shown, that phase corruption in the spectral domain is considerably less perceptible than a corruption of the amplitude in this domain [10].

Finally, the output signal can be transformed back into the time domain by using the iDFT:

$$y_{win}(n) = \frac{1}{N}\sum_{k=0}^{N-1} Y_{win}(k) \cdot e^{j2\pi n\frac{k}{N}}. \tag{15}$$

Thereby the output signal is as long as the input signal and consists of N samples.

3. Spectral Subtraction as a Time-Variant System

The Spectral Subtraction can be considered as a time-variant system with a varying processing and parameters that can change from window to window.

Hence, we are able to write the subtraction in the spectral domain from (13) as a multiplication:

$$\begin{aligned} |Y_{win}(k)| &= \left(|X_{win}(k)|^p - |U_{win}(k)|^p \cdot \upsilon(k,p)\right)^{\frac{1}{p}} \\ &= |X_{win}(k)| \cdot \left(1 - \frac{|U_{win}(k)|^p}{|X_{win}(k)|^p} \cdot \upsilon(k,p)\right)^{\frac{1}{p}}. \end{aligned} \tag{16}$$

Then the amplitude response of this system is:

$$H_0(win,k) = \left(1 - \frac{|U_{win}(k)|^p}{|X_{win}(k)|^p} \cdot \upsilon(k,p)\right)^{\frac{1}{p}}, \tag{17}$$

and the frequency response of each window is:

$$H_{win}(k) = H_0(win,k) \cdot e^{j\angle H_{win}(k)}. \tag{18}$$

Assuming $\angle H_{win}(k) = \angle X_{win}(k)$, the equations (17) and (18) leads us to:

$$H_{win}(k) = \left(1 - \frac{|U_{win}(k)|^p}{|X_{win}(k)|^p} \cdot \upsilon(k,p)\right)^{\frac{1}{p}} \cdot e^{j\angle X_{win}(k)}. \tag{19}$$

Like the input signal $X_{win}(k)$, the frequency response consists of N DFT-coefficients. Thus, the spectral output signal $Y_{win}(k)$ can be computed as a product of the spectral input signal and the frequency response $H_{win}(k)$:

$$Y_{win}(k) = X_{win}(k) \cdot H_{win}(k) \tag{20}$$

A multiplication in the spectral domain corresponds to a convolution of the equivalent signals in the time domain [2]:

$$\begin{aligned} y_{win}(n) &= x_{win}(n) * h_{win}(n) \\ &= \sum_{m=0}^{N_{IR}-1} x_{win}(n) \cdot h_{win}(n-m), \end{aligned} \tag{21}$$

where $h_{win}(n)$ describes the impulse response of the system and N_{IR} is the length of this impulse response.

4. The Fundamental Problem

4.1. The Length of the Output Signal

The length of the output signal of a convolution is [2]:

$$N_{conv} = N_{input} + N_{IR} - 1, \tag{22}$$

where N_{input} is the length of the input signal, N_{IR} is the length of the impulse response and N_{conv} is the length of the convolved signal.

Considering the convolution in (21), both the input signal and the impulse response are of length N. Therefore, the output signal consists of $2N-1$ samples.

This means, that the output signal computed using convolution in the time domain is nearly twice as long as the output signal which is computed using Spectral Subtraction in the spectral domain and which has N samples. Thus, the output signal $y_{win}(n)$ in (21) cannot be the same as the output signal in (15) with (13) and (14), as we can see in Fig. 6.

The reason for this is the static signal length in the spectral domain and the periodicity of the DFT. The periodicity presupposes a continuous repetition of the finite output signal. The modifications of the DFT coefficients cause an extension of the signal when transformed back into the time domain. The part of the processed signal after the Nth sample will be continued at the beginning of the window. Since the STFT does not take this repetition at the recombination of the windows into account, an error inevitably occurs. We receive an overlap with a signal part, which is inserted at the wrong time position. This error becomes apparent when the signal is compared directly with the output signal, which is computed by convolution in the time domain. In Fig. 6 we can see the differences between the output signal of the Spectral Subtraction and the output signal of the convolution.

4.2. Zero Padding is no Solution

By using zero padding, we can reduce the effective length of the input signal in relation to the length of the window $N_{input} + L$. Consequently, the length of the frequency response $H_{win}(k)$ increases and for this reason the length of the impulse response $h_{win}(n)$ will increase up to the extended window length of $N_{input} + L$ samples.

The constraint that the output signal fits into the window without an overlap is only fulfilled in the case of $N_{input} = 1$:

$$\begin{aligned} N_{input} + N_{IR} - 1 &\leq N_{input} + L \\ 2N_{input} + L - 1 &\leq N_{input} + L \\ N_{input} &\leq 1. \end{aligned} \tag{23}$$

This case is unusable for Fourier analysis.

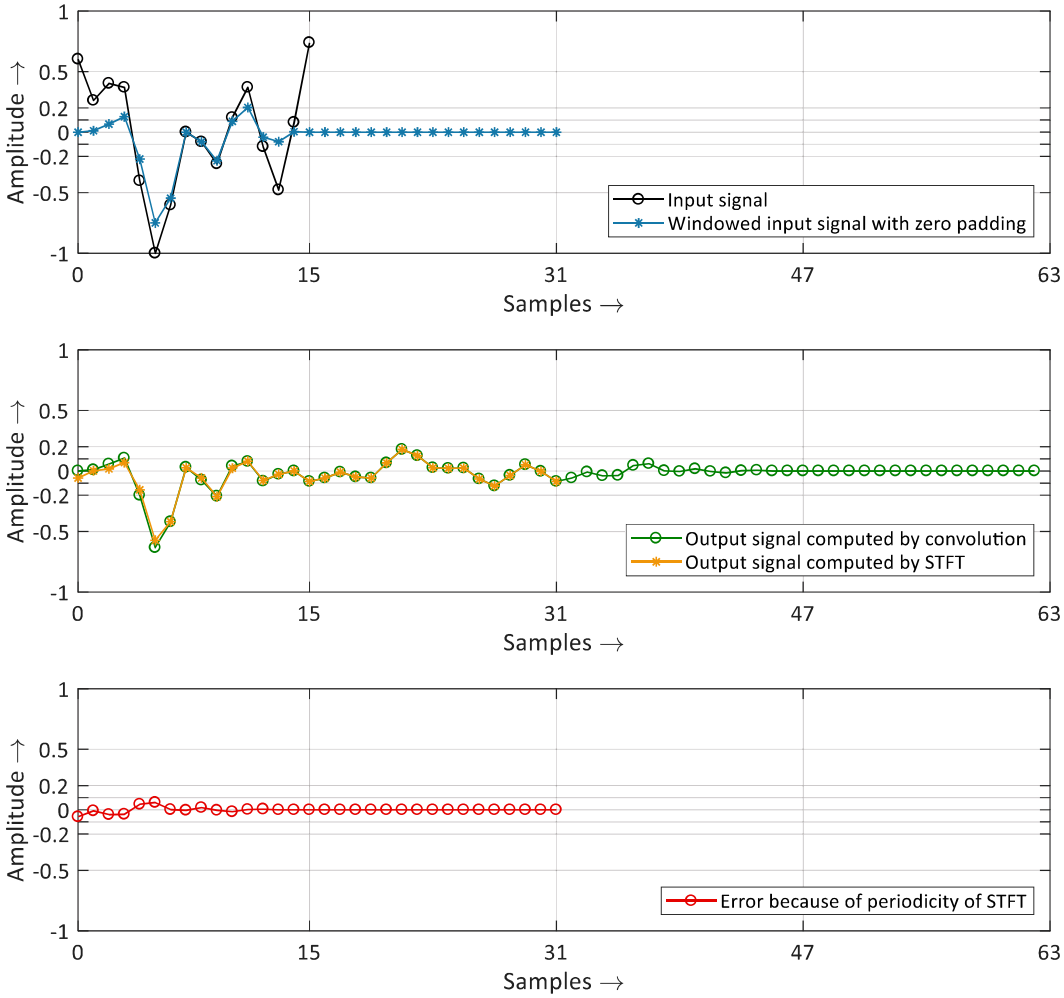


Fig. 6: Comparison of Spectral Subtraction using STFT and the equivalent processing with a convolution in the time domain. A windowed test signal of 16 samples is processed with Spectral Subtraction in the spectral domain and with a convolution in the time domain. The last diagram shows the signal part which is at the wrong position in the output signal when processed with the Spectral Subtraction.

4.3. An Example to Illustrate

To illustrate the behaviour of the DFT in combination with Spectral Subtraction we generate a window of a synthetic input signal:

$$x(n) = \cos\left(2\pi\frac{n}{N}\right) + \sin\left(4\pi\frac{n}{N}\right) + \cos\left(8\pi\frac{n}{N}\right), \quad (24)$$

with $N = 16$ and $n = 0,1,2,\ldots,N-1$. The result is the black graph in Fig. 6. We multiply this input signal with the von-Hann window function from (3):

$$x_{win}(n) = x(n) \cdot w_{hann}(n)\,. \quad (25)$$

To get a better frequency resolution we add 16 zeros:

$$\tilde{x}_{win}(n) = \begin{cases} x_{win}(n) & for\ 0 \le n \le 15 \\ 0 & for\ 16 \le n \le 31\,. \end{cases} \quad (26)$$

We receive the windowed input signal with zero padding, as illustrated by the blue graph of Fig. 6.
As an example, we reduce the third, fifth and ninth DFT coefficients by about 70%, using Spectral Subtraction and (4), (10) and (14). The result is the output signal of the Spectral Subtraction, shown as the orange graph. Now we compare this result with the equivalent processing using convolution in the time domain. By means of (19) with $p = 1$ and (21), we receive the green graph. The difference of these two output signals (red graph) shows the wrongly inserted part of the signal, occurring due to the periodicity of the DFT.

5. Analysis of the Impulse Response

If we look to the impulse response $h_{win}(n)$ of the Spectral Subtraction, which is the inverse Fourier transform of $H_{win}(k)$:

$$h_{win}(n) = \frac{1}{N}\sum_{k=0}^{N-1} H_{win}(k) \cdot e^{j2\pi n\frac{k}{N}}, \quad (27)$$

it becomes apparent, that the maximum of the impulse response is located at the first sample $n = 0$, as we can see in Fig. 7.

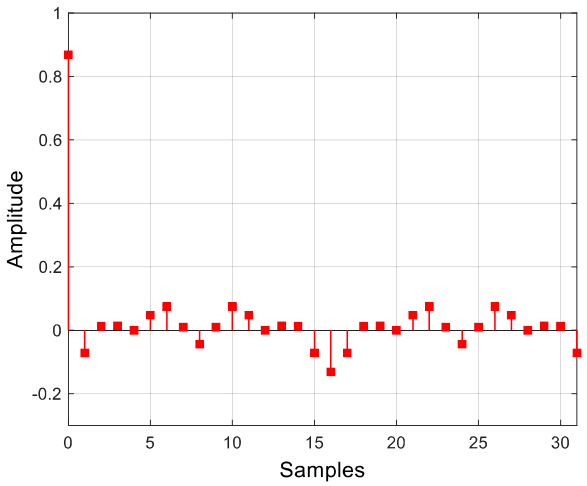


Fig.7: Impulse response of the Spectral Subtraction, computed with (10) and (14). The third, fifth and ninth DFT coefficients are reduced by ~70%.

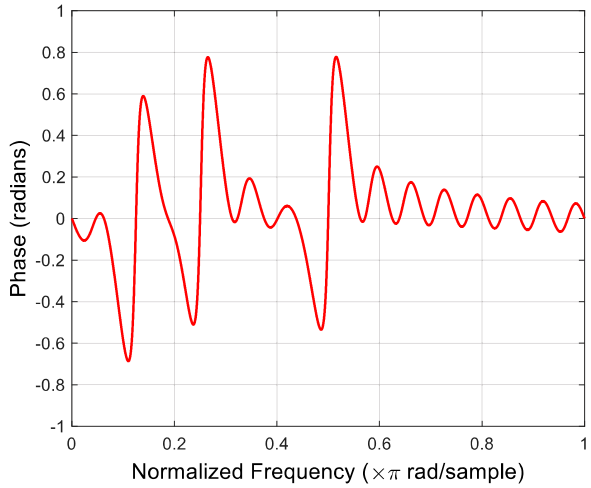


Fig. 8: Phase response of the Spectral Subtraction, computed with (10) and (14). The third, fifth and ninth DFT coefficients are reduced by ~70%.

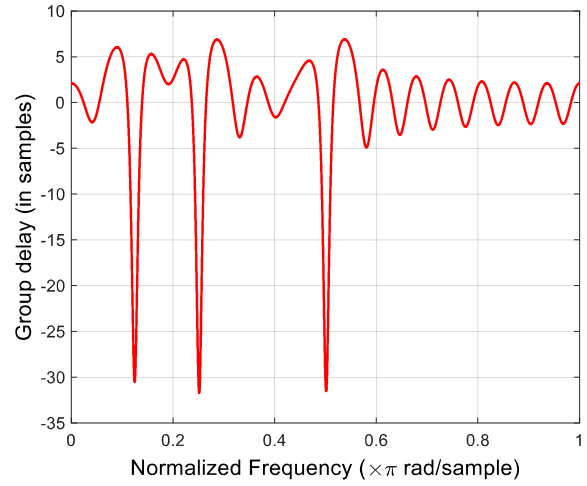


Fig. 9: Group delay response of the Spectral Subtraction, computed with (10) and (14). The third, fifth and ninth DFT coefficients are reduced by ~70%.

Furthermore, the samples $n = 1$ to $n = 31$ are axis-symmetric to $n = 16$. This impulse response behaves as if multiplied with the Heaviside step function:

$$\mathfrak{H}(n) = \begin{cases} 0 & for\ n < 0 \\ 1 & for\ n \geq 0 \end{cases}, \tag{28}$$

and shows a nonlinear phase shift, as we can see in Fig. 8 and a strong varying group delay depending on frequency, as we can see in Fig. 9. We obtain the strongest group delay at the three processed DFT coefficients.

To prevent nonlinear phase shifting and an inconstant group delay, we must shift the phase within the processing in the spectral domain, depending on frequency. The phase of the DFT coefficients representing high frequencies with a short wavelength have to be shifted more than the phase of DFT coefficients representing low frequencies. For an impulse response with an even length and an even symmetry we obtain the phase difference [2][9]:

$$\theta(k) = -\frac{N-1}{2}\Omega, \tag{29}$$

where $\Omega = 2\pi k/N$ is the normalized complex angular frequency. If we include this phase difference in (14), we receive:

$$Y_{win}(k,\theta) = |Y_{win}(k)| \cdot e^{j\angle X_{win}(k)} \cdot e^{j\theta}. \tag{30}$$

We can call this enhanced algorithm 'Advanced' Spectral Subtraction.

In Fig. 10–12 we can see the symmetric impulse response, the linear phase response and the constant group delay of the Advanced Spectral Subtraction using (10) and (30).

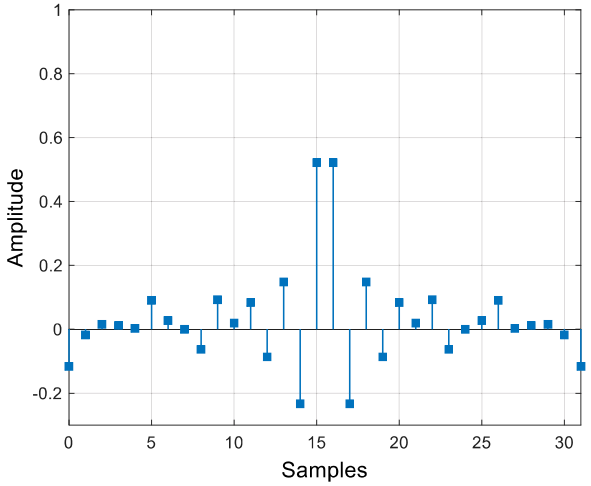


Fig. 10: Impulse response of the Spectral Subtraction, computed with (10) and (30). The third, fifth and ninth DFT coefficients are reduced by ~70%.

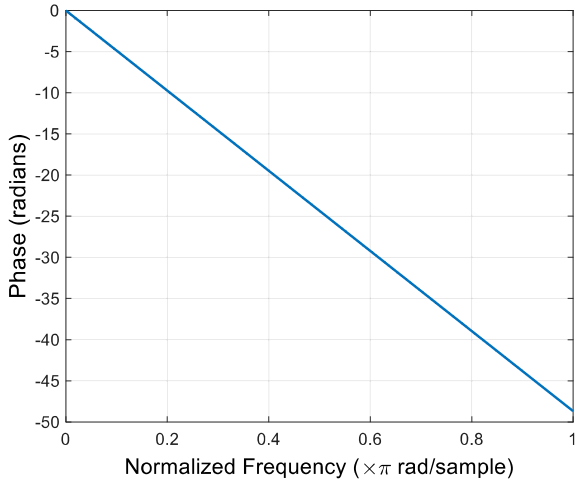


Fig. 11: Phase response of the Spectral Subtraction, computed with (10) and (30). The third, fifth and ninth DFT coefficients are reduced by ~70%.

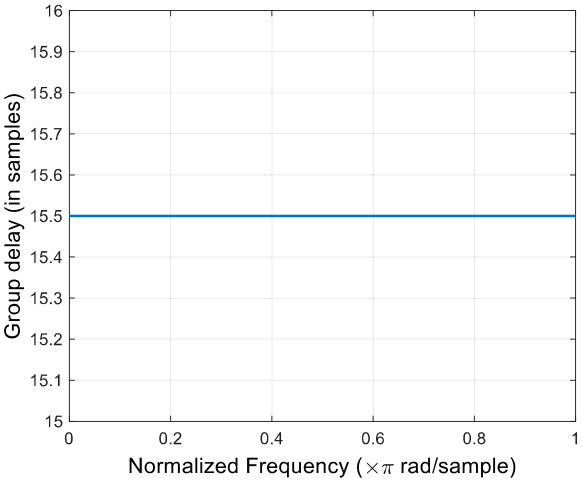


Fig. 12: Group delay response of the Spectral Subtraction, computed with (10) and (30). The third, fifth and ninth DFT coefficients are reduced by ~70%.

As we can see in Fig. 12, the Advanced Spectral Subtraction results in a constant group delay, which also means that the processing has a latency of one half window length.

Finally, we can take a look at the two magnitude responses, computed by Spectral Subtraction using (10) and (14) and by the Advanced Spectral Subtraction using (10) and (30).

The two magnitude responses show strong similarity. We can see the three attenuations, with the red one providing a slightly narrower band width. It also becomes apparent that the Spectral Subtraction with linear phase has a low-pass behaviour at very high frequencies. This is the result of an impulse response with an even length and an even symmetry [2][9]. In the vast majority of cases, this behaviour is of little to no consequence. For example, in digital audio signal processing with a sampling frequency of $f_s = 48kHz$, the cut off is located above the upper limit of human perception.

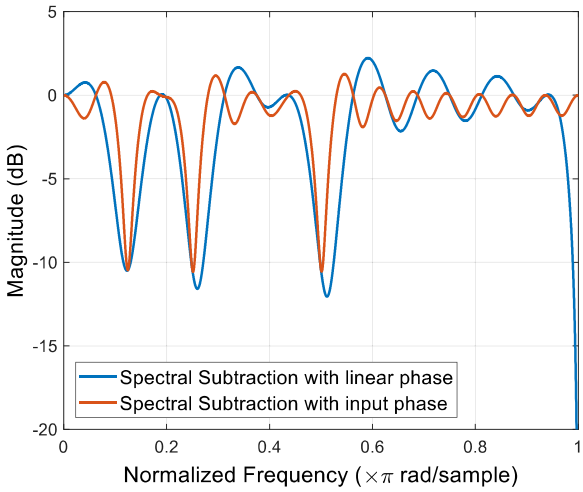


Fig. 13: Magnitude response of the Spectral Subtraction. The blue graph is computed with (10) and (30) and the red graph is computed with (10) and (14). The third, fifth and ninth DFT coefficients are reduced by ~70% within the processing.

6. Conclusion

We can state that processing in the frequency domain makes the signal longer. The signal part by which the output signal is longer than the input signal corresponds to the transient effect and decay process of the impulse response. The crucial point is to arrange the transient and decay parts at the correct time position in the output signal.
If we do the processing in the spectral domain via STFT, because of the periodicity, we receive an overlap in the output signal during resynthesis. This means, that we have a signal part at the wrong time position. Since the STFT does not take this repetition into account, an error inevitably occurs.

The subjective perception of this error is relatively small. Furthermore, it is not the reason of the artefact called 'musical noise'. Presumably, the resulting error is covered by stronger artefacts like the aforementioned 'musical noise', which can occur because of a dynamic processing in the spectral domain, too.

Irrespective of this, it is recommended to work around this error. For example, the resulting amplitude response can be smoothed. This approach minimizes the error, but it does not completely prevent it. To obtain the correct output signal, the frequency response can be generated. By means of the iFFT, we receive the impulse response of the processing. Now it is possible to compute the output signal with convolution of the windowed input signal and the impulse response in the time domain. This means, that the algorithm has more calculating steps and needs more time for the processing. However, with the fast convolution we have a fast-acting tool, which uses the fast Fourier Transform FFT.

The question arises as to why the fast convolution can compute the output signal without an error while still using the DFT. When we use the fast convolution, we have the

windowed input signal and the complete processing information within the impulse response. We don't have to generate the frequency response in the spectral domain. The fast convolution fills up the windowed input signal and the impulse response with enough zeros to fit the entire output signal into the window.
This is still not possible if we generate the frequency response of the Fourier transformed window with the input signal in the spectral domain, like the Spectral Subtraction does. In this case, the frequency response and for this reason the impulse response are always as long as the transformed window. Therefore, the output signal does never fit into the window.

We can conclude that Spectral Subtraction has a fundamental problem within its approach. But it is possible to work around this weak spot and prevent the occurring error. Furthermore, we can use a phase shift within the processing, so that the 'Advanced' Spectral Subtraction does not have any nonlinear phase response or inconstant group delay.

7. References

[1] Benesty, J.; Chen, J.; Habets, E.A.P.: Speech Enhancement in the STFT Domain, Springer Briefs in Electrical and Computer Engineering. Springer, Berlin, 2011.

[2] Bellanger, M.G.: Digital Processing of Signals - Theory and Practice. John Wiley and Sons Ltd, Chichester, 2000.

[3] Boll, S. F.: Suppression of Acoustic Noise in Speech Using Spectral Subtraction. IEEE Tran. on Acoustics, Speech and Signal Processing ASSP-27, 2, 1979, S. 113-120.

[4] Briggs, W. L.; Van Emden, H.: The DFT - An Owners' Manual for the Discrete Fourier Transform. Society for Industrial and Applied Mathematics, Philadelphia, 1995.

[5] Etter, W., Moschytz, G. S.: Noise reduction by noise-adaptive spectral magnitude expansion. Journal of the Audio Engineering Society 42 (1994), S. 341-349.

[6] Groh, J.: Verringerung von Kammfilterverzerrungen bei Multimikrofonaufnahmen. Tagungsbericht 26. Tonmeistertagung (2010), S. 616-625.

[7] Küpfmüller, K.; Kohn, G.: Theoretische Elektrotechnik und Elektronik. Springer-Verlag, Berlin, Heidelberg, 2000.

[8] Neubauer, A.: DFT – Diskrete Fourier-Transformation. Springer Vieweg, Wiesbaden, 2012.

[9] Oppenheim, A.V.; Schafer, R.W.: Digital Signalprocessing. Prentice Hall, Englewood Cliffs, 1975.

[10] Vary, P.: Noise suppression by spectral magnitude estimation-mechanism and theoretical limits. Signal Processing 8 (1985), S. 387-400.

Listening Tests in the Process of Microphone Development

Hendrik Paukert[1], Jonathan Ziegler[1,2]
[1]Hochschule der Medien Stuttgart, Germany, Email: paukert@hdm-stuttgart.de
[2]Eberhard Karls Universität Tübingen, Germany Email: zieglerj@hdm-stuttgart.de

Abstract

Preliminary listening tests play a key role in the development of novel types of digitally enhanced microphone arrays. The assessment of different types of noise and signal degradation can lead to a better understanding of which factors need the most attention in future development of signal processing algorithms. Along with the results, the principles of the listening test will be discussed, as well as the creation of suitable sound files.

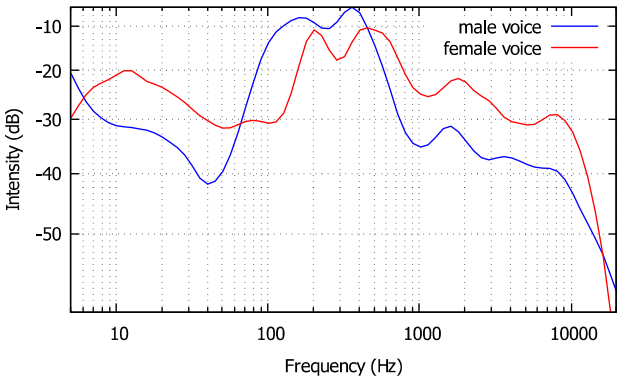
1. Preparation

The entire listening test was programmed using Max/MSP, an object oriented programming environment created by Cycling'74 [1]. This was chosen for its powerful and straightforward audio manipulation capabilities and the ability to quickly design a GUI for the test subjects. The test was comprised of pair comparisons [2], rankings [3] and active evaluations. To match the DSP algorithms for which the listening test was devised, the selected noise sources are jitter, compression artifacts and various colors of random noise. The creation of degraded speech signals took place using band pass filters, gates, limiters and audio clipping. All test subjects experienced the test on the same laptop with Beyerdynamic DT770 headphones [4] and data was acquired automatically. The set of test subjects consisted of audio professionals between 28 and 55 years of age. Special thanks go to Johanna Zehendner and Jo Jung for the generous contribution of speech samples [5-7].

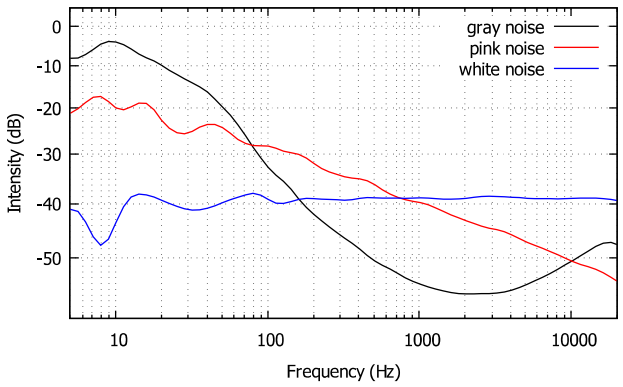
2. Synthesis of noise samples

White and pink noise were created using the noise generators integrated in Max/MSP. In addition, a type of noise was introduced to the test, which matches the spectral sensitivity of human hearing and thus should be more tolerable to an average listener. This was achieved by modeling white noise with appropriate equalization. As seen in Image 1, gray noise has a strong attenuation around 2000 Hz, which matches the heightened sensitivity of human hearing at this frequency [8].

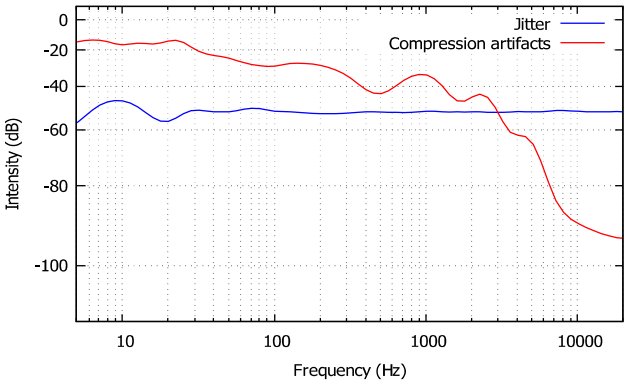
Jitter was captured by recording the signal of a damaged Toslink optical ADAT cable connecting a digital console to a recording interface. Compression artifacts were created using a specifically designed Max/MSP patch. Peak and RMS levels of the signals were analyzed using Audacity [9].



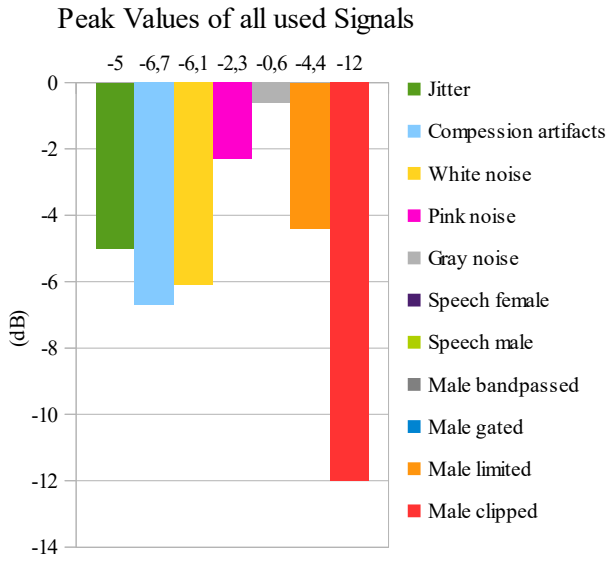
Img. 2: Spectral energy distribution of the speech samples used in tests 3-6.



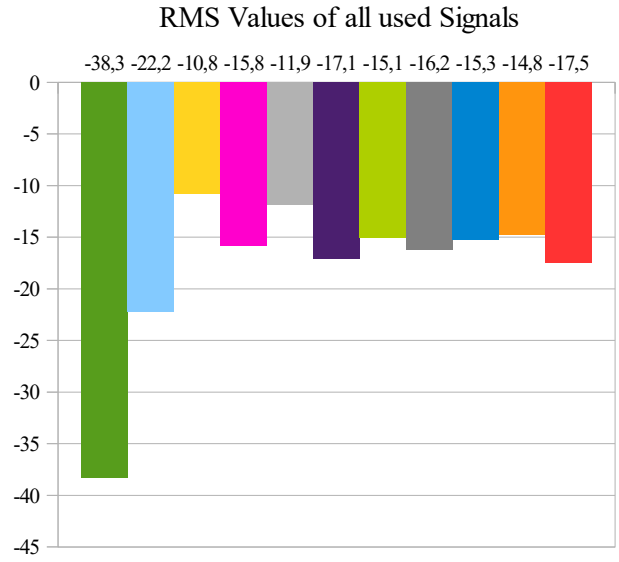
Img. 1: Overlay of white, pink and gray noise. White noise shows an even energy distribution over the entire audible spectrum, whereas pink and gray noise have specific spectral attributes.



Img. 3: Spectral analysis of compression artifacts and jitter. The compression artifacts contain little energy above 5000 Hz.



Img. 4: Peak values of all signals used in the listening tests. Before evaluation of the results individual gain matching was applied. 'Speech female', 'Speech male', 'Male band passed' and 'Male gated' have 0dB peak level.



Img. 5: RMS values of used signals. While jitter and white noise have similar peak levels, the difference in mean energy is significant.

3. Execution of listening tests

The program in use consists of 6 separate experiments and returns a total of 52 parameters for each test subject. Before the test is started, the subject is reminded that the listening test is devised for a speech-specific microphone and that therefore a focus should be placed on sound quality in regard to such signals.

Test 1: "Identical Disturbance Level"

Initially the test subject is asked to start a calibration signal to set the listening volume to a comfortable level. This process ensures that every listener is evaluating the signals within his or her own listening comfort zone.
Due to a variety of sonic differences in the noise samples, a direct comparison is not possible. A recording of white noise will be perceived to be louder than, for example, jitter at the same peak level. Therefore, every subject is asked to set the noise samples to a subjectively identical level. These gain values are consequently incorporated into the pair comparison tests.

Test 2: "Pair Comparison"

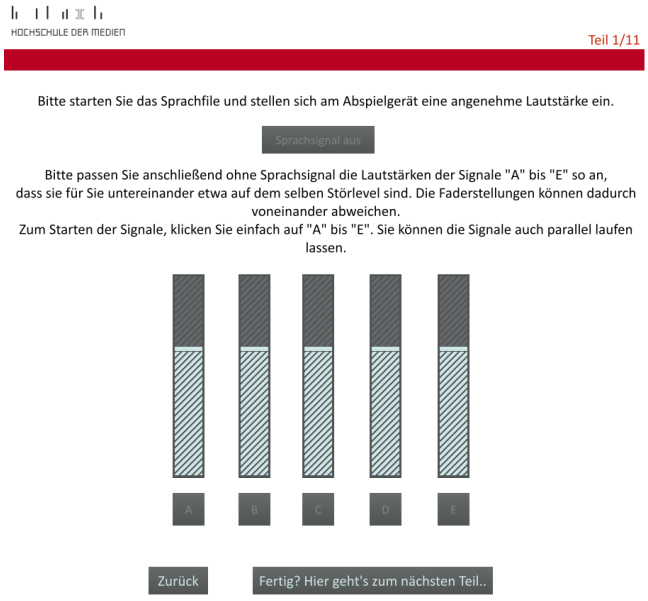
All noise signals are evaluated in pairs and the signal with a higher disturbance potential is chosen. Due to the fact that the test subject previously matched all signals to a subjectively identical level, the decision is based more on spectral and temporal energy distribution than on a general difference in volume between the signals.

Test 3: "Perception Threshold"

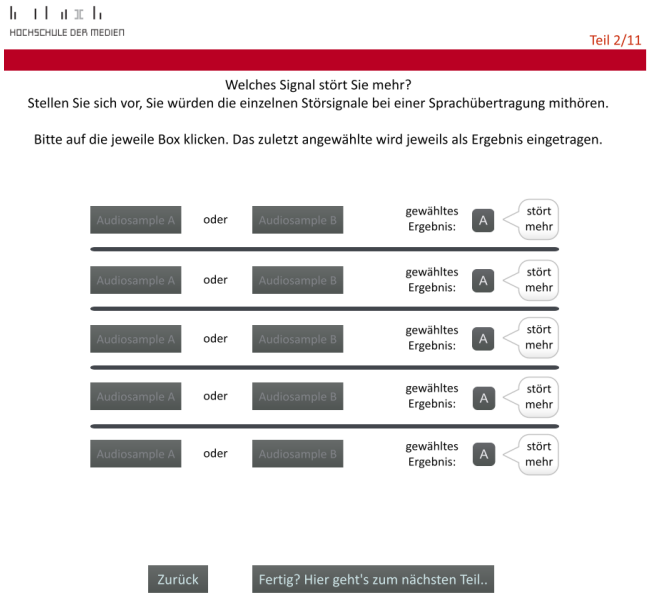
The signals are now examined based on an individual threshold of perception. In addition to the noise samples, a vocal sample is played back. Both male and female speech are used in order to examine a difference in signal masking. The user interface is similar to the one used in test 1.

Test 4: "Disturbance Threshold"

The disturbance threshold for each signal is determined using the same methodology as in the previous test.



Img. 6: "Identical Disturbance Level." The test subject is asked to set the noise signals to identical levels using the provided faders and on/off buttons.



Img. 7: “Pair Comparison.” The test subject is asked to determine the more bothersome signal within a pair.

Test 5: “Ranking”

The male speech sample is now played back in a variety of modifications, created with a gate, a band pass filter, a limiter and a clipper. All signals were modified to a similar degree. This insures that the character of a modification is perceived rather than its intensity. The subject is asked to rank the sound samples according to perceived signal quality. This test contains a blind reference.

Img. 8: “Ranking.” The subject compares the modified speech signals by clicking the tiles A-E and assigning a numerical rank to each sample.

Test 6: “Clipping”

In this test the amplitude of the audio samples of male and female speech are clipped. The subject is asked to set a tolerable level of clipping using a number box, as shown in Image 9. To prevent habits of audio professionals from influencing their decisions, no level meters are supplied.
The factor is converted to decibels for the evaluation process and can be compared to the fixed clipping from test 5.

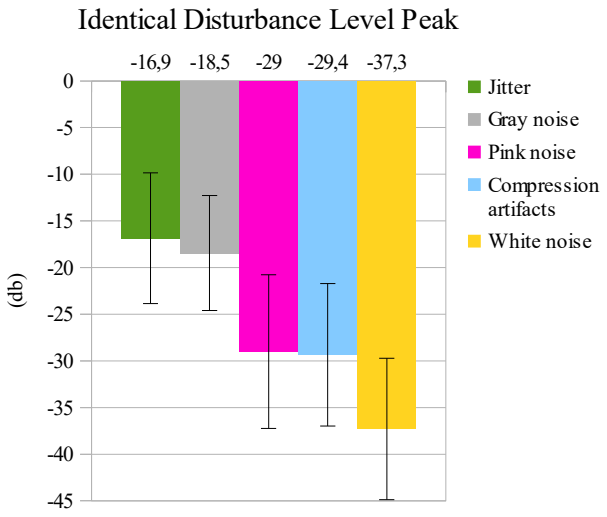
Img. 9: “Clipping.” Test subjects set a tolerable level of clipping in male and female voice samples using a number box.

4. Results

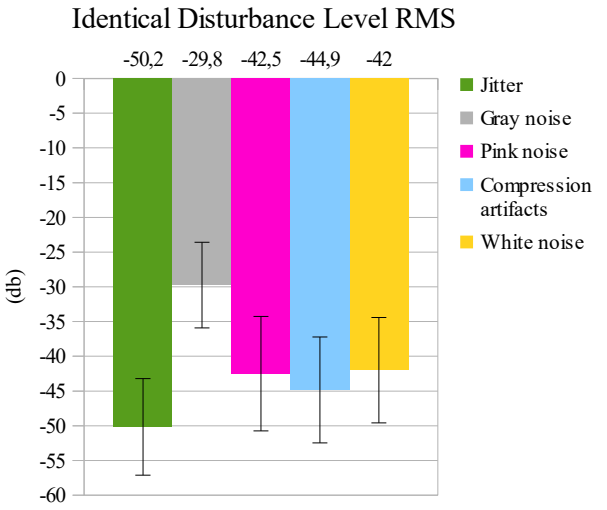
All results are gain-corrected using the peak level offsets shown in Image 4. By compensating for differences in peak and RMS levels, a uniform evaluation of all tests is achieved. The depicted results are gathered using averages of all test subjects.

Test 1: “Identical Disturbance Level”

Image 10 makes it quite clear that gray noise and jitter have a very high disturbance threshold. Thus, much higher peak levels can be tolerated than, for example, with white noise. The test subjects set pink noise and compression artifacts to similar peak levels. This could very likely be due to the spectral similarity of the two signals. RMS values for the chosen levels show more similarity. This can be observed in Image 11. The only exception is gray noise, where much higher RMS values are chosen due to reduced signal energy in the frequency bands most sensitive in human perception. The average standard deviation is 7.31 dB.



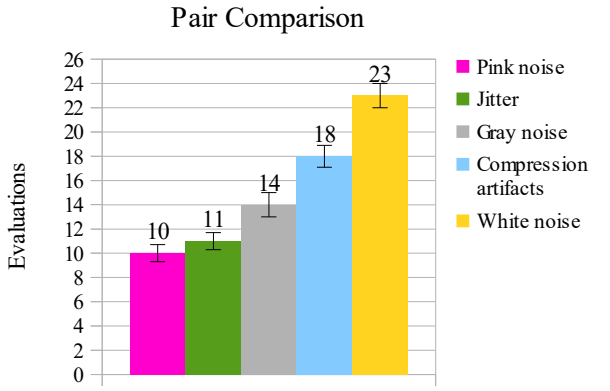
Img. 10: Results of "Identical Disturbance Level Peak." Jitter is set to the highest, white noise is set to the lowest peak level with a difference of approximately 20 dB.



Img. 11: Results of "Identical Disturbance Level RMS." Compared to peak levels in Image 5, RMS levels show much higher correlation.

Test 2: "Pair Comparison"

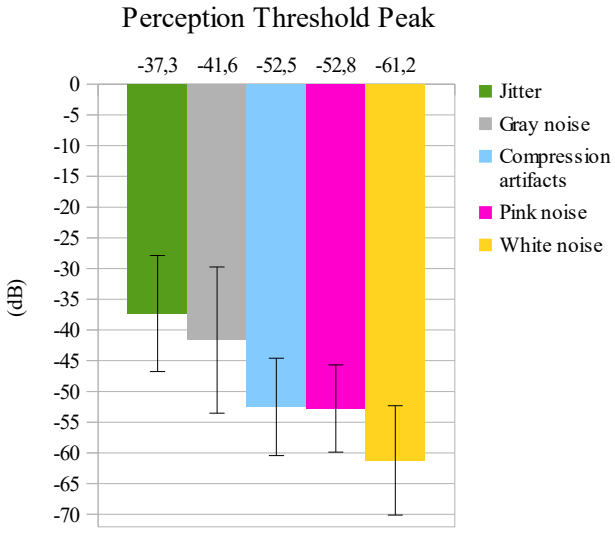
Pair comparison tests are run using the resulting disturbance levels from test 1, thus a consistent level of audible noise is achieved. Each noise sample is compared to every other noise sample, resulting in 10 choices per test subject. During evaluation the number of losing pair decisions is calculated per sample, determining the most disturbing source. As seen in Image 12, white noise clearly leads the list, followed by compression artifacts and gray noise. Pink noise and jitter were perceived to be the least bothersome.



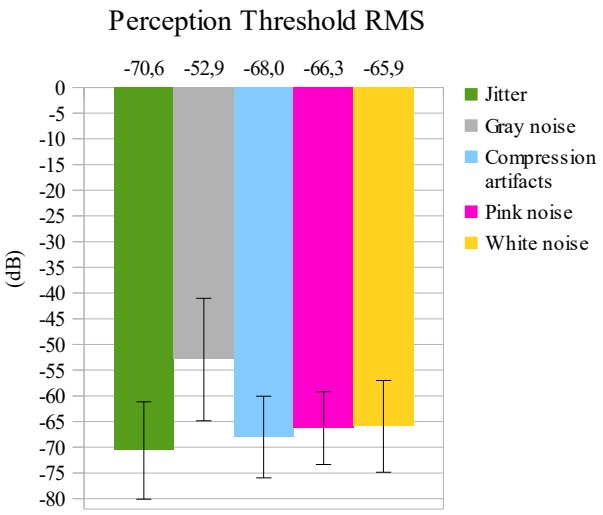
Img. 12: Results of "Pair Comparison." The list of most disturbing noise sources is clearly led by white noise. Pink noise and jitter were perceived as the least disturbing.

Test 3: "Perception Threshold" - Part I

The perception thresholds of jitter and gray noise are highest, followed by compression artifacts and pink noise. White noise, on the other hand, can already be heard at very low signal levels. The average standard deviation is 9.1 dB. Pink noise shows the lowest, and gray noise the highest standard deviation of the tested signals. This could be due to hearing capability of the test subjects. Gray noise has the highest spectral energy in low and high frequency ranges. The high frequency sensitivity of human hearing is decreased with age and over-exposition to high sound pressure levels. This can cause a higher fluctuation in perceived noise levels.



Img. 13: Results of "Perception Threshold Peak." Jitter can be added at the highest peak level without disturbance, while white noise can be detected at very low levels.



Img. 14: Results of "Perception Threshold RMS." A convergence of measured values can be detected. The highest standard deviation is found within gray noise.

Test 3: "Perception Threshold in Female Speech" - Part II

In contrast to the previous test, the order of the noise samples is changed: compression artifacts trade places with pink noise. The average standard deviation is 8.3 dB. As shown in Image 20, the perception threshold of the noise samples in female speech is on average slightly below the results in male speech. This can be attributed to a 2 dB higher RMS level of the male speech sample.

Comparative Analysis

When comparing perception thresholds of pure noise samples and noise in added speech, it becomes apparent that especially pink and gray noise can be increased in volume. Jitter and compression artifacts are masked least.

Perception threshold in female speech	Difference due to masking (dB Peak)	SD
Jitter	9.9	8.6
Compression artifacts	9.9	8.7
White noise	11.0	6.7
Pink noise	12.8	6.6
Gray noise	12.4	10.9
Average	*11.2*	*8.3*

Tab. 1: Results of "Perception Threshold in Female Speech." Pink noise profits most from masking effects and additionally shows the smallest standard deviation.

Test 3: "Perception Threshold in Male Speech" - Part III

The order of the samples is analogous to part 2 of this test and standard deviation is 7.8 dB.

Comparative Analysis

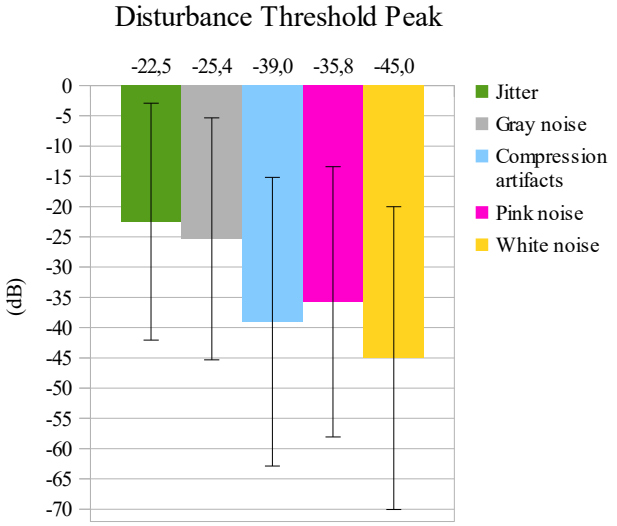
Masking effects are least apparent with jitter. As with female speech, pink and gray noise show the strongest masking characteristics. Additionally, pink noise shows an increase in masking of 1.5 dB compared to female speech.

Perception threshold in male speech	Difference due to masking (dB Peak)	SD
Jitter	9.2	9.0
Compression artifacts	9.8	8.5
White noise	11.8	7.0
Pink noise	14.3	5.1
Gray noise	12.7	9.4
Average	*11.6*	*7.8*

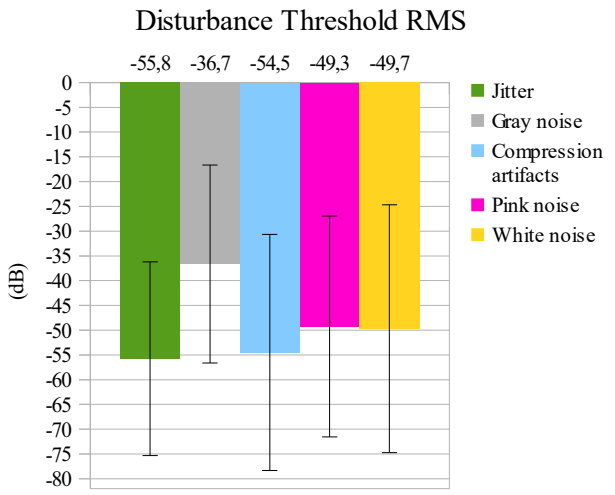
Tab. 2: Results of "Perception Threshold in Male Speech." As with female speech, pink noise shows both the most effective masking and lowest standard deviation.

Test 4: "Disturbance Threshold" - Part I

Compared to the tests concerning perception thresholds, pink noise shows a higher disturbance threshold and thus is slightly less bothersome. The highest levels are set for jitter, indicating the lowest relative disturbance among the compared signals. Compression artifacts and white noise show the lowest tolerance level. On average, the disturbance threshold is 15.6 dB above the perception threshold. The standard deviation of 22.2 dB on average is 13.1 dB higher than for the perception threshold. This could be due to a missing reference, unclear definitions of disturbance or diverging sensitivity for noise among test subjects.



Img. 15: Results of "Disturbance Threshold Peak." A significant difference is the greatly increased standard deviation.



Img. 16: Results of „Disturbance Threshold RMS." As with peak results, a significant fluctuation within the group of test subjects is registered.

Test 4: "Disturbance Threshold in Female Speech" - Part II

The average standard deviation is reduced through the introduction of a speech signal (female voice) from 22.2 dB to 15.6 dB. This level is still 6.5 dB above those of the tests concerning perception thresholds. The order of signals remains unchanged.

Comparative Analysis

Pink noise is masked most in the hearing tests with speech signals while the disturbance threshold changes most for jitter. Also, no direct correlation between the disturbance threshold and the perception threshold can be detected.

Disturbance threshold in female speech	Difference due to masking (dB Peak)	SD
Jitter	6.5	14.1
Compression artifacts	5.2	16.2
White noise	3.4	16.7
Pink noise	4.3	16.0
Gray noise	4.8	15.1
Average	*4.8*	*15.6*

Tab. 3: Results of "Disturbance Threshold in Female Speech." Concerning disturbance thresholds, jitter profits most from masking effects. The most notable difference to the tests concerning perception thresholds are the much higher standard deviations.

Test 4: "Disturbance Threshold in Male Speech" - Part III

The levels set by the test subjects are up to 2.2 dB higher than with female speech, thus confirming tendencies of the perception threshold tests. This is due to the 2 dB higher RMS value of the male speech sample. The order stays unchanged and standard deviation is at 15.5 dB.

Comparative Analysis

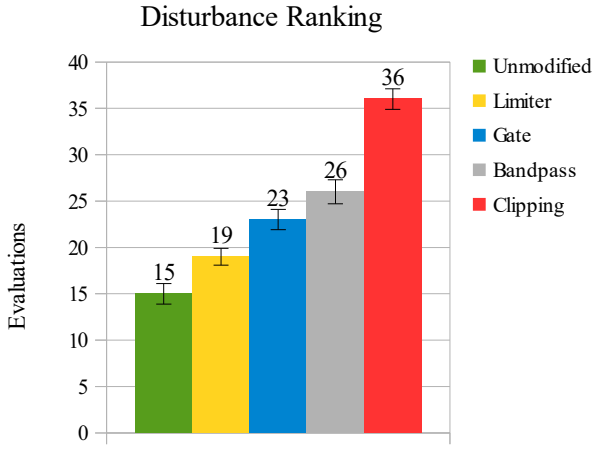
On average, the difference in masking is increased by 1.4 dB. In comparison to the perception threshold, masking occurs less, especially for gray noise. Jitter and compression artifacts profit most from masking effects.

Disturbance threshold in male speech	Difference due to masking (dB Peak)	SD
Jitter	7.0	14.1
Compression artifacts	7.3	18.3
White noise	5.5	16.4
Pink noise	6.5	13.8
Gray noise	5.0	14.9
Average	*6.2*	*15.5*

Tab. 4: Results of "Disturbance Threshold in Male Speech." Compression artifacts and jitter profit most from masking effects with male speech.

Test 5: "Ranking"

The unmodified sound sample is ranked highest with the signal treated with a limiter coming in second. The worst marks are given to the signals treated with clipping and band pass filters.



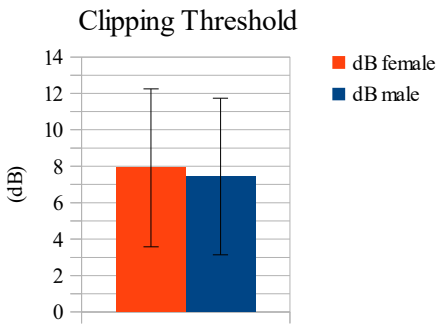
Img. 17: Results of "Disturbance Ranking." The original sample is ranked highest and the signal with clipping lowest.

Test 6: "Clipping Threshold"

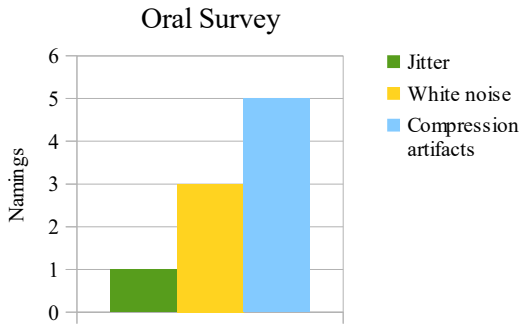
Clipping thresholds set by the test subjects are nearly identical between male and female speech with a difference of 0.5 dB. The higher RMS of the male speech sample could result in more noticeable clipping effects and would explain the lower threshold. Standard deviation is 4.3 dB for male speech and female speech. This indicates a very individual perception of disturbance.

Concluding Oral Survey

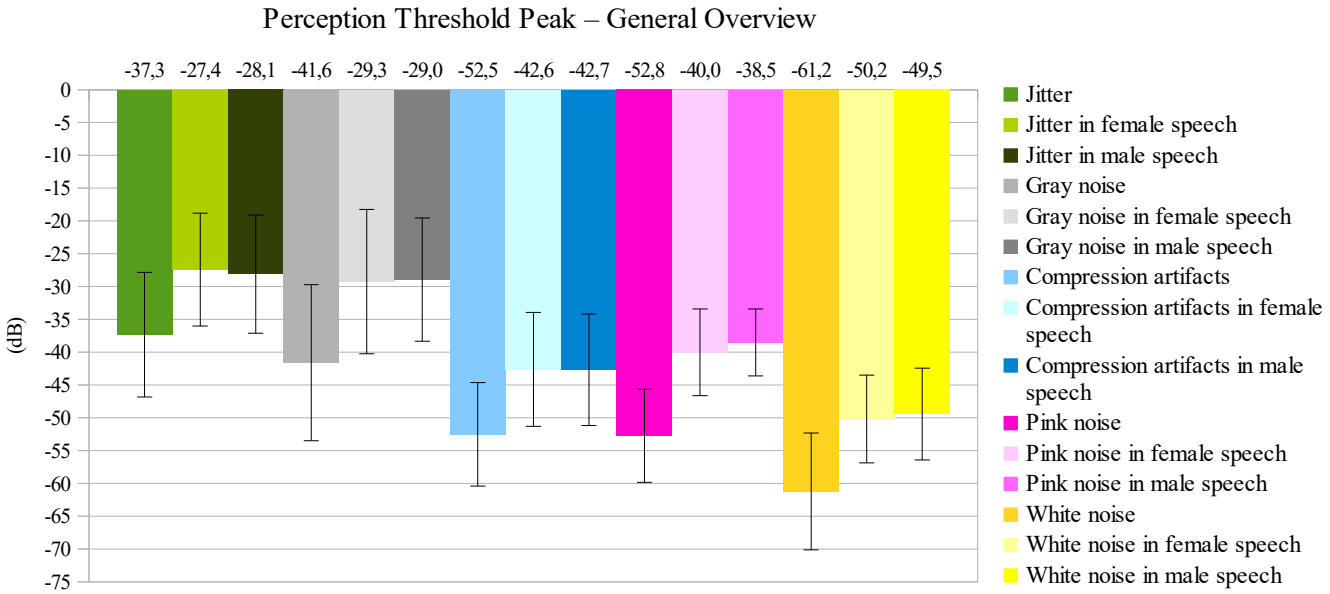
After completion of the test, the subjects were asked to name the least pleasant signal. The results of the survey can be seen in Image 19.



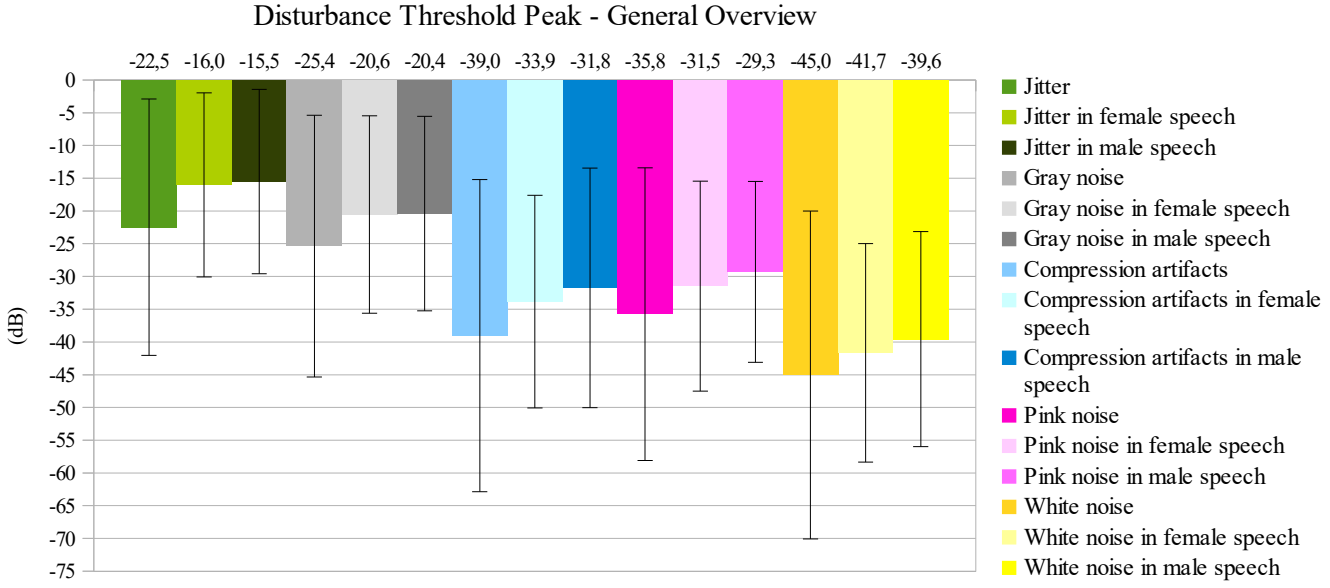
Img. 18: Results of "Clipping Threshold." Differences between male and female speech are below statistical relevance.



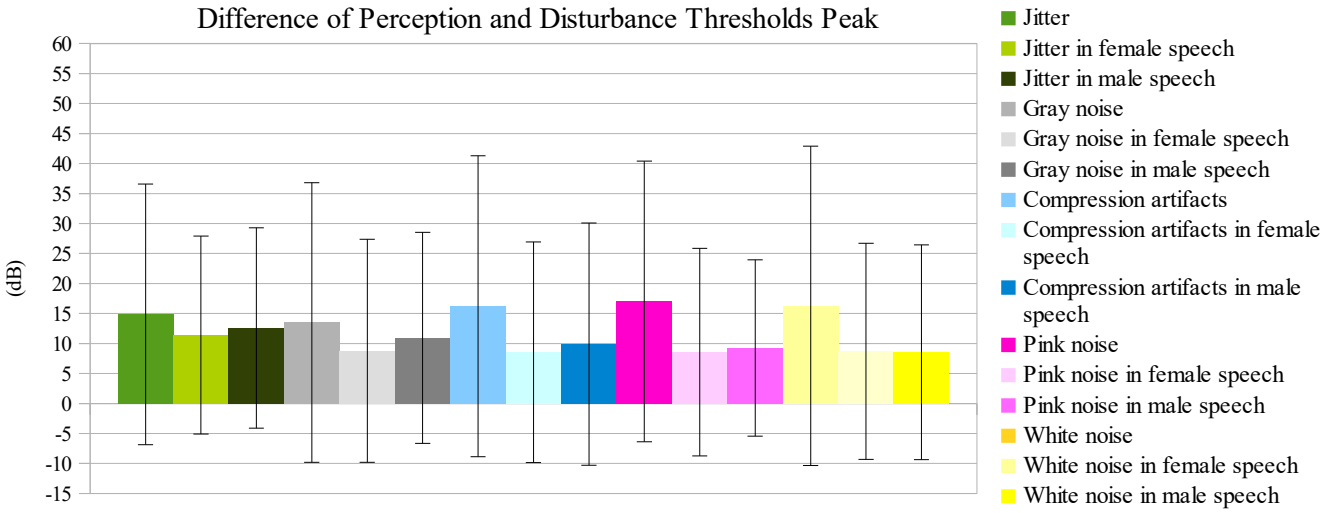
Img. 19: Results of "Oral Survey." Compression artifacts were perceived as the most disturbing. Of the five options, gray and pink noise were not mentioned at all.



Img. 20: Results of "Perception Threshold Peak." In general, white noise has the lowest and jitter the highest perception threshold. Pink noise profits most from masking effects by speech signals and has the lowest standard deviation.



Img. 21: Results of "Disturbance Threshold Peak." Jitter provides the least potential for disturbance, while white noise has the lowest disturbance threshold. The high standard deviation shows a wide variety of sensitivity towards noise in the test subjects.



Img. 22: Results of "Difference of perception and disturbance thresholds Peak." Gray noise shows the smallest level difference between perception and disturbance thresholds, thus being felt as disturbing relatively quickly after its perception. Pink noise has a threshold interval of 17dB and therefore is tolerated at levels well above the perception threshold. The pure noise samples uniformly show larger differences between perception and disturbance. The combined standard deviations of perception and disturbance thresholds create a larger spread.

5. Conclusion

In regard to peak levels, jitter provides the highest tolerability of all tested noise samples and can be present at the highest signal to noise ratio without being considered disturbing. White noise is detected and perceived as disturbing at much lower levels. Subjective determination of disturbance thresholds results in a significant spread of the results. The same occurs in perception thresholds of signals with a high percentage of spectral energy at the upper and lower end of human hearing.
Heavily clipped audio samples were considered to be of the poorest signal quality, while the unmodified signals were ranked highest. Additionally, modifications which increase intelligibility, such as noise gates and limiters did not significantly reduce the perceived signal quality.

6. Outlook

The described listening tests compose a foundation for further, more complex examinations within a larger project. The results will be used for prioritization within the development of DSP algorithms and for the creation of more detailed testing environments. If needed, various combinations of modified signals and noise sources can be surveyed within similar listening tests. Additional focus can be placed on the analysis of age distribution among test subjects.
More detailed evaluation of near-production prototypes will take place following the suggestions of ITU-R BS.1116[10] and ITU-R BS.1534[11]. In addition, listening tests with hidden reference and anchor (ABC/HR or MUSHRA) will be used.

7. References

[1] CYCLING '74, Max 7 perpetual licence, URL: https://www.cycling74.com

[2] HEAD Acoustics Application Note, Page 2, URL:https://www.headacoustics.de/de/nvh_application_notes_jury_evaluation.htm

[3] HEAD Acoustics Application Note, Page 1, URL: https://www.headacoustics.de/de/nvh_application_notes_jury_evaluation.htm

[4] Beyerdynamik GmbH & Co. KG, URL: http://www.beyerdynamic.de/shop/dt-770-pro.html

[5] Johanna Zehender, Radioplay „Menschlich ist", URL: https://www.johannazehendner.allyou.net

[6] Jo Jung, Radioplay „Menschlich ist", URL: http://www.jo-jung.eu

[7] A. Hummel, V. Kuptsov, M. Köhler, S. Kreuzer, H. Paukert: Radioplay „Menschlich ist", recorded at University of applied Science Stuttgart, URL: https://www.hdm-stuttgart.de/ URL: https://www.facebook.com/menschlichist/

[8] Michael Dickreiter, Volker Dittel, Wolfgang Hoeg, Martin Wöhr: Handbuch der Tonstudiotechnik, Band 1, 7. Auflage (2008), Saur Verlag, ISBN 979-3-598-11765-7, Seite 100, Abb.3/4

[9] Audacity, open source audio software, URL: http://www.audacityteam.org/

[10] International Telecommunication Union, URL: http://www.itu.int/rec/R-REC-BS.1116

[11] International Telecommunication Union, URL: http://www.itu.int/rec/R-REC-BS.1534

Submitted Patents

LICHTI

PATENTE · MARKEN · DESIGN

Lawo Holding AG
Am Oberwald 8
76437 Rastatt

14. Februar 2020
26136.9 Ka/jk

Extraktion eines Audioobjektes

Die Erfindung betrifft ein Verfahren zur Extraktion von mindestens einem Audioobjekt aus mindestens zwei Audio-Eingangssignalen, die jeweils das Audioobjekt enthalten. Ferner betrifft die Erfindung ein System zur Extraktion eines Audioobjektes und ein Computerprogramm mit Programmcodemitteln.

Im Sinne der Erfindung sind Audioobjekte Audiosignale von Objekten, wie beispielsweise das Geräusch beim Abschießen eines Fußballs, Klatschgeräusche eines Publikums oder der Vortrag eines Gesprächsteilnehmers. Die Extraktion des Audioobjektes im Sinne der Erfindung ist demgemäß die Separation des Audioobjekts von übrigen, störenden Einflüssen, die im Folgenden als Störschall bezeichnet sind. Beispielsweise wird bei der Extraktion eines Schussgeräuschs beim Fußballspiel das reine Schussgeräusch als Audioobjekt von den Geräuschen der Spieler und des Publikums separiert, so dass das Schussgeräusch schließlich als reines Audiosignal vorliegt.

2020000188

POSTFACH 410760
D-76207 KARLSRUHE

BERGWALDSTR. 1
D-76227 KARLSRUHE

TEL: +49 (0)721 9 43 28 15
FAX: +49 (0)721 9 43 28 40

eMAIL: mail@lichti.eu

Aus dem Stand der Technik sind gattungsgemäße Verfahren bekannt, die Extraktion von Audioobjekten vorzunehmen. Eine grundlegende Herausforderung ist dabei, dass üblicherweise die Mikrofone zur Quelle des Audioobjekts unterschiedlich beabstandet sind. Daher befindet sich das Audioobjekt an unterschiedlichen zeitlichen Positionen der Audio-Eingangssignale, was die Auswertung erschwert und verlangsamt.

Es ist bekannt, die Audio-Eingangssignale derart zu synchronisieren, damit sich das Audioobjekt insbesondere an der jeweils gleichen zeitlichen Position der Audio-Eingangssignale befindet. Dies wird üblicherweise auch als Laufzeitkompensation bezeichnet. Übliche Verfahren nutzen diesbezüglich neuronale Netzwerke. Dabei ist es erforderlich, dass das neuronale Netzwerk auf sämtliche mögliche Mikrofonabstände zur Quelle des Audioobjektes trainiert werden muss. Gerade bei dynamischen Audioobjekten, wie im Falle von Sportveranstaltungen, ist ein effektives Training des neuronalen Netzes aber nicht durchführbar.

Ferner sind gattungsgemäße Verfahren bekannt, bei denen zur Synchronisierung der Audio-Eingangssignale deren Korrelation, beispielsweise deren Kreuzkorrelation, analytisch berechnet wird, was zwar die Geschwindigkeit des Verfahrens steigert, aber die Zuverlässigkeit der nachfolgenden Extraktion des Audioobjekts beeinträchtigt, da die Korrelation stets unabhängig von der Art des Audioobjekts berechnet wird. Dabei werden aber oft für die nachfolgende Extraktion des Audioobjekts störende Effekte, insbesondere Störschall, verstärkt.

Es ist daher die Aufgabe der Erfindung, die genannten Nachteile aus dem Stand der Technik zu beseitigen und insbesondere die Zuverlässigkeit der Extraktion des Audioobjektes zu verbessern bei gleichzeitiger Optimierung der Geschwindigkeit des Verfahrens.

Die Aufgabe wird gelöst durch ein Verfahren mit den Merkmalen des Anspruchs 1, der ein Verfahren zur Extraktion von mindestens einem Audioobjekt aus mindestens zwei Audio-Eingangssignalen vorsieht, die jeweils das Audioobjekt enthalten, mit den folgenden Schritten: Synchronisieren des zweiten Audio-Eingangssignals mit dem ersten Audio-Eingangssignal unter Erhalt eines synchronisierten zweiten Audio-Eingangssignals, Extrahieren des Audioobjekts durch die Anwendung von mindestens einem trainierten Modell auf das erste Audio-Signal und auf das synchronisierte zweite Audio-Eingangssignal und Ausgabe des Audioobjekts,
wobei der Verfahrensschritt des Synchronisierens des zweiten Audio-Eingangssignals mit dem ersten Audio-Eingangssignal die folgenden Verfahrensschritte umfasst: Generieren von Audio-Signalen durch Anwendung eines ersten trainierten Operators auf die Audio-Eingangssignale, analytische Berechnung einer Korrelation zwischen den Audio-Signalen unter Erhalt eines Korrelationsvektors, Optimieren des Korrelationsvektors mit Hilfe eines zweiten trainierten Operators unter Erhalt eines Synchronisationsvektors und Bestimmen des synchronisierten zweiten Audio-Eingangssignals mit Hilfe des Synchronisationsvektors.

Ferner wird die Aufgabe durch ein System zur Extraktion eines Audioobjektes aus mindestens zwei Audio-Eingangssignalen mit einer Steuereinheit gelöst, die dazu ausgebildet ist, das erfindungsgemäße Verfahren durchzuführen.

Überdies wird die Aufgabe durch ein Computerprogramm mit Programmcodemitteln gelöst, das dazu ausgestaltet ist, die Schritte des erfindungsgemäßen Verfahrens durchzuführen, wenn das Computerprogramm auf einem Computer oder einer entsprechenden Recheneinheit ausgeführt wird.

Die Erfindung basiert auf der Grundüberlegung, dass durch die analytische Berechnung der Korrelation, beispielsweise der Kreuzkorrelation, die Qualität des extrahierten Audioobjekts, also die Signaltrennungsqualität des Verfahrens, verbessert wird. Gleichwohl wird durch die Ausbildung des ersten und des zweiten trainierten Operators eine Möglichkeit geschaffen, mit Hilfe von trainierten Komponenten die Zuverlässigkeit der nachfolgenden Extraktion des Audioobjektes zu verbessern. Insofern stellt die Erfindung ein neuartiges Verfahren dar, das die Extraktion des Audioobjektes zuverlässig und schnell durchführt. Dadurch ist das Verfahren auch bei komplexen Mikrofongeometrien, wie beispielsweise großen Mikrofonabständen einsetzbar.

Der erste trainierte Operator kann eine insbesondere trainierte Transformation der Audio-Eingangssignale in einen Merkmalsraum umfassen, um die nachfolgenden Verfahrensschritte zu vereinfachen. Der zweite trainierte Operator kann mindestens eine Normierung des Korrelationsvektors umfassen, um die Genauigkeit der Berechnung des synchronisierten zweiten Audio-Eingangssignals zu verbessern. Ferner kann der zweite trainierte Operator eine zur Transformation des ersten trainierten Operators inverse Transformation des synchronisierten zweiten Audio-Eingangssignals, insbesondere zurück in den Zeitraum der Audio-Eingangssignale, vorsehen.

Vorzugsweise weist der zweite trainierte Operator insbesondere ein iteratives Verfahren mit endlich vielen Iterationsschritten auf, wobei insbesondere in jedem Iterationsschritt ein Synchronisationsvektor, vorzugsweise ein optimierter Korrelationsvektor, insbesondere ein optimierter Kreuzkorrelationsvektor, bestimmt werden, was eine Beschleunigung des erfindungsgemäßen Verfahrens bewirkt. Die Anzahl der Iterationsschritte des zweiten trainierten Operators kann benutzerseitig definierbar sein, um das Verfahren benutzerseitig zu konfigurieren.

In jedem Iterationsschritt des zweiten trainierten Operators erfolgt vorzugsweise eine gestreckte Faltung des Audio-Signals mit mindestens einem Teil des Synchronisationsvektors, insbesondere des optimierten Korrelationsvektors. In jedem Iterationsschritt kann eine Normierung des Synchronisationsvektors und/oder eine gestreckte Faltung des synchronisierten Audio-Eingangssignals mit dem Synchronisationsvektor erfolgen, um die Signaltrennungsqualität des Verfahrens zu verbessern.

In einer weiteren Ausgestaltung der Erfindung sieht der zweite trainierte Operator die Bestimmung mindestens einer akustischen Modellfunktion vor. Im Sinne der Erfindung entspricht die akustische Modellfunktion insbesondere dem Zusammenhang zwischen dem Audioobjekt und dem aufgenommenen Audio-Eingangssignal. Damit gibt die akustische Modellfunktion beispielsweise die akustischen Eigenschaften der Umgebung, wie etwa akustische Reflexionen (Hall), frequenzabhängige Absorptionen und/oder Bandpass-Effekte wieder. Außerdem beinhaltet die akustische Modellfunktion insbesondere die Aufnahmecharakteristik mindestens eines Mikrofons. Insofern ist durch den zweiten trainierten Operator im Rah-

men der Optimierung des Korrelationsvektors die Kompensation unerwünschter akustischer Effekte auf das Audiosignal, bedingt etwa durch die Umgebung und/oder die Aufnahmecharakteristik des mindestens einen Mikrofons möglich. Neben der Kompensation der Laufzeit ist damit auch die Kompensation störender akustischer Einflüsse, beispielsweise bedingt durch den Propagationsweg des Schalls, möglich, was die Signaltrennungsqualität des erfindungsgemäßen Verfahrens verbessert.

Das trainierte Modell zum Extrahieren des Audioobjektes kann mindestens eine Transformation des ersten Audio-Eingangssignals und des synchronisierten zweiten Audio-Eingangssignals jeweils in einen insbesondere höherdimensionalen Darstellungsraum vorsehen, was die Signaltrennungsqualität verbessert. Im Sinne der Erfindung weist der Darstellungsraum eine im Vergleich zu dem in der Regel eindimensionalen Zeitraum der Audio-Eingangssignale höhere Dimensionalität auf. Indem die Transformationen als Teile eines neuronalen Netzwerks ausgebildet sein können, können die Transformationen spezifisch hinsichtlich des zu extrahierenden Audioobjektes trainiert sein.

Das trainierte Modell des Extrahierens des Audioobjekts kann die Anwendung mindestens einer trainierten Filtermaske auf das erste Audio-Eingangssignal und auf das synchronisierte zweite Audio-Eingangssignal vorsehen. Die trainierte Filtermaske ist vorzugsweise spezifisch auf das Audioobjekt trainiert.

Das trainierte Modell des Extrahierens des Audioobjekts kann mindestens eine Transformation des Audioobjekts in den Zeitraum der Audio-Eingangssignale vorsehen, um insbesonde-

re eine vorausgegangene Transformation in den Darstellungsraum rückgängig zu machen.

Die Verfahrensschritte des Synchronisierens und/oder des Extrahierens und/oder der Ausgabe des Audioobjektes sind vorzugsweise einem einzigen neuronalen Netzwerk zugeordnet, um ein spezifisches Training des neuronalen Netzwerks hinsichtlich des Audioobjektes zu ermöglichen. Durch die Ausgestaltung eines einzigen neuronalen Netzwerks wird die Zuverlässigkeit des Verfahrens und dessen Signaltrennungsqualität insgesamt verbessert.

Vorzugsweise wird das neuronale Netzwerk mit Soll-Trainingsdaten trainiert, wobei die Soll-Trainingsdaten Audio-Eingangssignale und dazu korrespondierende vordefinierte Audioobjekte umfassen, mit den folgenden Trainingsschritten: Vorwärtsspeisen des neuronalen Netzwerks mit den Soll-Trainingsdaten unter Erhalt eines ermittelten Audioobjekts, Bestimmen eines Fehlerparameters, insbesondere eines Fehlervektors zwischen dem ermittelten Audioobjekt und dem vordefinierten Audioobjekt und Ändern von Parametern des neuronalen Netzwerks durch Rückwärtsspeisen des neuronalen Netzwerks mit dem Fehlerparameter, insbesondere mit dem Fehlervektor, falls ein Qualitätsparameter des Fehlerparameters, insbesondere des Fehlervektors, einen vordefinierten Wert übersteigt.

Das Training ist dabei auf das spezifische Audioobjekt ausgerichtet; mindestens zwei Parameter der trainierten Komponenten des erfindungsgemäßen Verfahrens können wechselseitig voneinander abhängig sein.

Vorzugsweise ist das Verfahren derart ausgestaltet, dass es kontinuierlich abläuft, was auch als "Online-Betrieb" bezeichnet ist. Im Sinne der Erfindung werden dabei ständig, insbesondere ohne Benutzereingabe, Audio-Eingangssignale eingelesen und zur Extraktion von Audioobjekten ausgewertet. Dabei können beispielsweise die Audio-Eingangssignale jeweils Teile von insbesondere kontinuierlich eingelesenen Audio-Signalen mit insbesondere vordefinierter Länge sein. Dies wird auch als "Buffering" bezeichnet. Besonders vorzugsweise kann das Verfahren derart ausgebildet sein, dass die Latenz des Verfahrens höchstens 100 ms, insbesondere höchstens 80 ms, vorzugsweise höchstens 40 ms beträgt. Latenz ist im Sinne der Erfindung die Laufzeit des Verfahrens, gemessen ab dem Einlesen der Audio-Eingangssignale bis zur Ausgabe des Audioobjektes. Ein Betrieb des Verfahrens ist daher in Echtzeit möglich.

Das erfindungsgemäße System kann ein erstes Mikrofon zum Empfangen des ersten Audio-Eingangssignals und ein zweites Mikrofon zum Empfangen des zweiten Audio-Eingangssignals vorsehen, wobei die Mikrofone jeweils mit dem System derart verbindbar sind, dass die Audio-Eingangssignale der Mikrofone der Steuereinheit des Systems zuführbar sind. Das System kann insbesondere als Komponente eines Mischpults ausgestaltet sein, mit dem die Mikrofone verbindbar sind. Besonders vorzugsweise ist das System ein Mischpult. Die Verbindung des Systems mit dem Mikrofonen kann kabelgebunden und/oder kabellos sein. Das Computerprogramm zur Durchführung des erfindungsgemäßen Verfahrens ist vorzugsweise auf einer Steuereinheit des erfindungsgemäßen Systems ausführbar.

Weitere Vorteile und Merkmale der Erfindung ergeben sich aus den Ansprüchen und der nachfolgenden Beschreibung, in der Ausgestaltungen der Erfindung unter Bezugnahme auf die Zeichnungen im Einzelnen erläutert sind. Dabei zeigen:

Fig. 1 Ein erfindungsgemäßes System in einer schematischen Ansicht;

Fig. 2 eine Übersicht eines erfindungsgemäßen Verfahrens in einem Ablaufdiagramm mit modellhaften Signalen;

Fig. 3 ein Ablaufdiagramm zum Verfahrensschritt einer Synchronisierung von Audio-Eingangssignalen mit modellhaften Signalen;

Fig. 4 ein Ablaufdiagramm zu einem iterativen Verfahren der Synchronisierung;

Fig. 5 ein Ablaufdiagramm zum Extrahieren des Audioobjektes und

Fig. 6 ein Ablaufdiagramm zum Trainieren des erfindungsgemäßen Verfahrens.

Fig. 1 zeigt eine Ausgestaltung eines erfindungsgemäßen Systems 10 zur Extraktion eines Audioobjektes 11 in einer schematischen Darstellung, wobei das System 10 ein Mischpult 10a ist. Audioobjekte 11 im Sinne der Erfindung sind akustische Signale, die einem Ereignis und/oder einem Objekt zugeordnet sind. Im vorliegenden Ausführungsbeispiel der Erfindung ist das Audioobjekt 11 das Geräusch 12 eines abgeschossenen, in Fig. 1 nicht dargestellten Fußballs.

Das Geräusch 12 wird von zwei Mikrofonen 13, 14 aufgenommen, die jeweils ein Audio-Eingangssignal a1, a2 erzeugen, so dass die Audio-Eingangssignale a1, a2 das Geräusch 12 enthalten. Aufgrund der unterschiedlichen Distanzen der Mikrofone 13, 14 zum Geräusch 12 befindet sich das Geräusch 12 an unterschiedlichen zeitlichen Positionen der Audio-Eingangssignale a1, a2. Zusätzlich unterscheiden sich die Audio-Eingangssignale a1, a2 aufgrund der akustischen Eigenschaften der Umgebung voneinander und weisen daher jeweils auch unerwünschte Anteile auf, die beispielsweise durch die Propagationsstrecken des Schalls bis zu den Mikrofonen 13, 14 etwa in Form von Hall und/oder unterdrückten Frequenzen, verursacht sind, und die im Sinne der Erfindung als Störschall bezeichnet werden. Im Sinne der Erfindung gibt eine erste akustische Modellfunktion M1 die akustischen Einflüsse der Umgebung und der Aufnahmecharakteristik des Mikrofons 13 auf das aufgenommene Audio-Eingangssignal a1 des ersten Mikrofons 13 wieder. Das Audio-Eingangssignal a1 entspricht mathematisch insofern einer Faltung des Geräuschs 12 mit der ersten akustischen Modellfunktion M1. Analog gilt dies für eine zweite akustische Modellfunktion M2 und für das aufgenommene Audio-Eingangssignal a2 des zweiten Mikrofons 14.

Die Mikrofone 13, 14 sind mit dem Mischpult 10a verbunden, so dass die Audio-Eingangssignale a1, a2 an eine Steuereinheit 15 des Systems 10 übermittelt werden, damit die Steuereinheit 15 die Audio-Eingangssignale a1, a2 auswertet und das Geräusch 12 aus den Audio-Eingangssignalen a1, a2 mit Hilfe des erfindungsgemäßen Verfahrens extrahiert und zur weiteren Verwendung ausgibt. Bei der Steuereinheit 15 zur Extraktion des Audioobjektes 11 handelt es sich um einen

Mikrokontroller und/oder um einen Programmcodeblock eines entsprechenden Computerprogramms. Die Steuereinheit 15 umfasst ein trainiertes neuronales Netzwerk, das mit Audio-Eingangssignalen a1, a2 insbesondere vorwärts gespeist wird. Das neuronale Netzwerk ist dazu trainiert, das spezifische Audioobjekt 11, also im vorliegenden Falle das Geräusch 12, aus den Audio-Eingangssignalen a1, a2 zu extrahieren und insbesondere von Störschall-Anteilen der Audio-Eingangssignale a1, a2 zu trennen. Im Wesentlichen werden dabei die Auswirkungen der akustischen Modellfunktionen M1, M2 auf das Geräusch 12 in den Audio-Eingangssignalen a1, a2 kompensiert.

Fig. 2 veranschaulicht eine Ausgestaltung des erfindungsgemäßen Verfahrens in einer Übersicht als Flussdiagramm mit modellhaften Audio-Eingangssignalen a1, a2, an denen das Verfahren durchgeführt wird. In einem ersten Schritt V1 erfolgt ein Synchronisieren des zweiten Audio-Eingangssignals a2 mit dem ersten Audio-Eingangssignal a1, so dass im Ergebnis ein synchronisiertes zweites Audio-Eingangssignal a2' erhalten wird. Im Sinne der Erfindung weist das synchronisierte zweite Audio-Eingangssignal a2' insbesondere das Geräusch 12 an im Wesentlichen der gleichen zeitlichen Position auf wie das erste Audio-Eingangssignal a1, was die nachfolgenden Verfahrensschritte maßgeblich beschleunigt und vereinfacht. Insofern entspricht die Synchronisierung V1 der Audio-Eingangssignale a1, a2 insbesondere einer Kompensation der Laufzeitdifferenzen zwischen den Audio-Eingangssignalen a1, a2.

Anschließend erfolgt gemäß Fig. 2 das Extrahieren V2 des Geräuschs 12 durch die Anwendung eines trainierten Modells auf das erste Audio-Eingangssignal a1 und auf das synchro-

nisierte zweite Audio-Eingangssignal a2', so dass im Ergebnis das Geräusch 12 als Audiosignal erhalten wird. Das trainierte Modell ist dem neuronalen Netzwerk zugeordnet und ist als ein Teil von diesem auf die Extraktion des spezifischen Audioobjekts 11, hier des Geräuschs 12, trainiert. Im nachfolgenden Verfahrensschritt erfolgt die Ausgabe V3 des Geräuschs 12 als Audio-Ausgangssignal Z.

Die Verfahrensschritte des Synchronisierens V1, des Extrahierens V2 des Geräuschs 12 und dessen Ausgabe V3 sind einem einzigen, trainierten neuronalen Netzwerk zugeordnet, so dass das Verfahren als End-to-End-Verfahren ausgebildet ist. Dadurch ist es als Ganzes trainiert und läuft automatisch und kontinuierlich ab, wobei die Extraktion des Geräuschs in Echtzeit, also mit einer Latenz von höchstens 40 ms erfolgt.

Fig. 3 zeigt einen Verfahrensablauf des Synchronisierens V1 der Audio-Eingangssignale a1, a2 in einem Flussdiagramm mit modellhaften Audio-Eingangssignalen a1, a2 zur Veranschaulichung der Verfahrensschritte. In einem ersten Verfahrensschritt V4 der Fig. 3 wird ein erster trainierter Operator des neuronalen Netzwerks jeweils auf die Audio-Eingangssignale a1, a2 angewendet, um Audio-Signale m1, m2 zu generieren. In einer Ausgestaltung der Erfindung werden die Audio-Eingangssignale a1, a2 durch den ersten trainierten Operator des neuronalen Netzwerks in einen im Vergleich zu den Audio-Eingangssignalen a1, a2 höherdimensionalen Merkmalsraum in der Zeitdomäne zu den Audio-Signalen m1, m2 transformiert, um die nachfolgenden Berechnungen zu vereinfachen und zu beschleunigen. Je nach Art des Audioobjekts 11 erfolgt bereits bei der Transformation eine Bearbeitung

der Audio-Signale m1, m2. Die transformierten Audio-Signale m1, m2 sind in Fig. 3 modellhaft dargestellt.

Im zweiten Verfahrensschritt V5 der Fig. 3 erfolgt die analytische Berechnung der Kreuzkorrelation als Korrelation zwischen den Audio-Signalen m1, m2, die mathematisch wie folgt definiert ist:

$$(m_1 \star m_2)[t] \quad \hat{=} \sum_{n=-\infty}^{\infty} m_1[n]\, m_2[n+t]$$

Die Berechnung V5 resultiert in einen Kreuzkorrelationsvektor k, der modellhaft in Fig. 3 dargestellt ist. Im dritten Verfahrensschritt V6 wird der Kreuzkorrelationsvektor k mit Hilfe eines zweiten trainierten Operators des neuronalen Netzwerks optimiert, wobei mittels des zweiten trainierten Operators die Berechnung der akustischen Modellfunktion M erfolgt, um deren Auswirkungen auf die Audio-Signale m1, m2 zu kompensieren. Der zweite trainierte Operator dient damit beispielsweise als akustischer Filter und sieht im Ausführungsbeispiel der Fig. 3 insbesondere eine Normierung des Kreuzkorrelationsvektors k vor, beispielsweise mittels einer Softmax-Funktion. Der dadurch erhaltene Synchronisationsvektor s ist modellhaft in Fig. 3 dargestellt.

Im vierten Verfahrensschritt der Fig. 3 erfolgt die Berechnung V7 des synchronisierten zweiten Audio-Eingangssignals a2' durch die Faltung des Synchronisationsvektors s mit dem zweiten Audio-Eingangssignal a2.

Das synchronisierte zweite Audio-Eingangssignal a2' ist in Fig. 3 modellhaft dargestellt. Im Vergleich zum ursprünglichen Audio-Eingangssignal a2 ist erkennbar, dass im hier betrachteten, stark vereinfachten Modell eine Kompensation der Laufzeitdifferenz als zeitlicher Offset erfolgt ist. Das synchronisierte zweite Audio-Eingangssignal a2' wird anschließend, wie bereits beschrieben, für die Extraktion V2 des Audioobjekts 11 verwendet.

Fig. 4 zeigt eine weitere Ausgestaltung der Synchronisierung V1 der Audio-Eingangssignale a1, a2, bei der ein iteratives Verfahren zur Beschleunigung der Berechnung vorgesehen ist, wobei die Anzahl der Iterationsschritte I benutzerseitig festgelegt ist. Im ersten Iterationsschritt erfolgt eine Berechnung des Korrelationsvektors zwischen den Audio-Signalen m1, m2 ähnlich dem Verfahren gemäß Fig. 3 bis zur Berechnung V7 des synchronisierten Audio-Eingangssignals a2', wobei der Synchronisationsvektor s_i des aktuellen Iterationsschritts i aber nun im Rahmen der Optimierung V6 bei jedem Iterationsschritt i mittels der maxpool-Funktion beschränkt wird. Anschließend erfolgt - in jedem Iterationsschritt i - die Berechnung V8 des iterativen Audio-Signals $m2_i$ für die Iterationsstufe i mittels einer gestreckten Faltung, die mathematisch wie folgt definiert ist:

$$(a_2 *_{d_i} s)(t) = \sum_{n=-d_i}^{d_i} a_2(d_i \cdot n)\, s\,(n+t)$$

Der Faktor d_i entspricht dabei dem Maß der Beschränkung des Kreuzkorrelationsvektors für den Iterationsschritt i, wobei die Summierung über den +/- den Faktor d_i erfolgt. Dieser

Vorgang wird so lange wiederholt, bis die benutzerseitig vorgegebene Anzahl an Iterationsschritten I durchgeführt wurde. Schließlich erfolgt eine gestreckte Faltung V9 des Audio-Signals m2 mit dem zuletzt berechneten Synchronisationsvektor S_i, woraufhin das synchronisierte zweite Audio-Signal a2'berechnet und ausgegeben wird V7. Durch die Berechnung des Synchronisationsvektors s auf der Basis des Teilbereichs der im vorigen Iterationsschritt ermittelten Parameter reduziert sich die Komplexität der Berechnungen, was die Laufzeit des Verfahrens beschleunigt, ohne dessen Genauigkeit zu beeinträchtigen.

Fig. 5 zeigt eine Ausgestaltung der Extraktion V2 des Audioobjektes 11 aus dem Audio-Eingangssignal a1 und dem synchronisierten zweiten Audio-Eingangssignal a2' in einem Flussdiagramm. In einem ersten Verfahrensschritt V10 werden die Audio-Eingangssignale a1, a2' durch die Anwendung eines ersten trainierten Modells des neuralen Netzwerks jeweils in einen höherdimensionalen Darstellungsraum transformiert, um die nachfolgenden Berechnungen zu vereinfachen. Beispielsweise weist das erste trainierte Modell eine gängige Filterbank mit insbesondere einer Terzbandfilterbank und/oder einer Mel-Filterbank auf, wobei die Parameter der Filter durch das vorausgegangene Training des neuronalen Netzwerks optimiert worden sind.

Im zweiten Verfahrensschritt V11 erfolgt die Separation des Audioobjekts 11 von den Audio-Eingangssignalen a1, a2' durch Anwendung eines zweiten trainierten Modells des neuronalen Netzwerks auf die Audio-Eingangssignale a1, a2'. Auch die Parameter des zweiten trainierten Modells wurden durch das vorausgegangene Training optimiert und sind insbesondere von dem ersten trainierten Modell des vorangehen-

den Verfahrensschrittes V10 abhängig. Im Ergebnis dieses Verfahrensschrittes V11 wird das Audioobjekt 11 aus den Audio-Eingangssignalen a1, a2' erhalten und befindet sich noch im höherdimensionalen Darstellungsraum.

Im dritten Verfahrensschritt V12 der Fig. 5 wird das separierte Audioobjekt 11 durch die Anwendung eines dritten trainierten Modells des neuronalen Netzwerks auf das Audioobjekt 11 in den ursprünglichen, eindimensionalen Zeitraum der Audiosignale a1, a2 transformiert, wobei die Parameter des dritten trainierten Modells von jenen der übrigen trainierten Modelle abhängig sind und durch das vorausgegangene Training gemeinsam optimiert wurden. Insofern ist das dritte trainierte Modell der Transformation gemäß dem dritten Verfahrensschritt V12 der Fig. 5 funktional als Komplement zur Transformation V10 gemäß dem ersten trainierten Modell zu sehen. Falls beispielsweise im ersten trainierten Modell des ersten Verfahrensschrittes V10 eine eindimensionale Faltung vorgesehen ist, erfolgt in der Rücktransformation V12 eine transponierte eindimensionale Faltung.

Damit das neuronale Netzwerk das Audioobjekt 11 zuverlässig aus den Audio-Eingangssignalen a1, a2 extrahieren kann, muss es vor dem Einsatz trainiert werden. Dies geschieht beispielweise durch die nachfolgend beschriebenen Trainingsschritte V13 bis V19, die in Fig. 6 in einem schematischen Ablaufdiagramm gezeigt sind. In den betrachteten Ausführungsbeispielen des erfindungsgemäßen Verfahrens sind die genannten Verfahrensschritte einem einzigen neuronalen Netzwerk zugeordnet und jeweils differenzierbar, so dass mit dem nachfolgend beschriebenen Trainingsverfahren V13

sämtliche trainierten Komponenten spezifisch hinsichtlich des Audioobjekts 11 trainiert werden.

Vordefinierte Audioobjekte 16 werden mittels vordefinierter Algorithmen zu vorgegebenen Audio-Eingangssignalen a1, a2 generiert V14. Die vordefinierten Audioobjekte 16 sind stets vom gleichen Typ, so dass das Verfahren spezifisch hinsichtlich eines Typs von Audioobjekten 16 trainiert wird. Die generierten Audio-Eingangssignale a1, a2 durchlaufen das erfindungsgemäße Verfahren gemäß Fig. 2 und werden dabei insbesondere durch das neurale Netzwerk vorwärts gespeist V15. Das dadurch ermittelte Audioobjekt 17 wird mit dem vordefinierten Audioobjekt 16 verglichen, um auf dieser Grundlage einen mathematischen Fehlervektor P zu bestimmen V16. Danach erfolgt eine Abfrage V17, ob ein Qualitätsparameter des Fehlervektors P einen vordefinierten Wert unterschreitet und das ermittelte Audioobjekt 17 hinreichend gut extrahiert wurde.

Überschreitet der Qualitätsparameter den vordefinierten Wert, ist das Abbruchkriterium nicht erfüllt und es wird im nächsten Verfahrensschritt V18 der Gradient des Fehlervektors P bestimmt und rückwärts durch das neuronale Netzwerk gespeist, so dass sämtliche Parameter des neuronalen Netzwerks angepasst werden. Anschließend wird das Trainingsverfahren V13 mit weiteren Datensätzen solange wiederholt, bis der Fehlervektor P einen hinreichend guten Wert erreicht und die Abfrage V17 ergibt, dass das Abbruchkriterium erfüllt wurde. Dann wird der Trainingsprozess V13 abgeschlossen V19 und das Verfahren kann auf reale Daten angewendet werden. Idealerweise werden als vordefinierte Audioobjekte 16 in der Trainingsphase jene Audioobjekte 11 verwendet, die in der Anwendung des Verfahrens auch ermittelt werden

sollen, beispielsweise bereits aufgezeichnete Schussgeräusche 12 von Fußbällen.

Lawo Holding AG
Am Oberwald 8
76437 Rastatt

14. Februar 2020
26136.9 Ka/jk

Patentansprüche

1. Verfahren zur Extraktion von mindestens einem Audioobjekt (11) aus mindestens zwei Audio-Eingangssignalen (a1, a2), die jeweils das Audioobjekt (11) enthalten, mit den folgenden Schritten:

   - Synchronisieren (V1) des zweiten Audio-Eingangssignals (a2) mit dem ersten Audio-Eingangssignal (a1) unter Erhalt eines synchronisierten zweiten Audio-Eingangssignals (a2'),

   - Extrahieren (V2) des Audioobjekts (11) durch die Anwendung von mindestens einem trainierten Modell auf das erste Audio-Signal (a1) und auf das synchronisierte zweite Audio-Eingangssignal (a2') und

   - Ausgabe (V3) des Audioobjekts (11),

   wobei der Verfahrensschritt des Synchronisierens (V1) des zweiten Audio-Eingangssignals (a2) mit dem ersten Audio-Eingangssignal (a1) die folgenden Verfahrensschritte umfasst:

POSTFACH 410760
D-76207 KARLSRUHE

BERGWALDSTR. 1
D-76227 KARLSRUHE

TEL: +49 (0)721 9 43 28 15
FAX: +49 (0)721 9 43 28 40

eMAIL: mail@lichti.eu

- Generieren (V4) von Audio-Signalen (m1, m2) durch Anwendung eines ersten trainierten Operators auf die Audio-Eingangssignale (a1, a2),

- Analytische Berechnung (V5) einer Korrelation zwischen den Audio-Signalen (m1, m2) unter Erhalt eines Korrelationsvektors (k),

- Optimieren (V6) des Korrelationsvektors (k) mit Hilfe eines zweiten trainierten Operators unter Erhalt eines Synchronisationsvektors (s) und

- Bestimmen (V7) des synchronisierten zweiten Audio-Eingangssignals (a2') mit Hilfe des Synchronisationsvektors (s).

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass der erste trainierte Operator eine insbesondere trainierte Transformation der Audio-Eingangssignale (a1, a2) in einen Merkmalsraum umfasst.

3. Verfahren nach einem der Ansprüche 1 oder 2, dadurch gekennzeichnet, dass der zweite trainierte Operator mindestens eine Normierung des Korrelationsvektors (k) umfasst.

4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass der zweite trainierte Operator insbesondere ein iteratives Verfahren mit endlich vielen Iterationsschritten (I) aufweist, wobei insbesondere in jedem Iterationsschritt ein Synchronisationsvektor (s) bestimmt wird.

5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, dass die Anzahl der Iterationsschritte (I) des zweiten trainierten Operators benutzerseitig definierbar ist.

6. Verfahren nach einem der Ansprüche 4 oder 5, dadurch gekennzeichnet, dass in jedem Iterationsschritt (i) des zweiten trainierten Operators eine gestreckte Faltung des Audio-Signals (m2) mit mindestens einem Teil des Synchronisationsvektors (s) erfolgt.

7. Verfahren nach einem der Ansprüche 4 bis 6, dadurch gekennzeichnet, dass in jedem Iterationsschritt eine Normierung des Synchronisationsvektors (s) und/oder eine gestreckte Faltung des synchronisierten Audio-Eingangssignals (a2') mit Synchronisationsvektor (s') erfolgt.

8. Verfahren nach einem der Ansprüche 1 bis 7, dadurch gekennzeichnet, dass der zweite trainierte Operator die Bestimmung mindestens einer akustischen Modellfunktion (M) vorsieht.

9. Verfahren nach einem der Ansprüche 1 bis 8, dadurch gekennzeichnet, dass das trainierte Modell des Extrahierens (V2) des Audioobjekts (11) mindestens eine Transformation des ersten Audio-Eingangssignals (a1) und des synchronisierten zweiten Audio-Eingangssignals (a2') jeweils in einen insbesondere höherdimensionalen Darstellungsraum vorsieht.

10. Verfahren nach einem der Ansprüche 1 bis 9, dadurch gekennzeichnet, dass das trainierte Modell des Extrahierens (V2) des Audioobjekts (11) die Anwendung mindestens einer gelernten Filtermaske auf das erste Audio-Eingangssignal (a1) und auf das synchronisierte zweite Audio-Eingangssignal (a2') vorsieht.

11. Verfahren nach einem der Ansprüche 9 oder 10, dadurch gekennzeichnet, dass das trainierte Modell des Extrahierens (V2) des Audioobjekts (11) mindestens eine Transformation des Audioobjekts (11) in den Zeitraum der Audio-Eingangssignale (a1, a2) vorsieht.

12. Verfahren nach einem der Ansprüche 1 bis 11, dadurch gekennzeichnet, dass die Verfahrensschritte des Synchronisierens (V1) und/oder des Extrahierens (V2) und/oder der Ausgabe (V3) des Audioobjekts (11) einem einzigen neuronalen Netzwerk zugeordnet sind.

13. Verfahren nach Anspruch 12, dadurch gekennzeichnet, dass das neuronale Netzwerk mit Soll-Trainingsdaten trainiert wird, wobei die Soll-Trainingsdaten Audio-Eingangssignale (a1, a2) und dazu korrespondierende vordefinierte Audioobjekte (16) umfassen, mit den folgenden Trainingsschritten:

- Vorwärtsspeisen (V15) des neuronalen Netzwerks mit den Soll-Trainingsdaten unter Erhalt eines ermittelten Audioobjekts (17),

- Bestimmen (V16) eines Fehlervektors (P) zwischen dem ermittelten Audioobjekt (17) und dem vordefinierten Audioobjekt (16) und

- Ändern von Parametern des neuronalen Netzwerks durch Rückwärtsspeisen (V18) des neuronalen Netzwerks mit dem Fehlervektor (P), falls ein Qualitätsparameter des Fehlervektors (P) einen vordefinierten Wert übersteigt.

14. Verfahren nach einem der Ansprüche 1 bis 13, dadurch gekennzeichnet, dass das Verfahren derart ausgestaltet ist, dass es kontinuierlich abläuft.

15. Verfahren nach einem der Ansprüche 1 bis 14, dadurch gekennzeichnet, dass die Audio-Eingangssignale (a1, a2) jeweils Teile von insbesondere kontinuierlich eingelesenen Audio-Signalen (b1, b2) mit insbesondere vordefinierten zeitlichen Längen sind.

16. Verfahren nach einem der Ansprüche 1 bis 15, dadurch gekennzeichnet, dass das Verfahren derart ausgestaltet ist, dass die Latenz des Verfahrens höchstens 100 ms, insbesondere höchstens 80 ms, vorzugsweise höchstens 40 ms beträgt.

17. System (10) zur Extraktion eines Audioobjektes (11) aus mindestens zwei Audio-Eingangssignalen (a1, a2) mit einer Steuereinheit (15), die dazu ausgebildet ist, ein Verfahren nach einem der Ansprüche 1 bis 16 durchzuführen.

18. System nach Anspruch 17, dadurch gekennzeichnet, dass ein erstes Mikrofon (13) zum Empfangen des ersten Audio-Eingangssignals (a1) und ein zweites Mikrofon (14) zum Empfangen des zweiten Audio-Eingangssignals (a2) jeweils mit dem System (10) derart verbindbar sind, dass die Audio-Eingangssignale (a1, a2) der Mikrofone (13, 14) der Steuereinheit (15) zuführbar sind.

19. System nach einem der Ansprüche 17 oder 18, dadurch gekennzeichnet, dass das System (10) als Komponente eines Mischpults (10a) ausgeschaltet ist.

20. Computerprogramm mit Programmcodemitteln, das dazu ausgestaltet ist, die Schritte eines Verfahrens nach einem der Ansprüche 1 bis 16 durchzuführen, wenn das Computerprogramm auf einem Computer oder einer entsprechenden Recheneinheit ausgeführt wird, insbesondere auf einer Steuereinheit (15) eines Systems (10) nach einem der Ansprüche 17 bis 19.

Lawo Holding AG
Am Oberwald 8
76437 Rastatt

14. Februar 2020
26136.9 Ka/jk

Zusammenfassung

Die Erfindung betrifft ein Verfahren zur Extraktion von mindestens einem Audioobjekt aus mindestens zwei Audio-Eingangssignalen, die jeweils das Audioobjekt enthalten. Erfindungsgemäß sind die folgenden Schritte vorgesehen: Synchronisieren des zweiten Audio-Eingangssignals mit dem ersten Audio-Eingangssignal unter Erhalt eines synchronisierten zweiten Audio-Eingangssignals, Extrahieren des Audioobjekts durch die Anwendung von mindestens einem trainierten Modell auf das erste Audiosignal und auf das synchronisierte zweite Audio-Eingangssignal und Ausgabe des Audioobjekts. Ferner ist vorgesehen, dass der Verfahrensschritt des Synchronisierens des zweiten Audio-Eingangssignals mit dem ersten Audio-Eingangssignal die folgenden Verfahrensschritte umfasst: Generieren von Audiosignalen, analytische Berechnung einer Korrelation zwischen den Audiosignalen, Optimieren des Korrelationsvektors und Bestimmung des synchronisierten zweiten Audio-Eingangs-signals mit Hilfe des optimierten Korrelationsvektors. Ferner sieht die Erfindung ein System mit einer Steuereinheit vor, die dazu ausgebildet ist, das erfindungsgemäße Verfahren durch-

POSTFACH 410760
D-76207 KARLSRUHE

BERGWALDSTR. 1
D-76227 KARLSRUHE

TEL: +49 (0)721 9 43 28 15
FAX: +49 (0)721 9 43 28 40

eMAIL: mail@lichti.eu

zuführen. Außerdem ist ein Computerprogramm mit Programmcodemitteln vorgesehen, das dazu ausgestaltet ist, die Schritte des erfindungsgemäßen Verfahrens durchzuführen.

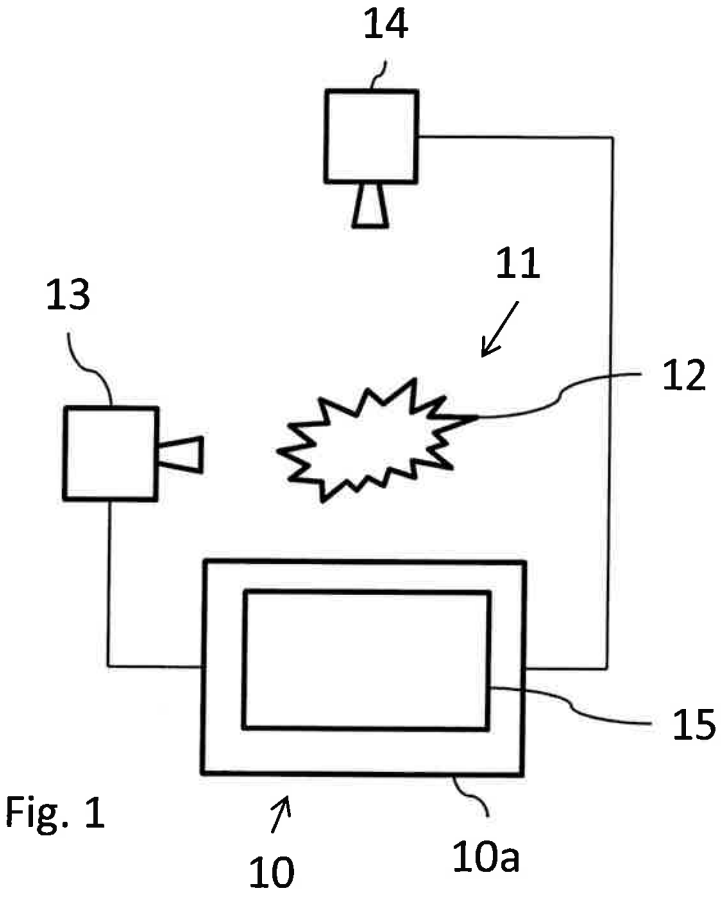


Fig. 1

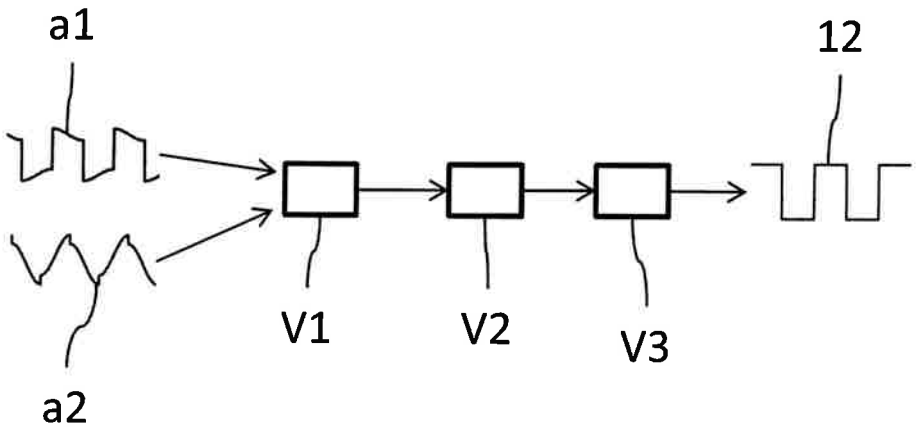


Fig. 2

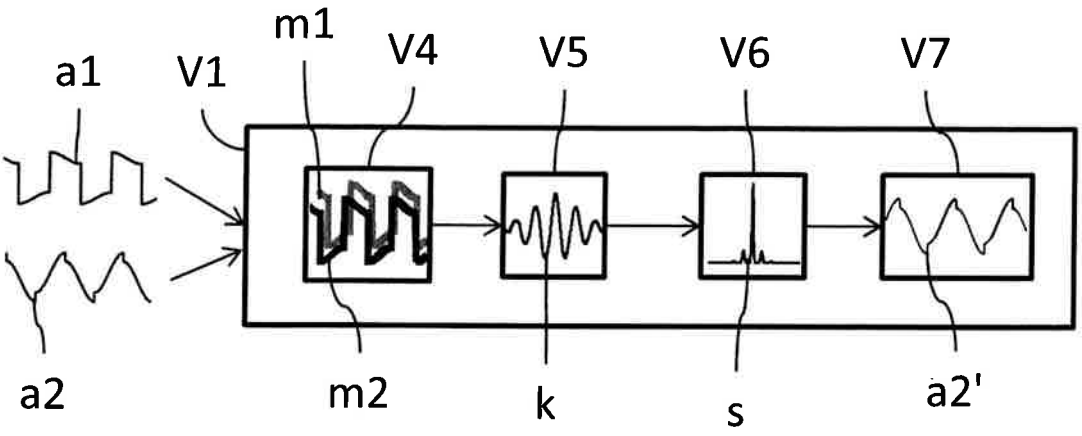


Fig. 3

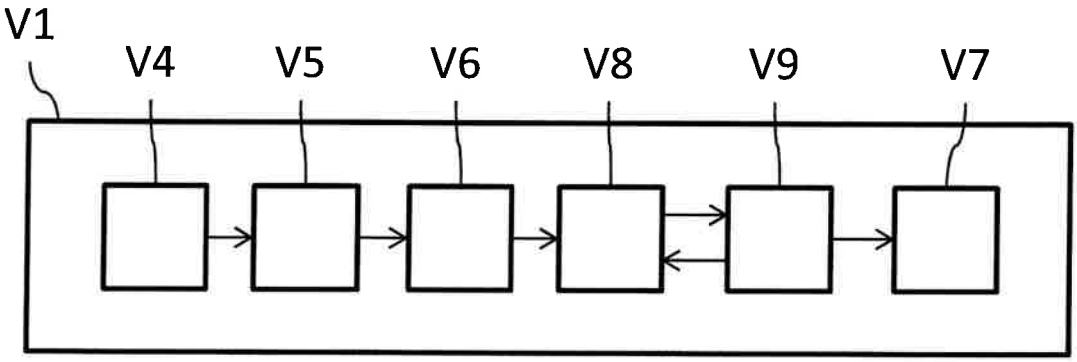


Fig. 4

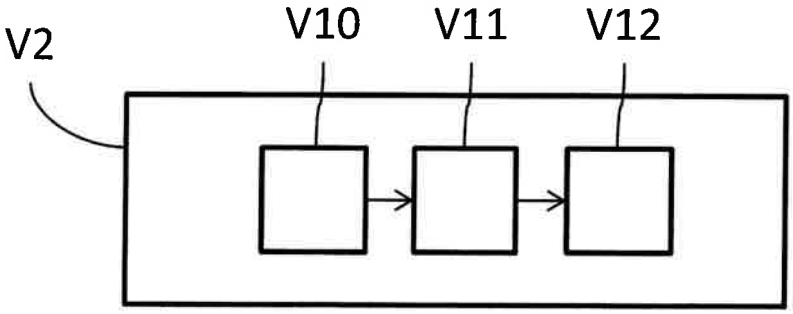


Fig. 5

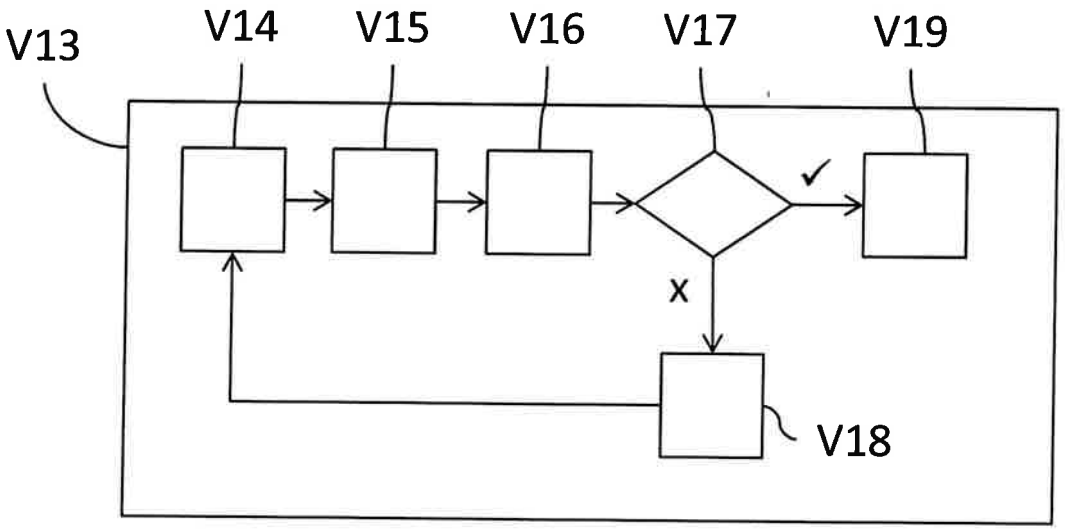


Fig. 6

Preprints and Manuscripts

Spatially Informed Neural Beamforming with Distributed Microphone Arrays

Jonathan D. Ziegler *†, Leon Schröder†, Andreas Koch† and Andreas Schilling*
* Institute for Visual Computing, Eberhard Karls University of Tübingen
†Institute for Applied Artificial Intelligence, Stuttgart Media University

***Abstract*—Robust real-time audio signal enhancement increasingly relies on multichannel microphone arrays for signal acquisition. Sophisticated beamforming algorithms have been developed to maximize the benefit of multiple microphones. With the recent success of deep learning models created for audio signal processing, the task of Neural Beamforming remains an open research topic. This paper presents a Neural Beamformer architecture capable of performing spatial beamforming with microphones randomly distributed over very large areas, even in negative signal-to-noise ratio environments with multiple noise sources and reverberation. The proposed method combines adaptive, nonlinear filtering and gating methods with the computation of spatial relations within a fully differentiable End-to-End neural network. Combining a small number of principle building blocks, the method is capable of low-latency, domain-specific beamforming even in challenging environments.**

I. Introduction

High-quality noise-free audio has become of ever greater importance with increases in human-computer interaction, telecommunication, and web conferencing. Ease of communication without personal contact has become a vital part of modern society. The enhancement of signals with respect to specific signal components such as speech can be performed with a wide variety of methods [1]. In addition to adaptive filtering of microphone signals, the use of microphone arrays is a popular approach to signal enhancement. When using distributed arrays, a common approach to multichannel signal processing is Delay-and-Sum (DS) or Filter-and-Sum (FS) beamforming [2]. By time-aligning and optionally filtering a set of microphone signals with respect to a defined signal source, surrounding noise and reverberation can be attenuated in the output signal. One of the largest challenges for DS and FS beamformers is the determination of the correct Time Delays of Arrival (TDOA), especially for spontaneous, large aperture microphone arrays of unknown configuration [3]. Commonly, the Generalized Cross Correlation with Phase Transform (GCC-PHAT) is used to compute the TDOA for every microphone with respect to a defined reference microphone [4]. While applying weighting factors such as the Phase Transform presents a nominal improvement over the standard cross correlation, accurately computing the TDOA of specific signals remains unstable, especially in situations with low signal-to-noise ratios (SNR).

In recent years, machine learning has had a large impact on audio signal processing, redefining the state of the art in many areas. The task of source separation can be approached in numerous ways, with adaptive, nonlinear filtering on individual audio channels being the most prominent and easily available to date [5, 6, 7]. Recently, multi-channel, deep-learning based array processing has become increasingly relevant [8]. The ability for neural beamforming networks to detect correlations of domain-specific signal components robustly and generate the appropriate spatial filters is of great value. Current systems can generally be categorized into two approaches, in which the spatial information is either precomputed analytically and fed into the beamforming network [9, 10, 11], or multiple neural networks are used independently to achieve source separation [12, 13]. Although investigations into End-to-End approaches have been remarkably successful [14, 15, 16], some basic limitations remain. Both referenced approaches directly use convolutions as beamforming filters. While Sainath et al. use dedicated convolutional layers to generate static spatial filters resulting in learnable "look-directions", Luo et al. use Temporal Convolutional Networks [17] or Dual-Path RNN [18] for the adaptive estimation of filters based on a pre-processed reference channel, raw auxiliary channels and the cosine similarity. The method introduced in the following sections circumvents said limitations by using iterative downsampling and upsampling elements to adaptively produce long beamforming filters. The architecture is capable of dynamically processing large-aperture microphone array signals in real time, outperforming the baseline approaches for microphone distances up to $75\,\mathrm{m}$.

II. Neural Beamforming

The approach to Neural Beamforming described in the following sections tackles the task of generating appropriate spatial beamforming filters via the encoding of audio signals and spatial information into a shared latent space from which the beamforming filters are decoded. Additionally, adaptive filtering of the audio channels prior to the computation of spatial relations is incorporated in order to enable the system to filter the input signals specifically for the task of spatial analysis. The use of learnable filter elements prior to the computation of pairwise cross-correlations creates the capability for the computation of domain-specific spatial filters, which can greatly increase the beamformer's robustness to noise and reverberation. In section II-A, an appropriate signal model is presented, sections II-B and III describe the method and implemented network architectures in detail. Results are presented in section V.

A. Problem Definition

Spatial beamforming can be achieved using M audio channels m_ν recorded by $M \geq 2$ transducers. The recorded channels consist of I signal and J noise components:

$$m_\nu = \sum_{i=1}^{I} \tilde{s}_i^\nu + \sum_{j=1}^{J} \tilde{n}_j^\nu. \tag{1}$$

Every signal $\tilde{s}_i^\nu$ and noise $\tilde{n}_j^\nu$ consist of the corresponding signal and noise sources s_i and n_j, propagated from their respective source position to the transducer m_ν. The propagation transformations are expressed as convolutions with corresponding transfer functions $H_{s_i}^\nu$ and $H_{n_j}^\nu$ [2]:

$$\tilde{s}_i^\nu = s_i * H_{s_i}^\nu, \quad \tilde{n}_j^\nu = n_j * H_{n_j}^\nu, \tag{2}$$

resulting in

$$m_\nu = \sum_{i=1}^{I} \left(s_i * H_{s_i}^\nu\right) + \sum_{j=1}^{J} \left(n_j * H_{n_j}^\nu\right). \tag{3}$$

The goal of Filter-and-Sum beamforming is to find additional beamforming filters H^ν that maximize s_i in the summed combination of all m_ν:

$$\varsigma_i = max_{s_i} \left\{ \sum_{\nu,i} m_\nu * H^{\nu i} \right\}. \tag{4}$$

Additional beams can be synthesized for subtractive noise reduction, resulting in

$$\varsigma_i = max_{s_i} \left\{ \sum_{\nu,i,j} m_\nu * H^{\nu ij} \right\}. \tag{5}$$

Finding optimal $H^{\nu ij}$ is difficult and computationally expensive. For the simplified FS approach, $H^{\nu ij}$ is approximated by a time shift $\delta^{\nu i}$ and a FIR filter $h^{\nu i}$ of relatively short length.

B. Differentiable Adaptive Generalized Cross Correlation

For microphone arrays of known spatial distribution, finding the correct time shifts $\delta^{\nu i}$ can be solved either geometrically, if the desired beam direction is known, or by means of a Steered Response Power source localization method, such as SRP-PHAT [19]. For arrays of unknown spatial distribution, pairwise time delay estimation can be performed by means of the cross correlation of the signals. Considering two input vectors $m_x[i]$, the discrete correlation, represented with the $\star$ operator, can be expressed as

$$(m_1 \star m_2)[i] = \sum_{j=-\infty}^{\infty} \overline{m_1[j]} \cdot m_2[j+i]. \tag{6}$$

Using the convolution theorem, (6) can be expressed as

$$(m_1 \star m_2) = \mathcal{F}^{-1} \left\{ \overline{\mathcal{F}\{m_1\}} \cdot \mathcal{F}\{m_2\} \right\}, \tag{7}$$

with $\mathcal{F}\{\ \}$ representing the Fourier transform and $\overline{[\ ]}$ the complex conjugation. Applying Phase Transform weighting leads to the computation of GCC-PHAT:

$$\text{GCC-PHAT}_{m_1,m_2} = \mathcal{F}^{-1} \left\{ \frac{\overline{\mathcal{F}\{m_1\}} \cdot \mathcal{F}\{m_2\}}{\left|\overline{\mathcal{F}\{m_1\}} \cdot \mathcal{F}\{m_2\}\right|} \right\}. \tag{8}$$

In the absence of interference and reverberation and for a single source s, the main peak of the GCC vector indicates the Time Delay of Arrival of the source with respect to the two input channels:

$$\delta^\nu = \texttt{argmax}\left[\text{GCC}_{m_0,m_\nu}\right]. \tag{9}$$

In real-world scenarios with multiple target and noise sources, TDOA estimation becomes unreliable, especially when using short audio buffers required for real-time applications. The use of weighting filters such as PHAT improves performance, but TDOA computations generally fail in reverberant and noisy environments. Addressing this problem from a data-driven perspective can improve the performance with domain knowledge and can enable accurate, domain-specific beamforming in noisy environments.

Performing the time alignment of the individual microphone signals within a neural network is a non-trivial task. As (9) is not differentiable, it cannot be incorporated into an End-to-End network architecture. Instead, spatial filters $H^{\nu ij}$ are generated by the model, using latent representations of the input signals and the respective GCC vectors. The microphone signals are then correlated[1] with the generated filter. While the process of DS beamforming can be implemented in a strictly analytical way, adaptive prefiltering and pattern enhancement within the spatial filter greatly improve performance over conventional methods. Additionally, the spatial relations for domain-specific signal classes can be computed, while GCC strictly extracts the correlation of the signal source with the highest sound pressure level within the recorded signals.

III. Network Architecture

The main principle of the proposed End-to-End architecture is to filter and synchronize multiple channels in order to maximize the signal-to-distortion ratio of the weighted sum of the channels. Cross correlation on adaptively-filtered input signals is used within the network to extract feature-dependent, domain-specific spatial relations between channel pairs. Figure 1 shows the top-level model architecture. A total of M microphones, with one defined reference channel is processed. Every pair of signals m_0 and m_ν is passed to an Aligner block which is discussed in detail in section III-A and can be seen in Figure 2. The output of the Aligner block consists of the processed channel and a latent vector containing information on signal shape, spatial relations, and amplitude statistics. Upsampling blocks named *GenerateWaveform*, discussed in section III-D, use the average latent vector of all Aligner

[1]Mathematically, convolution is the correct operation. As this requires an additional inversion of the filters along the time axis and considering the filters are learned by the network, the correlation operation can be used.

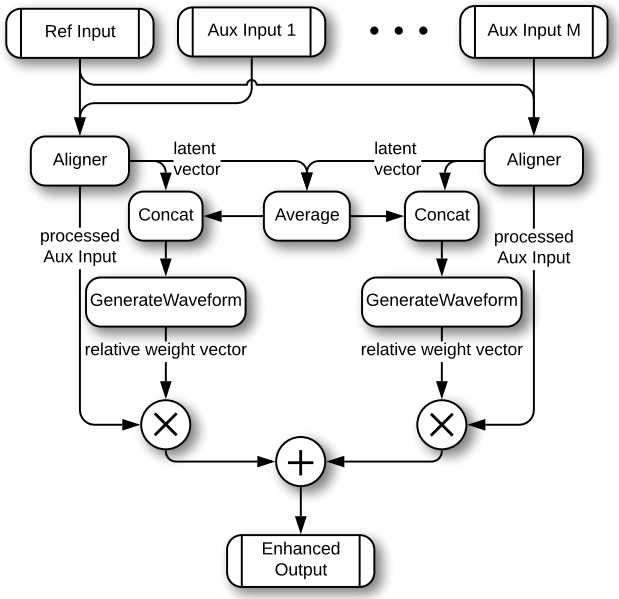


Fig. 1. Model Architecture. The reference and individual auxiliary channels are passed into Aligner blocks for processing. Once spatial filtering is applied, the channels are combined according to the sample-wise weight vectors estimated by the GenerateWaveform blocks.

outputs, combined with the individual latent vector of the respective channel, to create a per-sample weight vector. These vectors are subsequently applied to create the weighted sum over all processed channels, resulting in the final enhanced output of the model. In the following sections, the individual components and the motivation behind the design choices are discussed.

A. Channel Synchronization - Aligner

Dedicated modules are implemented to transform the auxiliary channels with respect to the reference channel. One important requirement for the proposed method is the ability to extract the spatial relations of domain-specific signal components. Signal classes, for example *speech*, are to be enhanced in the signal prior to the computation of spatial relations. This enables the model to better make use of beamforming capabilities in environments that contain high levels of interference and noise. To achieve this, adaptive, nonlinear filtering of the input signals is implemented: The signals are encoded using *GenerateWaveform* blocks (subsection III-C) and passed to FilterBlocks (subsection III-B) for masking and filtering. The filtered signals are correlated to extract the spatial relation of the channels with respect to the desired signal components. The correlation vectors are then encoded and concatenated to the latent vectors of both input channels. This combined latent vector is then passed to an additional FilterBlock in combination with the original Aux input. Within this block, channel synchronization and masking are performed to generate the transformed Aux signal.

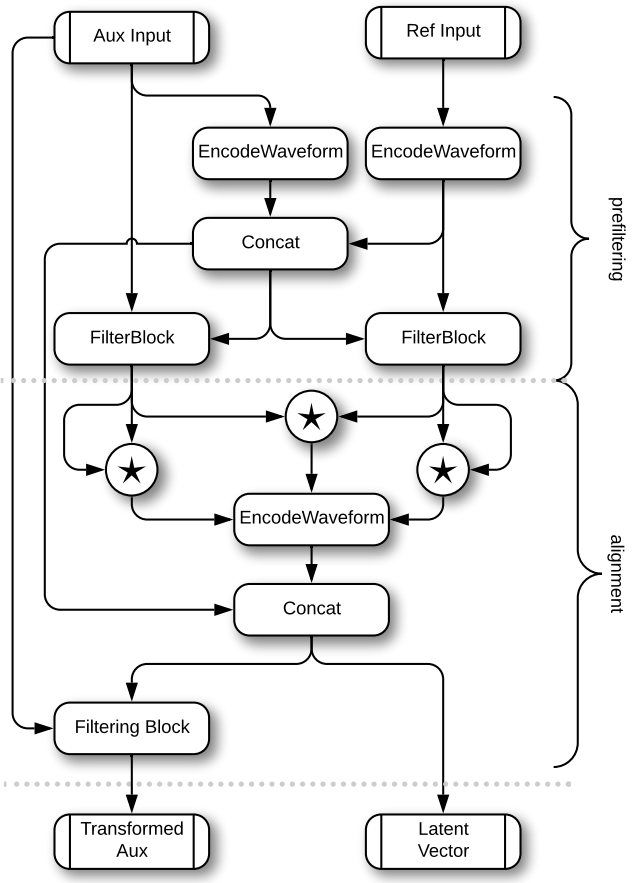


Fig. 2. This module, named Aligner, presents the core of the model. Individual auxiliary and reference microphone pairs are processed with respect to the reference microphone. Spatial relations are computed for the filtered signals via cross correlation and auto correlation. Subsequently, the signal and correlation vectors are encoded and concatenated to create the combined latent vector. This vector is then used to generate the beamforming filter, which is later applied to the original auxiliary microphone signal.

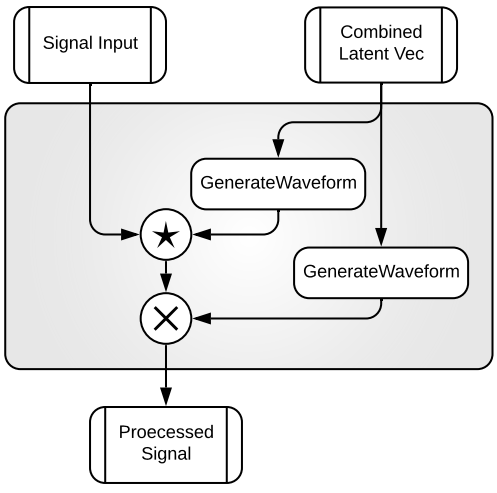


Fig. 3. Filter block capable of generating convolutive filters and per-sample masks from latent signal representations. Two individual upsampling blocks are used for the generation of the required filter vectors.

B. Filtering and Masking - FilterBlock

The FilterBlock, shown in Figure 3, combines two main filtering approaches, namely convolutive filters and per-sample filter masks. Two individual decoders (*GenerateWaveform*, subsection III-D) are used to generate filter vectors from the latent vector input. The filters are then applied to the input signal. This flexible architecture provides capabilities for masking, deconvolution, FIR-filtering, scaling, and any desired combination of the aforementioned methods. For the configuration used to generate the results presented in this paper, the prefiltering blocks within the Aligner only use the masking components. The alignment filter generation exclusively relies on convolutive filtering to compensate spatial propagation and enhance the signal's spectral content.
The alignment filter presents an exception from the other FilterBlocks: prior to the application of the filter, it is normalized to zero mean and unit variance. Additionally, a small multilayer perceptron is used to estimate the mean and variance of desired signal components from the latent vector. These scalars are then added to the latent vector which is later used for the weighted average operation in the outer model structure.

C. Encoder - EncodeWaveform

To accommodate variable signal lengths with the same general architecture, the encoders are constructed using an iterative core, shown in Figure 4. After normalization to zero mean and unit variance, activations and a multiplicative gate as used in WaveNet architectures [20] are created using convolutional layers. After passing through LayerNorm normalization [21], the output and the previous input are concatenated and passed to a convolutional down-sampling layer which services as the input of the next iteration. Once the required number of iterations has been performed, a final dense layer transforms the activations into the latent vector, which is then concatenated with the standard deviation and the mean of the input signal prior to normalization.

D. Decoder - GenerateWaveform

The decoder blocks are designed with several purposes in mind and present the inverse operation of the encoder. Such blocks can be used to create filter masks, spatial filters, gating vectors and the ensemble weight vectors. Although the general architecture is identical for all applications, the iterative core of the Gating block enables a variable definition of the desired output size.

IV. Experimental Setup

A. Training Data

Test and training data were created by simulating virtual acoustic environments using the image method [22], as implemented in [23]. To initially reduce the complexity of the problem, data simulation was limited to a fairly low order, concentrating on direct sound and early reflections and omitting diffuse reverb tails. Details on the motivation for this are discussed in section VI-C. As this Neural Beamformer architecture is uniquely capable of processing large time delays, room geometries were synthesized ranging from $8\,\mathrm{m}$ to $50\,\mathrm{m}$ per dimension. One *signal*, a random number from three to seven *noise* sources, and a fixed number of microphones depending on the network configuration were randomly placed in the synthetic virtual environment. For training, 500000 multichannel buffers were randomly sampled from 500000 virtual recordings. Each buffer contains a reference channel, spanning 4096 samples and M microphone signals, spanning the same 4096 samples, preceded by a context window of 12288 samples[2], resulting in a total buffer length of 16384 samples. For the *signal* class, speech recordings from the VCTK corpus were used, *noise* samples were extracted from the ESC50 corpus [24, 25].

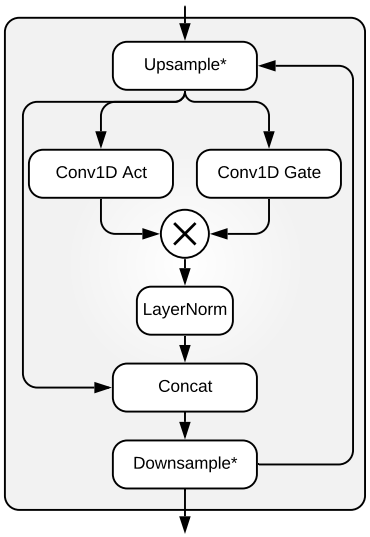


Fig. 4. Gating block used for up-sampling and down-sampling. Two paths within the graph combine to a multiplicative attention mechanism. Only one of the up-sampling/down-sampling layers are present for any Gating module, depending on the module in which it is used.

B. Training Process

Training was performed using standard MSE loss and the Ranger optimization algorithm [26], representing a combination of the Rectified Adam (RAdam) optimization algorithm [27] and Lookahead [28]. Standard Adam and Lookahead parameters were used, combined with a learning rate of 10^{-4} and a warm-up period of one 1000 steps, in which the learning rate is linearly ramped from near zero to the final learning rate. The final five epochs were trained with a reduced learning rate of $5 \cdot 10^{-6}$.
The training process was enhanced with a form of channel dropout in which individual input channels were randomly set to 0 or filled with audio of the same signal statistics but with no correlation to the other channels. This was simply performed by swapping individual buffers along the batch axis. The motivation of this dropout was to force the network to completely reject individual input channels, if necessary.

[2]Considering the maximum possible distance of approximately $78\,\mathrm{m}$, the given vector lengths still suffice for time alignment.

TABLE I
PERFORMANCE COMPARISON OF THE NEURAL BEAMFORMER WITH ORACLE AND GCC-PHAT BEAMFORMERS.

max mic dist in m	SNR in dB	GCC SDRi in dB	oracle BF SDRi in dB	NBF SDRi in dB
	-6	-0.85±2.17	4.13±0.82	6.26±2.18
24.9	0	0.09±1.78	4.15±0.82	5.05±1.65
	6	0.09±1.62	4.15±0.8	1.72±1.75
	-6	-0.95±2.28	4.19±0.86	6.53±1.87
55.8	0	0.07±1.83	4.16±0.89	4.78±1.93
	6	-0.04±1.43	4.03±1.01	0.81±2.23
	-6	-1.11±1.82	3.68±0.9	6.42±1.82
97.0	0	-0.36±1.4	3.55±1.0	4.98±1.56
	6	-0.69±1.06	3.09±1.38	1.42±1.89
	-6	-0.81±1.68	3.05±0.99	6.39±1.82
139.4	0	-0.29±1.25	2.79±1.16	4.51±1.88
	6	-1.07±1.12	2.0±1.72	0.52±2.25

C. Baseline Methods

Conventional Delay-and-Sum beamforming is referenced to better gauge the performance of the proposed method compared to algorithmic approaches. In most scenarios using distributed microphone arrays of large dimensions, the exact spatial relations between the individual microphones and noise and target sources remain unknown. For this reason, TDOA estimation using GCC-PHAT is implemented, using the full length of the Aux input buffers. To compare the Neural Beamformer with optimal DS beamforming, the results using oracle time delays are supplied as well.

To the knowledge of the authors, no multichannel deep learning architecture is currently capable of efficiently performing beamforming on randomly distributed microphone arrays over great distances.

V. RESULTS

Overall separation performance is monitored by means of SDRi, the improvement of the signal-to-distortion ratio, compared to the reference receiver. This commonly used metric is computed using the *mir_eval* toolbox [29]. In Table I, an evaluation of the SDRi of the proposed method using five channels is presented under varying conditions, compared with two baseline approaches described in section IV-C. The methods were compared on ten $40\,\mathrm{s}$ room simulations per configuration. Audio material, room dimensions, and both microphone and source positions were randomly sampled for each iteration. Each example was converted to blocks of 4096 + 12288 samples with $50\,\%$ overlap and passed to the individual processors. The resulting signals were Hann-windowed [30] and recombined for evaluation.

Audio excerpts of the comparison can be found at https://zieglerj.home.hdm-stuttgart.de/nbf.html.

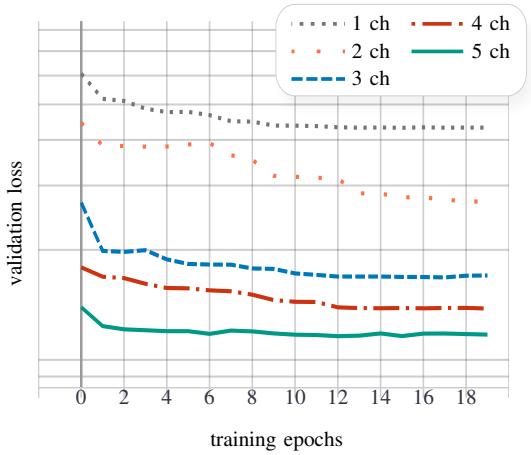


Fig. 5. Comparison of the proposed architecture using between one and five input channels. After training for 20 epochs, a clear trend favoring larger channel counts can be observed.

A. Separation Quality with Respect to Number of Microphones

To investigate the degree of beamforming applied within the model, the same network architecture was trained for 20 epochs on a subset of 50000 training examples using one to five input channels. Evaluation was performed on 10000 examples of unseen data synthesized as described in section IV. Figure 5 show the epoch loss of the five models over the training process. Both one- and two-channel models perform badly, while training performance increases noticeably as soon as the number of channels becomes larger than the number of reference channels. This is the case starting at $M = 3$, because the reference channel is identically present as one of the microphone inputs with the context frame. The motivation behind this is discussed in section VI. For larger channel counts, the difference in performance is less pronounced but a clear correlation between number of channels and validation loss can be observed.

In order to evaluate if training larger models has any advantage over expanding the models after training, three-channel and five-channel models were expanded after the training process was completed. As the Aligner blocks share weights over every channel and the ensemble weight is computed using only the mean latent vector and the individual latent vector of the channel, expanding or shrinking the model after training requires no additional training iterations. Figure 6 shows a comparison of relative SDRi between the original models and reconstructed models containing one to twelve channels. No clear correlation between channel count and SDRi can be observed for either model.

B. Separation Quality with Respect to Reverberation Level

As the signal quality of algorithmic beamformers based on TDOA estimations computed through GCC degrades with increasing reverberation, analyzing the effect on model performance is of particular interest. Figure 7 shows separation quality of both the Neural Beamformer and the baseline approach under variation of reverberation levels. Ten $40\,\mathrm{s}$ scenes

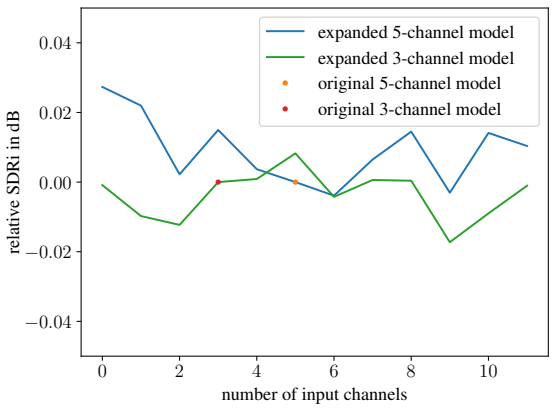


Fig. 6. Relative performance gain by expanding pre-trained three-channel and five-channel models. No clear correlation between channel count and relative SDRi can be observed. The mean relative SDRi over 1000 simulated scenes is shown.

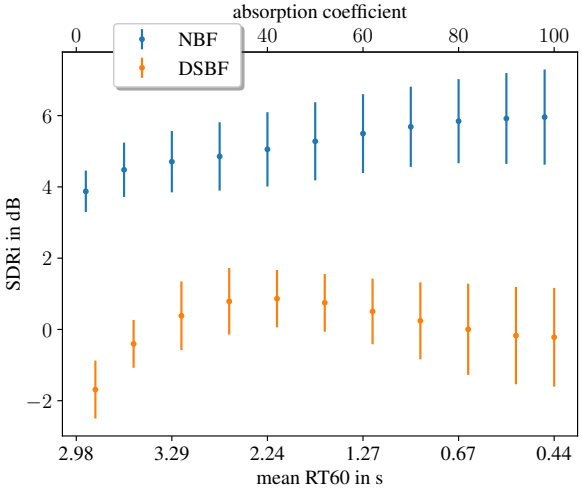


Fig. 7. Model performance with respect to the reverberation level. For most real-world reverberation patterns, relatively constant model performance can be expected.

were synthesized, using fixed room dimensions and a fixed SNR of 0 dB. Reverberation was manipulated by changing the absorption coefficients of the virtual room, ranging from 0.02 to 0.98. The model performs well over large ranges of reverberation levels, outperforming the baseline for every configuration. These results are particularly notable, as the entire training process was performed on virtual scenes in which the diffuse reverberation components were omitted. A detailed explanation of the motivation behind this approach is given in the discussion in section VI.

VI. Discussion

A. Use of Reference Channel

The explicit use of a reference receiver is not directly required for the experiments presented in this paper. The first microphone input channel could be defined as the reference

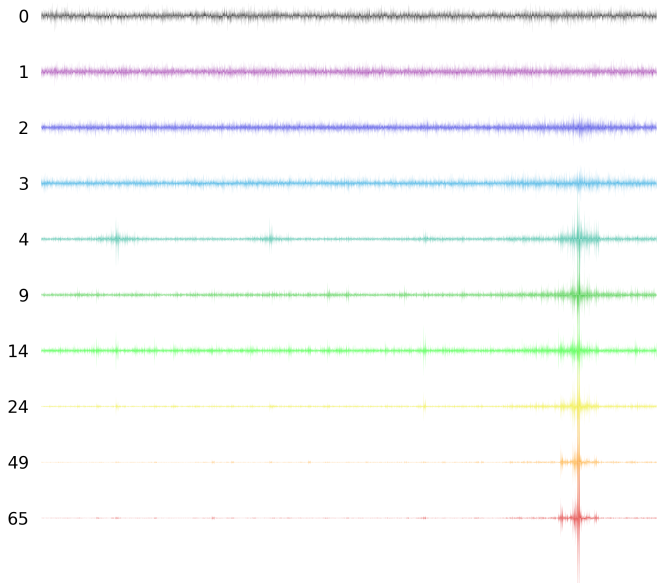


Fig. 8. Exemplary development of alignment filter generated in an Aligner block. During the early epochs, direct propagation paths are learned. In later epochs, early reflections are explored and increasingly included in the processing.

receiver and the Model would perform identically. The motivation behind the nomenclature of auxiliary and reference channel stems from the desire to optionally provide additional information to the network. In application scenarios in which it is possible to define that the target signal is always closest to the reference channel, this would present vital spatial and spectral information. As the baseline methods are not capable of processing such additional information and a comparison to existing methods was of key interest for this paper, the reference microphone in the experiments above was randomly sampled from the available microphones.

B. Evaluation of Alignment Filters

Based on the considerations in section II-A, it is expected that the network learns to generate filters that resemble transfer functions in the Aligner blocks. To confirm this, network activations for the output of the alignment filters were monitored during the training of a Neural Beamformer model. Figure 8 shows the filter generated for a variety of training epochs. For later models which produce positive SDRi, the basic structure of a transfer function becomes evident: in addition to the time-alignment of the direct sound, echos, early reflections, or periodic components are either enhanced or attenuated by means of additional peaks within the alignment filter. This clearly demonstrates that the network applies beamforming techniques to obtain the enhanced output signal.

C. Training with Simplified Simulated Data

As mentioned in sections IV-A and V-B, data simulation was performed without diffuse reverberation. As previously stated, the exact regression of optimal transfer Functions $H^{\nu ij}$

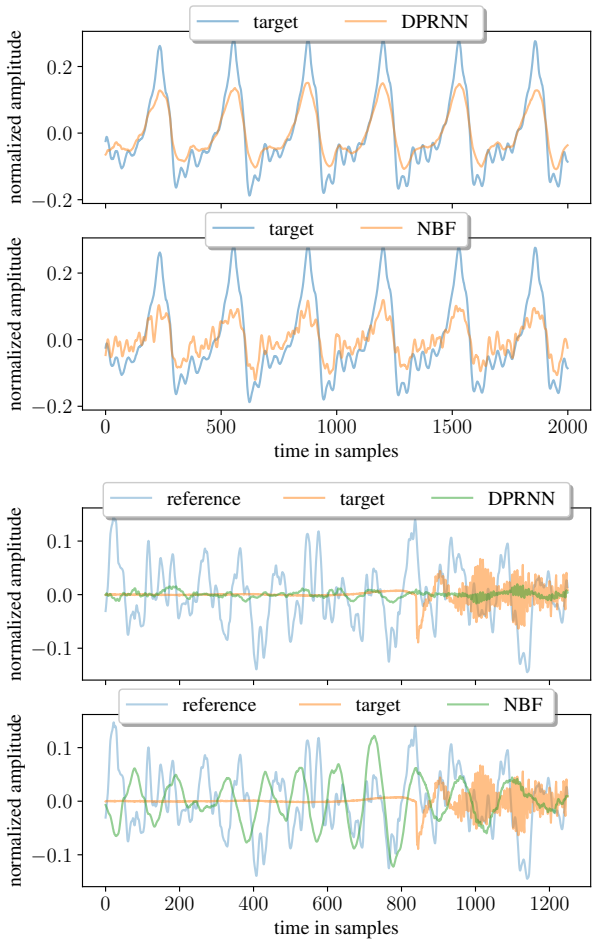


Fig. 9. Waveform comparison of DPRNN and NBF output, trained on the same dataset. While the DPRNN is better equipped to remove noise, the NBF can preserve high frequency content more effectively.

is extremely complex, preventing reliable training convergence of the proposed model. Concentrating on the main contributing factors for beamforming and excluding the task of dereverberation presents an option for efficient and reliable training of Neural Beamformers. Section V-B shows that such models are capable of performing well on data that contains a wide range of reverberation levels. Future iterations may include a multi-step training process in which a training on full simulations follows the current training process.

D. Subjective Versus Objective Performance

While comparable neural mask-based approaches, referenced in section I, outperform the Neural Beamformer in many tested scenarios with respect to absolute SDR improvement on speech enhancement, the different processing approaches are clearly audible. Directly masking the output signal results in more noticeable processing, presenting challenges to some applications. While speech recognition and general intelligibility-related tasks would see no detrimental effects from the mask-based processing, using the actual audio output of the system in recordings, broadcasting or other high-quality applications would not be possible. The beamforming approach completely omits direct masking of the output signal and guarantees a consistent, if less processed result. Figure 9 shows two excerpts of the audio generated during model evaluation that highlight two main differences between the approaches. While a mask-based model, such as a DPRNN, is capable of effectively removing noise, preserving high-frequency content is more challenging. The Neural Beamformer shows less overall noise reduction but better high-frequency retention.

VII. Conclusion

This paper presents a physics-informed End-to-End system for multichannel array processing, aimed at extracting domain-specific signals from a noisy mixture. The high accuracy of the model for short input vectors and an inference time in the order of $10\,\mu s$ on consumer-level CPU enable the system to be used in real time applications. When comparing with the algorithmic Delay-and-Sum beamformer, the advantage of multiple inputs and the inherent spatial information is apparent, even for very large microphone distances of up to approximately $75\,\mathrm{m}$.

Acknowledgments

This research was in part funded by the *Zentrales Innovationsprogramm Mittelstand*, a grant from the *Bundesministerium für Wirtschaft und Energie*.

References

[1] J. Benesty, I. Cohen, and J. Chen, *Fundamentals of Signal Enhancement and Array Signal Processing*. Singapore: John Wiley & Sons Singapore Pte. Ltd, Dec. 2017. [Online]. Available: http://doi.wiley.com/10.1002/9781119293132

[2] J. Benesty, C. Jingdong, and Y. Huang, *Microphone Array Signal Processing*. Springer Berlin Heidelberg, 2008. [Online]. Available: https://doi.org/10.1007/978-3-540-78612-2_3

[3] Ying Yu and H. F. Silverman, "An improved TDOA-based location estimation algorithm for large aperture microphone arrays," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 2004, pp. iv–iv.

[4] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, Aug. 1976.

[5] "Krisp | Noise Cancelling App," library Catalog: krisp.ai. [Online]. Available: https://www.krisp.ai/

[6] "Real-Time Noise Suppression Using Deep Learning," Oct. 2018, library Catalog: developer.nvidia.com. [Online]. Available: https://developer.nvidia.com/blog/nvidia-real-time-noise-suppression-deep-learning/

[7] M. Moussallam, "Releasing Spleeter: Deezer R&D source separation engine," Feb. 2020, library Catalog: deezer.io. [Online].

Available: https://deezer.io/releasing-spleeter-deezer-r-d-source-separation-engine-2b88985e797e

[8] J. Barker, S. Watanabe *et al.*, "The fifth'CHiME'speech separation and recognition challenge: dataset, task and baselines," *arXiv preprint arXiv:1803.10609*, 2018.

[9] X. Xiao, S. Watanabe *et al.*, "Deep beamforming networks for multi-channel speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Shanghai: IEEE, Mar. 2016, pp. 5745–5749. [Online]. Available: http://ieeexplore.ieee.org/document/7472778/

[10] R. Gu, L. Chen *et al.*, "Neural Spatial Filter: Target Speaker Speech Separation Assisted with Directional Information," in *Interspeech 2019*. ISCA, Sep. 2019, pp. 4290–4294. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2019/abstracts/2266.html

[11] K. Qian, Y. Zhang *et al.*, "Deep Learning Based Speech Beamforming," *arXiv:1802.05383 [cs, eess]*, Feb. 2018, arXiv: 1802.05383. [Online]. Available: http://arxiv.org/abs/1802.05383

[12] Z.-Q. Wang and D. Wang, "All-Neural Multi-Channel Speech Enhancement," in *Interspeech 2018*. ISCA, Sep. 2018, pp. 3234–3238. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2018/abstracts/1664.html

[13] Y. Koyama and B. Raj, "W-Net BF: DNN-based Beamformer Using Joint Training Approach," *arXiv:1910.14262 [cs, eess]*, Oct. 2019, arXiv: 1910.14262. [Online]. Available: http://arxiv.org/abs/1910.14262

[14] T. N. Sainath, R. J. Weiss *et al.*, "Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. Scottsdale, AZ, USA: IEEE, Dec. 2015, pp. 30–36. [Online]. Available: http://ieeexplore.ieee.org/document/7404770/

[15] ——, "Multichannel Signal Processing With Deep Neural Networks for Automatic Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 965–979, May 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7859320/

[16] Y. Luo, E. Ceolini *et al.*, "FaSNet: Low-latency Adaptive Beamforming for Multi-microphone Audio Processing," *arXiv:1909.13387 [cs, eess]*, Sep. 2019, arXiv: 1909.13387. [Online]. Available: http://arxiv.org/abs/1909.13387

[17] Y. Luo and N. Mesgarani, "TasNet: Surpassing Ideal Time-Frequency Masking for Speech Separation," *arXiv:1809.07454 [cs, eess]*, Sep. 2018, arXiv: 1809.07454. [Online]. Available: http://arxiv.org/abs/1809.07454

[18] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation," *arXiv:1910.06379 [cs, eess]*, Oct. 2019, arXiv: 1910.06379. [Online]. Available: http://arxiv.org/abs/1910.06379

[19] J. H. DiBiase, "A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays," Ph.D. dissertation, Brown University, 2000.

[20] A. v. d. Oord, S. Dieleman *et al.*, "WaveNet: A Generative Model for Raw Audio," *arXiv:1609.03499 [cs]*, Sep. 2016, arXiv: 1609.03499. [Online]. Available: http://arxiv.org/abs/1609.03499

[21] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *arXiv:1607.06450 [cs, stat]*, Jul. 2016, arXiv: 1607.06450. [Online]. Available: http://arxiv.org/abs/1607.06450

[22] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[23] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A Python package for audio room simulations and array processing algorithms," *arXiv e-prints*, p. arXiv:1710.04196, Oct. 2017.

[24] C. Veaux, J. Yamagishi *et al.*, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2017.

[25] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM '15. New York, NY, USA: ACM, 2015, pp. 1015–1018, event-place: Brisbane, Australia. [Online]. Available: http://doi.acm.org/10.1145/2733373.2806390

[26] L. Wright, "New Deep Learning Optimizer, Ranger: Synergistic combination of RAdam + LookAhead for the best of both." Sep. 2019, library Catalog: medium.com. [Online]. Available: https://medium.com/@lessw/new-deep-learning-optimizer-ranger-synergistic-combination-of-radam-lookahead-for-the-best-of-2dc83f79a48d

[27] L. Liu, H. Jiang *et al.*, "ON THE VARIANCE OF THE ADAPTIVE LEARNING RATE AND BEYOND," in *Proc. ICLR2020*, 2020, p. 13.

[28] M. Zhang, J. Lucas *et al.*, "Lookahead Optimizer: k steps forward, 1 step back," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle *et al.*, Eds. Curran Associates, Inc., 2019, pp. 9597–9608. [Online]. Available: http://papers.nips.cc/paper/9155-lookahead-optimizer-k-steps-forward-1-step-back.pdf

[29] C. Raffel, B. McFee *et al.*, "MIR_eval: A Transparent Implementation of Common MIR Metrics." in *ISMIR*, H.-M. Wang, Y.-H. Yang, and J. H. Lee, Eds., 2014, pp. 367–372. [Online]. Available: http://dblp.uni-trier.de/db/conf/ismir/ismir2014.html#RaffelMHSNLE14

[30] J. O. S. III, *Spectral Audio Signal Processing*. W3K Publishing, Dec. 2011. [Online]. Available: https://www.xarg.org/ref/a/0974560731/