Thèse n° 7215

Sparse and Parametric Modeling with Applications to Acoustics and Audio

Présentée le 24 janvier 2020

à la Faculté des sciences et techniques de l’ingénieur
Laboratoire de traitement des signaux 2
Programme doctoral en informatique et communications

pour l’obtention du grade de Docteur ès Sciences

par

Helena PEIĆ TUKULJAC

Acceptée sur proposition du jury

Prof. P. Frossard, président du jury
Prof. P. Vandergheynst, Dr H. Lissek, directeurs de thèse
Dr P. Balazs, rapporteur
Prof. L. Daudet, rapporteur
Prof. J.-Ph. Thiran, rapporteur

École
polytechnique
fédérale
de Lausanne

2020

> As an academic, I wish I could tell you that the tendency to fall in love with our own ideas never happens in the clean, objective world of science. After all, we like to think that scientists care most about evidence and data and that they all work collectively, without pride or prejudice, toward a joint goal of advancing knowledge. This would be nice, but the reality is that science is carried out by human beings. As such, scientists are constrained by the same 20-watts-per-hour computing device (the brain) and the same biases (such as a preference for our own creations) as other mortals. In the scientific world, the Not-Invented-Here bias is fondly called the "toothbrush theory". The idea is that everyone wants a toothbrush, everyone needs one, everyone has one, but no one wants to use anyone else's.
>
> — Dan Ariely

To all those people that stood by my side in the good and especially in the not so good times.

Acknowledgements

I would like to take a moment to reflect on the great academic and professional support I have received from numerous people across different countries and cultures over the course of my PhD studies.

First of all, I would like to thank my supervisor, professor Pierre Vandergheynst. By introducing me to the right people in Roscoff in 2017 and also by giving me all the freedom of choice and financial support, he has made many great things possible. Also, establishing connections through the TransformTECH program, in which he was one of the initiators, has helped me to expand my horizons and understand the perception of people on data science and artificial intelligence. I would also like to thank my co-advisor, Hervé Lissek, that gave me his trust at the point when I was transiting from Martin Vetterli's laboratory (LCAV - The Audiovisual Communications Laboratory) to LTS2 - The Signal Processing Laboratory 2.

I would like to thank Dr. Benjamin Ricaud and Dr. Nicolas Aspert for all the effort they have invested in SpectroBank and also for being great people.

Over the course of the PhD I have done two internships - one in French institute INIRIA, in PANAMA (Parsimony and New Algorithms for Audio and Signal Modeling) team (Rennes) and the another in Microsoft Research, on a joint project between Spatial Audio Team and Interactive Media Group (Redmond). In INRIA I would like to thank Dr. Antoine Deleforge for his enthusiasm, sense of humor and most of all - patience and hunger for scientific discoveries, and also professor Rémi Gribonval, for supporting my internship in INRIA. In Microsoft I would like to thank Dr. Nikunj Raghuvanshi for the incredible understanding he has of audio and computer graphics and for the devotion to work and patience he had for me, throughout the internship.

I would like to thank people from Logitech that participated in Master thesis co-supervision for our students.

I would like to thank all the collaborators that have provided their contribution in written or oral form.

I would like to thank all the students for trusting me as an advisor. I would like to thank the CHIC student start up project for all the inspiring interaction that we had with young students-enterpreneurs.

On the other hand, for invaluable support and unconditional love, I need to thank my mother Ružica, my father Emil, and my sister Lidija. All of this would have never been possible without all the hours you spent with me during our Sunday Skype sessions. You have never doubted whether I can make it this far.

I thank all my friends, from all around the globe, regardless of the fact if they have joined me for this incredible PhD journey for just a couple of months or for most of the way.

Lausanne, October 1, 2019

Abstract

Recent advances in signal processing, machine learning and deep learning with sparse intrinsic structure of data have paved the path for solving inverse problems in acoustics and audio. The main task of this thesis was to bridge the gap between the powerful mathematical tools and challenging problems in acoustics and audio. This thesis consists out of two main parts.

The first part of the thesis focuses on the questions related to acoustic simulations that comply with the "real world" constraints and the acoustic data acquisition inside of closed spaces. The simulated and measured data is used to solve various types of inverse problems with underlying sparsity. By using the technique of compressed sensing, we estimate the room modes, localize sound sources in a room and also estimate room's geometry. The Finite Rate of Innovation technique is coupled with non-convex optimization for the task of blind deconvolution in the context of echo retrieval. We also invent a new statistical measure for the echo density for the purpose of detecting the type of acoustic environment from its acoustic impulse response, even beyond fully closed spaces. These types solutions can have an application in the blooming domain of virtual, augmented and mixed reality for sound compression and rendering.

The second part of the thesis focuses on the recent trends in machine learning that are centered around deep learning. Large scale data acquisition of acoustic impulse responses is still a challenging and very expensive task. Also, the existing databases tend to be too heterogeneous to be merged, due to the lack of the standardization of the acquisition procedure, and also the available metadata tends to be incomplete. In order to keep up with the recent trends and avoid the difficulties that come from the lack of large scale acoustical data, the last part of research in this thesis has diverged from the rest and is devoted to deep learning applied to classification problems in audio with the focus on speech and environmental sounds. The learning procedure is parametrized, which results in an off-grid learning procedure for audio classification. Learned trends align with perceptual trends, which helps the interpretation of the achieved results.

Keywords: *Acoustics, audio, classification, compressed sensing, deep learning, finite rate of innovation, inverse problems, localization, optimization, parametric models, perceptual models, room acoustics, sensing matrix design, signal processing, simulations, sparsity.*

Résumé

Les progrès récents en traitement du signal et en apprentissage automatique profond sur des données parcimonieuses ont ouvert la voie à la résolution des problèmes inverses dans les domaines de l'acoustique et de l'audio. Le sujet principal de cette thèse est d'utiliser ces puissants outils mathématiques pour résoudre des problèmes complexes de ces domaines.

La première partie se concentre sur des questions liées à des simulations acoustiques physiquement réalistes et l'acquisition de données dans les espaces fermés. Les données simulées et mesurées sont utilisées pour résoudre plusieurs types de problèmes inverses faisant appel à la parcimonie sous-jacente. En utilisant des méthodes d'acquisition comprimée, il est possible d'estimer les modes d'une salle, de localiser les sources sonores dans celle-ci, ou encore d'en estimer la géométrie. Les techniques liées aux signaux à taux d'innovation finie et l'optimisation non-convexe sont couplées pour estimer les échos par déconvolution "aveugle". Une nouvelle mesure statistique de la densité des échos est introduite pour détecter le type d'environnement à partir de la réponse impulsionnelle acoustique. Ces techniques ont des applications prometteuses dans les domaines de la réalité virtuelle, augmentée et mélangée, notamment pour la compression des sons et leur rendu.

La deuxième partie de cette étude porte sur les récentes avancées dans le domaine de l'apprentissage automatique, et plus spécifiquement de l'apprentissage profond. L'acquisition à grande échelle de données de réponses impulsionnelles acoustiques reste une tâche coûteuse et complexe. Les différentes bases de données existantes sont trop hétérogènes pour être fusionnées, du fait du manque de standardisation des procédures d'acquisition et des métadonnées incomplètes. Pour pouvoir suivre les récents développements et éviter les difficultés provoquées par ce manque de données acoustiques à grande échelle, la dernière partie de ce travail est dédiée à l'apprentissage prodond appliqué à des problèmes de classification audio, plus précisément sur des données de bruits environnementaux et de parole. La procédure d'apprentissage est paramétrisée de manière continue. Les résultats de l'apprentissage suivent certaines caractéristiques psychoacoustiques, ce qui simplifie leur interprétation.

Mots clés : *Acquisition comprimée, acoustique, acoustique des salles, apprentissage profond, audio, classification, localisation, matrice d'acquisition, modèles perceptuels, modèle paramétrique, optimisation, parcimonie, problème inverse, simulation, taux d'innovation finie, traitement du signal*

Contents

Acknowledgements v

Abstract vii

List of Mathematical Notations xv

List of Acoustical Symbols xvii

List of Figures xix

List of Tables xxi

I Introduction and Contribution 1

Introduction 3

Contribution 7

II Inverse problems in acoustics with underlying sparsity 9

1 Acoustic background 11
- 1.1 Acoustic waves 11
- 1.2 Room acoustics 12
  - 1.2.1 The Plenacoustic Function 12
  - 1.2.2 Room acoustics of a rectangular room 14
  - 1.2.3 Plane wave decomposition of sound pressure in a room 16
  - 1.2.4 Image-source model 17
- 1.3 Acoustic modeling 17
  - 1.3.1 Acoustic simulators 19
  - 1.3.2 Acoustic measurements 21
- 1.4 Inverse problems in acoustics with underlying sparsity 21

2 Mathematical Background 25
- 2.1 Inverse problems with underlying sparsity 25
  - 2.1.1 Sparse regularizers 26

2.1.2 Sparse problems' framework 27

2.2 On-grid modeling of sparsity 29

2.2.1 Synthesis vs analysis approach 29

2.2.2 (On-grid) Compressed (Compressive) sensing 29

2.2.3 Cosparsity 33

2.2.4 A comparison of sparse and cosparse model 33

2.2.5 Basis mismatch - How the on-grid models became deprecated 34

2.3 Off-grid modeling of sparsity 36

2.3.1 How to be less greedy and less griddy? 36

2.3.2 Off-grid Compressed Sensing 37

2.3.3 Finite Rate of Innovation 39

2.4 Relationship between compressed sensing and Finite Rate of Innovation 41

III Sparse models for room acoustics: Compressed sensing and room modes 43

3 Localization of Sound Sources in a Room with One Microphone 45

3.1 Modal representation of the sound pressure and its low-frequency properties . 46

3.1.1 Room Transfer Function at different positions across the room 46

3.1.2 Ambiguities that exist in the terms of uniqueness of the attenuation across the room 47

3.2 Compressed sensing and sound source localization 48

3.2.1 Sparse representation of the position of sources 48

3.2.2 Problem formulation within compressed sensing approach 48

3.2.3 A sensing matrix for sound source localization in a room 49

3.3 Designing an efficient sensing matrix 50

3.3.1 Coherence 50

3.3.2 Battle of the grids 51

3.4 Results 52

3.4.1 The recovery of signal's support in a highly coherent dictionary 52

3.4.2 Precision and basis mismatch 55

3.4.3 Requirements and limitations 56

3.5 Conclusion 56

4 Joint Estimation of Room Geometry and Modes with Compressed Sensing 57

4.1 Problem Setup 58

4.1.1 Spherical search space of plane wave approximation 58

4.1.2 Periodicity of the wave vector grid 59

4.2 Parameter estimation with partial compressed sensing for structured data . . . 59

4.2.1 Reconstruction procedure 60

4.3 Results 62

4.3.1 Reconstruction of the k-space of a rectangular room 62

4.4 Reconstruction of the room shape . . . 63
4.5 Conclusion . . . 64
4.6 Future work . . . 64

IV Parametric models for echo estimation: Location, weight and density 65

5 MULAN: A Blind and Off-Grid Method for Multichannel Echo Retrieval 67
5.1 The signal and measurement models . . . 69
5.2 Existing methods in channel identification . . . 71
5.3 Off-grid echo retrieval by multichannel annihilation . . . 73
5.3.1 The non-blind case . . . 73
5.3.2 MULAN: an iterative scheme . . . 74
5.3.3 Identifiability and ambiguities . . . 75
5.4 Experiments . . . 76
5.4.1 On-grid vs. off-grid echo retrieval . . . 76
5.4.2 Influence of K, M, F on recovery rate . . . 77
5.4.3 A discussion on the minimum separation of Diracs . . . 78
5.5 Conclusion . . . 79

6 Estimating Early Acoustic Echoes from Noisy Speech with Multichannel Structured Low-Rank Optimization 81
6.1 Cadzow denoising algorithm . . . 81
6.2 Data model and Cadzow denoising . . . 82
6.3 Weighted cross-relation for a multichannel case . . . 83
6.3.1 Binaural (2-channel) case . . . 83
6.3.2 Multichannel case . . . 84
6.4 An alternating projections algorithm: MUSHU . . . 86
6.5 Discussion on results and data collection . . . 86
6.6 Conclusion . . . 88

7 A Sparsity Measure for Echo Density Growth in General Environments 89
7.1 Echo density measure . . . 90
7.1.1 Local energy normalization . . . 91
7.1.2 Sorted Density (SD) . . . 91
7.2 Statistical model . . . 92
7.2.1 Analytical motivation . . . 92
7.2.2 Model fitting . . . 94
7.3 Results . . . 95
7.3.1 Experimental data . . . 95
7.3.2 Validation on simple enclosures . . . 96
7.3.3 Indoor to outdoor scene modification . . . 97
7.3.4 Varying volume with fixed reverberation time . . . 97
7.3.5 Indoor versus outdoor location in urban area . . . 98

7.4 Conclusion and future work 99
7.5 Acknowledgements 100

V Parametric models for audio classification 101

8 Audio Representations for Deep Learning 103
8.1 Audio representations with fixed parameters 104
8.1.1 The most common parameters of fixed representations 105
8.2 Learning without an intermediate representation: End-to-end approaches in audio 106
8.3 Audio representations with learnable parameters 106
8.4 Audio datasets for audio classification tasks 107

9 SpectroBank: A Filter-bank Convolutional Layer for CNN-based Audio Applications 109
9.1 Learnable filter banks (SpectroBank) 110
9.2 Experiments and results 113
9.2.1 The impact of SpectroBank parameters 114
9.2.2 *AudioMNIST* Results 115
9.2.3 *GoogleSpeechCommands* Results 116
9.2.4 *UrbanSound8K* Results 117
9.2.5 Properties of learned filters 118
9.3 Interpretation of network's decision making 120
9.4 Conclusion 121

VI Conclusion and Future Work 123

Conclusion 125

A Proof: Binaural weighted norm as an objective function 127

B Network architectures 129

C Perceptually motivated models: ERB and Bark scales 133

Bibliography 135

Curriculum Vitae 151

List of Mathematical Notations

x, X	scalars
$x(\cdot), X(\cdot)$	functions
$\boldsymbol{x}$	vector
$\boldsymbol{x}^*$	conjugate of vector $\boldsymbol{x}$
$\boldsymbol{x}_i$	i^{th} vector in a list of vectors
$\boldsymbol{x}[i]$	entry of vector $\boldsymbol{x}$ at the i^{th} position
$\boldsymbol{x}^{(i)}$	value of vector $\boldsymbol{x}$ in i^{th} iteration of an algorithm
$\mathbf{X}$	matrix
$\mathbf{X}^\dagger$	Moore-Penrose pseudo inverse
$\mathbf{X}^*$	conjugate of matrix $\mathbf{X}$
$\mathbf{X}^T$	transpose of matrix $\mathbf{X}$
$\overline{\mathbf{X}}$	Hermitian conjugate of matrix $\mathbf{X}$
$\mathbf{X}_i$	i^{th} matrix in a list of matrices
$\mathbf{X}[i,j]$	entry of matrix $\mathbf{X}$ at the i^{th} row and j^{th} column
$\mathbf{X}^{(i)}$	value of matrix $\mathbf{X}$ in i^{th} iteration of an algorithm
$\operatorname{diag}(\boldsymbol{x})$	matrix whose diagonal is equal to vector $\boldsymbol{x}$
$\operatorname{diag}(\mathbf{X})$	vector formed by extracting diagonal of matrix $\mathbf{X}$
$\operatorname{trace}(\mathbf{X})$	sum of elements on the diagonal of matrix $\mathbf{X}$
$\operatorname{Toep}(\boldsymbol{x})$	Toeplitz matrix formed out of element of vector $\boldsymbol{x}$
$P_{\boldsymbol{x}}$	polynomial constructed out of $\boldsymbol{x}$ as its coefficients
$\mathscr{P}_{\mathscr{S}}$	projection to a space $\mathscr{S}$
$\mathbf{I}_N$	$N \times N$ identity matrix
$\mathbb{R}$	set of real numbers
$\mathbb{Z}$	set of integer numbers
$\mathbb{N}, \mathbb{N}_0$	set of natural numbers excluding and including zero
$\mathbb{M}$	set of indices
$\mathscr{A}$	set of points or matrices
$\operatorname{conv}(\mathscr{A})$	convex hull of points in set $\mathscr{A}$
$\\|\boldsymbol{x}\\|_p$	ℓ_p norm of vector $\boldsymbol{x}$, $\\|\boldsymbol{x}\\|_p = (\sum_n \lvert x_n\rvert^p)^{\frac{1}{p}}, p \in [0,\infty]$
$\partial^n x$	n^{th} order partial derivative of function x
∇x	gradient of function x
Δx	Laplacian operator $\Delta = \nabla^2$ of function x
e, j, π	constants

List of Acoustical Symbols

p	sound pressure
c	sound celerity (speed)
ρ	air density
Q	volume flow velocity of the sound source
t	time
$\boldsymbol{r}$	spatial coordinate; usually $\boldsymbol{r} \in \mathbf{R}^3$, $\boldsymbol{r} = [x, y, z]^T$
T	linear period
f	linear frequency, $f = \frac{1}{T}$
λ	wavelength, $\lambda = cT$
ω	angular frequency, $\omega = 2\pi f$
Δ_t	temporal sampling step
f_s	temporal sampling frequency, $f_s = \frac{1}{\Delta_t}$
f_{sch}	Schroeder frequency
Δ_i	spatial sampling step, $i \in \{x, y, z\}$
φ_i	spatial sampling frequency, $\varphi_i = \frac{2\pi}{\Delta_i}$, $i \in \{x, y, z\}$
ξ	room mode damping
κ	wave number, $\kappa = (\omega + j\xi)/c$
$\boldsymbol{k}$	wave vector, $\boldsymbol{k} \in \mathbf{R}^3$, $\boldsymbol{k} = [k_x, k_y, k_z]^T$
Ξ	room mode
t_{60}	reverberation time
$\Box p$	d'Alembert operator $\Box = \frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \nabla^2$ of p
$\boldsymbol{x}$	vector in discrete time domain
$\hat{\boldsymbol{x}}$	vector in discrete frequency domain

List of Figures

1 Schlieren experiment: gun shot . . . 3
2 Schlieren experiment: firecracker . . . 3
3 Schlieren experiment: book on a table . . . 3

1.1 Wave properties . . . 12
1.2 Plenacoustic function . . . 13
1.3 Plane waves inside a rectangular room . . . 15
1.4 Eigenvalue space of a rectangular room with rigid walls . . . 15
1.5 Image-source model: true and virtual source . . . 18
1.6 Image-source model: repetitions . . . 18
1.7 Room modes below Schroeder frequency . . . 19
1.8 Room modes above Schroeder frequency . . . 19
1.9 Sound propagation . . . 20
1.10 Acoustic simulation: pyroomacoustics . . . 20
1.11 Acoustic simulation: Triton . . . 20
1.12 Acoustic measurements: The measurement setting . . . 21
1.13 Acoustic measurements: The microphone positions . . . 21
1.14 Sparsity in acoustics: Room modes . . . 22
1.15 Sparsity in acoustics: Spectrogram . . . 22
1.16 Sparsity in acoustics: Room Impulse Response . . . 22
1.17 Sparsity in acoustics: Direction of Arrival . . . 22

2.1 ℓ_p norms . . . 26
2.2 An illustration of ℓ_0 against ℓ_1 norm . . . 26
2.3 Compressed sensing . . . 30
2.4 Basis mismatch problem: Ground truth . . . 34
2.5 Basis mismatch problem: Compressed sensing . . . 34
2.6 Finite Rate of Innovation: Samples in time . . . 39
2.7 Finite Rate of Innovation: Annihilating filter . . . 39
2.8 Annihilating polynomials of different degrees . . . 40
2.9 Temporal evolution of sparse methods . . . 41

3.1 Room modes in a room transfer function . . . 47
3.2 Room transfer function . . . 48

3.3 Room modes for a rigid wall case 49
3.4 Basic room modes and their amplitudes 50
3.5 Battle of the grids 51
3.6 Results: Localization of sources inside a room 54
3.7 Spatial and feature grid subsampling for sound source localization 55

4.1 Plane waves in search space 59
4.2 Periodicity of the wave vector grid 60
4.3 k-space 63

5.1 On and off grid Dirac train 69
5.2 MULAN: Phase transition diagram of location retrieval for $nF = 201$ 78
5.3 MULAN: Phase transition diagram of weight retrieval for $nF = 201$ 78
5.4 MULAN: Phase transition diagram of location retrieval for $nF = 401$ 79
5.5 MULAN: Phase transition diagram of weight retrieval for $nF = 401$ 79

7.1 From acoustic impulse response to normalized echogram 90
7.2 From normalized echogram to echo density profile 90
7.3 Echo density trends for various types of space 93
7.4 Parametric model for echo density trend 94
7.5 Echo density model for various types of space 95
7.6 Simulated shoebox rooms and echo density extraction 96
7.7 Acoustic simulations in convex polyhedral rooms and their echo density trends 97
7.8 Acoustic simulations in a room with sliding ceiling and their echo density trends 98
7.9 Acoustic measurements of three rooms with the same energy decay, but different echo density trends 99
7.10 Acoustic measurements in Microsoft campus 100

8.1 Audio representations with fixed parameters for deep learning 105
8.2 Dataset energy distribution per class and corresponding labels 108

9.1 Filter banks in time domain 111
9.2 Filter banks in frequency domain 111
9.3 SpectroBank: Influence of layer properties on the network accuracy 114
9.4 SpectroBank: Bandwidth and frequency of the learned Gammatone filters . . . 119
9.5 Cumulative frequency energy distribution for learned filters on AudioMNIST dataset, SpectroBank-XS network trained with Gammatone (order 4) and SincNet first layer, with 32 filters. 120
9.6 SpectroBank: Filter convergence illustration on synthetic data 121

List of Tables

2.1 Properties of the sparse (synthesis) and cosparse (analysis) approach 34
2.2 Overview of the sensing methods 42

5.1 Dirac recovery trends with MULAN 78

6.1 Room impulse response databases with and without labels for early reflections 87

8.1 Typical window types for fixed audio representations 105
8.2 Class statistics of *AudioMNIST*, *GoogleSpeechCommands v2* and *UrbanSound8K* 107

9.1 SpectroBank: filter parameter number and type 110
9.2 *AudioMNIST* mean test accuracy for networks with and without a SpectroBank layer 116
9.3 *GoogleSpeechCommands* mean test accuracy for networks with and without a SpectroBank layer 117
9.4 *UrbanSound8K* mean test accuracy for networks with and without a SpectroBank layer 118

B.1 SpectroBank custom architecture for *AudioMNIST*. 129
B.2 SpectroBank-XS custom architecture for AudioMNIST. 130
B.3 M3-SpectroBank custom architecture for *UrbanSound8K*. 130
B.4 M5-SpectroBank custom architecture for *UrbanSound8K*. 131
B.5 SampleCNN-SpectroBank basic block 131
B.6 SampleCNN-SpectroBank architecture for *GoogleSpeechCommands* or *Urbansound8K* 132

Introduction and Contribution Part I

Introduction

In the recent years there have been different solutions to tackle high-dimensional problems in a lower dimensional domain. Most of the challenges in engineering lay within the need for reducing the costs. On one hand, we want to reduce the amount of data we need to collect and still arrive to a meaningful conclusion and on the other hand, we want to reduce the number of steps when processing the data, in order to arrive to conclusions faster and keep them up to date. In the spirit of Occam's razor principle, we want to design and build solutions that have minimal complexity, but still serve the original purpose.

The content of this thesis will be evolving around sound, which is just a variation of air density. In a very interesting experiment called Schlieren experiment[1], we are able to visualize a varying density of a medium. Some experimenters have used this experimental setting to show how sound propagates in a visual manner, as can be seen in Figures 1, 2 and 3. In all three examples we have a point source: a gun shot, a firecracker and a book falling onto a table. As can be seen, the sound propagates in a bubble-like formation uniformly in all directions. We will be processing the sound in two contexts: in the context of its propagation inside an environment (acoustics) and also in the context of its characteristics (audio), for example - sound classification.

When observing the intensely blooming domain of acoustics and audio engineering, we can see a lot of changes in the recent years. After focusing on efficient sound reproduction for many decades, acoustic engineering became omnipresent in the consumer electronics industry. A need for having the ability to interact with devices in a more human-like manner has emerged,

---

[1] by Mike Hargather https://www.youtube.com/watch?v=px3oVGXr4mo

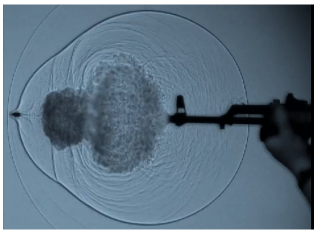


Figure 1 – Schlieren: gun shot.

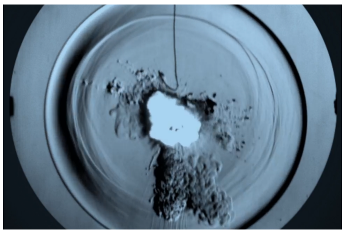


Figure 2 – Schlieren: firecracker.

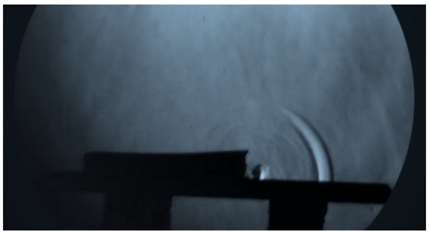


Figure 3 – Schlieren: book on a table.

so corporations had to push their devices towards a speech-based interaction mode. Recent trends also include: audio surveillance of the house, generation of music and human speech, audio event and genre tagging etc. Therefore, Spike Jonze's movie *Her* from 2013 might need to update its genre tag from science-fiction to something more appropriate, because the line that separated the possible from impossible keeps moving further and further on a daily basis. There is a strong competition on the market in the domain of the creation of conversational bots and lucrative competitions such as Alexa Prize [2] indicate a strong need of the market for such solutions.

So one may ask: why are we trying to focus on the recent trends in industry instead of science? The main motivating fact is that within every product in the consumer electronics device, there lay numerous algorithms to make the device fully functional. As an example, we will take the Amazon's device - Alexa. Some user might want to ask Alexa what is the weather going to be like tomorrow and would like to get a response in the audio form.

This thesis will be addressing the problems of acoustic signal processing and audio through the evolution of the available techniques in the last four years. Over the course of this thesis, the general trend has moved from griddy and greedy approaches to continuous and efficient approaches, but with potentially high number of tunable parameters and non-convex problem formulations. So the focus of this thesis will be on marrying parsimonious methods with the high-dimensional acoustical data, with an evolutionary perspective of these methods.

Therefore there is a need to give an introduction to the problem formulation and the state of the art of both sides - acoustical and mathematical. In order to motivate the application of such techniques to this type of data, we will start with an acoustical introduction, followed up by the identification of intrinsic sources of sparsity that exist within, together with the description of the plethora of available methods that have emerged for tackling such problems.

The main goal of this research is to increase the efficiency of existing methods for acoustical and audio data processing (by reducing the number of required measurements or improving the computational cost) and also to investigate new methods for inverse problems in acoustics and audio with underlying sparsity, such as sound source localization, room mode decomposition of room transfer function, detection of early reflections in room impulse response etc.

This thesis consists out of six main parts:

Part I: Gives an introduction to the problems that will be addressed throughout the thesis.

Part II: Establishes the background on the physical properties of acoustical data that lead to highly accurate simulators and gives examples of underlying sparsity hidden in the acoustical data that pave the path for parsimonious data processing. Once the sparsity has been recognized, we give an overview of various approaches that deal with data whose underlying

---

[2] https://developer.amazon.com/alexaprize

structure is approximately sparse.

Part III: Focuses on the acoustical behavior of rooms in the *low* frequency domain, below the Schroeder frequency where resonator type of behavior prevails. Here we use the sparse representation of room transfer function in the terms of room modes in order to localize sound sources inside rooms and also to estimate the dimensions of rectangular rooms along each one of the axis.

Part IV: Focuses on the acoustical behavior of rooms in *mid-high* frequency domain, above the Schroeder frequency where a room shows diffuse behavior. In this domain reflections from the walls behave like a billiard table. Here we model the room impulse response as an array of noisy Dirac pulses. We will be focusing on the estimation of the location and weight of early reflections and also on the estimation of the evolution of echo density within acoustic impulse responses, for the characterization of the acoustic environments even beyond rooms.

Part V: Focuses on the exploration of the recent *deep learning* techniques and their application on the audio classification task. Due to the fact that large room impulse response databases that could be used for these types of approaches are still unavailable[3] [60], we will be making a slight shift from parametric data exploration for acoustics to parametric data exploration for audio.

Part VI: Gives concluding remarks and discusses the potential future work.

---

[3] most of the available audio/acoustic datasets are listed here: http://www.cs.tut.fi/~heittolt/datasets.html

Contribution

During my four-year stay at EPFL, from 2015 to 2019, I have authored and co-authored[4] the following publications and technical reports (sorted in chronological order):

1. **H. Peic Tukuljac**, I. Dokmanic, J. Ranieri, M. Vetterli, *Time-varying FRI Theory for Sound Source Localization*, Technical report, EPFL, 2015. [132]
2. **H. Peic Tukuljac**, H. Lissek, P. Vandergheynst, *Localization of Sound Sources in a Room with One Microphone*, Wavelets and Sparsity XVII, SPIE 2017 [133]
3. **H. Peic Tukuljac**, T. Pham Vu, H. Lissek, P. Vandergheynst, *Joint Estimation of the Room Geometry and Nodes with Compressed Sensing*, ICASSP 2018 [136]
4. **H. Peic Tukuljac**, A. Deleforge, R. Gribonval, *MULAN: A Blind and Off-Grid Method for Multichannel Echo Retrieval*, NeurIPS 2018 [131]
5. **H. Peic Tukuljac**, V. Pulkki, H. Gamper, K. Godin, I. J. Tashev, N. Raghuvanshi, *A Sparsity Measure for Echo Density Growth in General Environments*, ICASSP 2019 [134]
6. B. Inan, M. Cernak, H. Grabner, **H. Peic Tukuljac**, R. C. G. Pena, B. Ricaud, *Evaluating Audiovisual Source Separation in the Context of Video Conferencing*, Interspeech 2019 [201]
7. **H. Peic Tukuljac**, B. Ricaud, N. Aspert, *SpectroBank: A Filter-bank Convolutional Layer for CNN-based Audio Applications*, submitted to ICLR 2020 [135]

Over the course of this thesis, collaboration has been established with INRIA institute in France, Microsoft Research in the USA and Logitech in Switzerland. In all the cases where there were no proprietary issues, the code was released in order to comply with EPFL's Open Science initiative.

Throughout all of these collaborations, I have had the role of the main contributor and was assisted by different groups of people mostly for discussion, state of the art collection and also in the terms of formulating the outcomes of the research.

---

[4] https://www.researchgate.net/profile/Helena_Peic_Tukuljac

Within the scope of this thesis, the content is coupled with publications in the following manner: Chapter 3 - [133], Chapter 4 - [136], Chapter 5 - [131], Chapter 7 - [134], Chapter 9 - [135], while Chapter 6 covers unpublished work and extension to [131].

Part II

Inverse problems in acoustics with underlying sparsity

1 Acoustic background

Acoustics is a branch of physics that deals with the study of all the mechanical waves in gases, liquids and solids, including topics such as vibration, sound, ultrasound and infrasound. Therefore, it focuses on modeling and processing of sound from a physical point of view. On the other hand, *audio engineering* is focused on the sound itself and uses techniques to recognize type of sound (classification), mix sound to create music or generate new sounds. In this thesis we will be applying parsimonious methods to both of these domains.

1.1 Acoustic waves

Sound is a physical phenomenon of change of the pressure inside of the medium of propagation. Propagation of sound through the means of mechanical waves has been studied in classic texts on theoretical acoustics [114], room acoustics [91] and partial differential equations [56, 72].

As can be seen in Figure 1.1.[1] waves are described by many properties. Within the scope of the thesis we will encounter:

- f (linear frequency) - number of oscillations/samples per second,
- ω (angular frequency) - number of radians per second,
- T (linear period) - length of periodicity in time,
- λ (linear wavelength) - length of periodicity in space, and
- k (angular wave number) - number of oscillations/samples per unit length.

Other types of wave properties include: amplitude and phase.

---

[1] Source: https://commons.wikimedia.org/wiki/File:Commutative_diagram_of_harmonic_wave_properties.svg.

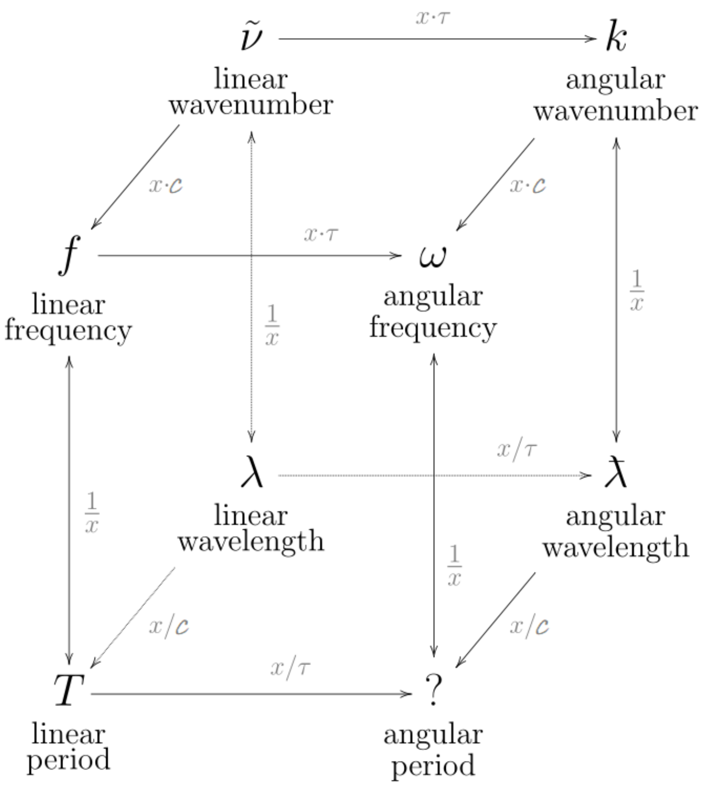


Figure 1.1 – Wave properties, $\tau = 2\pi$.

1.2 Room acoustics

1.2.1 The Plenacoustic Function

The behavior of waves is defined by the acoustic-wave equation. It is a second order partial differential equation [91]:

$$\nabla^2 p(t,\boldsymbol{r}) - \frac{1}{c^2}\frac{\partial^2 p(t,\boldsymbol{r})}{\partial t^2} = \begin{cases} 0, \text{if no source at the location } \boldsymbol{r} \\ s(t,\boldsymbol{r}), \text{if source at the location } \boldsymbol{r} \end{cases} \tag{1.1}$$

where $p(t,\boldsymbol{r})$ is sound pressure at observed location $\boldsymbol{r}$, c is celerity (speed) of sound and the right hand side represents the contribution to the sound pressure field. $s(t,\boldsymbol{r})$ represents a distribution of sources located in space. Sometimes this relationship is expressed by using the d'Alembert operator: $\Box = \frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \nabla^2 = \frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} - \frac{\partial^2}{\partial z^2}$. The speed of sound depends on the type of the environment where the sound propagates and also its temperature. The speed of sound in air is given by the following expression: $c(T) = (331.4 + 0.6T[C^o])\,\frac{\mathrm{m}}{\mathrm{s}}$.

The solution of the wave equation is called the plenacoustic function (PAF). For a specific case where the input signal is a Dirac pulse, we have a Green function case. The *plenacoustic function* is a set of all the impulse responses for all the possible (source position, receiver

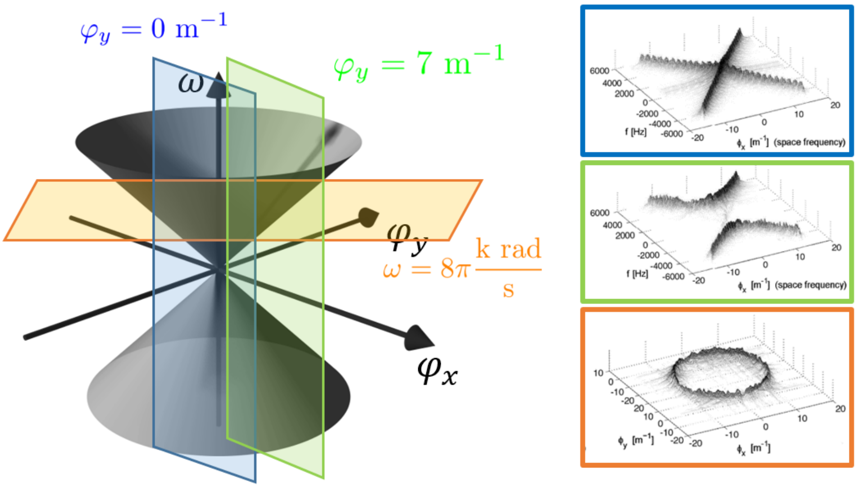

$\varphi_y = 0\ \mathrm{m}^{-1}$
$\varphi_y = 7\ \mathrm{m}^{-1}$
ω
φ_y
$\omega = 8\pi \frac{\mathrm{k\ rad}}{\mathrm{s}}$
φ_x
f [Hz]
ϕx [m⁻¹] (space frequency)
ϕy [m⁻¹]
ϕx [m⁻¹]

Figure 1.2 – The plenacoustic function against 2 spatial (x and y direction) and one temporal frequency. Note that this is a case for $2D$ spatial coordinates. For the real $3D$ case our data lays on a surface of a hypercone in $4D$. Images from the right hand side are from [112].

position) pairs inside a given room. It was defined in [9] and confirmed by measurements in [112], as shown in Figure 1.2.

When sampling sound we need to take into account two types of possible sources of aliasing: temporal and spatial. Therefore, we need to decide on the values for the temporal and spatial sampling frequencies: $\omega = \frac{2\pi}{\Delta_t} = 2\pi f_s$, $\varphi_x = \frac{2\pi}{\Delta_x}$, $\varphi_y = \frac{2\pi}{\Delta_y}$ and $\varphi_z = \frac{2\pi}{\Delta_z}$, where Δ_t is the temporal sampling step and $\Delta_x, \Delta_y, \Delta_z$ are spatial sampling steps. Depending on the highest frequency that we want to capture $f_{\max}$, we define our temporal sampling step Δ_t in such a way that the sampling frequency satisfies $f_s = \frac{1}{\Delta_t} \geq 2 f_{\max}$, which is a constraint given by Nyquist [120]. Once the temporal sampling step is fixed, we determine the appropriate sampling step in space either by the limits imposed by the Courant–Friedrichs–Lewy condition [97] for Finite Difference Time Domain (FDTD) schemes, or by a contemporary view of the problem observed through the sampling of the PAF [9]. In order to have an even more precise model for sound propagation, recent studies increase the dimensionality of this function by taking into account the direction of arrival for sound. One of the approaches is called DiRaC (Directional Audio Coding) [92], which offers parameterization similar to the high parameterization of the plenoptic function [5].

The support of the spectrum of the PAF $\hat{p}(\omega_s, \varphi_x, \varphi_y, \varphi_z)$, where $\omega = \frac{2\pi}{\Delta t} = 2\pi f_s$ is the temporal angular sampling frequency [rad.s^{-1}] and $\varphi_i = \frac{2\pi}{\Delta_i}$ is the spatial angular frequency [rad.m^{-1}] over each of the i^{th} observed axis, lays inside a hypercone [9]:

$$\varphi_x^2 + \varphi_y^2 + \varphi_z^2 \leq \frac{\omega^2}{c^2}, \tag{1.2}$$

where c is the celerity of sound. In case of 1D sampling, we have the following condition for the sampling step over the axis of interest: $\Delta_i > \frac{\pi c}{\omega_{\max}}$, $i \in \{x, y, z\}$ and $\omega_{\max}$ is the maximal frequency we want to capture ($\omega \geq 2\omega_{\max}$) [112]. As the Figure 1.2 shows, the set of the possible spectrums lays within a butterfly-like formation, for all the possible temporal and spatial sampling frequencies.

If we go back to the CFL conditions [97], we can state the following: if a wave travels across a discrete spatial grid and we want to compute its amplitude at discrete time steps of equal duration, then this duration must be less than the time for the wave to travel to adjacent grid points:

$$\Delta_t \sum_{i=1}^{D} \frac{c}{\Delta_i} \leq C_{\max}, \tag{1.3}$$

where D is the number of observed dimensions, $D \in \{1, 2, 3\}$ and $C_{\max}$ is usually equal to 1. Therefore we have: $\frac{c\Delta_t}{\Delta_i} \leq \frac{1}{D}$. This is in agreement with the PAF theory.

Spatial resolution is always more costly than the temporal resolution, since for taking more samples in space we either have to have a moving sensor or expensive sensor arrays, and the temporal resolution is usually not a problem, since even cheap microphones have high sampling frequencies. This is where the sparse models come into play. In order to fully exploit the potential of sparse models, we need to rely on the underlying structure of our data.

1.2.2 Room acoustics of a rectangular room

If we observe the room in a time independent form, its Helmholtz equation [91] reads:

$$\Delta p + \kappa^2 p = 0, \tag{1.4}$$

where $\kappa = \frac{\omega}{c}$ is the wave number, that is - the eigenvalue of the Laplacian, which is coupled with an eigenfunction called a *room mode*.

For a rectangular room of dimensions $L_x \times L_y \times L_z$, the eigenvalues and eigenfunctions of the Laplace operator have a closed form expression. So for eigenvalues we define the *resonant frequencies*:

$$\omega_{n_x n_y n_z} = c\pi \sqrt{\left(\frac{n_x}{L_x}\right)^2 + \left(\frac{n_y}{L_y}\right)^2 + \left(\frac{n_z}{L_z}\right)^2} \tag{1.5}$$

where $\{n_x, n_y, n_z\} \in \mathbb{N}_0$.

In a rectangular room, each eigenfunction (eigenmode of the Laplacian) represents a sum of 8 plane waves that share a wave number. The (n_x, n_y, n_z) room mode that represents a 3D standing wave, and at a position $\boldsymbol{r} = [x, y, z]^T$ is given by:

$$\Xi(\mathbf{k}_n, \mathbf{r}) = \sum_{i=1}^{8} a_i e^{j(\mathbf{S}(:,i)\odot\mathbf{k}_n)\cdot\mathbf{r}_m}, \tag{1.6}$$

where $\odot$ is a Hadamard product, $\mathbf{S}_{3\times 8}$ is a sign matrix whose columns alternate from $[1,1,1]^T$ to $[-1,-1,-1]^T$ (that is - $[\pm 1, \pm 1, \pm 1]^T$), $\mathbf{k}_n = [\frac{n_x\pi}{L_x}, \frac{n_y\pi}{L_y}, \frac{n_z\pi}{L_z}]^T$, $(n_x, n_y, n_z) \in \mathbb{N}_0^3 \setminus (0,0,0)$, is the eigenvalue of the wave equation for the i^{th} room mode (wave vector), and $\mathbf{r}$ is a position inside the room.

As can be seen in Figures 1.3 and 1.4, these wave vectors are just corners of a parallelepiped, $\mathbf{k} = [\pm k_x, \pm k_y, \pm k_z]^T$. We can also notice the periodicity of the wave vector grid: $\frac{\pi}{L_x}$, $\frac{\pi}{L_y}$, $\frac{\pi}{L_z}$, along each of the axes.

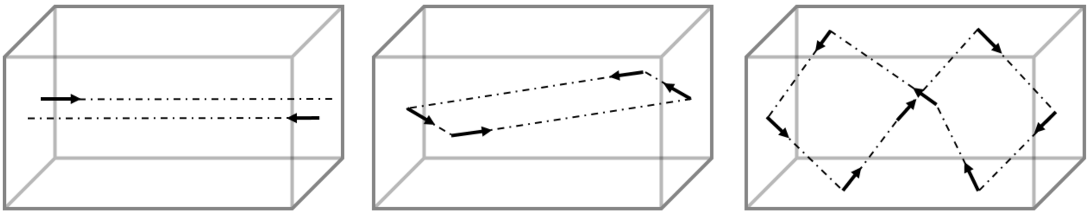


Figure 1.3 – Plane waves inside a rectangular room. From left to right: x-axial mode, xy-tangential mode and oblique mode. Wave vector is perpendicular to the plane wave.

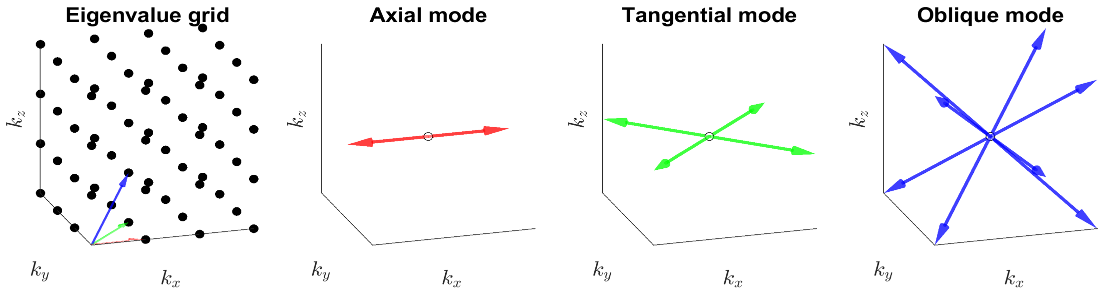


Figure 1.4 – Eigenvalue space of a rectangular room with rigid walls. The left-hand side shows just one octant because of the symmetry that exists (there are 8 plane waves for each wave number, or 4 for a tangential mode and 2 for an axial mode). The length of the wave vector is proportional to the eigenvalue of the Laplacian.

In a case of rigid walls in a rectangular room, where the damping can be neglected ($\xi \approx 0$, since $\xi \ll \omega$), we have:

$$\Xi(\boldsymbol{k}, \boldsymbol{r}) = C\cos\left(\frac{n_x\pi}{L_x}x\right)\cos\left(\frac{n_y\pi}{L_y}y\right)\cos\left(\frac{n_z\pi}{L_z}z\right) = C\cos(k_x x)\cos(k_y y)\cos(k_z z), \tag{1.7}$$

where C is an arbitrary constant and $\boldsymbol{k} = [k_x\ k_y\ k_z]^T$ are the coordinates of the wave vectors.

1.2.3 Plane wave decomposition of sound pressure in a room

In order to introduce schemes for more efficient sampling of the sound field, we will be observing structured representations of sound field in a room. We will observe the decomposition of sound pressure into N room modes with W plane waves per room mode (where $W = 8$ in a rectangular room). A more general discussion of plane waves will be included, covering cases with damping, $\boldsymbol{\xi}[n] < 0$. Therefore, we will be holding our wave numbers in a vector $\boldsymbol{\kappa}[n] = \frac{\boldsymbol{\omega}[n]+j\boldsymbol{\xi}[n]}{c}$, $\boldsymbol{\kappa} \in \mathbb{C}^N$ and the corresponding wave vectors in a matrix $\mathbf{K} \in \mathbb{R}^{N\times W\times 3}$.

For a given position of the microphone $\boldsymbol{r}_{\text{mic}}$ and position of the sound source $\boldsymbol{r}_{\text{ss}}$, we can define the behavior of the room in the frequency domain with the Room Transfer Function (RTF) [91]:

$$H(\omega, \boldsymbol{r}_{\text{mic}}, \boldsymbol{r}_{\text{ss}}) = \rho c^2 \omega Q \sum_n \frac{\Xi(\mathbf{K}[n,1,:], \mathbf{r}_{\text{mic}})\Xi(\mathbf{K}[n,1,:], \mathbf{r}_{\text{ss}})}{\boldsymbol{g}[n]\Big(2\xi[n]\tilde{\boldsymbol{\omega}} + j(\omega^2 - \tilde{\boldsymbol{\omega}}[n]^2)\Big)} \tag{1.8}$$

where ρ is the density of the propagating medium (air), c is the sound celerity, Q is the volume flow velocity of the sound source, $\Xi_n(\cdot)$ are the eigenfunctions, $\boldsymbol{g}$ is the gain, ξ_n is the damping coefficient and $\tilde{\boldsymbol{\omega}}$ are the resonant frequencies.

We can notice an interesting underlying symmetry that exists in this equation: position of the microphone $\mathbf{r}_{\text{mic}}$ and position of the sound source $\mathbf{r}_{\text{ss}}$ are interchangeable (acoustic reciprocity), meaning that if we exchange these positions, the expression will remain the same [91]. In the literature this is know as the *reciprocity principle.* This is sometimes used to decrease the computational costs of some acoustic renderings [149]. For example, inside games usually the player's head stays always at the same height or takes just a few values - when the player is standing or crawling, so player's head becomes a source. With reciprocity we transform the problem from a multiple-source single-listener into multiple-listener single-sources case which introduces savings due to the limited freedom of listener's position along z-axis.

Solution of the wave equation can be approximated in the low-frequency domain as a discrete sum of damped complex harmonics [113]:

$$p(t, \boldsymbol{r}) = \sum_{n\in\mathbb{W}} A_n \Xi_n(\boldsymbol{r}) g_n(t) = \sum_n \sum_w \mathbf{A}[n, w] e^{j(\boldsymbol{\kappa}[n]\mathrm{c}t + \mathbf{K}[n,w,:]\cdot\boldsymbol{r})} \tag{1.9}$$

where $\mathbb{W} \subset \mathbb{Z}$, $\mathbf{A}$ contains expansion coefficients, $\Xi_n(\boldsymbol{r})$ represents the spatial dependency of mode shape (illustrated in Figure 3.3) and $g_n(t) = e^{j\boldsymbol{\kappa}[n]ct}$ is corresponding time evolution of the mode [91], due to the damping from the air over the course of propagation. Temporal functions are orthogonal. This expression emphasizes the separability of the analysis and the estimation of the temporal and spatial parameters, which can greatly reduce the computational complexity of the parameter retrieval [112].

Although the modal behavior is obvious for the case of rectangular rooms, the plane wave and room mode decompositions holds also for other convex room shapes. This approximation holds sufficiently far from the walls, where evanescent waves can be neglected and gives good result within the region where the measurements were performed [112, 11].

1.2.4 Image-source model

When observing the behavior of a room in temporal domain, we usually employ the image-source model [10, 21]. The concept is illustrated in Figure 1.5: once a sound is emitted by the real source (black) next to a wall, it first reaches the microphone directly, and slightly later microphone receives the reflection from the wall. The time that it takes the sound to travel on a source$_{\text{real}}$-wall-microphone path is the same as on the source$_{\text{image}}$-microphone, that is the same as if the sound was emitted by an image source (grey) that we get by mirroring the real source against the wall. We need to pay attention to the attenuations that happen to the sound due to the propagation and due to the losses at the wall.

If we keep mirroring the true and all the image sources, we will get the image-source model illustrated in Figure 1.6. Therefore, if the sound source emits a Dirac pulse inside a room, at the microphone position we will have:

$$p(t,\boldsymbol{r}) = \sum_{i=0}^{S} c_i \frac{\delta\left(t - \|\mathbf{s}_i - \mathbf{r}\|\right)}{4\pi \|\mathbf{s}_i - \mathbf{r}\|}, \tag{1.10}$$

where $\boldsymbol{r}$ is the position of the microphone, $\boldsymbol{s}_i$ are positions of real and image-sources and S is the number of image-sources. So we will be receiving a train of attenuated Dirac pulses. c_i model the losses from walls and the denominator models the losses related to propagation.

As can be seen, the higher the order of an image-source, the higher the number of such sources. For example, in a 2D case we have four 1$^{\text{st}}$ order image-sources (corresponding to first order reflections), eight 2$^{\text{nd}}$ order image-sources etc. In the early part of the impulse response these reflections are sparse, but they get denser and as soon as the impulse response reaches the *mixing time* [103], starting from where reflections can not be distinguished any more (since the room transits to diffuse behavior), as can be seen in Figure 1.16. When the sound is rendered for a virtual environment, the part of impulse response after the mixing time t_{mix} is usually modeled as Gaussian noise [149].

1.3 Acoustic modeling

As was previously introduced, the Room Impulse Response (RIR) is a response of a room to a Dirac pulse emitted at a sound source position $\boldsymbol{r}_{\text{ss}}$ and received at a microphone position $\boldsymbol{r}_{\text{m}}$ inside a given room. RIR describes the behavior of the room over the temporal axis. For characterization of the room in the frequency domain we use Room Transfer Function (RTF) that is just a Fourier transform of RIR. When a certain room exists, we can bring our equipment and measure its acoustic behavior. On the other hand, for room that do not exist or might be

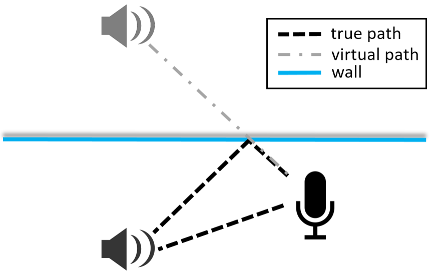


Figure 1.5 – Image-source model: true and virtual source.

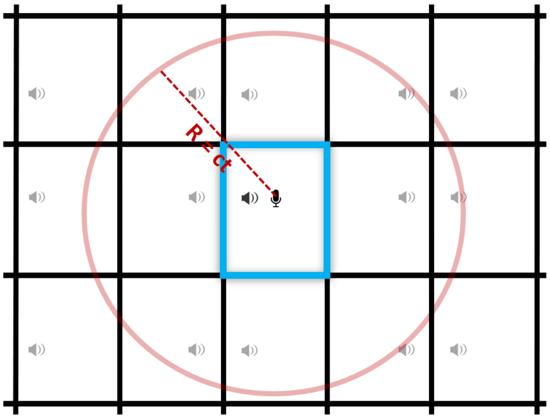


Figure 1.6 – Image-source model: repetitions.

built in the future, we need to prepare simulations that given perceptually relevant models of acoustical behavior of such spaces.

Back in 1954, Schroeder referred to the frequency at which a room goes from being a resonator to being a reflector/diffusor as the *crossover frequency*. We now call it the Schroeder frequency:

$$f_{\text{sch}} = 2000\sqrt{\frac{t_{60}}{V}} \tag{1.11}$$

where: t_{60} - reverberation time (time until the signal drops by 60dB) and V - room volume. Acoustics of a space is modelled in two different ways: below the Schroeder frequency (around 200Hz) we usually use Finite Element Method (FEM) [93] or Finite Difference Time Domain (FDTD) method [66, 81], and for the medium and high frequency range, acoustics is usually modeled with image-source [10, 21] method or ray-tracing [90]. FDTD is just a discretization scheme for the wave equation. The most common discretization scheme is leap-frog scheme [81], which for a $2D$ case reads:

$$\begin{aligned}\frac{\partial^2 p(t,r(i,j))}{\partial x^2} + \frac{\partial^2 p(t,r(i,j))}{\partial y^2} - \frac{1}{c^2}\frac{\partial^2 p(t,r(i,j))}{\partial t^2} = \frac{p(t,r(i-1,j)) - 2p(t,r(i,j)) + p(t,r(i+1,j))}{\Delta_x^2} + \\ + \frac{p(t,r(i,j-1)) - 2p(t,r(i,j)) + p(t,r(i,j+1))}{\Delta_y^2} - \frac{1}{c^2}\frac{p(t-1,r(i,j)) - 2p(t,r(i,j)) + p(t+1,r(i,j))}{\Delta_t^2}\end{aligned} \tag{1.12}$$

where $r(i,j) = [i\Delta_x, j\Delta_y]^T$ therefore, the sound pressure at temporal-spatial point $(t, r(i,j))$ depends only on the sound pressure values of the direct neighbors in time and in space. In a usual setting, RTFs are used for low frequency modeling and RIRs are used for medium and high frequency range.

On an example of a room transfer function and its room mode decomposition, we see that be-

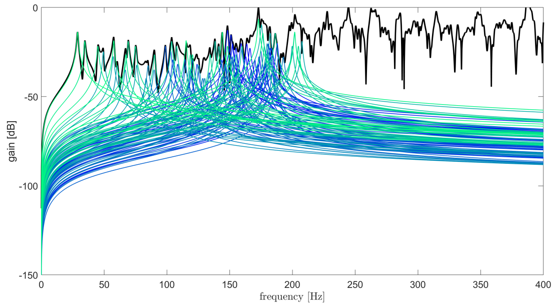


Figure 1.7 – Room modes below Schroeder frequency.

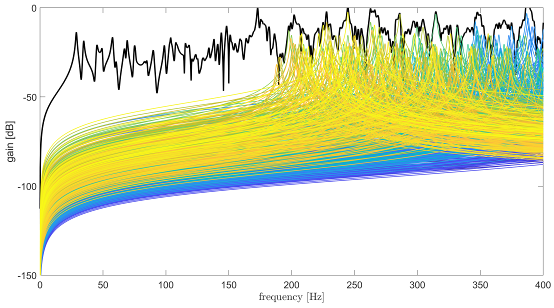


Figure 1.8 – Room modes above Schroeder frequency.

low the Schroeder frequency (Figure 1.7) the room modes are sparse and above the Schroeder frequency (Figure 1.8), room modes become very dense. This all follows from (eq. 1.9).

1.3.1 Acoustic simulators

There are different types of acoustic simulators. For accurate acoustic simulations in the low-frequency domain we rely on computational acoustics and in the mid-high-frequency domain on geometrical acoustics. It is not possible to say what kind of acoustic simulator provides the best performance and what kind of errors that simulators introduce when relying on assumptions about behavior of sound would not prevent some algorithm's deployment in the real world after the evaluation on the data produced by simulators. Figure 1.9. shows an example of sound propagation inside a fortress scene in a simulated environment. The main advantage of the simulator is high level of control of the environment where data is acquired, but it also comes with a drawback - simulations can be cumbersome and could potentially take long time to execute and finally, they usually give bandlimited results. There are two types of acoustic simulators that were used for the creation of this thesis:

1. pyroomacoustics [163], and
2. Triton [151].

pyroomacoustics (shown in Figure 1.10) is a simulator that relies on an image-source model [10, 21] when generating room impulse responses and is mainly limited to rooms where the walls are orthogonal to the floor. This simulator is open-source and the new versions of the software might include ray-tracing [90]. Blue points correspond to the position of image-sources that we get in a recursive manner: by reflecting the real and the image sources against the walls.

On the other hand, *Triton* [2] (shown in Figure 1.11.) is an acoustic simulator developed by

---

[2] https://www.microsoft.com/en-us/research/project/project-triton/

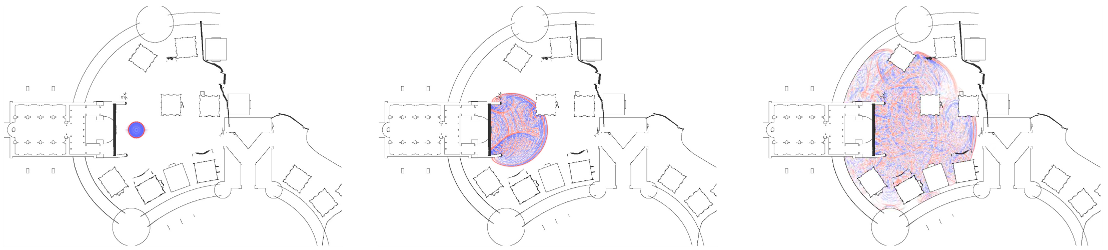


Figure 1.9 – Sound propagation at t_1, t_2 and t_3 ($t_3 > t_2 > t_1$) in a simulated environment (*Triton* simulator owned by Microsoft. This is an example of a simulation in *Citadel* scene).

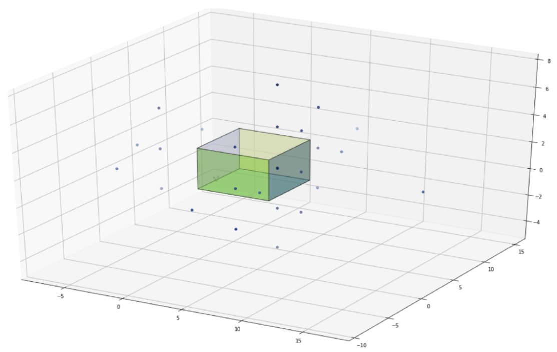


Figure 1.10 – Acoustic simulation: pyroomacoustics.

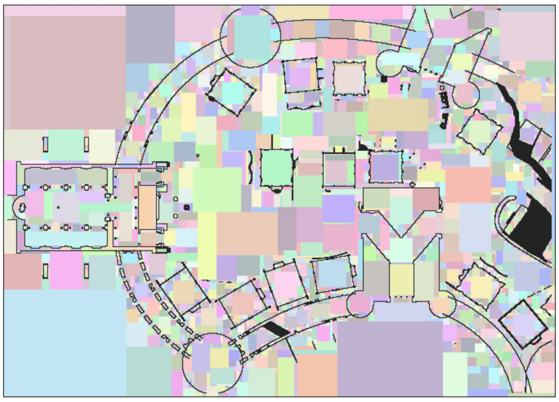


Figure 1.11 – Acoustic simulation: Triton.

Microsoft [146]. The domain of interest is first decomposed into rectangular domains (these domains are visualized with different colors). In each one of them the FDTD [66] technique was applied and sound propagation is modeled with wave propagation in rectangular domains. The latter simulator covers complex acoustic behavior that are a result of physics of the environment, such as: *obstruction* (sound is weakened when it diffracts around obstruction), *portaling* (doors funnel sounds and it should not be heard through the walls), *occlusion* (total reduction in loudness from geometry, including complex propagation and diffraction), *reverberance* (high reverberance is usually correlated with low clarity) and large rooms reverberate longer (therefore have a longer *decay time*). This type of simulator allows easy simulation within a space of an arbitrary shape, as long as we have a graphic model available for it.

Both simulators still need to be further developed to fully cover the potentially frequency dependent behavior of surfaces in different scenes. Once these optimizations are achieved, we can have an idea of the acoustic sound rendering in non-existing rooms (rooms that have been designed, but not built yet).

If for any reason someone finds it difficult to imagine sound propagation, a brilliant Schlieren experiment was performed to film how the density of air changes with sound propagation [35]. Since the sound travels at high speed, cameras that can capture a couple of thousands of

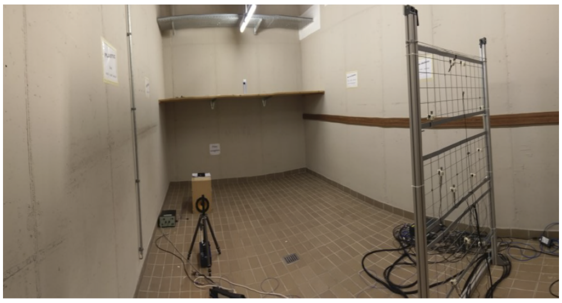


Figure 1.12 – Acoustic measurements: The measurement setting.

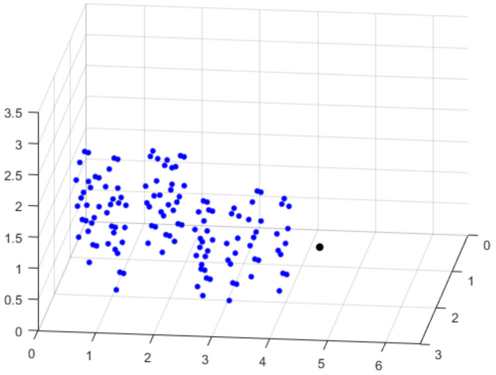


Figure 1.13 – Acoustic measurements: The microphone positions.

frames per second are needed for such a system to work.

1.3.2 Acoustic measurements

In order to evaluate the performance of our algorithms within the scope of this thesis on real data, we have recorded room impulse responses in a simple rectangular room of size 3.0m × 6.6m × 3.5m at École Polytechnique Fédérale de Lausanne, Switzerland. This is a simple rectangular room with concrete walls and wooden floor. Impulse responses were measured at 132 random locations and all the recorded data, together with the source and receiver locations is released on: Zenodo platform. This release contains detailed information about the data acquisition and data preprocessing. Figure 1.12 shows the measurement setting and Figure 1.13 shows the placement of the microphones (blue) and the sound source (black).

The room transfer functions are usually measured with freqsweeps [116] or white noise, in order to ensure the full coverage of the frequency range. We can conceptually illustrate this trade-off using extreme cases of two Fourier transform pairs involving Dirac delta functions: $\delta(t) \xrightarrow{\mathscr{F}} 1$ and $1 \xrightarrow{\mathscr{F}} 2\pi\delta(\omega)$. Therefore, in order to compute the impulse response of the system, we can excite it with broadband white noise in frequency and use the inverse Fourier transform to get the impulse response in time.

1.4 Inverse problems in acoustics with underlying sparsity

Modeling data in acoustics involves many challenging tasks: appropriate quantization of data, appropriate sampling in time and space (over all three spatial coordinates), selection of the best subset of recorded data for processing and real-time or near-real-time acquisition and processing. In order to build efficient algorithms with suitable approximations, we turn to sparse representations.

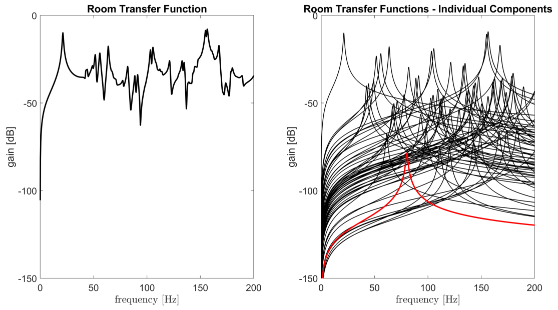


Figure 1.14 – Sparsity: Room modes.

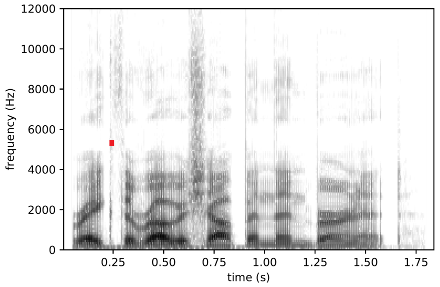


Figure 1.15 – Sparsity: Speech spectrogram.

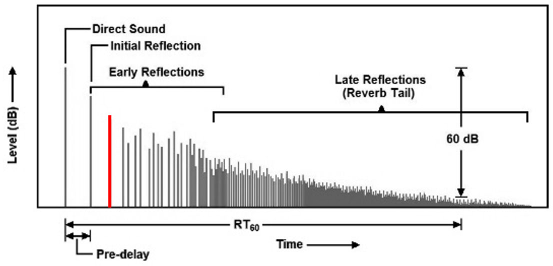


Figure 1.16 – Sparsity: RIR.

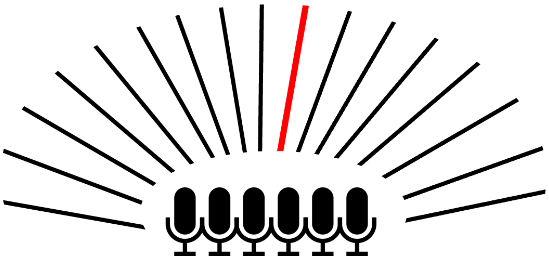


Figure 1.17 – Sparsity: Direction of Arrival.

The idea behind this thesis is to decrease the number of measurements required for different acoustical tasks and also to improve the computational complexity to reach a good speed-accuracy trade-off. Acoustical data is high-dimensional - data is measured in time at a certain position in 3D space at frequencies that are usually equal to a couple of tens of thousands of Hz.

In order to be able to apply the tools for sparse signal processing, optimization and machine learning with sparse priors, we need to identify the sources of sparsity in acoustics. Main types of sparsity (colored in red in the following Figures) in acoustics are:

1. room transfer function (RTF) has a sparse representation in the low-frequency domain (below Schroeder frequency, around 200Hz) in the terms of room modes (Figure 1.14),
2. speech spectrogram[3] has a sparse representation in the terms of the non-zero components in the time-frequency plane (this is usually used for speech enhancement and dereverberation [86]) (Figure 1.15) - spectral sparsity,
3. room impulse response (RIR) is sparse in the part with early reflections (Figure 1.16) - temporal sparsity, and
4. in the sound source locatization problem, we have the sparsity in the terms of spatial Fourier transform - usually only one or a few angles/pixels/voxels are occupied by sound sources (Figure 1.17) - spatial sparsity.

---

[3] https://github.com/drammock/spectrogram-tutorial

Spectrograms are time-frequency representations that show the evolution of the frequency components over time (having the time as the x-axis and frequency as the y-axis). It is usually computed over a sliding window with overlaps, over which Short Time Fourier Transform (STFT) is applied.

There is another interpretation of sparsity in room acoustics, given in a recent paper by Antonello et al. [11]. While working on the RIR interpolation, he uses the following sources of sparsity as the regularizers to his optimization problem:

1. spatial sparsity - because there are just a few sound sources,
2. spatio-spectral sparsity - for the case when the spatial sparsity is coupled with the Plane Wave Decomposition Method (PWDM), and
3. spatio-temporal sparsity - for the case when the spatial sparsity is coupled with the Time Equivalent Source Model (TESM).

Note that spatio-temporal sparsity is more applicable for a broadband case and the spatio-spectral sparsity holds only in the low-frequency domain, as was previously explained through the type of the acoustic simulators.

In this thesis we will tackle problems with different origins of sparsity.

The third part of the thesis will be devoted to exploiting the *spatio-spectral* sparsity inside of a room for the tasks of sound source localization, room mode detection and room shape estimation. Here we use the classical approaches of the compressed sensing with encoding information on a grid.

The forth part of the thesis will be devoted to exploiting the *spatio-temporal* sparsity inside the room and beyond for the tasks of the early reflection estimation and detection of the type of the space through the echo density trend estimation. In this part we avoid the grid and we process the information in the continuous domain by building appropriate sparse parametric models.

The fifth part of the thesis uses parametric learning for the problems of audio classification. Here we have a case of *sparse parametric modeling.* A small number of parametric filters is sufficient for classification of sounds of different types, as will be explained in detail later.

2 Mathematical Background

2.1 Inverse problems with underlying sparsity

An *inverse* problem is a type of a mathematical problem where we start with the observations and we want to estimate model parameters that produced them. The dual problems are *direct* problems. For clarification, we give an example: a direct problem is an estimation of the room impulse response starting from the known room and conditions within and the inverse problem would be the estimation of the room shape and properties starting from the room impulse response.

When solving an inverse problem, we need to understand what is recoverable and what is forever lost in the forward problem. Sometimes the solution can be found up to a certain set of ambiguities, for example translation, rotation and scaling.

A problem can be solved only when it is *well-posed* and *well-conditioned*. We say that a problem is *well-posed* when we know how many degrees of freedom there are (how many parameters are supposed to be identified) and in the term of what parameters is our problem defined, and a problem is *well-conditioned* if small errors in the initial data impose only small errors in the solutions.

By introducing a sparse regularization into our problems, not only do we ensure the well-posedness of our problems, but we also reduce the number of required measurements and reduce the computational time (by reducing the dimensionality of the search space for the solution). Sparse priors and parsimonious processing are so powerful that they can enable sub-Niquist sampling frequencies, as we will see later.

Our parsimonious processing will be governed by the Ockham's (Occam's) razor principle (the law of parsimony) that states: "Numquam ponenda est pluralitas sine necessitate" (Plurality must never be posited without necessity, that is: "Of two equivalent theories or explanations, all other things being equal, the simpler one is to be preferred"). This can be interpreted as "the underlying model should not be more complicated than necessary to accurately represent

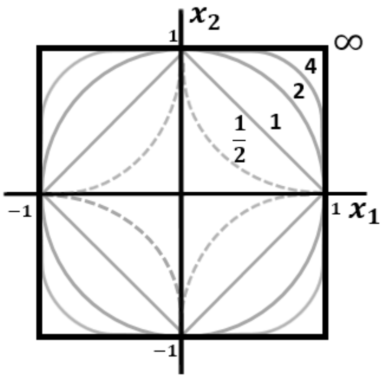


Figure 2.1 – ℓ_p norms in $\mathbb{R}^2$ [190].

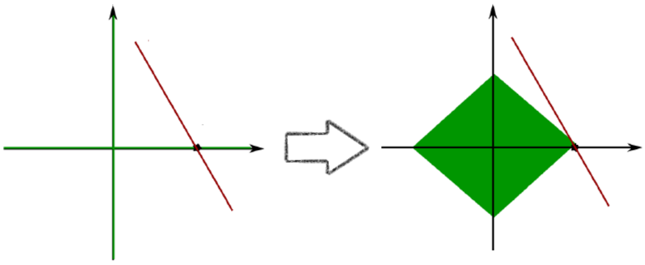


Figure 2.2 – An illustration of ℓ_0 against ℓ_1 norm.

the observations".

Within the context of this thesis we will recognize two types of sparse data retrieval: *on-grid* sparsity and *off-grid* sparsity. This type of method classification was chosen according to the nature of the search space where our sparsity is defined - on a discrete grid or on a continuous line.

2.1.1 Sparse regularizers

In order to ensure well-posedness of the problem at hand, we introduce regularizers which add information to our problem. Usually regularizers impose some constraints on the structure or the nature of data at hand. In the formulation of optimization problems we can find norms of signals of interest.

We can define ℓ_p-norms of a discrete signal $\boldsymbol{x} \in \mathbb{R}^N$ in the following way [190]:

$$\|\boldsymbol{x}\|_p = \left(\sum_n |\boldsymbol{x}_n|^p\right)^{\frac{1}{p}} \tag{2.1}$$

A few examples of ℓ_p norms for a $\mathbb{R}^2$ case are shown in Figure 2.1. Every norm is a context dependent metric of the given signal. When we deal with sparse signals, we usually use the ℓ_0 norm which is the norm that counts signal's non-zero elements. Unfortunately, this norm comes with a major drawback - it is not convex.

Norm relaxation for regularization of sparse problems is used to enable the convexity of the problem [32], as has been illustrated in the Figure 2.2. The red line illustrates a linear objective function. As we can see, the solution to the problem with an ℓ_0 and ℓ_1 norm regularization are equivalent. Norms that are usually used for enforcing sparsity:

1. ℓ_0-norm $= \|\boldsymbol{x}\|_0 = |\text{supp}(\boldsymbol{x})| = |j : \boldsymbol{x}[j] \neq 0|$,

2. ℓ_1-norm: $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{N} |\boldsymbol{x}[i]| = \sum_{i=1}^{K} |\boldsymbol{c}[i]|$, where $\boldsymbol{c}[i]$ are non-zero coefficients,
3. atomic norm: $\|x\|_{\mathscr{A}} = \inf\{t > 0 \mid x \in t \operatorname{conv}(\mathscr{A})\}$ [18]; here we make an assumption that the continuous signal x os a sparse non-negative combination of points from an arbitrary, possibly infinite set $\mathscr{A} \subset \mathbb{C}^n$, so $\operatorname{conv}(\mathscr{A})$ is the convex hull of points in $\mathscr{A}$, and
4. (generalized) total variation norm (ℓ_1 norm of the gradient): $\|\boldsymbol{x}\|_{\mathrm{TV}} = \|\nabla \boldsymbol{x}\|_1$; although it induces piecewise constant solutions, it can also be used as a sparse regularizer.

2.1.2 Sparse problems' framework

Although many people relate the beginning of sparse processing to the methods emerging in 21^{st} century with the Finite Innovation Theory from 2002 [191] and compressed sensing from 2006 [27, 52], the notion of sparse signal processing is a lot older. For example, in a paper from 1995 [196] about blind channel identification, we can read:

"The number of modes p, which is often referred to as the linear complexity, is a measurement of diversity of finite sequence."

There are various classes of computational techniques for solving sparse approximation problems [183]. All the algorithms available for handling sparse data can be classified in one of these groups:

1. **Convex relaxation** [180, 24] (the Basis Pursuit [182]): This type of approach uses ℓ_1 instead of ℓ_0 norm as the structure-inducing (sparsity promoting) function, replacing combinatorial problem with a convex optimization problem which enables the usage of the gradient or the interior point methods. The equivalency of these approaches is guaranteed by the Restricted Isometry Property [20] (Figure 2.2.). ℓ_0-norm minimization problem is equal to ℓ_1-norm minimization problem as long as the signals are *sufficiently sparse* and the sensing matrices have *sufficiently incoherent* columns [53, 57]. This type of relaxation comes with a cost of requiring higher number of measurements [29]. Convex relaxation techniques also include matrix-lifting [33] and semi-definite programming [194]. On the other hand, convex relaxation is cumbersome in higher dimensions and therefore is not used in practice for such cases.
2. **Greedy algorithms** [182, 119, 124]: These types of algorithms iteratively refine a sparse solution by successively identifying one or more components that yield the greatest improvement in quality of the approximation. These methods are time consuming, since they involve iterative projection and computation of the residual of the signal in every iteration.
3. **Parametric models**: In this group of algorithms, one of the most famous methods is Finite Rate of Innovation (FRI) [191] (with extended version for noisy observations [19]

incorporating Cadzow denoising [25]). These methods provide an off-grid approach for sparse problems where the sparsity level is known upfront. Although the initial requirements imposed having the access to uniform measurements, newer approaches relaxed the sampling constraint by enabling random at uniform sampling schemes [125]. They are based on annihilating filter and rooting of polynomials with unit norm zeros - a method from spectral analysis [175] and error-correction coding. These methods involve *matrix enhancement* [70] or *partition-and-stack* technique - stacking windowed version of the original vector into a Toeplitz or Hankel matrix.

There also exist other approaches, but they will not be explored in detail within the context of this thesis. These approaches include: **Bayesian framework** - It assumes a prior distribution for the unknown coefficients that favor sparsity; **Nonconvex optimization** - This approach relaxes the ℓ_0 problem to a related non-convex problem and attempts to identify a stationary point; **Brute force** - Rarely used in practice, this approach searches through all possible support sets, possibly using cutting-plane methods to reduce the number of possibilities.

2.2 On-grid modeling of sparsity

The most famous method for processing sparsity on a grid is *compressed sensing*. The main idea of compressed sensing is capturing the information that would survive compression, instead of capturing all the information and following up with a compression.

2.2.1 Synthesis vs analysis approach

Here we will define the two basic terms that will be used through this section: *sparsity* and *cosparsity*. Although the definitions will be established for matrices with entries in $\mathbb{R}$, the extension for a case of entries in $\mathbb{C}$ is straightforward.

Sparsity (synthesis approach): A sparse signal $\boldsymbol{y} \in \mathbb{R}^d$ can be constructed by a linear combination of a few column vectors (atoms) taken from a large matrix called the *dictionary* $\mathbf{D} \in \mathbb{R}^{n \times d}$ denoted:

$$\boldsymbol{y} = \mathbf{D}\boldsymbol{x}. \tag{2.2}$$

Therefore, signal $\boldsymbol{x} \in \mathbb{R}^n$ contains only $k \ll n$ non-zero elements, so although its ambient dimension is n, its intrinsic dimension is k if the appropriate representation basis is chosen. The indices of non-zero elements of $\boldsymbol{x}$ are called the *support*.

Cosparsity (analysis approach): Given a matrix $\mathbf{A} \in \mathbb{R}^{p \times n}$, a signal $\boldsymbol{x} \in \mathbb{R}^n$ is l-cosparse if the product $\mathbf{A}\boldsymbol{x}$ contains only $p - l$ non-zero components. In this case the ambient dimension of $\boldsymbol{x}$ is again n, but the intrinsic dimension is typically $n - l$. Or in other words, we assume that there exists an analysis operator $\mathbf{A}$, such that the following analysis representation:

$$\boldsymbol{x} = \mathbf{A}\boldsymbol{y} \tag{2.3}$$

of the signal $\boldsymbol{y}$ is sparse. A signal whose analysis representation contains l zero elements is said to be *l-cosparse*. The index set of the zero entries in $\boldsymbol{x}$, corresponding to the rows of $\mathbf{A}$ that are orthogonal to $\boldsymbol{y}$, is called its *cosupport*.

These models are equivalent only for a special case where: $\mathbf{A} = \mathbf{D}^{-1}$ that is - only if the analysis matrix and the dictionary are non-singular matrices (and therefore invertible). Both, the synthesis and the analysis operators, can be learned or carefully chosen according to the application and the nature of the data at hand.

2.2.2 (On-grid) Compressed (Compressive) sensing

Beginning from 2006, a group of scientists (Donoho, Candes, Romberg and Tao) have established a new direction in signal processing called *compressed sensing* [52, 27, 28, 14] that relies on sparse data. A representation of a certain phenomenon is recognized as being *sparse* if it can be faithfully represented as a linear combination of a small number of elements of certain

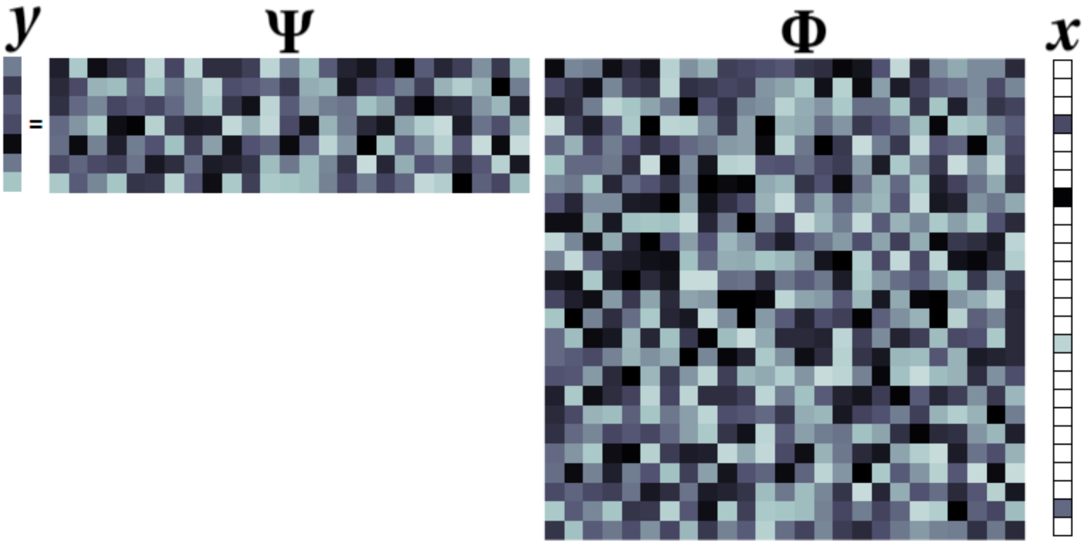


Figure 2.3 – Compressed sensing $\boldsymbol{y} = \mathbf{\Psi}\mathbf{\Phi}\boldsymbol{x}$.

representation basis or frame. Before compressed sensing has emerged, data was captured and then compressed before being stored. The key paradigm of compressed sensing is to capture only the data that would survive the compression step. Its main strengths are that it speeds up the acquisition process, but is also robust to noise. The data that was cumbersome to capture before now can be easily handled with the compressed sensing approach. For example, with the application of compressed sensing to magnetic resonance imaging (MRI) [75] has drastically reduced the amount of time patients have to spend inside a scanner.

The compressed sensing is illustrated in Figure 2.3. The sensing matrix (matrix that maps the signal to the observations) $\mathbf{D} = \mathbf{\Psi}\mathbf{\Phi}$ is the product of the matrix $\mathbf{\Phi}$, which *transforms* the signal from one domain to another (e.g., the inverse discrete Fourier transform (IDFT)), and the matrix $\mathbf{\Psi}$, which represents the *measurement* process (e.g., time sampling). The sampling matrix $\mathbf{\Psi}$ can have a random or deterministic form. In the case of deterministic form, it is equal to an identity matrix $\mathbf{I}$ subsampled over rows. This matrix is usually called a *mask*. While the transform matrices $\mathbf{\Phi}$ are determined by the *class* of signals, sensing matrices $\mathbf{D}$ are determined by the specific *application* [26].

The representation $\boldsymbol{x}$ of signals $\boldsymbol{y}$ in a given dictionary $\mathbf{D}$ is usually not exactly sparse, but has a fast decay of the ordered absolute value of the expansion coefficients. In this case we usually say that the signal is *compressible* rather than *sparse*.

Although many theoretical results emerged in the domain, the application side was mostly focusing on image processing and the state of the art on compressed sensing in acoustics at the start of this thesis was coarse. At first compressed sensing was based on random matrices. Later on, the better exploration of the phenomenon that we model or want to capture, together with the improvement of the sampling strategies, have lead to more advanced techniques for measurement matrix design. Recognizing the structure of the sparsity in the given problem (e.g. block, tree, graph) or introducing some assumptions can further improve the efficiency and therefore the robustness of the algorithm, because additional constraints usually reduce the search space. When defining a data model for the phenomenon we want to capture, we define a set of mathematical properties that the data is believed to satisfy. Therefore, we can tailor the sensing matrices according to the application to get the most of it. A thorough list of the solutions for compressed sensing in acoustics is given in [61, 16].

The atoms (columns) of dictionaries are usually unit norm (for dictionary **D** each column: $\|\boldsymbol{d}_j\|_2 = 1$), but they are not orthogonal, since $d \neq n$. So the sparse signals are just a linear combination of small number of elements of such a dictionary that is tailored or learned according to the nature of the data or the application. Due to the lack of orthogonality, we need to project the residual onto the space spanned by the columns of the dictionary through an iterative process instead of computing a fast projection with the information available in the signal.

Despite the fact that we have to deal with an underdetermined system of equations $\boldsymbol{y} = \mathbf{D}\boldsymbol{x}$, since we know that the underlying signal is sparse, we can recover almost as many coefficients as there are equations available. The guarantee that we will be able to reconstruct the signal is given by the RIP (Restricted Isometry Property) [20]:

$$(1-\delta)\|\mathbf{x}\|_2^2 \leq \|\mathbf{D}\boldsymbol{x}\|_2^2 \leq (1+\delta)\|\mathbf{x}\|_2^2 \tag{2.4}$$

where **D** is the measurement (sensing) matrix and for some small δ. **x** is a vector whose sparsity level is K ($\|\mathbf{x}\|_0 \leq K$). This matrix has to preserve angles and distances (lengths) when moving from one space to another. Since it is a fat rectangular matrix, it has a non-trivial nullspace.

It must provide an injective mapping for K-sparse vectors: if $\mathbf{x_1} \neq \mathbf{x_2}$ then $\mathbf{Dx_1} \neq \mathbf{Dx_2}$. Therefore:

$$\text{spark}(\mathbf{D}) > 2K. \tag{2.5}$$

Spark is the smallest number of linearly dependent columns of a matrix, where K is the number of non-zero elements in the vectors $\mathbf{x_1}$ and $\mathbf{x_2}$ (they are K-sparse). To achieve an injective mapping we need to make sure that there are no two K-sparse vectors that map to the same measurements. This implies that the rank of our sensing matrix has to be at least $2K$ which is tightly related to the restriction on the coherence of the dictionary.

Our dictionary must also have an appropriate *coherence statistic* (to avoid ill-posedness) - $\mu = \max_{i \neq j}\langle \boldsymbol{d}_i, \boldsymbol{d}_j \rangle = \max_{i \neq j}(\boldsymbol{d}_i^T \boldsymbol{d}_j / (\|\boldsymbol{d}_i\|_2 \|\boldsymbol{d}_j\|_2))$. The coherence is just the cosine of the acute angle between the closest pair of atoms in a given dictionary. The idea is to avoid having atoms that are almost "parallel", that is - that have *high* correlation. It can also be defined as normalized absolute inner product between any two columns of the sensing matrix. Usually:

$$\mu(\boldsymbol{d}_i, \boldsymbol{d}_j) < \frac{1}{2K-1}. \tag{2.6}$$

In the early stages of the development of compressed sensing, it has mostly relied on random dictionaries, since they have proven to have a low coherence.

Therefore, we want our sensing matrix to have a *large* spark and *low* coherence.

In acoustics our measurements for T time moments ($t \in [1, T]$), m microphones and n sound sources are of the form:

$$\mathbf{p} = \mathbf{\Phi a} + \varepsilon \tag{2.7}$$

where:
$\mathbf{p}$ - measurements from the microphones, $\mathbf{p} \in \mathbb{R}^{mT}$,
$\mathbf{\Phi}$ - measurement matrix, $\Phi \in \mathbb{R}^{mT \times nT}$ - usually represents the Green's function of the room, inverse or direct spatial or temporal Fourier transform of sound captured by microphones,
$\mathbf{a}$ - expansion coefficients, $\mathbf{a} \in \mathbb{R}^{nT}$, and
ε - noise (usually white Gaussian noise).

The usual formulation of the sparse problem is in the form of constrained LASSO (Least Absolute Shrinkage and Selection Operator):

$$\mathbf{a}_{\text{estim}} = \arg\min_{a} \|\mathbf{a}\|_0 \tag{2.8}$$

$$\text{subject to } \|\mathbf{p} - \mathbf{\Psi\Phi a}\|_2 \leq \varepsilon, \tag{2.9}$$

where $\mathbf{\Psi}$ is the subsampling matrix and $\mathbf{\Phi}$ is usually associated with the Green's function [91]. The objective function is usually relaxed to: $\arg\min_{a} \|\mathbf{a}\|_1$, as was previously discussed.

As will be shown, in the domain of acoustics the high dimensional data can not be fitted into such formulation with the relaxation of ℓ_0 norm, because it becomes extremely computationally demanding, so usually the greedy methods are used, such as: Basis Pursuit (BP), Orthogonal Matching Pursuit (OMP) [182], Compressive Sampling Matching Pursuit (CoSaMP) [119] and Orthogonal Matching Pursuit with Replacement (OMPR) [74]. For example OMP selects the top K atoms that are correlated with the residual the most, and the OMPR finds $2K$ vectors that are aligned with the residual the most and then selects K ones that give the smallest residual in the orthogonal space (the nullspace of our selected-atom-dictionary matrix).

2.2.3 Cosparsity

A complementary approach to sparsity is cosparsity [117]. In this approach there is no need to compute the measurement matrix (related to the Green's function of the room's behavior) which can be cumbersome in the non-trivial room shape case and can be of a high dimension, which leads to high computational cost. Usually the entries of the analysis operator **A** for this case contains the Finite Difference Time Domain entries with special entries that model the absorption of the boundaries. Within the acoustic application, the complexity of analysis formulation of the algorithms for 2D and 3D case is usually lower than for the synthesis approach, as shown in [83, 118].

In acoustics the cosparse formulation would be of the following form:

$$\mathbf{p}_{\text{estim}} = \arg\min_{p} \|\mathbf{\Omega}\mathbf{p}\|_0 \tag{2.10}$$

$$\text{subject to } \|\mathbf{p}_{\text{mes}} - \mathbf{\Psi}\mathbf{p}\|_2 \leq \varepsilon \tag{2.11}$$

where $\mathbf{\Omega}$ is the analysis operator the usually represents the Finite Difference Time Domain (FDTD) discretization scheme (potentially with initial and boundary conditions inside) and $\mathbf{p}_{\text{estim}}$ is the estimated sound pressure. $\mathbf{\Omega}$ is very sparse and has 7 or less non-zero elements per row. The objective function is usually relaxed to: $\mathbf{p}_{\text{estim}} = \arg\min_{p} \|\mathbf{\Omega}\mathbf{p}\|_1$, as was previously discussed. This type of formulation focuses directly on the estimation of the pressure rather than on the estimation of the expansion coefficients. Here the estimation of the Green's function is avoided. This approach is more intuitive, since it relies on the well known laws of physics [118] that govern the phenomenon that we model. The cosparse model focuses on the zeros rather that on the non-zeros in the sparse signal's representation.

An application of cosparsity to acoustics was investigated by Kitić et al. [83]. They investigate the cosparsity that lives inside the Helmholtz equation, given that only at the several places across the room where the sources are located, this equation is not homogeneous. The location of the sources is described as a point on a finite grid. Together with the discretization of the d'Alembertian operator with FDTD [66] discretization through leapfrog scheme, authors develop a theory for the localization of sound sources behind an obstacle [83, 82] and also for joint estimation of source location and boundary impedance [17].

2.2.4 A comparison of sparse and cosparse model

As has been shown in [117], the analysis (cosparsity) model contains many more low-dimensional subspaces than the synthesis (sparsity) model, but the situation reverses for high-dimensional subspaces. Reducing the size of the space of potential solutions reduces the computational complexity of the algorithm. For a k-sparse n-dimensional signal we have: for the synthesis operator $\mathbf{\Phi} \in \mathbb{R}^{n \times q}$ (usually a redundant dictionary), there are $\binom{q}{k}$ potential subspaces and, on the other hand, for the analysis operator $\mathbf{\Omega} \in \mathbb{R}^{p \times n}$, there are $\binom{p}{n-l}$ potential subspaces. So the cosparsity is defined as: $c = n - l = n - \|\mathbf{\Omega x}\|_0$. In cosparse approach we find a representation in the orthogonal complement of the nullspace of the operator, $\mathbf{\Omega}^T$.

In Table 2.1 we see an overview of the properties of the sparse and cosparse approach. As has been indicated earlier, sparse approach is more favorable for lower dimensions and the cosparse approach should be used for high-dimensional subspaces.

Table 2.1 – Properties of the sparse (synthesis) and cosparse (analysis) approach.

Model	**Subspace**	**Number of subspaces**	**Subspace dimension**
synthesis	$\mathbf{V}_T := \mathrm{span}\{\boldsymbol{v}_j, j \in T\}$	$\binom{q}{k}$	k
analysis	$\mathbf{W}_\Lambda := \mathrm{span}\{\boldsymbol{\omega}_j, j \in \Lambda\}^\perp$	$\binom{p}{\ell}$	$n - \ell$

Well-defined cosparse models have a faster rate of convergence than the sparse models, because the number of possible subspaces is lower. There is space for improvement in the proposed approach by building fast algorithms for dense grids. In the case of cosparse processing, by increasing the amount of available data we can reduce the temporal cost of the reconstructions, since it introduces more constraints into the search space. This is the opposite from the synthesis approach, which implies that the cosparse approach scales better.

2.2.5 Basis mismatch - How the on-grid models became deprecated

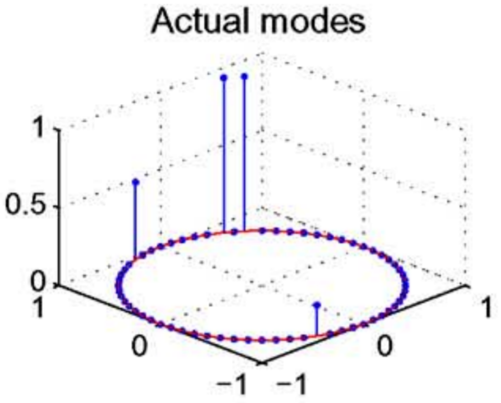


Figure 2.4 – Basis mismatch problem: Ground truth from [34].

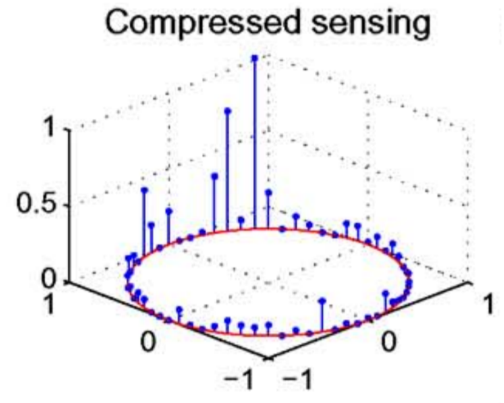


Figure 2.5 – Basis mismatch problem: Compressed sensing estimation from [34].

In a paper from 2011 Chi et al. [34] discussed the failure cases of processing on a grid in order to underline the main problems that happen in the case if a certain signal does not have a sparse representation in the terms of a certain dictionary. Even in the case of just a slight mismatch of the search grid and the ground truth grid on which the signal has a sparse representation, signals can appear to be incompressible and there would be a slow decay of their expansion coefficients in the perturbed version of the dictionary. Although the initial strength of the compressed sensing theory was robustness to noise, the basis mismatch problem was never alleviated.

As has been shown and proved in [34] and illustrated in Figures 2.4 and 2.5 on an example of a mode recovery through compressed sensing (the z axis is for mode's amplitude and x and y axis are for frequency and damping), the decay of the expansion coefficients in the perturbed dictionary comes for Fourier imaging with a slow trend that has a shape of a Dirichlet as shown in [34]. Here we see an example of the frequency grid mismatch which means that the true frequency grid does not coincide with the grid of the search space. Therefore, there is a discrepancy between the ground truth grid and the grid on which the data is retrieved, which causes a slow decay of the amplitudes of the retrieved data due to the spectral spillage.

Therefore, there was a need for building methods that will alleviate the constraints of on-grid processing. By removing the regularity of the captured or retrieved data, we should be able to asses the measurements for moving sources whose trajectory does not have to coincide with a grid of any granularity.

2.3 Off-grid modeling of sparsity

2.3.1 How to be less greedy and less griddy?

Although the processing on a grid has enabled fast acquisition of data and also the solutions that rely on a grid are usually redundant to noise, processing on a grid can cause basis mismatch problems that lead to spectral leakage and other types of loss or misinterpretation of data. Therefore, we need to create models that are able to alleviate these types of constraints.

The *less greedy* paradigm assumes reduction of requirements for an execution of an algorithm and data acquisition in the terms of computational resources and memory requirements. The *less griddy* paradigm assumes reduction of requirements to process data on a grid, always outputting data that can be accurate up to ϵ, where ϵ is the precision of the grid on which the data was acquired. Increasing the resolution of the equispaced grid by decreasing ϵ increases the coherence of the dictionary and becomes cumbersome to handle with compressed sensing technique. This type of approach would potentially resolve the *basis mismatch* problem [34].

On the other hand, the grid-free approach will open the door to arbitrary placement of sources as well as to the moving sources, which are important constraints for a dynamic system. So the questions that follow naturally are: If we want to avoid sampling or retrieval on a grid, what kind of constraints do we need to relax? What is the trade-off?

As can be seen in the literature, most of the off-grid methods establish their theory for the recovery of band-limited signals. They can be found under the following lines of work: spectral estimation on a line [177, 67], off-grid compressed sensing [178, 194] and sampling signals with Finite Rate of Innovation (FRI) [191, 19, 125]. FRI proposes sub-Nyquist sampling schemes for signals that have finite number of parameters that fully define them (for example: train of Diracs, splines, piece-wise polynomials etc.). All of the methods have roots in the spectral analysis by Prony from 1795 [141]. Also a thorough list of spectral methods has been made by Stoica [174]. Although these methods do not require uniform subsequent samples, they require random at uniform sampling and still are a step away from true random sampling.

We define atoms $a(f,\phi) \in \mathbb{C}^{|J|}$, where $f \in [0,1]$ is the normalized frequency and $\phi \in [0,2\pi)$ is the phase, as:

$$[a(f,\phi)]_i = e^{j(2\pi f i+\phi)}, i \in J \tag{2.12}$$

or in matrix-vector form:

$$\hat{\boldsymbol{x}} = \sum_{k=1}^{K} |\boldsymbol{c}[k]|\, a\big(\boldsymbol{f}[k], \boldsymbol{\phi}[k]\big). \tag{2.13}$$

In most formulations the phase gets absorbed into the expansion coefficient. So the recovery procedure focuses on the recovery of $2K$ coefficients - $\{(\boldsymbol{c}[k], \boldsymbol{f}[k])\}_{k=1}^{K}$ pairs.

2.3.2 Off-grid Compressed Sensing

The off-grid compressed sensing has appeared in 2012 and 2013 [178]. In this case the sparsity is defined on a continuous domain, so x is a continuous variable:

$$\|x\|_{\mathscr{A}} = \inf\{t > 0 : x \in t\,\mathrm{conv}(\mathscr{A})\} = \inf_{\boldsymbol{c}[k]\geq 0, \boldsymbol{f}[k]\in[0,1], \boldsymbol{\phi}[k]\in[0,2\pi)} \left\{ \sum_k \boldsymbol{c}[k] : x = \sum_k \boldsymbol{c}[k]\, a\big(\boldsymbol{f}[k], \boldsymbol{\phi}[k]\big) \right\} \tag{2.14}$$

where we follow the earlier established definition of the atomic norm from 2.1.1.

In the original formulation, this type of problem is defined in the following way:

$$\begin{aligned} &\text{minimize}_x \quad \|x\|_{\mathscr{A}} && (2.15)\\ &\text{subject to} \quad x(j) = x(j)^{\star}, j \in \mathbb{T} && (2.16) \end{aligned}$$

where $\mathbb{T} \subset \mathbb{J}$ is the index set of the observed entries and $\mathbb{J}$ is the index set of all the entries. In this case the optimization variable is continuous, so we need to reformulate the problem in order to enable its implementation.

Sparse recovery in the continuous domain is redefined in the dual problem formulation [24]. In order to arrive to a discrete optimization variable and countable number of constrains, the authors of [178, 194] use the theory for bounded trigonometric polynomials, unit norm polynomial rooting and Schur complement in order to finally arrive to semidefinite program formulation.

Therefore, the equivalent semidefinite program formulation of the dual problem [24] is given by:

$$\begin{aligned} &\text{minimize}_{\boldsymbol{u},\boldsymbol{x},\boldsymbol{t}} && \frac{1}{2|\mathbb{J}|}\mathrm{trace}(\mathrm{Toep}(\boldsymbol{u})) + \frac{1}{2}\boldsymbol{t} && \text{(2.17a)}\\ &\text{subject to} && \begin{bmatrix} \mathrm{Toep}(\boldsymbol{u}) & \boldsymbol{x} \\ \boldsymbol{x}^{*} & \boldsymbol{t} \end{bmatrix} \succeq 0 && \text{(2.17b)}\\ & && \boldsymbol{x}[j] = \boldsymbol{x}[j]^{\star}, j \in \mathbb{T} && \text{(2.17c)} \end{aligned}$$

where $\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{t}$ are discrete optimization variables.

This type of formulation has shown to be successful [194], although the parametric dictionary does not satisfy the Restricted Isometry Property.

In the case of reconstruction of a sparse spectrum with sparse components on $\boldsymbol{f}$-axis, the reconstruction bounds are given with: $\Delta_f = \min_{k \neq j} \left| \boldsymbol{f}[k] - \boldsymbol{f}[j] \right| \geq \frac{1}{\lfloor (n-1)/4 \rfloor}$, where this is a circular distance in a periodic sequence (a wrap-around distance on a circle) and n is the number of samples.

This technique was observed in order to connect all the pieces of the state of the art and will not be further investigated within the scope of this thesis.

2.3.3 Finite Rate of Innovation

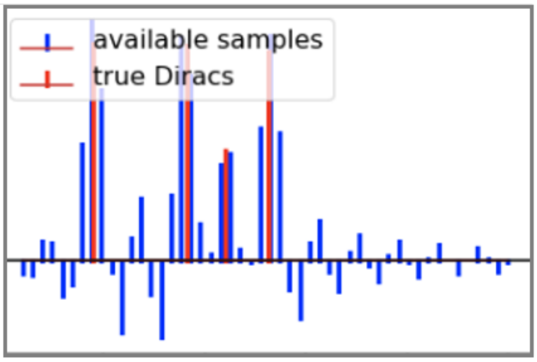


Figure 2.6 – FRI: Samples in time.

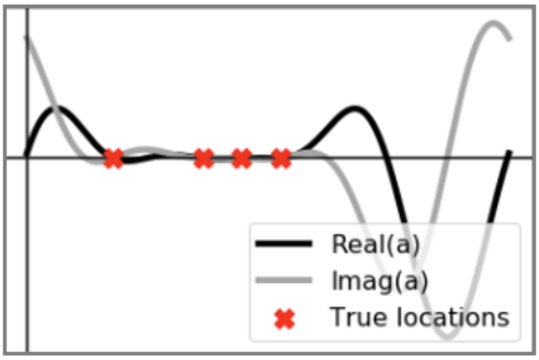


Figure 2.7 – FRI: Annihilating filter.

While defining the Finite Rate of Innovation theory (FRI) [191, 19], authors go back to the basics of signal processing [190]. Assuming that we are dealing with a bandlimited signal x (bandlimited to $[-B/2, B/2]$) in a continuous time domain, sampling of the signal encapsulates convolution with a bandlimited kernel defined as $\operatorname{sinc}(t) = \frac{\sin \pi t}{\pi t}$:

$$x(t) = \sum_{k \in \mathbb{Z}} \boldsymbol{x}[k] \operatorname{sinc}(Bt - k), \tag{2.18}$$

where $\boldsymbol{x}[k] = \langle B \operatorname{sinc}(Bt - k), x(t) \rangle = x(k/B)$. According to Nyquist sampling rate [120], we would need to sample such a signal at a sampling rate of sampling period $T = 1/B$.

In order to relax this requirements, two assumptions are introduced: we are dealing with shift-invariant signals that have a period T, and our signal consists only out of K Dirac pulses per period T which is its *rate of innovation*:

$$x(t) = \sum_{k \in \mathbb{Z}} \boldsymbol{x}[k] \delta(t - \boldsymbol{t}[k]). \tag{2.19}$$

Therefore this signal is completely defined by a set of K pairs of $(\boldsymbol{x}[k], \boldsymbol{t}[k])$ values, therefore $\{\boldsymbol{x}, \boldsymbol{t}\} \in \mathbb{R}^K$, so it can be sampled at its rate of innovation. So we can rewrite the signal for a τ-periodic case $\boldsymbol{t}[k] \in [0, \tau[$:

$$x(t) = \sum_{k=1}^{K} \sum_{k' \in \mathbb{Z}} \boldsymbol{x}[k] \delta\left(t - \boldsymbol{t}[k] - k'\tau\right). \tag{2.20}$$

Our measurements take the following form:

$$\boldsymbol{y}[n] = \langle x(t), \operatorname{sinc}(B(nT - t)) \rangle = \sum_{k=1}^{K} \boldsymbol{x}[k] \varphi\left(nT - \boldsymbol{t}[k]\right). \tag{2.21}$$

where: $\varphi(t) = \sum_{k' \in \mathbb{Z}} \operatorname{sinc}\left(B\left(t - k'\tau\right)\right) = \frac{\sin(\pi B t)}{B\tau \sin(\pi t/\tau)}$ is the sampling kernel. If we look at Figure 2.6, the red shows the ground truth Diracs and when sampled, we obtain the blue samples.

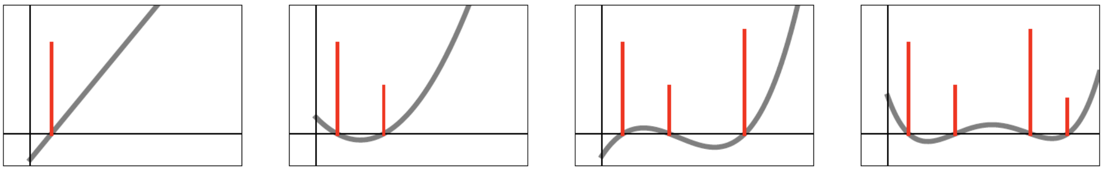


Figure 2.8 – Annihilating filters of different degrees (from left to right: first to fourth order).

A periodized stream of Diracs can be modeled through the Fourier transform as:

$$x(t) = \sum_{k=0}^{K-1} \boldsymbol{x}[k] \sum_{n\in\mathbb{Z}} \delta\left(t - \boldsymbol{t}[k] - n\tau\right) = \sum_{k=0}^{K-1} \boldsymbol{x}[k] \frac{1}{\tau} \sum_{m\in\mathbb{Z}} e^{j(2\pi m(t-\boldsymbol{t}[k])/\tau)} \tag{2.22}$$

From Poisson's summation formula [190], we have:

$$x(t) = \sum_{m\in\mathbb{Z}} \underbrace{\frac{1}{\tau}\left(\sum_{k=0}^{K-1} \boldsymbol{x}[k] e^{-j(2\pi m\boldsymbol{t}[k]/\tau)}\right)}_{\hat{\boldsymbol{x}}[m]} e^{i(2\pi mt/\tau)}, \tag{2.23}$$

so we arrive to the Fourier series representation. In the original formulation of FRI [191], the frequency grid is sampled uniformly and $m \in [-\frac{K}{2}+n,...,\frac{K}{2}+n[$, where $n \in \mathbb{Z}$ is an arbitrary constant.

Annihilating filter technique: The original requirement for establishing a uniform grid comes from the annihilating filter requirement. When using an arithmetic progression of numbers inside exponents (the indices m of the Fourier series coefficients), they transform the expression into a geometric progression of the form $e^{-i(2\pi m\boldsymbol{t}[k]/\tau)}$ that enables annihilation. The *annihilating filter* technique dates back in 1997 [175]. Although the original version of the FRI theory [191, 19] required consecutive $2K+1$ uniform frequencies for the noiseless recovery, newer extensions relax the requirement to non-uniform frequencies [125], by requiring random at uniform samples.

As has been illustrated in Figure 2.8, for any set of Diracs we can design an annihilating filter that can annihilate it, regardless of the height of the Dirac. Therefore, the annihilating filter has to be of the form:

$$A(z) = \prod_{k=0}^{K-1} \left(1 - e^{-j(2\pi\boldsymbol{t}[k]/\tau)} z^{-1}\right), \tag{2.24}$$

with the zeros at the positions of the Diracs in the exponential context: $e^{-i(2\pi\boldsymbol{t}[k]/\tau)}$. Also in Figure 2.7 we see an example of the annihilating filter for the sequence in Figure 2.6.

This all follows from:

$$\left[1, -e^{-j(2\pi\boldsymbol{t}[k]/\tau)}\right] \star \left[\dots, e^{j(2\pi\boldsymbol{t}[k]/\tau)}, 1, e^{-j(2\pi\boldsymbol{t}[k]/\tau)}, e^{-j(4\pi\boldsymbol{t}[k]/\tau)}, \dots\right] = 0 \tag{2.25}$$

The coefficients of the annihilating filter can be determined from the system of linear equations. This Yule-Walker system follows from the Vandermonde definition of all the convolutions and choosing $\hat{\boldsymbol{a}}[0] = 1$, since the set of roots of $P_{\boldsymbol{a}}$ is invariant to global scaling of the coefficients (in vector $\boldsymbol{a}$):

$$\begin{bmatrix} \hat{\boldsymbol{x}}[0] & \hat{\boldsymbol{x}}[-1] & \cdots & \hat{\boldsymbol{x}}[-K+1] \\ \hat{\boldsymbol{x}}[1] & \hat{\boldsymbol{x}}[0] & \cdots & \hat{\boldsymbol{x}}[-K] \\ & & \ddots & \\ \hat{\boldsymbol{x}}[K-1] & \hat{\boldsymbol{x}}[K-2] & \cdots & \hat{\boldsymbol{x}}[0] \end{bmatrix} \cdot \begin{bmatrix} \hat{\boldsymbol{a}}[1] \\ \hat{\boldsymbol{a}}[2] \\ \vdots \\ \hat{\boldsymbol{a}}[K] \end{bmatrix} = - \begin{bmatrix} \hat{\boldsymbol{x}}[1] \\ \hat{\boldsymbol{x}}[2] \\ \vdots \\ \hat{\boldsymbol{x}}[K] \end{bmatrix} \tag{2.26}$$

or with the least squares approach. The recovery of the coefficients $\boldsymbol{x}[k]$ is straightforward, once the $\boldsymbol{t}[k]$'s have been retrieved from this system of equation and the previously introduces parametric model in (eq. 2.24).

Reconstruction bounds: The Finite Rate of Innovation theory gives bounds on the possibility of recovery of Diracs related to the bandwidth of the available frequencies and the distance between Diracs. Also there are Cramer-Rao bounds given for the amount of noise in the observed signal [19].

This theory is applicable to the following families of signals: stream of Diracs, nonuniform splines, derivative of Diracs and piecewise polynomials. The key property of all of these families is a finite number of parameters that fully define signal over their period τ or over their length, in the case of finite length signals. In this thesis we will be focusing on the stream of Diracs case. Although we were focusing mostly on the retrieval of Diracs in time domain, we can also retrieve Diracs in spatial domain [126] as long as the sources are monochromatic.

2.4 Relationship between compressed sensing and Finite Rate of Innovation

In this thesis we will focus on the application of compressed sensing and Finite Rate of Innovation to the problems in acoustics.

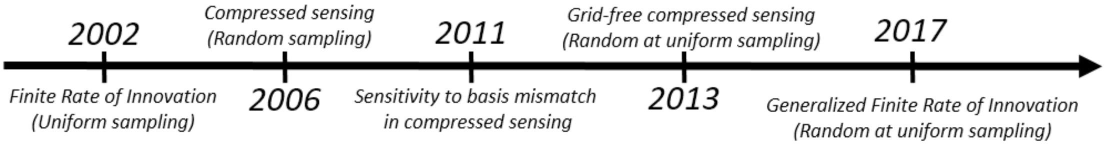

2002
Finite Rate of Innovation
(Uniform sampling)
Compressed sensing
(Random sampling)
2006
2011
Sensitivity to basis mismatch
in compressed sensing
Grid-free compressed sensing
(Random at uniform sampling)
2013
2017
Generalized Finite Rate of Innovation
(Random at uniform sampling)

Figure 2.9 – Temporal evolution of sparse methods.

To get the idea of the history of described methods, we have put their initial papers on a timeline in Figure 2.9 together with their key properties summarized in Table 2.2.

Table 2.2 – Overview of the sensing methods.

Method	**Sampling (Sensing)**	**Search space**
On-grid compressed sensing	random	grid
Off-grid compressed sensing	(random at) uniform	continuous
Uniform Finite Rate of Innovation	uniform grid	continuous
Non-uniform Finite Rate of Innovation	(random at) uniform	continuous

Authors of Finite Rate of Innovation compare their method to compressed sensing in [19], and the other authors do it the other way round in [27], finally reaching a common conclusion: compressed sensing needs slightly more samples $C \cdot N_t \log N$ versus $C \cdot N_t$ (where N_t is the level of sparsity). On the other hand, the original version of compressed sensing is limited to uniform (consecutive) samples and compressed sensing allows random sampling schemes. The compressed sensing technique also comes with an advantage of being robust to noise.

Part III

Sparse models for room acoustics: Compressed sensing and room modes

3 Localization of Sound Sources in a Room with One Microphone

In the last decade the theory of compressed sensing [52, 27] has arised in the domain of acoustic signal processing. There was always a need for finding a structure in the high dimensional acoustical data that was cumbersome to handle. In 2015 Boche et al. [20] provided a detailed state of the art for the application of compressed sensing in the domains of image and acoustic signal processing.

Estimation of the location of sound sources is usually done using microphone arrays [195, 126]. Settings with multiple microphones provide an environment where we know the difference between the received signals among different microphones in the terms of phase or attenuation, which enables localization of the sound sources. In our solution we exploit the properties of the room transfer function in order to localize a sound source inside a room with only one microphone. The shape of the room and the position of the microphone are assumed to be known. The design guidelines and limitations of the sensing matrix are given. Implementation is based on the sparsity in the terms of voxels in a room that are occupied by a source. What is especially interesting about our solution is that we provide localization of the sound sources not only in the horizontal plane, but in the terms of the 3D coordinates inside the room.

Instead of estimating the position of the sound sources from time difference of arrival between different microphones in an array [109, 110], we aim to rely only on one microphone and combine the sparsity that exists in the term of the voxels of a room occupied by the sound sources and the low-frequency room modes in the room transfer function (RTF) toward successful localization. As has been discussed earlier, the sparsity of the room modes may be exploited in the low-frequency range of the RTFs [112], as shown in Figure 1.7. By RTF we denote the relationship between the received and emitted signal inside a given room in the Fourier domain. To this end, we will analyze the transfer functions below the so called Schroeder frequency, which is defined as: $f_{\text{sch}} = 2000\sqrt{\frac{t_{60}}{V}}$, where V is the volume of the room and t_{60} is the reverberation time [91], which is usually around 200Hz for a typical room. Details around this specific frequecy have been previously given in section 1.3.

The combination of sparsity in the term of locations occupied by sound sources and the room mode sparsity in the low-frequency domain should result in a fast localization of sound sources by only one microphone as will be further explained.

We start by discussing the sparsity that exists in the low frequency domain of room transfer functions. An overview of the available techniques for coupling the compressed sensing and the localization of sound sources is given. We design the sensing matrix for such a particular setting and discuss its limitations. Since the search space is defined on a regular grid, we discuss a technique for subsampling the regular grid as the means of lowering the coherence of the sensing matrix at hand. At the end we conclude and give a some remarks on the potential future work.

3.1 Modal representation of the sound pressure and its low-frequency properties

In the further development of our approach, we are going to rely on two facts: the room shape is known and the microphone position is known. These assumptions imply that we know the resonant frequencies of the room and the room modes related to the microphone's positions.

This chapter will rely on the formulation of the acoustic behavior of a rectangular room developed in Section 1.2.2, and the decomposition of the Room Transfer Function (eq. 1.8).

In Figure 3.1 we can see a segment of RTF for an arbitrary set of positions $\mathbf{r}_{mic}$ and $\mathbf{r}_{ss}$ below Schroeder frequency and its decomposition into the room modes. This has been previously discussed and observed in Figure 1.7. The sharpness of the peaks of room modes is dependent on the damping properties of walls of the room. Peaks of the room modes are aligned with the resonant frequencies of the room.

3.1.1 Room Transfer Function at different positions across the room

Each RTF is characterized by a set of parameters: resonant frequencies (eigenfrequencies) $\boldsymbol{\omega}[n]$, which are aligned with the position of the peaks of room modes, damping $\boldsymbol{\xi}[n]$, attenuation and phase. For different positions of the microphones/sound sources across the room, some parameters stay the same - *common parameters*: eigenfrequencies which depend on the room shape, and the room mode damping, which depends on the damping of the walls. The attenuation and the phase of the room modes are position dependent parameters - *specific parameters.*

Figure 3.2 illustrates the difference between the attenuation and the phase of the RTFs across the room at the resonant frequencies. White point shows the fixed and known position of the microphone and colorful points are the positions of sound sources that should be estimated. As can be observed, although all the positions of the sound sources result in the peaks at the same set of frequencies (the resonant frequencies of the room), the set of the heights of these

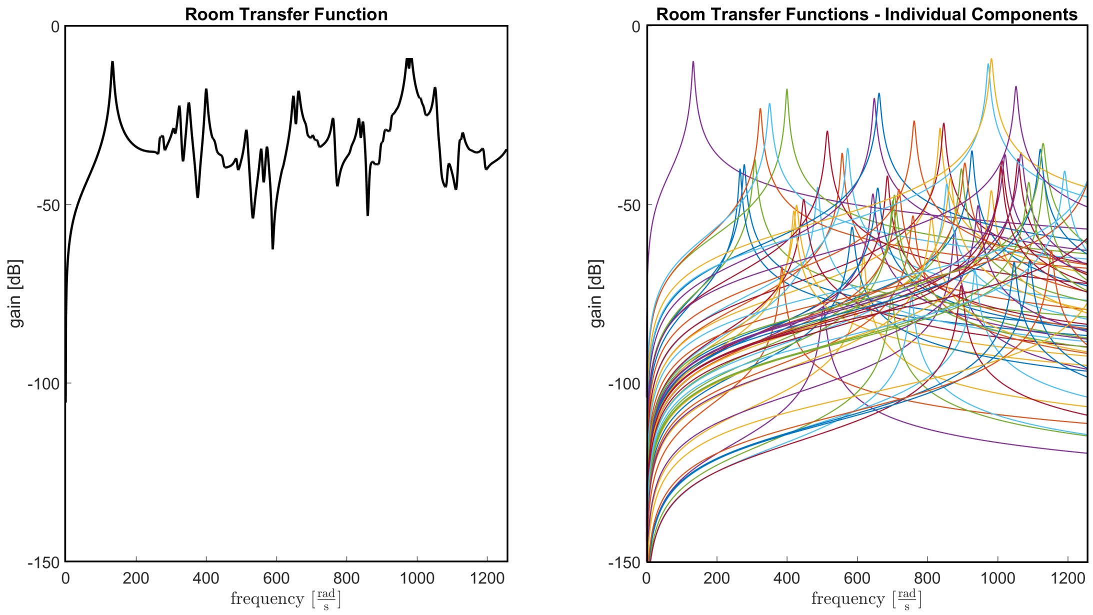


Figure 3.1 – Individual components of the RTF are called room modes. As illustrated, room modes can be simply modeled as second order bandpass filters.

peaks seems unique, as will be further observed in the next section. This means that each pair of the positions of a sound source and a microphone could potentially result in a unique set of attenuation factors at the resonant frequencies. An example of room mode for a room with rigid walls is given in Figure 3.3 [91].

Although there exists a uniqueness of phase for each room mode, since we plan to use only one microphone and white noise sources, this is irrelevant for our case but has a potential for some other type of room characterization. We have decided to investigate the potential of unique representation of a (position of the sound source, position of a microphone) pair, within the room with the set of attenuations of RTF at resonant frequencies. Therefore we have established a valuable reasoning for the design of our sensing matrix.

3.1.2 Ambiguities that exist in the terms of uniqueness of the attenuation across the room

We will observe the basic axial modes in Figure 3.4 in order to illustrate that relying only on them would not be sufficient to have a unique position representation. First row shows the x- and y-axial modes (everything that will be said applies analogously to z-axial modes as well). We can see that these two modes form pairs of points that result in a unique location identifier. But, since we have decided to explore the special case with only one microphone, we need to neglect the phase of the RTF. As seen in the second row of the same figure, this introduces ambiguity - there exists a unique representation, but only in $\frac{1}{8}$ of the room.

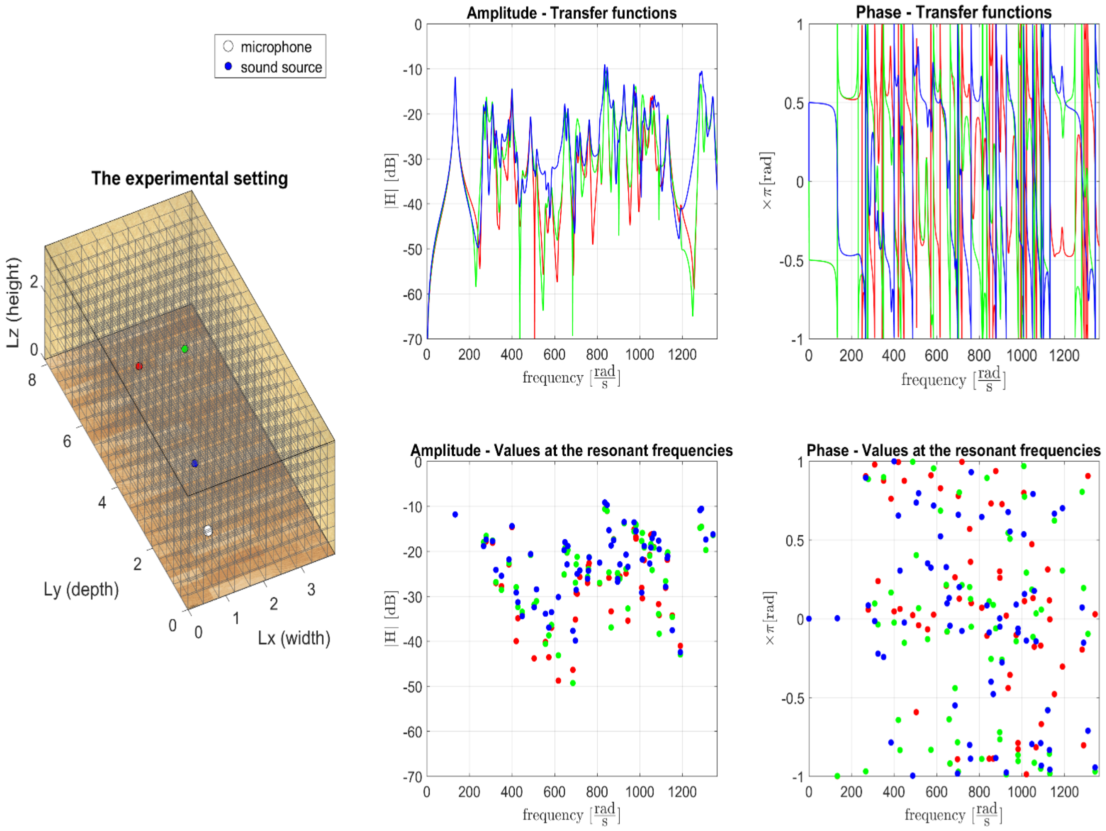


Figure 3.2 – Values of the RTF across the room vary in the terms of attenuation and phase value at the resonant frequencies. We exploit only the difference in the attenuation because in our target experimental setting there exists only one microphone and the sources will emit white noise.

3.2 Compressed sensing and sound source localization

3.2.1 Sparse representation of the position of sources

In sound source localization problems the domain of interest is usually divided into an angular grid such that the sources occupy just a few of these angles. Since our sources are positioned inside a room, we will divide the room into voxels and assume that the number of voxels occupied by a source is small. We recognize that this is a problem with underlying sparsity. These problems are usually solved by using the theory of compressed sensing.

3.2.2 Problem formulation within compressed sensing approach

We will be following the reasoning established with the Figure 2.3. Therefore, we have:

$$\hat{\boldsymbol{y}} = \boldsymbol{\Psi}\boldsymbol{\Phi}\boldsymbol{x} \tag{3.1}$$

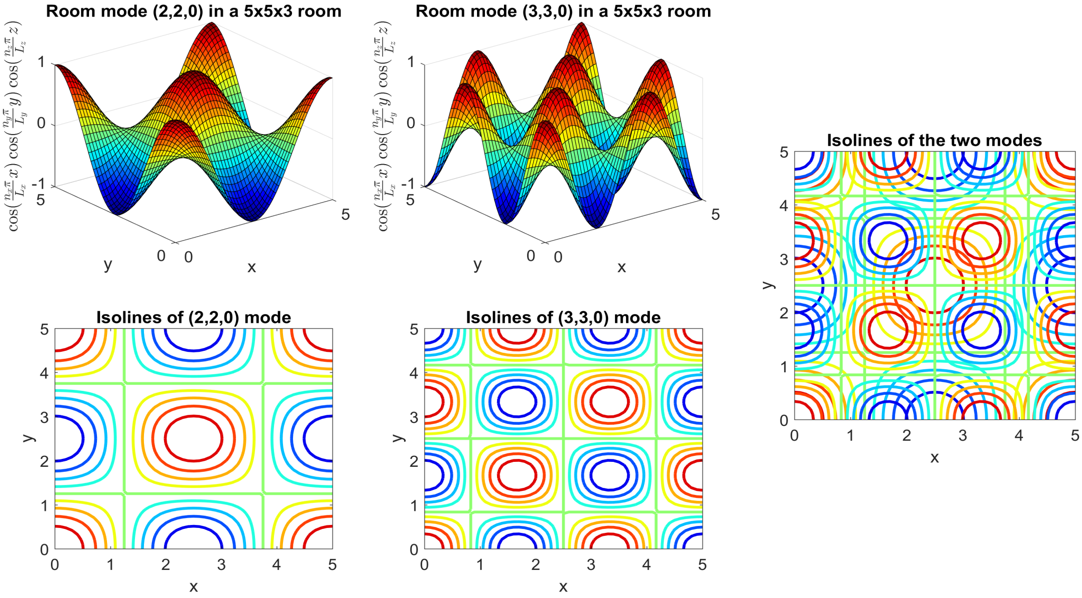


Figure 3.3 – An example of $(n_x, n_y, n_z) \in \{(2,2,0),(3,3,0)\}$ room modes in a 5m × 5m × 3m room with rigid walls. We can notice that the isolines of different modes intersect in just a few locations, which supports our assumption of different height of sets of peaks in the RTF.

where $\hat{\boldsymbol{y}}$ is the measurement of sound pressure at a known location inside a known room, $\boldsymbol{\Psi}$ is the row-wise subsampled identity matrix (used for pruning), $\boldsymbol{\Phi}$ is a representational basis with the RTFs as columns and $\boldsymbol{x}$ are the sparse expansion coefficients. The product $\mathbf{A} = \boldsymbol{\Psi\Phi}$ is usually referred to as the sensing matrix. $\mathbf{x}$ is K-sparse, which means that it contains at most K non-zero elements. We are facing an underdetermined system of equations with a sparse regularization.

3.2.3 A sensing matrix for sound source localization in a room

The following question rises: How to tailor a simple incoherent dictionary (along the definitions from Section 2.2.2) for fast localization of sources inside the room? In order to have a well-posed problem we introduce the following assumptions:

1. the shape of the room and the reverberation time are known,
2. the position of the microphone is known, and
3. all the sound sources have a flat spectrum in the observed frequency range.

For each of the potential positions of sound sources and a fixed position of the microphone we have one atom in the dictionary which consists out of the heights of the peaks in the RTFs at the resonant frequencies. The height of the dictionary is proportional to the number of the resonant frequencies in the observed frequency range. The number of resonant frequencies

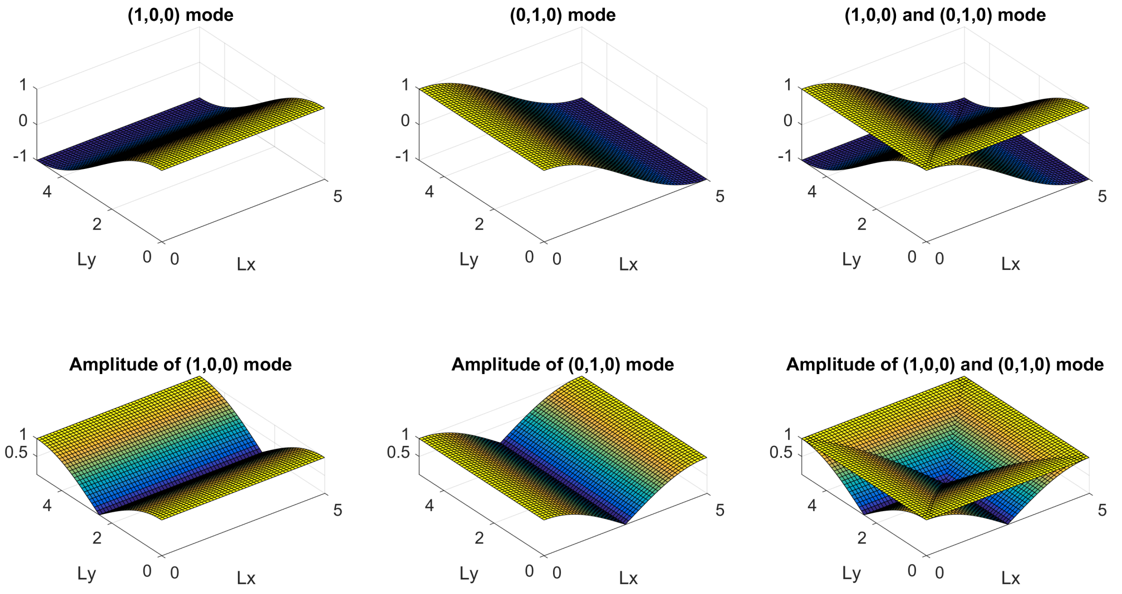


Figure 3.4 – Basic modes and their attenuation values.

below a given frequency f [91] can be computed by: $N(f) = \frac{4}{3}\pi V\left(\frac{f}{c}\right)^3 + \frac{1}{4}\pi S\left(\frac{f}{c}\right)^2 + \frac{1}{2}L\frac{f}{c}$, where $V = L_x L_y L_z$, $S = 2(L_x L_y + L_y L_z + L_z L_x)$ and $L = L_x + L_y + L_z$. The width of the dictionary is proportional to the number of observation points on the predefined grid.

In order to localize the sources, we search for a subset of atoms that give the best fitting for the signal recorded by the microphone. Once we discover which atoms of our sensing matrix have the highest expansion coefficients in the sparse representation, we can easily recover the position of the sound sources in the room, because we know which atom corresponds to which position, since we have tailored the dictionary ourselves.

3.3 Designing an efficient sensing matrix

3.3.1 Coherence

Coherence of a dictionary can be seen from the maximum off-diagonal element of the coherence Gram matrix $\mu = \max_{i \neq j} \mathbf{G}_{ij}$. In our case where $\mathbf{\Psi}$ is the row-wise subsampled identity matrix (used for pruning), $\mathbf{\Phi}$ is a representational basis with the RTFs as columns, the Gram matrix has the following form:

$$\mathbf{G} = |\mathbf{A}^H \mathbf{A}| = |(\mathbf{\Psi\Phi})^H \mathbf{\Psi\Phi}| = |\mathbf{\Phi}^H \mathbf{\Psi}^H \mathbf{\Psi\Phi}|. \tag{3.2}$$

From the definition of the matrices we have: $\mathbf{\Psi}^H \mathbf{\Psi} = \mathbf{I}$, so the Gram matrix has a simple form: $\mathbf{G} = \mathbf{\Phi}^H \mathbf{\Phi}$.

Therefore we observe the coherence of the sensing matrix by focusing on the discretization of the room transfer function. Since our exponentials in the plane wave representation are not equidistant, we can not apply the Dirichlet kernel sum to our case to simplify the expression (an approach common for many solutions [19, 195, 34]).

For a uniform case, the off-diagonal elements of our Gram matrix at position $\boldsymbol{r} = [r_x, r_y, r_z]^T$ and for wave vector $\boldsymbol{k} = [k_x, k_y, k_z]^T$ are proportional to:

$$\mathbf{G}_{ij} \sim cos(k_x r_x) cos(k_x(r_x \pm m\Delta_x)) + cos(k_y r_y) cos(k_y(r_y \pm n\Delta_y)) + cos(k_z r_z) cos(k_z(r_z \pm o\Delta_z)), \tag{3.3}$$

where $(m, n, o) \in \mathbb{R}^3$ and $[r_x \pm m\Delta_x, r_y \pm n\Delta_y, r_z \pm o\Delta_z]^T$ are potential positions on the grid in the room.

It results in a complex form of the elements of Gram matrix. Some observations have shown that we are dealing with highly correlated atoms. Therefore we need to find a workaround in order to have a successful source localization. Due to the smoothness of cosine function, the points on the potential sound source position grid that lay close, result in similar heights of the peaks in RIR.

3.3.2 Battle of the grids

Our problem has two degrees of freedom and both of them represent a selection process of the nodes on a uniform grid. We have a grid of wave vectors - *features* and a grid of potential positions of sound sources - *samples*. In Figure 3.5 the grid on the left-hand side repeats in all 6 directions and the one on the right-hand side repeats in 3 directions. For wave vectors we will use the matrix form as defined earlier: $\mathbf{K} \in \mathbb{R}^{N \times W \times 3}$, where N is the number of room modes and W is the number of plane waves per wave number.

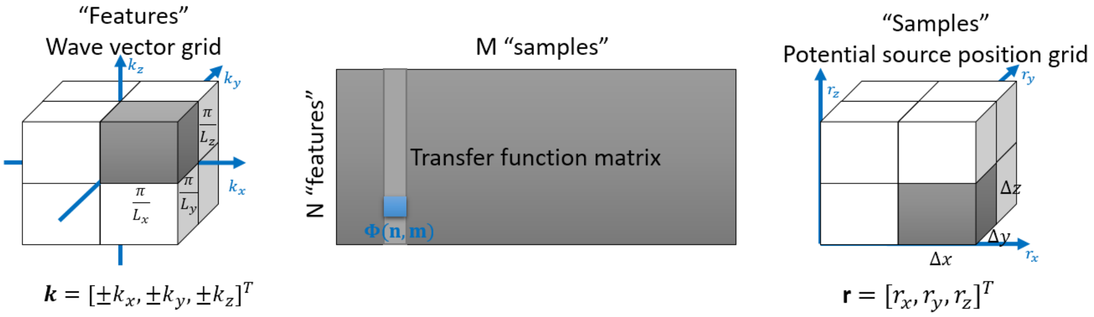


Figure 3.5 – Two grids that represent two degrees of freedom that we have for designing the sensing matrix.

We will observe the room transfer function in a matrix form at the resonant frequencies. If we go back to equation (1.8) and introduce $\omega = \tilde{\boldsymbol{\omega}}[n]$, we get that each of the entries of our sensing matrix $\mathbf{\Phi}$ is of the following form:

$$\Phi(n,m) = \frac{\rho c^2 Q_m}{2\boldsymbol{g}[n]\boldsymbol{\xi}[n]}\Xi(\mathbf{K}[n,1,:],\boldsymbol{r}_{\text{mic}})\Xi(\mathbf{K}[n,1,:],,\boldsymbol{r}_m) \tag{3.4}$$

which corresponds to n^{th} wave vector and m^{th} potential sound source position. The only coefficients that differ among the atoms of the dictionary are represented in blue. The difference due to the volume velocity of the sound source Q_m will not affect our approach, since we assume that we are observing our sound sources in a linear regime. This parameter has an effect only on the expansion coefficients of the sparse representation. Therefore we focus on the sound sources' positions that produces different attenuation of room modes.

So the RTF matrix has the following decomposition:

$$\mathbf{\Phi} = \frac{\rho c^2}{2}\begin{bmatrix} \frac{\Xi(\mathbf{K}[1,1,:],\mathbf{r}_{\text{mic}})}{\boldsymbol{g}[1]\boldsymbol{\xi}[1]} & \cdots & \frac{\Xi(\mathbf{K}[1,1,:],\mathbf{r}_{\text{mic}})}{\boldsymbol{g}[1]\boldsymbol{\xi}[1]} \\ \vdots & \ddots & \vdots \\ \frac{\Xi(\mathbf{K}[N,1,:],\mathbf{r}_{\text{mic}})}{\boldsymbol{g}[N]\boldsymbol{\xi}[N]} & \cdots & \frac{\Xi(\mathbf{K}[N,1,:],\mathbf{r}_{\text{mic}})}{\boldsymbol{g}[N]\boldsymbol{\xi}[N]} \end{bmatrix} \odot \begin{bmatrix} Q_1\Xi(\mathbf{K}[1,1,:],\mathbf{r}_1) & \dots & Q_M\Xi(\mathbf{K}[1,1,:],\mathbf{r}_{\text{M}}) \\ \vdots & \ddots & \vdots \\ Q_1\Xi(\mathbf{K}[N,1,:],\mathbf{r}_1) & \dots & Q_M\Xi(\mathbf{K}[N,1,:],\mathbf{r}_{\text{M}}) \end{bmatrix}. \tag{3.5}$$

As we have seen earlier, our rigid wall room modes are of the form (eq. 1.6), where $\mathbf{K}[n,1,:]$ belongs to the positive octant of the left-hand side grid from Figure 3.5.

3.4 Results

In our solution we will rely on the greedy approaches such as Orthogonal Matching Pursuit (OMP) [182] and Compressive Sampling Matching Pursuit (CoSaMP) [119]. These methods select up to K atoms of a dictionary that give the smallest approximation error. CoSaMP is a faster contemporary method which works by selecting multiple atoms at every iteration. The main drawback of these methods is that the sparsity of the signal has to be known upfront.

3.4.1 The recovery of signal's support in a highly coherent dictionary

Candès et al. [30] discuss the potential of recovery of data that has a sparse representation in a coherent dictionary. Coherent dictionaries can give guarantees only on the recovery of the sparse signal, but not on the recovery of the set of indices of atoms in sparse representation. That is because if we have pairs of atoms that are extremely coherent (almost collinear), e.i. we are far away from satisfying $\mu \leq \frac{1}{2K-1}$, where K is the level of sparsity, therefore we can not tell which one of them will be used for our sparse representation when projecting to a lower-dimension space. Schnass et al. have approached this problem by introducing a complementary dictionary of the same size, but with low coherence, which maintains the sparse support of the measurements [167]. Our approach will be in the spirit of random subdictionary selection [181]. There have been some approaches with subsampling of dictionaries over rows and columns in order to increase the speed of the convergence of greedy methods [130, 124], but using such subsampling methods for coherent dictionaries is still unexplored. Authors of these papers named one of these methods as StoCoSaMP (Stochastic CoSaMP) [124].

We restate our problem in the following manner: Recover sparse signal $\boldsymbol{x}$ from the following:

$$\mathbf{S}_{rf}\hat{\boldsymbol{y}} = \mathbf{S}_{rf}(\boldsymbol{\Phi}\mathbf{S}_{sp})\mathbf{x} \tag{3.6}$$

where $\boldsymbol{y}$ is the measured signal in frequency domain, $\mathbf{S}_{rf}$ is a resonant frequency selector that defines which points on the wave vector grid we observe and $\mathbf{S}_{sp}$ is a sound source position selector that defines which points on the potential source position grid we observe. Both matrices, $\mathbf{S}_{rf}$ and $\mathbf{S}_{sp}$, are just submatrices of an identity matrix. The first one is constructed from selected rows and the second one is constructed from selected columns. We could characterize our case as a highly sparse case, since the number of sources to be localized is going to be small. Only one or a few voxels in the room will be having a source inside.

Support of $\boldsymbol{x}$ shows which of the positions on the grid are the most probable positions of the sources. Without subsampling of the coherent dictionary, this support is usually wrongly estimated due to the ill-conditioness of the problem coming form the high coherence of the dictionary.

Here is the description of the algorithm ($\mathbf{I}$ is the identity matrix):

Algorithm 1 Localization of sound sources in a room with one microphone

```
Input: Measurements in frequency domain $\hat{\boldsymbol{y}}$, highly coherent room mode dictionary $\boldsymbol{\Phi}$ and the number of sources $K$.
Output: Reconstructed positions of the sound sources.
do
    Generate random subsampling matrices $\mathbf{S}_{rf} \underset{row}{\subset} \mathbf{I}$ and $\mathbf{S}_{sp} \underset{column}{\subset} \mathbf{I}$.
    Subsample the dictionary: $\boldsymbol{\Phi}_{ss} = \mathbf{S}_{rf}(\boldsymbol{\Phi}\mathbf{S}_{sp})$ and the measured signal $\hat{\boldsymbol{y}}_{ss} = \mathbf{S}_{rf}\hat{\boldsymbol{y}}$.
    Try to estimate the positions of the sound sources by estimating the support of $\mathbf{x}$ on $\boldsymbol{\Phi}_{ss}$ using CoSaMP for the given measured signal $\hat{\boldsymbol{y}}_{ss}$ knowing the level of sparsity $K$.
while CoSaMP [119] sparse representation does not converge (has norm of the residual significantly greater than zero)
```

Figure 3.6. shows a reconstruction example for a case with 3 sound sources. Grey circles are the potential positions taken into account in the current iteration, blue circles are the true positions and light blue points are the reconstructed positions. The red point represents the known position of the microphone. This algorithm has no problems with identifying position of sources that are close, as can be seen from the right hand side of the Figure 3.6.

We will observe how different subsampling schemes effect the success and speed of our sparse support estimation.

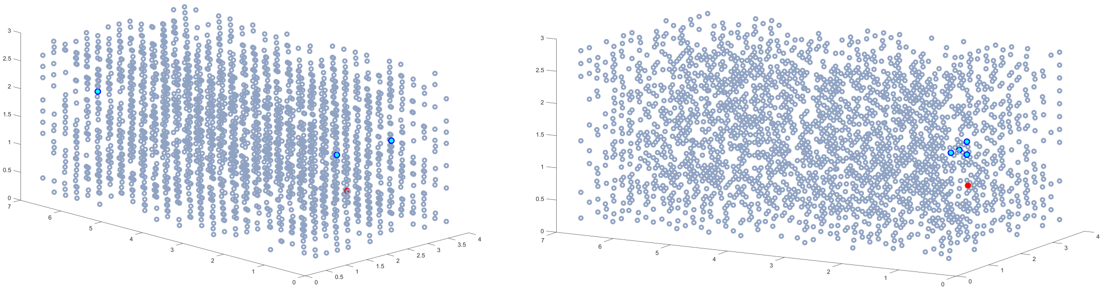


Figure 3.6 – These are the results for localization of 3 sound sources inside a 4m × 7m × 3m room for a uniformly undersampled 10 × 15 × 10 grid.

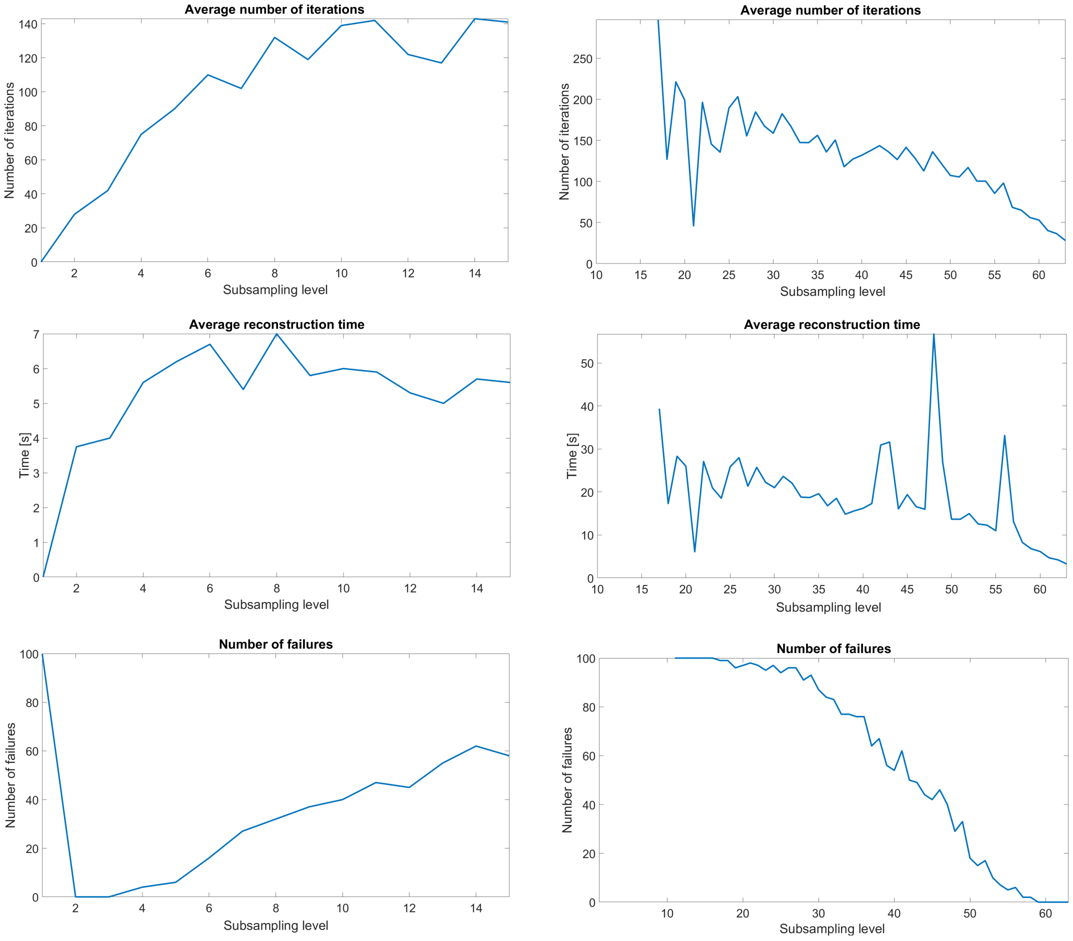
We have performed 100 Monte Carlo simulations for each set of parameters and for the estimation of the position of two sound sources. Experiments were performed on a single core of Intel Xeon processor at 2.8GHz of a computer with 16GB of RAM. If the algorithm did not converge within 300 iterations, we would consider that to be a failure. If we do not bound the number of iterations, the algorithm always converges but sometimes it needs a few thousands of iterations. Reconstruction time does not include the time needed for constructing the dictionary.

We have applied two types of subsampling schemes: subsampling over the spatial grid and also subsampling over the feature grid (resonant frequencies). The purpose of the spatial subsampling is to decrease the coherence of the dictionary by reducing the number of atoms. The second type of subsampling has the goal of decreasing the computational costs and increasing the speed of convergence of the algorithm.

In Figure 3.7a we can see results for no subsampling over resonant frequencies (first 63 resonant frequencies were taken into account - room modes between $(1,0,0)$ and $(3,3,3)$) and different subsamplings over the potential sound source positions. There were no successful reconstruction attempts when the whole grid was taken into account. Subsampling two or three times showed the best performance with the convergence within the predefined 300 iterations. Average number of iterations and average reconstruction time were computed only for the successful quick reconstructions.

In Figure 3.7b we can see results for subsampling level of 2 over the potential positions of sound sources and different subsets of resonant frequencies have been taken into account (from 11 up to 63 out of 63). If we choose a subset of below 17 resonant frequencies, the algorithm never converges. If we had average results over more than 100 simulations, the curves in the results would have been smoother. We see that we can not subsample a lot such a small set of resonant frequencies.

Therefore, we have to subsample the sound source position grid since we are dealing with a highly coherent dictionary. By increasing the level of subsampling over columns of the



(a) Potential sound source position grid subsampling (from no subsampling up to subsampling 15 times).

(b) Resonant frequency grid subsampling (from selecting 11 up to selecting all 63 resonant frequencies)

Figure 3.7 – Spatial and feature grid subsampling for sound source localization.

dictionary, we decrease the probability that the atoms that we are searching for are present in the subset. On the other hand, the resonant frequency grid should not be too oversampled in order to achieve a quick convergence (below predefined 300 iterations or similar).

3.4.2 Precision and basis mismatch

Due to the smoothness of the room mode functions, there is a small variation in the value between the close points. This supports the idea of similarity of the atoms of the dictionary of the spatially close positions.

Compressed sensing usually assumes the existence of a grid with finite density and our signals of interests can fail to coincide with the nodes of the predefined grid, especially in the case of moving sources. As shown in [34] this can cause that sparse signals appear incompressible.

The work we have observed before [195] has an extension to a continuous case [194] by applying the semi-definite programming [189]. In our observations we have assumed that our grid of the potential positions of the sources is dense enough to avoid the spectral leakage. Continuous approaches are left for future work.

3.4.3 Requirements and limitations

In a setting where we have multiple sound sources and a microphone, the sound received is equal to the linear combination of the convolution of sounds emitted by the sound sources and the transfer functions that correspond to their positions. Therefore we need the following assumption: we can efficiently localize sources that are wide-band, such that we target the resonant frequencies where the room modes are.

In order to avoid ill-conditioness the microphone and the sound sources should lie off the planes of symmetry.

3.5 Conclusion

By observing the sound source localization problem through the theory of compressed sensing, we have enabled localization of multiple sound sources in a room using only one microphone. Unlike most of the localization algorithms, this approach guaranties the localization in 3D, without neglecting the elevation angle, which is rarely estimated. The simplicity of our solution lays in the low required prior knowledge about the room - only the height of the peaks in the RTF at the resonant frequencies should be know. *matlab* code used for generating each of the figures in this paper as well as the acoustical room mode framework is available for download[1].

Our solution has the potential of being applied to the optimization of the quality of the hearing aids - once the location of source is estimated we can introduce weighting on the reception side, as well as in robotics for monoaural localization. The emerging field of virtual reality would be just another domain of potential application.

Future work will include estimation *off the grid* in order to avoid the basis mismatch and the challenging computational costs. Removal of the assumption on the level of sparsity should also be investigated further.

Another possible extension would be encoding the position in the room in the term of *relative transfer function* (RTF) [100] after adding another microphone into the room. This would remove the requirements for white noise sound sources, since RTF is invariant to the input signal.

---

[1] https://github.com/epfl-lts2/room_transfer_function_toolkit

4 Joint Estimation of Room Geometry and Modes with Compressed Sensing

*

Acoustical behavior of a room for a given position of microphone and sound source is usually described using the room impulse response. If we rely on the standard uniform sampling, the estimation of room impulse response for arbitrary positions in the room requires a large number of measurements, because we can not know upfront which microphones might be set at the nodal lines of room modes (where room mode value is zero). In order to lower the required sampling rate, some solutions have emerged that exploit the sparse representation of the room's wave field in the terms of plane waves in the low-frequency domain. The plane wave representation has a simple form in rectangular rooms. We will observe the basic axial modes of the wave vector grid for extraction of the room geometry and then we propagate the knowledge to higher order modes out of the low-pass version of the measurements.

In 2006 Ajdler et al. [9] have defined the Plenacoustic function (PAF) as the function that contains the room impulse responses (RIRs) for all the possible pairs of microphone and source positions in a room with the given acoustical properties. Without having any prior knowledge involved, it is extremely hard to estimate the PAF. As shown by Moiola et al. [113] the acoustical behavior of the room can be described by a discrete sum of plane waves that can exist inside a given room which are tightly related to the resonant frequencies as described in Section 1.2.2. This plane wave approximation holds for any star-convex room and is independent of boundary conditions, domain of propagation, type of the source or proximity to the source or the walls [112].

Sparse plane wave approximation in the low frequency domain introduces an assumption required for sparse analysis of room's complex wave field which further opens the door to compressed sensing [52, 27]. Mignot et al. [112] have started the trend of the sparse modal analysis for room acoustics. They have designed a greedy approach which uses space decomposition based on iterative alternating projections for the estimation of the wave number and wave vectors that fully determine the acoustical behaviour of the given room. Due to the high dimensionality of data acquired by microphones, greedy methods such as Simultaneous

---

[1] Work done with Thach Pham Vu at École polytechnique fédérale de Lausanne.

Orthogonal Matching Pursuit (SOMP) [182] (simultaneous, since we are fitting measurements from multiple microphones at once) have shown better performance than the relaxation of the minimization of ℓ_0 norm [183].

Our solution focuses on the structured sparsity of the plane wave representation for the reconstruction of parameters of the Room Transfer Function (RTF). In literature, sparse plane wave representation has been used not only for the representation of the wave field in a room in low frequency domain, but also for efficient storage of highly correlated recordings of dense microphone arrays [88]. Besides sparse plane wave representation an interesting sparse approach to the estimation of RTF is a recent approach with orthonormal basis functions based on infinite impulse response filters (IIR) [185]. On the other hand, although not exploring plane wave sparsity, the solution relying on the weighted spatio-temporal representation [11] also gives promising room impulse response interpolations.

In general, the solutions for estimating the shape of the room usually rely on knowing the location of early reflections [89, 50], but finding the true reflections within an echogram is not a trivial problem and is still an open research question.

4.1 Problem Setup

The goal of this research will be an attempt to acquire data below the temporal and spatial constraints of PAF and CFL conditions established earlier in Section 1.2.1 by exploiting the plane wave sparsity in room acoustics. Also, we will be relying on the definition of structured Room Transfer Function from (eq. 1.8) and will be focusing on its parameters estimation in a lightly damped setting.

4.1.1 Spherical search space of plane wave approximation

In the case of a room with *light damping* ($\boldsymbol{\xi}[n] \ll \boldsymbol{\omega}[n]$), the length of the wave vectors can be approximated by the real part of the corresponding wave number: $\|\mathbf{K}[n, w, :]\| = |\boldsymbol{\kappa}[n]|$, since in that case $\boldsymbol{\kappa}[n] \approx \frac{\boldsymbol{\omega}[n]}{c}$. This builds an intuition for the spherical vector search as can be seen in Figure 4.1 which will be explained more in detail later. For a rectangular room the wave vectors are on the vertices of a parallelepiped inscribed into the sphere with radius of $\frac{\boldsymbol{\omega}[n]}{c}$.

If we rely on the modal decomposition (eq. 1.9) and plane wave approximation, we need to estimate the following parameters: resonant frequencies $\boldsymbol{\omega}$, damping factors $\boldsymbol{\xi}$, wave vectors $\mathbf{K}$ and expansion coefficients $\mathbf{A}$. This will be done through an iterative procedure that relies on alternating recovery of temporal and spatial parameters, as will be explained further.

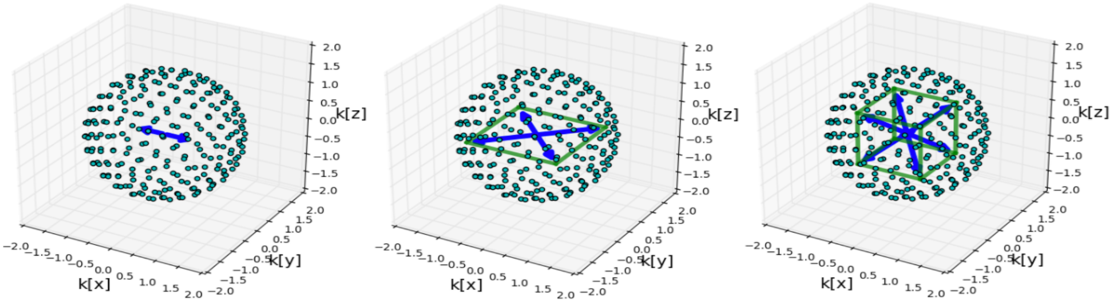


Figure 4.1 – Structured sparsity of wave vectors for different plane wave types. In theory these wave vectors form a parallelepiped inscribed into a sphere with radius $\frac{\boldsymbol{\omega}[n]}{c}$, resulting in structured sparsity. From left to right: x-axial mode, xy-tangential mode and oblique mode. Tightly related to Figure 1.3.

4.1.2 Periodicity of the wave vector grid

In our solution we will be focusing only on the rectangular rooms with the regular wave vector grid (regular eigenvalue lattices in the wave vector space) [91] as shown in Figure 4.2. The *k-space* is an array of numbers representing spatial frequencies. According to the theory, as long as we know the periodicity of the grid over each of the axes, it will provide us the knowledge on the room geometry as well as the values of the wave vectors of higher order. So the goal of our approach is the estimation of these three periods along each of the axes. Under the assumptions that the room is lightly damped, the three fundamental axial modes can be used as a basis to find all higher order modes. This will reduce the cutoff frequency of the analyzed data, which further reduces the density of the required grid of microphones, due to the dependencies between the temporal and spatial sampling as shown earlier.

4.2 Parameter estimation with partial compressed sensing for structured data

There are two key questions for our parameter estimation procedure: how many room modes N do we expect up to a given cutoff frequency f_c and what are their approximate resonant frequencies $\boldsymbol{\omega}[n]$? These parameters are dependent on the room shape and size [91]. An approximate number of modes up to the cutoff frequency f_c is given by: $\tilde{N}_{f_c} \approx \frac{4\pi}{3} V \left(\frac{f_c}{c}\right)^3$ where $V = L_x L_y L_z$. This is because most of the modes in a room are oblique modes and we follow the formulation from [91].

In our solution we will be using the curve fitting algorithm[2] from 1985 by Richardson et al. [154] that allows the reconstruction of the RTF curve from discrete measurements using room mode shaped functions as basic fitting elements. Since we will be evaluating our algorithm with real measurements, the curve fitting algorithm will help us to retrieve some of the parameters and

---

[2] https://ch.mathworks.com/matlabcentral/fileexchange/3805-rational-fraction-polynomial-method?focused=5049537&tab=function

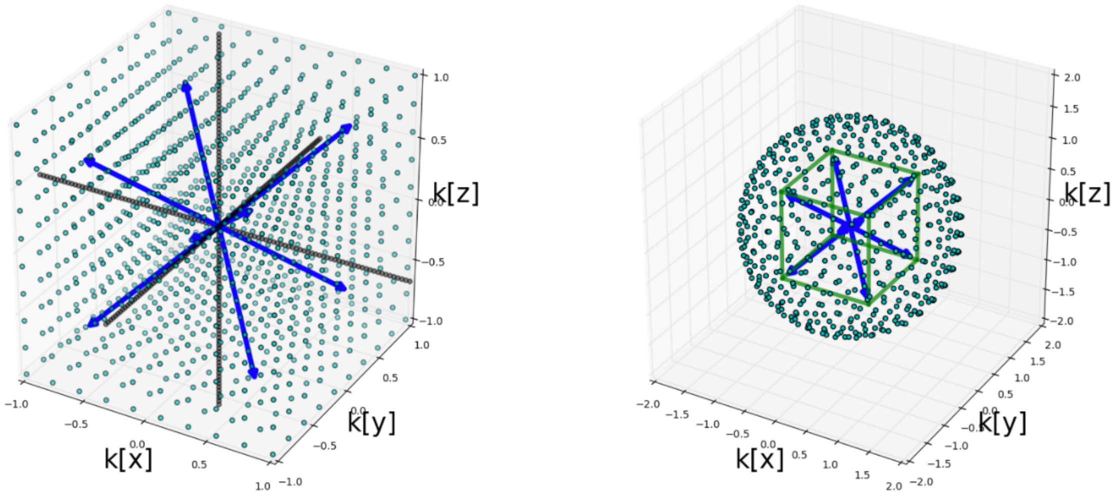


Figure 4.2 – The left hand side shows the periodicity of the wave vector grid with respect to $\mathbf{K}[n, w, :] = [\pm k_x, \pm k_y, \pm k_z]$ with period over the axes equal to: $\frac{\pi}{L_x}$, $\frac{\pi}{L_y}$ and $\frac{\pi}{L_z}$. Here we see an example of an oblique wave vector. The right hand side shows the structured search space on our uniformly sampled sphere.

use them as the ground truth.

Of special interest will be the basic axial resonant frequencies: $\boldsymbol{\omega}_{[k_x,k_y,k_z]=[1,0,0]} = \frac{\pi c}{L_x}$, $\boldsymbol{\omega}_{[k_x,k_y,k_z]=[0,1,0]} = \frac{\pi c}{L_y}$ and $\boldsymbol{\omega}_{[k_x,k_y,k_z]=[0,0,1]} = \frac{\pi c}{L_z}$, because they will provide the data about the shape of the room.

4.2.1 Reconstruction procedure

We will be relying on the acoustical properties of rectangular room as described in Section 1.2.2. Our goal is to reconstruct spatial periods of the wave vector grid from low-pass room impulse responses over each of the axes. The size of the room is assumed to be unknown and is jointly estimated. All measured signals are separated into two components: low-pass $\mathbf{X}_l$ and high-pass $\mathbf{X}_h$. Analysis procedure is first applied to the low-pass component, which includes the estimation of the wave numbers and corresponding wave vectors. The bandwidth of this low-pass analysis is chosen in such a way that it covers reasonable sizes of rooms and removes the false modes that can appear below the first mode in RTF. With $f \in [20, 70]$Hz we cover room dimensions $L_x, L_y, L_z \in [2.45, 8.575]$m for $c = 343\frac{\text{m}}{\text{s}}$. This can easily be adjusted for rooms of unusual sizes.

Estimation of $\boldsymbol{\omega}[i_l]$, $\boldsymbol{\xi}[i_l]$ and $\boldsymbol{\xi}[i_h]$

In the *low* part of the frequency domain observations we define a unit-norm temporal dictionary with atoms of form: $\boldsymbol{\Theta}[:, i] = \frac{\boldsymbol{\theta}[i]}{\|\boldsymbol{\theta}[i]\|}$, where $\boldsymbol{\theta}[i] = e^{\boldsymbol{\xi}[i]t} e^{j\boldsymbol{\omega}[i]t}$ and i is an index on a 2D grid

Algorithm 2 ReSEMblE algorithm (Algorithm for the joint estimation of Room SizEs and ModEs)

Input: A set of measurements at M *known* locations $\boldsymbol{r} = [x, y, z]^T$ in space and T points in time. $\mathbf{X} \in \mathbb{C}^{T\times M}$ are measurements in a matrix form and $\boldsymbol{x} \in \mathbb{C}^{TM}$ are measurements in a vectorized form. f_p is frequency that separates data into 2 analysis procedures.

Output: Estimated room size $(\tilde{L}_x, \tilde{L}_y, \tilde{L}_z)$ and estimated room transfer function parameters:

- expansion coefficients $\{\mathbf{A}[n, w]\}_{n=1,w=1}^{N,W}$,
- resonant frequencies $\{\boldsymbol{\omega}[n]\}_{n=1}^{N}$ and damping $\{\boldsymbol{\xi}[n]\}_{n=1}^{N}$, and
- wave vectors $\{\mathbf{K}[n, w, 1:3]\}_{n=1,w=1}^{N,W}$

N: number of modes, W: number of wave vectors per wave number.

```
Separate the measurements with $f_p$: $\mathbf{X} = \mathbf{X}_l + \mathbf{X}_h$ and $\boldsymbol{x} = \boldsymbol{x}_l + \boldsymbol{x}_h$.
for $i_l \in \{1, ..., N_l\}$ do
    step 1: estimate $(\boldsymbol{\omega}[i_l], \boldsymbol{\xi}[i_l])$ from $\mathbf{X}_l^{(i_l)}$ with Fast Fourier Transform;
    step 2: estimate $\mathbf{K}[i_l, :, :]$ from $\boldsymbol{x}_l^{(i_l)}$ with inscribed parallelepiped search;
    step 3: compute new residuals: $\mathbf{X}_l^{(i_l+1)}$ and $\boldsymbol{x}_l^{(i_l+1)}$;
end for
Recover the room size $\tilde{L}_x, \tilde{L}_y, \tilde{L}_z$ from basic axial room modes.
for $i_h \in \{N_l + 1, ..., N\}$ do
    step 1: get $\boldsymbol{\omega}[i_h]$ and $\mathbf{K}[i_h, :, :]$ from the wave vector grid;        // See Fig. 4.2
    step 2: estimate $\boldsymbol{\xi}[i_h]$ from $\mathbf{X}_h^{(i_h)}$;
    step 3: compute new residuals: $\mathbf{X}_h^{(i_h+1)}$ and $\boldsymbol{x}_h^{(i_h+1)}$;
end for
Estimate the expansion coefficients $\{\mathbf{A}[n, w]\}_{n=1,w=1}^{N,W}$ using least squares approach.
return $(\tilde{L}_x, \tilde{L}_y, \tilde{L}_z)$, $\{\boldsymbol{\omega}[n]\}_{n=1}^{N}$,$\{\boldsymbol{\xi}[n]\}_{n=1}^{N}$, $\{\mathbf{K}[n, w, 1:3]\}_{n=1,w=1}^{N,W}$ and $\{\mathbf{A}[n, w]\}_{n=1,w=1}^{N,W}$.
```

of possible $(\boldsymbol{\omega}[i_l], \boldsymbol{\xi}[i_l])$, $\boldsymbol{\omega}[i_l] \in [0, \pi f_s]$ and $\boldsymbol{\xi}[i_l] \in [10\xi_0, 0.1\xi_0]$, $\xi_0 = -3\frac{\ln 10}{t_{60}}$. The reverberation time can be computed through Sabine's law $t_{60} \approx 0,163\frac{V}{A}$, where: $A = \sum_i \alpha_i = \sum_i a_i S_i$, a_i is the absorption coefficient of the i^{th} wall and S_i is its surface. The atoms with the highest correlation contains the solution pair.

In the *high* part the frequency is known, so we have only a 1D grid of possible values for the damping, which leads to a much simplified search.

Estimation of $\mathbf{K}[i_l, :, :]$

The estimation of wave vectors is done with a structured group sparsity assumption - after estimating the wave number, we construct a sphere with a radius $\frac{\boldsymbol{\omega}[i_l]}{c}$ which follows from the assumption of lightly damped modes. We define a non-unit-norm spatio-temporal dictionary with atoms of form: $\boldsymbol{\Sigma}[:, i] = e^{\boldsymbol{\xi}[i_l]t} e^{j\boldsymbol{\omega}[i_l]t} e^{j\tilde{\mathbf{K}}[i,:]\cdot \boldsymbol{r}}$, where $\tilde{\mathbf{K}}$ are samples on this uniformly sampled sphere[3] [169].

On the surface of this sphere we search for a group of 8 wave vectors $[\pm k_x, \pm k_y, \pm k_z]^T$ which form a parallelepiped and which are aligned with the residual the most. In a case of tangential modes, the parallelepiped collapses over 1 dimension and shrinks to 4 wave vectors (e.g. $[\pm k_x, \pm k_y, 0]^T$), and axial modes are defined by 2 wave vectors (e.g. $[\pm k_x, 0, 0]^T$).

In each iteration the best subgroup of 8 atoms has been estimated by applying a simultaneous version of matching pursuit (MP) [108] and the new residual is estimated by an orthogonal projection onto the space spanned by the union of all of the subgroups that were previously selected.

4.3 Results

4.3.1 Reconstruction of the *k*-space of a rectangular room

In our solution we have relied on two types of structured sparsity expected in theory [91]: wave vector sparsity as nodes of parallelepiped and wave vector periodicity in the wave vector grid. How does this structured approach affect the data retrieval? As shown in [112, 11] efficient interpolation of the sound field is expected only within the part of the room surrounded by microphones used for training of the parameters.

We will present the performance of our approach on measurements made in a rectangular room with an approximate size 3m × 5.6m × 3.53m. Properties of the chosen room are observed in [23]. Microphones are distributed randomly inside a 1m side cube in one half of the room and the sound source is in the other half of the room. Random placement serves the purpose of reducing the coherence of the captured data. Since we were processing real measurements, in order to have an idea about the approximate value of some of the parameters we want to estimate, we have applied the rational fraction polynomial curve fitting [154] based on the room mode shaped polynomials as basic fitting elements. In this way we have retrieved approximate ground truth values of resonant frequencies and mode damping factors. During the curve fitting process, our wave numbers $\boldsymbol{\kappa}[n] = \frac{\omega[n]+j\xi[n]}{c}$ appear in the poles of the fitted function [155] as in (eq. 1.8). The retrieved values of the axial room modes are in accordance with the laser measurements for the room dimensions of the given room related through (eq. 1.5).

Figure 4.3. shows the results for the estimation of the room mode resonant frequencies and their position in the k-space in the *low* part of the algorithm with 20 microphones. Here the f_p frequency was set to be 70Hz. The basic axial modes are easily recognized and they give a fine approximation of the room size up to a few cm away from ground truth. We can notice that the k_x and k_y component of the estimated wave vectors give a good approximation, but

---

[3] https://ch.mathworks.com/matlabcentral/fileexchange/37004-suite-of-functions-to-perform-uniform-sampling-of-a-sphere?s_tid=prof_contriblnk

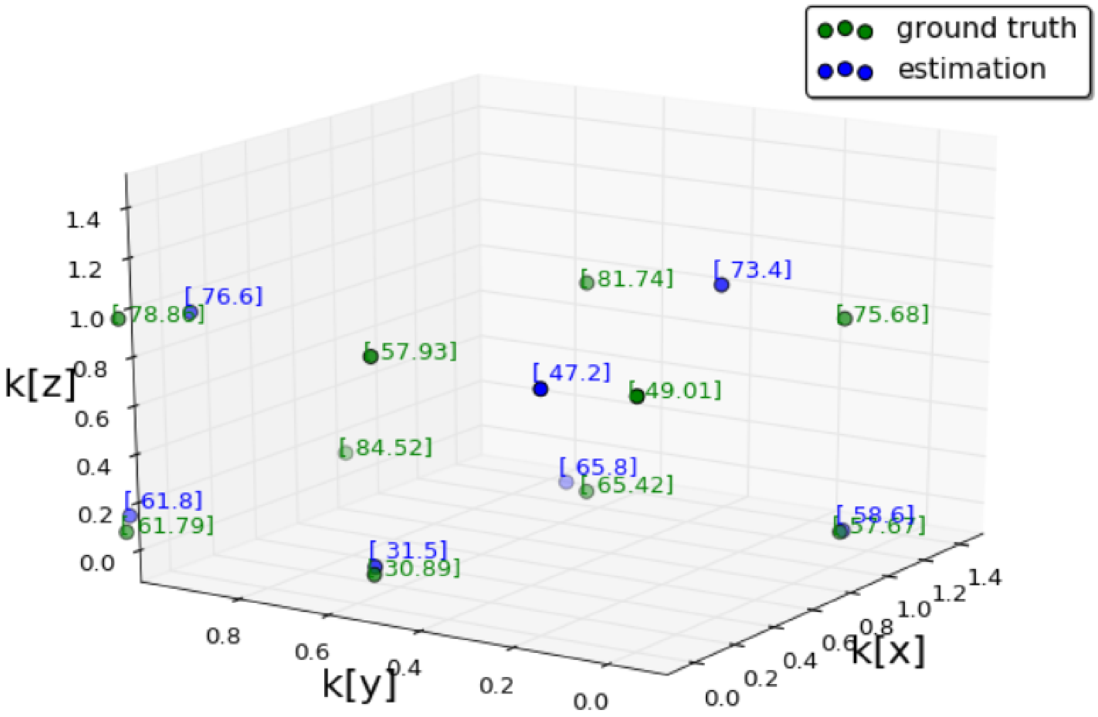


Figure 4.3 – The estimation of wave vectors in k-space. The numbers next to the points indicate the corresponding eigenfrequencies (in Hz). What we expect from theory in a case with perfectly rigid walls is plotted against the values we get from the measurements.

there is a slight deviation in the k_z direction. This is attributed to the fact that in the room where the measurements were performed the floor is made from wood and ceiling is made from concrete. Also the slight deviation of the eigenfrequencies can be attributed to the fact that the search of the wave vectors was performed with a rigid wall model $\|\mathbf{K}[n, w, :]\| = |\boldsymbol{\kappa}[n]|$.

After applying the *high* part of the algorithm, the Pearson correlation coefficient showed that the approximation is good (e.g. 82% for only 19-microphone setting and $f_c = 200\text{Hz}$), but it should be further improved once the nature of the deviation of the wave vectors in z direction is efficiently characterized.

4.4 Reconstruction of the room shape

An important line of work on room shape estimation was done by Dokmanic et al. [50]. Although originally the key weakness of this paper was estimation of the location of echoes from a bandlimited noisy recording, this line of research was followed by a method for improved echo detection and pruning [51]. This approach and the one we propose, operate in different domains - their approach is in temporal and ours is in frequency domain. Despite the fact that both methods offer recovery within up to a few centimeters, the main differences are that our method requires more microphones, but does not require any preprocessing in the term of peak picking and echo pruning. Also, we are not limited to high quality equipment, since the proposed algorithm operates in the low-frequency domain.

On the other hand, with the proposed method we have only covered the case of regular rectangular room. Since plane waves exist also in non-rectangular rooms, our method could be extended to more diverse cases with higher computational complexity when it comes to the structured search of the wave vector space.

4.5 Conclusion

The proposed solution is suitable only for rectangular shaped rooms that are lightly damped, which was confirmed by the experiments. Also, the sound source has to be put in a position such that it excites all the axial modes. Although the solution requires the number of modes N, the reverberation time t_{60} and the sound celerity c to be know, solution is not sensitive to their slight perturbation. The estimation of approximate structure of the k-space has lead to the reduction in the terms of number of required measurements and in the increase of the speed of the reconstruction without great losses of quality, but not for a broad range of frequencies. The higher we take the frequencies, the greater become the deviations. In the spirit of reproducible research, we have decided to open our data on the Zenodo platform [4] and the code on github [5].

4.6 Future work

Relying on the regularity of the modes in the terms of parallelepipedic shape resulted in a good approximation, but relying on the periodicity of the wave vector grid has shown medium results, especially for higher order modes. Future work will include further investigation on the characterization of the deviation of the periodic wave vector grid from its theoretically projected values imposed by the rigid wall model. We have noticed a higher deviation of the modes along the z-axis than along the x- and y-axis. This might be explained well by the temperature gradient that exists along this dimension that can largely affect the efficiency of the approximation, especially since the speed of sound is a function of temperature.

---

[4] https://zenodo.org/record/1169161

[5] https://github.com/epfl-lts2/joint_estimation_of_room_geometry_and_modes

Part IV

Parametric models for echo estimation: Location, weight and density

5 MULAN: A Blind and Off-Grid Method for Multichannel Echo Retrieval[1]

When a wave propagates from a point source through a medium and is reflected on surfaces before reaching sensors, the measured signals consist of mixtures of the direct path signal with delayed and attenuated copies of itself. This physical phenomenon is commonly referred to as *echoes* and has a wide range of applications in different areas of science, from sonars [84] to seismology [162], from acoustics [50, 41, 40] to ultrasounds [4]. For instance, in acoustics, it has been shown that precise knowledge of early echo timing enables the estimation of the positions of reflective surfaces in a room [50, 41, 193]. In [50], the approximate 3D geometry of Lausanne cathedral could be retrieved in this way. On the other hand, echoes' attenuation capture information about the acoustic *impedance* of surfaces, which is notoriously hard to measure or estimate in practice [12, 17]. In [51] and [164], it is shown that knowing the attenuation and timing of early echoes may improve beamforming and source separation performance, respectively. Systems using echoes for beamforming are commonly referred to as *rake receivers* in the wireless literature [140].

Retrieving echo properties when the emitted signal is known is referred to as *active echolocation* in biology, and is well exemplified by the sensory system of echoing bats. This principle is for instance at the heart of active sonar technologies. In the signal processing literature, this problem belongs to the category of *system* or *channel identification*, *i.e.*, estimating the filters from a known input to the observed output of a linear system. In the case of echoes, these linear filters consist of streams of Diracs in the continuous-time domain and are hence sparse in the discrete-time domain. The more challenging problem of estimating echoes/filters when the emitted signal is unknown is referred to as *passive echolocation* or *blind system identification* (BSI) [196, 71, 2, 8, 102, 77, 87, 40, 99, 96]. BSI is a long-standing and still active research topic in signal processing, notably due to its fundamental ill-posedness. In the general setting of arbitrary signals and filters, rigorous theoretical ambiguities under which the problem is unsolvable have been identified [196]. A number of methods for multichannel BSI with general signals and filters have been developed some time ago [196, 71, 2]. Some well-known limitations of these approaches are their sensitivity to the chosen length of filters, and their

---

[1] Work done as research intern at INRIA, Rennes. A collaboration with Antoine Deleforge.

intractability when the filters are too large. Following the compressed sensing wave [28], a number of methods extending these BSI methods to the case of sparse [8, 102, 77, 87, 40] or structured [96] filters have been developed. They generally extend classical methods using regularizers such as the ℓ_1-norm for sparsity or a bilinear constraint as in [96]. Similarly to classical filter estimation methods, they require knowledge of the filters' length and they work in the space of discrete-time filters which are typically thousands of samples long. Because they work in the discrete-time domain, the accuracy at which these methods can recover echo locations is fundamentally limited by the signal's frequency of sampling: the recovered echoes are *on-grid*. Moreover, the sparsity assumption on filters is invalid in practice due to smoothing and sampling effects at sensors that comes from finite sampling frequency of the device. Interestingly, [33] employs a continuous-time spike model for single-channel blind deconvolution but relies on a strong linear prior on the signal.

We propose a drastically different approach to blind echo retrieval based on the framework of Finite Rate of Innovation (FRI) sampling [191, 19, 197]. In stark contrast with existing methods, the approach directly operates in the space of continuous-time echoes, and is hence able to blindly recover their locations *off-grid*. The proposed method is shown to recover echo delays and attenuation with an accuracy far higher than what the sampling rate would normally allow, using noiseless multichannel discrete-time measurements of an unknown simulated speech emitter in a room. The method does not assume that the filters are finite-length and only requires the number of echoes.

In this chapter we will be addressing the general problem of *blind echo retrieval*, *i.e.*, given M sensors measuring in the discrete-time domain M mixtures of K delayed and attenuated copies of an unknown source signal, can the echo locations and weights be recovered? This problem has broad applications in fields such as sonars, seismology, ultrasounds or room acoustics. It belongs to the broader class of blind channel identification problems, which have been intensively studied in signal processing. Existing methods in the literature proceed in two steps: (i) blind estimation of sparse discrete-time filters and (ii) echo information retrieval by peak-picking on filters. The precision of these methods is fundamentally limited by the rate at which the signals are sampled: estimated echo locations are necessary *on-grid*, and since true locations never match the sampling grid, the weight estimation precision is impacted. This comes from the basis mismatch problem, as was discussed earlier, because we retrieve the weight at the retrieved location which causes both components of the (location, weight) pair to be incorrect. We propose a radically different approach to the problem, building on the framework of Finite Rate of Innovation sampling. The approach operates directly in the parameter-space of echo locations and weights, and enables near-exact blind and *off-grid* echo retrieval from discrete-time measurements. It is shown to outperform conventional methods by several orders of magnitude in precision.

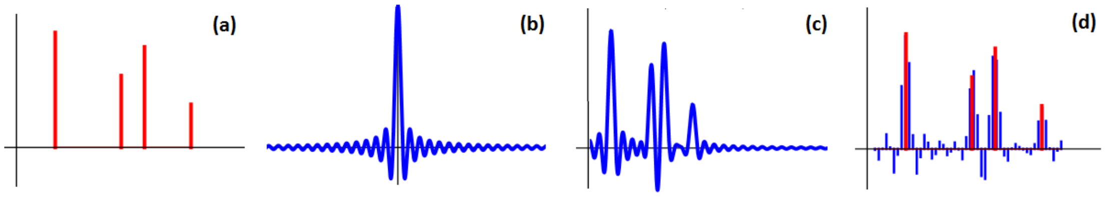


Figure 5.1 – (a) Continuous-time stream of Diracs $h(t)$, (b) sinc kernel $\phi(t)$, (c) smoothed stream $(\phi * h)(t)$, (d) original stream $h(t)$ (red) and its smoothed, sampled version $\boldsymbol{h} \in \mathbb{R}^L$ (blue).

5.1 The signal and measurement models

We start by defining the signal model in the continuous-time domain. Suppose a source emits a band-limited signal $s(t)$ which is reflected and attenuated K times before reaching M sensors. The continuous signal impinging at sensor m is

$$x_m(t) = (h_m * s)(t) \tag{5.1}$$

where $h_m(t)$ is a linear filter from the source to sensor m and $*$ denotes the continuous convolution operator defined by

$$(x * y)(t) = \int_{-\infty}^{+\infty} x(u)y(t-u)du. \tag{5.2}$$

The filter consists of the following stream of Diracs:

$$h_m(t) = \sum_{k=1}^{K} \boldsymbol{c}_m[k]\delta(t - \boldsymbol{\tau}_m[k]), \tag{5.3}$$

where δ denotes the Dirac delta function, $\{\boldsymbol{\tau}_m[k]\}_{k=1}^{K}$ denote the K propagation times from the source to sensor m in seconds, *i.e.* the *echo delays* or Dirac locations and $\{\boldsymbol{c}_m[k]\}_{k=1}^{K}$ denote the *echo attenuations* or Dirac weights. In practical applications, continuous time-domain signals are not accessible. They are measured by sensors and discretized to be stored in a computer's memory. Let $\boldsymbol{x}_m \in \mathbb{R}^N$ denote N consecutive discrete samples collected by sensor m. Most measurement models assume that the impinging signal undergoes an ideal low-pass filter with frequency support $[-f_s/2, f_s/2]$ before being regularly sampled at the rate f_s in Hz. This is expressed by

$$\boldsymbol{x}_m[n] = (\phi * x_m)(n/f_s),\ n = 0, \dots, N-1 \tag{5.4}$$

where $\phi = \sin(\pi t)/\pi t$ is the classical sinc sampling Kernel. The continuous-time model (5.1) can then be approximated in two different ways, described in the next two sub-sections.

Discrete time-domain model

First, model (5.1) can be approximated in the discrete, finite-time domain. Let $\boldsymbol{h}_m \in \mathbb{R}^L$ and $\boldsymbol{s} \in \mathbb{R}^{N+L-1}$ denote discrete, sampled versions of the filter $h_m(t)$ and signal $s(t)$ respectively. We then have

$$\boldsymbol{x}_m[n] \approx (\boldsymbol{h}_m \star \boldsymbol{s})[n] \tag{5.5}$$

where the discrete finite convolution operator $\star$ between two vectors $\boldsymbol{u} \in \mathbb{R}^L$ and $\boldsymbol{v} \in \mathbb{R}^D$ ($L \leq D$) is defined by

$$(\boldsymbol{u} \star \boldsymbol{v})[n] = \sum_{j=0}^{L-1} u[j]\, v[L-1+n-j],\ n = 0, \ldots, D-L. \tag{5.6}$$

The following convenient matrix notation will be used:

$$\boldsymbol{u} \star \boldsymbol{v} = \mathit{Toep}_0(\boldsymbol{u})\boldsymbol{v} = \mathit{Toep}(\boldsymbol{v})\boldsymbol{u} = \tag{5.7}$$

$$\begin{bmatrix} u_L & \ldots & u_1 & 0 & \ldots & \ldots & 0 \\ 0 & u_L & \ldots & u_1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & \ldots & 0 & u_L & \ldots & u_1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_D \end{bmatrix} = \begin{bmatrix} v_L & v_{L-1} & \ldots & v_1 \\ v_{L+1} & v_L & \ddots & v_2 \\ \vdots & \ddots & \ddots & \vdots \\ v_D & v_{D-1} & \ldots & v_{D-L+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_L \end{bmatrix},$$

where $\mathit{Toep}_0(\boldsymbol{u}) \in \mathbb{R}^{(D-L+1)\times D}$ and $\mathit{Toep}(\boldsymbol{v}) \in \mathbb{R}^{(D-L+1)\times L}$. The validity of approximation (5.5) depends on the way $h_m(t)$ and $s(t)$ are sampled. In [187](Proposition 2), it is showed that if $s(t)$ is band-limited with maximum frequency lower than $f_s/2$ and if we let the number of samples N and the filter length L grow to infinity, then model (5.5) is **exact** for the following sampling schemes:

$$\boldsymbol{s}[n] = s(n/f_s),\ n \in \mathbb{Z} \tag{5.8}$$

$$\boldsymbol{h}_m[n] = (\phi * h_m)(n/f_s),\ n \in \mathbb{Z}. \tag{5.9}$$

Here, it is important to note that contrary to intuition, even in the idealized case where an infinite number of samples are available, the discrete-time filters $\{\boldsymbol{h}_m\}_{m=1}^M$ involved in the measurement model are *never* streams of Diracs, but non-sparse, infinite-length filters consisting of decimated combinations of sinc functions. This is illustrated in Fig. 5.1. Recovering the original Dirac positions and coefficients from finitely many samples of such filters is a challenging task in itself.

Discrete frequency-domain model

Alternatively, one may approximate model (5.1) in the discrete finite-frequency domain. Let $\hat{\boldsymbol{x}}_m \in \mathbb{C}^F$ denote the discrete Fourier transform (DFT) of $\boldsymbol{x}_m$, defined by

$$\hat{x}_m(f) = \mathrm{DFT}(\boldsymbol{x}_m) = \sum_{n=0}^{N-1} \boldsymbol{x}_m[n] e^{-2\pi j f n / f_s} \tag{5.10}$$

where f belongs to a set of F regularly-spaced frequencies $\mathbb{F} = \{f_1, \ldots, f_F\} \subset]0, f_s/2]$ in Hz, therefore is a discrete variable. We then have the following approximate model:

$$\hat{\boldsymbol{x}}_m[f] \approx \hat{\boldsymbol{h}}_m[f] \hat{\boldsymbol{s}}[f] \approx \left(\sum_{k=1}^{K} \boldsymbol{c}_m[k] e^{-2\pi j f \boldsymbol{\tau}_m[k]} \right) \boldsymbol{s}[f] \tag{5.11}$$

where $\hat{\boldsymbol{h}}_m \in \mathbb{C}^F$ and $\hat{\boldsymbol{s}} \in \mathbb{C}^F$ denote the DFT of $\boldsymbol{h}_m$ and $\boldsymbol{s}$, respectively. Two approximations are made in (5.11). First, the time-domain convolution between $\hat{\boldsymbol{h}}_m$ and $\hat{\boldsymbol{s}}$ has been transformed into a multiplication through the DFT. This would be exact for a circular convolution, but the actual model is a linear convolution between infinite and non periodic signals, resulting in an approximation error. Second, the formula used for $\hat{\boldsymbol{h}}_m$ in the right hand side of (5.11) is the one that would result from the discrete-time Fourier transform (DTFT) of $\boldsymbol{h}_m$ which would require infinitely many samples N to be calculated exactly, as opposed to the DFT. Note that the smoothing sinc kernel $\phi(t)$ does not impact this formula, since only frequencies below $f_s/2$ are considered. Importantly, both approximations in (5.11) become arbitrarily precise as the number of samples N grows to infinity.

While both the discrete-time model (5.5) and the discrete-frequency model (5.11) become increasingly accurate when N becomes large, the latter directly incorporates the variables of interest $\{\boldsymbol{c}_m[k], \boldsymbol{\tau}_m[k]\}_{m,k=1}^{M,K}$, as opposed to the former. In the remainder of this chapter, it will be assumed that $s(t)$ is bandlimited with maximum frequency less than $f_s/2$ and that N is sufficiently large such that both models hold very well. This is a reasonable assumption in audio applications, where sensors typically acquire tens of thousands of samples per second. Moreover, we focus on situations where sensor noise is negligible. Hence, the approximation signs will be dropped for convenience.

5.2 Existing methods in channel identification

All existing methods in blind channel identification rely on the discrete-time model (5.5) [196, 71, 2, 8, 102, 77, 87, 40, 99, 96]. The case of general emitted signals and finite filters was studied both methodologically and theoretically in the 90s [196, 71, 2], where two main categories of methods emerged, which we briefly review here, focusing on the two-channel ($M = 2$) case for simplicity. First, the so-called *subspace methods* rely on the estimation of a time-domain $MP \times MP$ covariance matrix where P is a time-window length that must be larger than the filters' length L [2]. The filters are estimated by spectral decomposition of this matrix. Second, the more common *cross-relation* (CR) methods rely on the observation that under noiseless conditions we have $\boldsymbol{h}_m \star \boldsymbol{x}_l - \boldsymbol{h}_l \star \boldsymbol{x}_m = \mathbf{0}_{N-L+1}$ for $l \neq m \in \{1, \dots, M\}$, by associativity of the convolution. A common approach is therefore to solve a minimization problem of the form:

$$\hat{\boldsymbol{h}}_1^*, \hat{\boldsymbol{h}}_2^* = \underset{\hat{\boldsymbol{h}}_1[1]=1}{\operatorname{argmin}} \left\| \mathit{Toep}(\hat{\boldsymbol{x}}_2)\hat{\boldsymbol{h}}_1 - \mathit{Toep}(\hat{\boldsymbol{x}}_1)\hat{\boldsymbol{h}}_2 \right\|_2^2, \tag{5.12}$$

which is a simple least-square problem. The constraint $\hat{h}_1(1) = 1$ is used to avoid the trivial solution $\hat{\boldsymbol{h}}_1 = \hat{\boldsymbol{h}}_2 = \mathbf{0}_L$. Alternatively, the normalization $\|\hat{\boldsymbol{h}}_1\|_2^2 + \|\hat{\boldsymbol{h}}_2\|_2^2 = 1$ can be used, leading to a minimum eigenvalue problem.

In the case of interest where the goal is to retrieve echo information from the filters, both subspace [77] and to a larger extent CR [8, 102, 87, 40] methods have been extended in order to handle sparse filters. This approach requires two independent steps: first estimating sparse filters, second retrieving echo locations and weights from them, typically using a peak-picking technique. Following the compressed sensing idea [28], sparsity is usually promoted using an ℓ_1-norm penalty term on the filters. For instance in [102], the following LASSO-type [179] problem is considered:

$$\hat{\boldsymbol{h}}_1^*, \hat{\boldsymbol{h}}_2^* = \underset{\hat{\boldsymbol{h}}_1(1)=1}{\operatorname{argmin}} \left\| \mathit{Toep}(\hat{\boldsymbol{x}}_2)\hat{\boldsymbol{h}}_1 - \mathit{Toep}(\hat{\boldsymbol{x}}_1)\hat{\boldsymbol{h}}_2 \right\|_2^2 + \lambda(\|\hat{\boldsymbol{h}}_1\|_1 + \|\hat{\boldsymbol{h}}_2\|_1) \tag{5.13}$$

and a Bayesian-learning method for the automatic inference of λ is proposed. Several other approaches relying on similar schemes [8, 87, 40] have been proposed.

Four important bottlenecks of discrete time methods for echo retrieval can be identified:

- Although they rely on sparsity-enforcing regularizers, the filters are strictly-speaking non-sparse in practice, due to the sinc kernel (Fig. 5.1). This general bottleneck of compressed sensing has been referred to as *basis mismatch* and was notably studied in [34]. In particular, the true peaks of the filters do **not** correspond to the true echoes (Fig. 5.1), even for $N \to \infty$. Though, most existing methods rely on peak-picking [87, 40].
- For the same reason, these methods are fundamentally *on-grid*, namely, they can only output echo locations which are integer multiple of the sampling period $1/f_s$. This prevents subsample resolution, which may be important in applications such as room shape reconstruction from audio signals [50].
- These methods strongly rely on the knowledge of the length L of the filters. However, due to the sinc kernel (Sec. 5.1), the true filters are always infinite.
- The dimension of the search space is $ML - 1$, which is much larger in practice than the actual number $2MK$ of unknown variables. This makes the methods computationally demanding and sometimes intractable for large filter lengths (typically in the tens of thousands for acoustic applications).

5.3 Off-grid echo retrieval by multichannel annihilation

In this section, we introduce a novel method for echo recovery that makes use of the discrete-frequency model (5.11) and overcomes a number of shortcomings of existing approaches. Namely, it works directly in the parameter space, it does not rely on the filters' length but on the number of echoes, and it enables exact off-grid recovery of echoes' locations and weights in the noiseless case. The approach relies on the FRI sampling paradigm introduced in [191]. This is the first time this paradigm is applied to blind channel identification.

5.3.1 The non-blind case

We start by considering the non-blind case where the emitted signal $\boldsymbol{s} \in \mathbb{C}^F$ in the discrete frequency domain is known. We further assume throughout the chapter that this signal is nonzero on the considered frequency grid $\mathbb{F} = \{f_1, \dots, f_F\}$. We can then transform the discrete-frequency model (5.11) by writing:

$$\hat{\boldsymbol{h}}_m[f] = \hat{\boldsymbol{x}}_m[f]\boldsymbol{z}[f] = \sum_{k=1}^{K} \boldsymbol{c}_m[k] e^{-2\pi j f \boldsymbol{\tau}_m[k]} \tag{5.14}$$

where the *Fourier-inverted* signal $\boldsymbol{z} \in \mathbb{C}^F$ is defined by $\boldsymbol{z}[f] = \hat{\boldsymbol{s}}[f]^{-1}$. Our goal is to estimate $\{\boldsymbol{c}_m[k], \boldsymbol{\tau}_m[k]\}_{k=1}^{K}$ from $\hat{\boldsymbol{h}}_m = \hat{\boldsymbol{x}}_m \odot \boldsymbol{z}_m$, where $\odot$ denotes the Hadamard product. If we take our frequency indexes $\mathbb{F}$ to be in arithmetic progression with step Δ_f, then the exponential sequence $\{e^{-2\pi j f_i \boldsymbol{\tau}_m[k]}\}_{i=1}^{F}$ is a geometric progression with ratio $\boldsymbol{r}_m[k] = e^{-2\pi j \Delta_f \boldsymbol{\tau}_m[k]}$ for each m, k. Hence, $\hat{\boldsymbol{h}}_m$ is a weighted sum of geometric progressions. This enables us to use the so called *annihilating filter* technique [175]. This technique is based on the observation that

$$[1, -w] \star [w^0, w^1, w^2, \dots, w^{F-1}] = \mathbf{0}_{F-1}, \tag{5.15}$$

for any $w \in \mathbb{C}$ and $F \in \mathbb{N}$. We deduce that if we define the filter $\boldsymbol{a}_m = [a_{m,0}, \dots, a_{m,K}] \in \mathbb{C}^{K+1}$ as the following discrete convolution[2] of K filters of size 2:

$$\boldsymbol{a}_m = [1, -r_{m,1}] \star [1, -r_{m,2}] \star \cdots \star [1, -r_{m,K-1}] \star [\mathbf{0}_{K-1}, 1, -r_{m,K}, \mathbf{0}_{K-1}], \tag{5.16}$$

then $\boldsymbol{a}_m$ is an *annihilating filter* for $\boldsymbol{h}_m$, *i.e.*, $\boldsymbol{a}_m \star \boldsymbol{h}_m = \mathbf{0}_{F-K}$. Importantly, the number of echoes K has to be known upfront in order to define $\boldsymbol{a}_m$. Let us now define the *polynomial representation* of filter $\boldsymbol{a}_m$ by:

$$P_{\boldsymbol{a}_m}[y] = \sum_{k=0}^{K} \boldsymbol{a}_m[k] y^k. \tag{5.17}$$

Because $\boldsymbol{a}_m$ is an annihilating filter for $\boldsymbol{h}_m$, it follows from the classical interpretation of convolution as polynomial multiplication that $P_{\boldsymbol{a}_m}$ has exactly K roots, which are the ratios $\{\boldsymbol{r}_m[k]\}_{k=1}^{K}$. Hence, once an annihilating filter $\boldsymbol{a}_m$ for $\boldsymbol{h}_m$ has been found, the Dirac locations

---

[2] The chained discrete convolutions in (5.16) have to be taken from right to left to be compatible with (5.6).

$\{\boldsymbol{\tau}_m[k]\}_{k=1}^{K}$ can be deduced by rooting $P_{\boldsymbol{a}_m}$. Once the roots are known, reconstructing the weights is a simple linear problem involving a Vandermonde matrix $\mathbf{V}(\boldsymbol{r}_m) \in \mathbb{C}^{F\times K}$ (type of a matrix that contains geometric progression in each row), obtained by writing (5.14) in matrix form:

$$\begin{bmatrix} \hat{\boldsymbol{h}}_m[f_1] \\ \hat{\boldsymbol{h}}_m[f_2] \\ \vdots \\ \hat{\boldsymbol{h}}_m[f_F] \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ (\boldsymbol{r}_m[1])^1) & (\boldsymbol{r}_m[2])^1 & \dots & (\boldsymbol{r}_m[K])^1 \\ \vdots & \vdots & \ddots & \vdots \\ (\boldsymbol{r}_m[1])^{F-1} & (\boldsymbol{r}_m[2])^{F-1} & \dots & (\boldsymbol{r}_m[K])^{F-1} \end{bmatrix} \mathbf{D}_m \begin{bmatrix} \boldsymbol{c}_m[1] \\ \boldsymbol{c}_m[2] \\ \vdots \\ \boldsymbol{c}_m[K] \end{bmatrix} = \mathbf{V}(\boldsymbol{r}_m)\mathbf{D}_m\boldsymbol{c}_m. \tag{5.18}$$

where $\mathbf{D}_m = \operatorname{diag}(e^{-2\pi i f_1 \boldsymbol{\tau}_m}) \in \mathbb{C}^{K\times K}$. The least-square solution of this system is given by $\boldsymbol{c}_m = \mathbf{D}_m^{-1}\mathbf{V}(\boldsymbol{r}_m)^\dagger \hat{\boldsymbol{h}}_m$ where $\{\cdot\}^\dagger$ denotes the Moore-Penrose pseudo inverse. For a rectangular matrix $\mathbf{M}$, its Moore-Penrose psaeudo inverse is given by $\mathbf{M}^\dagger = (\overline{\mathbf{M}}\mathbf{M})^{-1}\overline{\mathbf{M}}$, if $\mathbf{M}$ has linearly independent columns, or $\mathbf{M}^\dagger = \overline{\mathbf{M}}(\mathbf{M}\overline{\mathbf{M}})^{-1}$, if $\mathbf{M}$ has linearly independent rows.

In practice, since positive weights are sought, the phases of this complex vector are discarded. General FRI theory [191] tells us that $F \geq 2K+1$ is enough to uniquely recover the exact K Dirac locations and weights using this method in a noiseless setting. In other words, the original echo retrieval problem has been reduced to that of finding an annihilating filter for $\hat{\boldsymbol{h}}_m = \hat{\boldsymbol{x}}_m \odot \boldsymbol{z}$. In practice, this can be done by solving the following minimization problem for $m = 1,\dots,M$:

$$\boldsymbol{a}_m^* = \operatorname*{argmin}_{\|\boldsymbol{a}_m\|_2^2=1} \left\| \mathit{Toep}(\hat{\boldsymbol{x}}_m \odot \boldsymbol{z})\boldsymbol{a}_m \right\|_2^2, \tag{5.19}$$

where the unit norm constraint is used to avoid the trivial solution $\boldsymbol{a}_m = \mathbf{0}_{K+1}$. The solution of this problem is the eigenvector associated to the lowest eigenvalue of $\mathit{Toep}(\hat{\boldsymbol{x}}_m \odot \boldsymbol{z})$ (later referenced as *min_eig_vec* solved through singular value decomposition (SVD)). Assuming that the true $\boldsymbol{z}$ is given, that model (5.11) holds exactly and that $F \geq 2K+1$, this eigenvalue will be unique and equal to 0.

5.3.2 MULAN: an iterative scheme

In the blind echo retrieval problem of interest, the emitted signal $\boldsymbol{s}$ and hence $\boldsymbol{z}$ are unknown. To solve for all unknown variables jointly, we introduce the following non-convex optimization problem:

$$\boldsymbol{z}^*, \boldsymbol{a}_1^*, \dots, \boldsymbol{a}_M^* = \operatorname*{argmin}_{\|\boldsymbol{z}\|_2^2=\|\boldsymbol{a}_1\|_2^2=\dots=\|\boldsymbol{a}_M\|_2^2=1} \sum_{m=1}^{M} \left\| \mathit{Toep}(\hat{\boldsymbol{x}}_m \odot \boldsymbol{z})\boldsymbol{a}_m \right\|_2^2. \tag{5.20}$$

Our strategy to tackle this problem is by alternated minimization with respect to each variable. Minimization with respect to each $\boldsymbol{a}_m$ is already covered by the previous section. Minimization with respect to $\boldsymbol{z}$ is also a minimum eigenvalue problem, since the cost function $C(\boldsymbol{z}, \boldsymbol{a})$ can

Algorithm 3 MULAN (MULtichannel ANnihilation)

Input: Frequency-domain multichannel measurements $\{\hat{\boldsymbol{x}}_{1:M}(f); f \in \mathbb{F}\}$ computed via DFT (5.10); *max_iter*; *conv_thresh*.

Output: Echo locations and weights $\{\boldsymbol{\tau}_m[k], \boldsymbol{c}_m[k]\}_{m,k=1}^{M,K}$.

```
iter := 0; z := random();      // i.i.d. standard complex Gaussian in $\mathbb{C}^F$
repeat
    iter := iter + 1;
    for $m = 1 \to M$ do: $\boldsymbol{a}_m := \text{min\_eig\_vec}(Toep(\hat{\boldsymbol{x}}_m \odot \boldsymbol{z}))$; end for
    $\boldsymbol{z} := \text{min\_eig\_vec}(\mathbf{Q})$;      // See eq. (5.22)
until iter=max_iter or $\mathrm{C}(\boldsymbol{z}, \boldsymbol{a})$ decreased by less than conv_thresh      // See eq. (5.21)
for $m = 1 \to M$ do
    $\boldsymbol{r}_m := \text{roots}(P_{\boldsymbol{a}_m})$; $\boldsymbol{\tau}_m := -\arg(\boldsymbol{r}_m)/(2\pi\Delta_f)$; $\boldsymbol{c}_m := \text{abs}(\mathbf{D}_m^{-1}\mathbf{V}(\boldsymbol{r}_m)^\dagger \hat{\boldsymbol{h}}_m)$;      // Sec. 5.3.1
end for
return shifted and scaled $\{\boldsymbol{\tau}_m[k], \boldsymbol{c}_m[k]\}_{m,k=1}^{M,K}$;      // See Sec. 5.3.3
```

be rewritten:

$$C(\boldsymbol{z}, \boldsymbol{a}) = \sum_{m=1}^{M} \left\| Toep(\hat{\boldsymbol{x}}_m \odot \boldsymbol{z})\boldsymbol{a}_m \right\|_2^2 = \sum_{m=1}^{M} \left\| Toep_0(\boldsymbol{a}_m)\mathrm{diag}(\hat{\boldsymbol{x}}_m)\boldsymbol{z} \right\|_2^2 = \|\mathbf{Q}\boldsymbol{z}\|_2^2, \tag{5.21}$$

$$\text{where } \mathbf{Q} = [Toep_0(\boldsymbol{a}_1)\mathrm{diag}(\hat{\boldsymbol{x}}_1); \ldots; Toep_0(\boldsymbol{a}_M)\mathrm{diag}(\hat{\boldsymbol{x}}_M)] \in \mathbb{C}^{M(K+1)\times F} \tag{5.22}$$

and $[\cdot;\cdot]$ denotes vertical concatenation. If the algorithm succeeds in bringing down the cost function to zero, it means that appropriate annihilating filters have been found for all channels for a given Fourier-inverted signal $\boldsymbol{z}$, and the locations and weights of all Diracs can be recovered up to a global shift of locations and global scaling of attenuations. We call this method MULAN for *MULtichannel ANnihilation.* Pseudo-code for the algorithm is given in Alg. 3. Since (5.20) is non-convex, the alternate minimization scheme is at best guaranteed to converge to a stationary point of the cost-function $C(\boldsymbol{z}, \boldsymbol{a})$. To alleviate this issue, we propose to initialize the method multiple times with random values of $\boldsymbol{z}$ and only keep the run with lowest final $C(\boldsymbol{z}, \boldsymbol{a})$.

5.3.3 Identifiability and ambiguities

The identifiability of blind channel identification for general discrete filters and signals has been studied some time ago [196]. It is known that the filters $\{\boldsymbol{h}_m\}_{m=1}^{M}$ cannot be recovered if their polynomial representations admit at least a common root or if the polynomial representation of the emitted signal $\hat{\boldsymbol{s}}$ has less than $2L+1$ roots. The latter is ruled out if the emitted signal has a rich enough spectral content (enough nonzero frequencies) which is usually the case for natural signals. The former has at least one consequence in our case: the problem is unidentifiable if the observed signals are scaled and delayed versions of each other, which may happen in practice. While other common roots may appear in the general setting, it is important to note that MULAN restricts the search of filters to those which are linear combinations of geometrical series in the frequency domain. There is no complete theoretical study on common roots in this case, to the best of our knowledge. The authors of [38] theoretically studied blind deconvolution of sparse signals, but their results do not apply here since our filters are not sparse (see Sec. 5.2). Another well-known ambiguity is that the filters can only be recovered up to a global time-shift and scaling, because a converse shifting and scaling of the emitted signal yields the same observations. We handle this by adopting the convention $\boldsymbol{\tau}_1[1] = 0$ and $\boldsymbol{c}_1[1] = 1$. Additionally, we assume that all echoes are located in the first half of temporal filters to avoid time-wrapping ambiguities. Finally, the proposed MULAN algorithm has an extra specific ambiguity. It can be easily shown that multiplying the roots of all polynomials $\{P_{\boldsymbol{a}_m}\}_{m=1}^{M}$ by a complex scalar γ while dividing the Fourier-inverted signal $\boldsymbol{z}$ element-wise by a geometric series of ratio γ does not change the cost function $C(\boldsymbol{z}, \boldsymbol{a})$. This can be handled by rescaling the roots of all annihilating filters to have unit modulus at each iteration. However, since only the complex arguments of the roots are used in the end, this appeared to be unnecessary in our experiments.

5.4 Experiments

5.4.1 On-grid vs. off-grid echo retrieval

We first emphasize the specific ability of the proposed method to recover echo locations off-grid by comparing it to conventional on-grid methods on a simulated room-acoustic scenario and on an artificial scenario with truly sparse discrete filters for reference. For the room-acoustic scenario, there is a point source emitting speech from the TIMIT dataset [59], and $M = 2$ microphones are randomly placed inside 100 random shoe-box rooms whose sizes vary from 4m × 6m × 8m to 5m × 7m × 9m. Simulations were performed using the *pyroomacoustics* library [163]. The absorption coefficient of each surface of the room is set to 0.2 to arrive to moderately damped early reflections. Only first-order reflections on the 6 surfaces and the direct path are simulated, resulting in $K = 7$ echoes per channel and filters shorter than 50 ms. For each experiment, it was ensured that the minimum separation of echoes was 1ms. The filters are simulated in the continuous-time domain using the image-source method [10]. They are then smoothed, sampled and convolved with the source signal at $f_s = 16$kHz according to the measurement model described in Sec. 5.1. The ground-truth echo locations and weights are saved in the time-domain before smoothing and are hence off-grid. The M-channel input signals used are 0.25s long, *i.e.*, $N = 0.25 f_s = 4000$ samples. On the other hand, for the artificial scenario, the speech source was discretely convolved with sparse filters of similar length with $K = 7$ nonzero elements each resulting in $N = 4000$ samples of M-channel observations. The ground-truth echo locations and weights are hence on-grid in this case. All weights take values between 0 and 1.

For MULAN, the DFT (eq. 5.10) is applied to each input signal using a grid $\mathbb{F}$ of $F = 401$ regularly spaced frequencies between 200 Hz and 2000 Hz. Such a choice of the frequency range avoids low-frequency bands which are often noisy in real scenario, while focusing on a typical spectral range for speech, but it can be easily adapted depending on the application. An odd number of frequencies was chosen, since it has proven to be a good practice [19]. We use 20 random initializations as a good compromise between global convergence and computing time, *max_iter*= 1000 and *conv_thresh*= 0.1%. The two baseline methods chosen are CR [196] as described in (5.12) and its LASSO-type extension [102] as described in (5.13). The filters' lengths L were always set to the true lengths (which never exceed $0.05 f_s$) and the sparsity parameter λ for LASSO was manually set to $\lambda = 10^{-3}$, which empirically showed best performance among the choices $\{10^{-6}, 10^{-5}, \ldots, 10^{2}\}$, although any value below 10^{-2} showed similar performance.

We used two distinct metrics to evaluate Dirac location estimation and Dirac weight estimation. For the first one, a test is counted as successful if the root mean squared error (RMSE) of the $7 \times 2 = 14$ Dirac locations is below 1 sample ($1/f_s$ seconds), and the success rate out of 100 tests is provided. This metric only counts fully successful channel recovery and penalizes tests where some Diracs are missed or completely off. For the second one, we provide the weight RMSE of successful tests only. This is to avoid counting weights estimated at wrong Dirac locations. These metrics for 100 on- and off-grid tests and all three methods are showed in Table 5.1. We can see that for the on-grid case, both CR and MULAN perform well, CR even achieving more location recoveries than MULAN. This is not too surprising since CR is based on the on-grid artificial model, while MULAN uses an off-grid model. We observed that LASSO struggles with the proximity of Diracs and did not perform as well. In terms of weight estimation MULAN yields errors which are 2 to 3 orders of magnitudes smaller than the two competing methods, which is very encouraging. In the more realistic off-grid scenario, we observed that localization errors of CR and LASSO drastically degrades with almost no successful channel estimation. Meanwhile, MULAN achieves near-exact full recovery of locations and weights in 70 out of 100 tests.

5.4.2 Influence of *K*, *M*, *F* on recovery rate

We now conduct further experiments to check the influence of parameters K, M and F on the ability of MULAN to fully recover Dirac locations and weights off-grid. We show results with 20 random initializations, $F = 201$ or $F = 401$ in the same frequency range as before, $M \in \{2, \ldots, 7\}$ and $K \in \{2, \ldots, 7\}$. The following RMSE thresholds were defined for success of recovery: 1 sample for locations as before and 10^{-2} for weights. 100 experiments were performed for every parameter set. Results for $F = 201$ can be seen in Figures 5.2 and 5.3, and for $F = 401$ in Figures 5.4 and 5.5. As can be seen, a higher recovery rate is generally observed when fewer echoes are present and more frequencies are used. On the other hand, the number of sensors does not significantly affect recovery performance. This is expected since $\mathcal{O}(KM)$ parameters are estimated from $\mathcal{O}(MF)$ observations, so by increasing the number of sensors we increase the search space. Increasing the number of random initializations also showed to increase success by alleviating the non-convexity of the problem, at the cost of an increased computational requirement.

case	method	full location recovery	weight RMSE
on-grid	CR [196]	**92** %	0.0390
	LASSO [102]	13 %	0.155
	MULAN (proposed)	59 %	**0.00016**
off-grid	CR [196]	1%	0.0442
	LASSO [102]	2%	0.0346
	MULAN (proposed)	**70** %	**0.00048**

Table 5.1 – Ratio of full Dirac location recovery (RMSE < 1 sample = $1/f_s$ seconds) and weight RMSE (successful cases only) for three methods over 100 on-grid and 100 off-grid tests. Weights take values between 0 and 1.

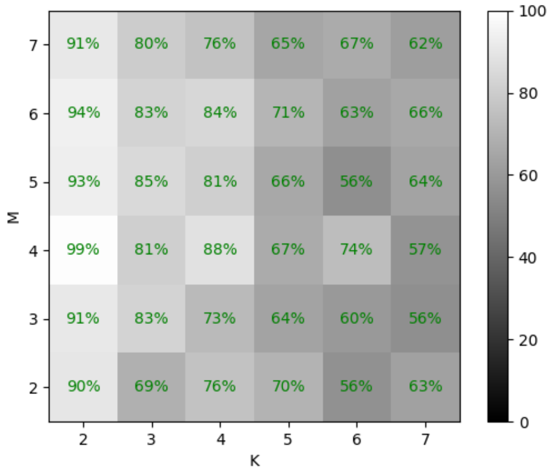


Figure 5.2 – Rate of location retrieval for $F = 201$.

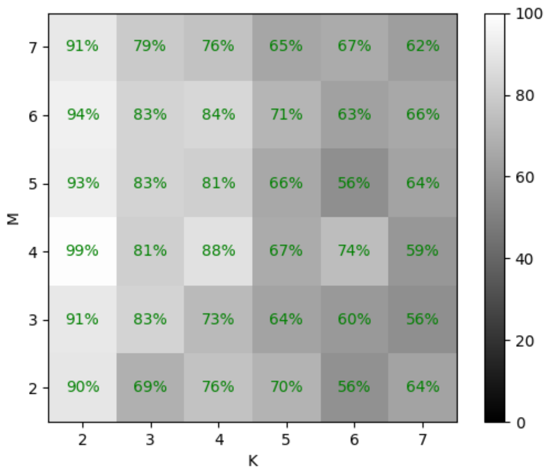


Figure 5.3 – Rate of weight retrieval for $F = 201$.

5.4.3 A discussion on the minimum separation of Diracs

Due to the fact that the proposed algorithm highly relies on the finite rate of innovation theory, the minimal separation of Diracs is determined by two key factors: the bandwidth and the amount of noise. In [19] the limits on the recovery of a Dirac with parameters (c, τ) are given with Cramér-Rao bound:

$$\frac{\Delta\tau}{T} \geq \frac{1}{\pi}\sqrt{\frac{3BT}{N\left(B^2T^2-1\right)}} \cdot \mathrm{PSNR}^{-1/2}, \tag{5.23}$$

where T is the periodicity of the signal (or length of the finite-length signal), B is the bandwidth of the measuring device, $PNSR$ is Peak Signal-to-Noise Ratio with $PSNR = \frac{|c|^2}{\sigma^2}$, N is the number of measurements and σ^2 is the noise power. Within the scope of this chapter we have taken Δt to be 1ms. Further evaluation of the recovery of these bounds is left for future work.

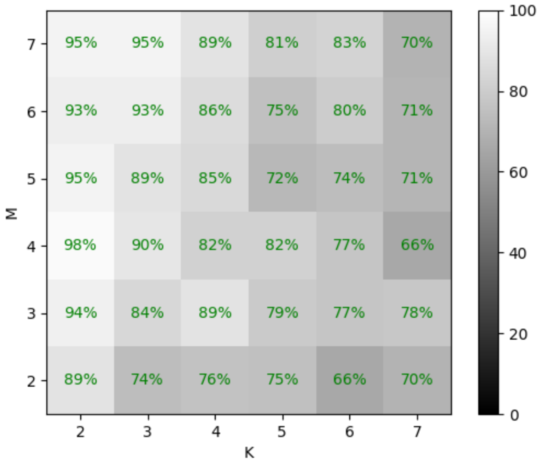


Figure 5.4 – Rate of location retrieval for $F = 401$.

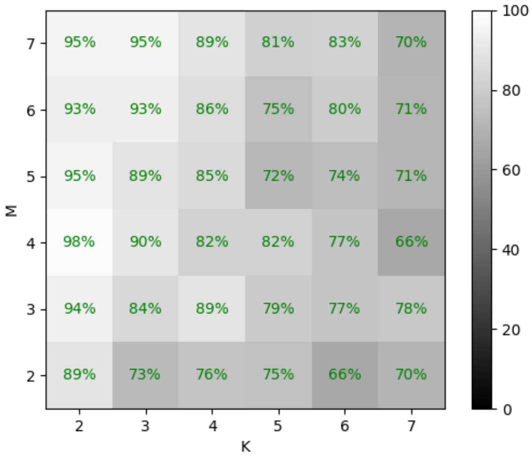


Figure 5.5 – Rate of weight retrieval for $nF = 401$.

5.5 Conclusion

This chapter introduced the first method enabling blind and off-grid recovery of echo locations and weights from discrete-time multichannel measurements. The code can be found on github[3]. In the next chapter we develop the extension of this approach to a multichannel noisy setting and also cover the case when the number of Diracs in our sparse representation is underestimated.

---

[3] https://github.com/epfl-lts2/mulan

6 Estimating Early Acoustic Echoes from Noisy Speech with Multichannel Structured Low-Rank Optimization[1]

In the previous chapter, our model has mostly relied on the clean data from a simulation with a known true number of Diracs. As we have discussed earlier, models relying on the Finite Rate of Innovation theory are known to show degraded performance in the presence of noise. In this chapter we extend our approach to more realistic cases. The scenarios that we will be focusing on include the case when the data is noisy and also the case when we want to retrieve the top K reflections, which is in the literature mostly know as the case of *underfitting* [39] (since we assume that our model is of a lower complexity than it truly is), which relaxes the constraint of knowing the level of sparsity upfront.

The main premise for our observations will be the fact that when the data is recorded simultaneously by multiple microphones, all the channels are correlated by the common variable - the input of the system [196], that is - the sound emitted by the source. This setting is common for blind deconvolution problem. Although there has been some significant work in 2018 on blind deconvolution with sparse priors [199, 101, 7, 6], most of the solutions are still relying on the on-grid recovery.

As in the previous chapter, we will solve the Dirac train recovery problem off-grid in time domain by moving to the on-grid problem definition in the frequency domain, which should enable arbitrary positions of Dirac pulses in the temporal/spatial domain.

6.1 Cadzow denoising algorithm

The main contribution of this piece of research is including the Cadzow denoising algorithm [25] into the story. This algorithm has been used when the Finite Rate of Innovation theory was extended to noisy cases [19]. The main contribution of this algorithm is an alternating scheme

---

[1] Work done with Antoine Deleforge from INRIA.

Algorithm 4 Cadzow denoising algorithm

Input: Noisy Toeplitz matrix $\tilde{\mathbf{T}}$, *tolerance*, *max_iter*
Output: Denoised Toeplitz matrix $\mathbf{T}$
***Note**: here we use the $_K$ notation to denote the K-rank matrix

```
$\mathbf{T}^{(1)} = \tilde{\mathbf{T}}$;
repeat
    $iter := iter + 1$;
    $\mathbf{U\Sigma V}^* = \mathbf{T}^{(iter)}$;      // singular value decomposition
    $\mathbf{T}_K^{(iter)} = \mathbf{U\Sigma}_K\mathbf{V}^*$;      // will be noted as a projection $\mathcal{P}_{\mathcal{R}_K}$
    $[h, w] = \text{size}(\mathbf{T}_K^{(iter)})$;
    for $i = -h \to w$ do      // will be noted as a projection $\mathcal{P}_{\mathcal{T}}$
        $\text{diag}(\mathbf{T}^{(iter+1)}, i) = \text{mean}(\text{diag}(\mathbf{T}^{(iter)}, i))$; // Toeplitization
    end for
until $iter < max\_iter$ and $\|\mathbf{T}^{(iter)} - \mathbf{T}^{(iter-1)}\| > tolerance$
return $\mathbf{T} = \mathbf{T}^{(iter)}$;
```

that can be used for removing the noise from data or data enhancement. We need to alternate between the properties that our data is known or hypothesized to possess, which would in our case be: rank K for the measurement matrix and its Toeplitz structure. The pseudocode of the original formulation of the algorithm is given in Algorithm 4.

The level of sparsity, K, is usually assumed to be known upfront and [19] also discusses what happens in cases when the number of Diracs is over- or underestimated. In an overestimated case, some spurious Diracs are retrieved which can be alleviated by introducing the minimum threshold for Diracs' weight. On the other hand, for the underestimated case, usually the Diracs with the K highest weights are retrieved.

This chapter is an extension of the previous, so we keep the definition of the data model for the continuous and observation (discrete) case.

6.2 Data model and Cadzow denoising

The original formulation of Cadzow denoising is non-convex and has no guarantees on the convergence of the algorithm. In a paper from 2014, Condat et al. [39] redefine this denoising into a convex formulation by denoising a weighted ℓ_2-norm of a Toeplitz structured matrix that has number of columns greater than the level of sparsity.

As was previously defined in (eq. 5.14), in the context of acoustic echoes retrieval, M filters that correspond to the room impulse response at the positions of our sensors have the following form in the frequency domain representation:

$$\hat{\boldsymbol{h}}_m[f] = \sum_{k=1}^{K} \boldsymbol{c}_m[k] e^{-2\pi j f \boldsymbol{\tau}_m[k]} \tag{6.1}$$

where $\{\boldsymbol{c}_m[k], \boldsymbol{\tau}_m[k]\}_{m=1}^{M}$ are the variables of interest. Following [39], this form can be equivalently enforced by the following constraint on $\hat{\boldsymbol{h}}_m$:

$$\operatorname{rank}(\operatorname{Toep}_P(\hat{\boldsymbol{h}}_m)) \leq K, \tag{6.2}$$

where for any $K \leq P < D/2$, the Toep_P operator maps a vector $\boldsymbol{v} \in \mathbb{C}^D$ to a matrix in $\mathbb{C}^{(D-P)\times(P+1)}$ as follows:

$$\operatorname{Toep}_P(\boldsymbol{v}) = \begin{bmatrix} v_{P+1} & v_P & \cdots & v_1 \\ v_{P+2} & v_{P+1} & \ddots & v_2 \\ \vdots & \ddots & \ddots & \vdots \\ v_D & v_{D-1} & \cdots & v_{D-P} \end{bmatrix}. \tag{6.3}$$

6.3 Weighted cross-relation for a multichannel case

6.3.1 Binaural (2-channel) case

Although built as an extension of MULAN, MUSHU will strongly rely on a cross-relation formulation of the problem. To make the algorithm more robust to both model and observation noise, a natural idea is to use the cross-relation cost function. We focus for now on the $M = 2$ (2-channel) case. In discrete time domain, cross-relation methods aim at minimizing the following cost function:

$$\|\boldsymbol{h}_1 \star \boldsymbol{x}_2 - \boldsymbol{h}_2 \star \boldsymbol{x}_1\|_2^2 \tag{6.4}$$

where $\boldsymbol{h}_m \in \mathbb{R}^L$ and $\boldsymbol{x}_m \in \mathbb{R}^N$ $(m = 1,2)$ are the discrete time-domain filters and signals, respectively. In the frequency domain, the following analog cost-function can be defined:

$$\|\hat{\boldsymbol{h}}_1 \odot \hat{\boldsymbol{x}}_2 - \hat{\boldsymbol{h}}_2 \odot \hat{\boldsymbol{x}}_1\|_2^2 \tag{6.5}$$

where $\odot$ denotes the Hadamard product (element-wise multiplication), $\hat{\boldsymbol{h}}_m \in \mathbb{C}^F$ and $\hat{\boldsymbol{x}}_m \in \mathbb{C}^F$ $(m = 1,2)$ are the discrete Fourier transform (DFT) of $\hat{\boldsymbol{h}}_m$ and $\hat{\boldsymbol{x}}_m$, respectively. Note that minimizing (6.4) or (6.5) is not equivalent. Indeed, (6.4) implicitly includes a strong constraint: the filters are of size L. This constraint is released in (6.5), making the latter highly ill-posed (for instance, $\hat{\boldsymbol{h}}_1 = \mathbf{1}$ and $\hat{\boldsymbol{h}}_2 = \hat{\boldsymbol{x}}_2 \oslash \hat{\boldsymbol{x}}_1$ is always a solution, where $\oslash$ denotes element-wise division).

We hence consider the following minimization problem:

$$\begin{array}{ll} \underset{\hat{\boldsymbol{h}}_1,\hat{\boldsymbol{h}}_2\in\mathbb{C}^F}{\operatorname{argmin}} & \|\hat{\boldsymbol{h}}_1 \odot \hat{\boldsymbol{x}}_2 - \hat{\boldsymbol{h}}_2 \odot \hat{\boldsymbol{x}}_1\|_2^2 \\ \text{such that} & \operatorname{rank}(\operatorname{Toep}_P(\hat{\boldsymbol{h}}_m)) \leq K,\ m = 1,2 \\ & \hat{\boldsymbol{h}}_1[1] = 1, \end{array} \tag{6.6}$$

where $\hat{\boldsymbol{h}}_1[1] = 1$ is here to avoid the trivial solution $\hat{\boldsymbol{h}}_1 = \hat{\boldsymbol{h}}_2 = \mathbf{0}$. Again following [39], (6.6) can be equivalently rewritten in matrix form as follows:

$$\begin{array}{ll} \underset{\mathbf{H}_1,\mathbf{H}_2 \in \mathbb{C}^{(F-P)\times(P+1)}}{\operatorname{argmin}} & \|\mathbf{H}_1 \odot \mathbf{X}_2 - \mathbf{H}_2 \odot \mathbf{X}_1\|_W^2 \\ \text{such that} & \mathbf{H}_m = \operatorname{Toep}_P(\hat{\boldsymbol{h}}_m), \\ & \operatorname{rank}(\mathbf{H}_m) \le K,\ m = 1,2 \\ & \mathbf{H}_m \in \mathcal{T}_{F,P},\ m = 1,2 \\ & \mathbf{H}_1[1, P+1] = 1 \end{array} \tag{6.7}$$

where $\mathcal{T}_{F,P}$ denotes the set of $(F-P)\times(P+1)$ Toeplitz matrices (F is the cardinality of the frequency set where the data was observed: $F = |\mathbb{F}|$), $\mathbf{X}_m = \operatorname{Toep}_P(\hat{\boldsymbol{x}}_m)$ for $m = \{1,2\}$ (we avoid using the hat sign above matrices for convenience). The weighted Frobenius norm $\|\cdot\|_W$ of a matrix $\mathbf{A} \in \mathbb{C}^{(F-P)\times(P+1)}$ is defined by:

$$\|\mathbf{A}\|_W^2 = \sum_{i=1}^{F-P}\sum_{j=1}^{P+1} \mathbf{W}[i,j]|\mathbf{A}[i,j]|^2 \tag{6.8}$$

and the following weights $\mathbf{W} \in \mathbb{C}^{(F-P)\times(P+1)}$ are used [39]:

$$\mathbf{W}[i,j] = \begin{cases} 1/(i-j+P+1) \text{ if } i-j \le 0, \\ 1/(P+1) \text{ if } 1 \le i-j \le F-2P-1, \\ 1/(j-i+F-P) \text{ if } i-j \ge F-2P. \end{cases} \tag{6.9}$$

We propose to alternately minimize (6.7) with respect to $\mathbf{H}_1$ and $\mathbf{H}_2$ using Cadzow denoising. For a fixed $\mathbf{H}_2$, minimization with respect to $\mathbf{H}_1$ can be written as follows:

$$\begin{array}{ll} \underset{\mathbf{H}_1 \in \mathbb{C}^{(F-P)\times(P+1)}}{\operatorname{argmin}} & \|\mathbf{H}_1 - \mathbf{H}_2 \odot \mathbf{X}_1 \oslash \mathbf{X}_2\|_{\mathbf{W}\odot|\mathbf{X}_2|^{\odot 2}}^2 \\ \text{such that} & \operatorname{rank}(\mathbf{H}_1) \le K \\ & \mathbf{H}_1 \in \mathcal{T}_{F,P}. \end{array} \tag{6.10}$$

Note that the third constraint has been dropped because it suffices to have $\mathbf{H}_2 \neq \mathbf{0}$ to avoid the trivial solution. A proof of the equivalence between (eq. 6.7) and (eq. 6.10) is given in Appendix A. (6.10) has the form of a *structured low-rank approximation* (SLRA) problem with target $\mathbf{H}_2 \odot \mathbf{X}_1 \oslash \mathbf{X}_2$ for which Cadzow denoising or the method proposed in [39] can be applied. Given an appropriate initialization of $\mathbf{H}_2$, we propose to alternate between one Cadzow iteration to update $\mathbf{H}_1$ and one Cadzow iteration to update $\mathbf{H}_2$ (analogously) until convergence. We terminate the algorithm when almost no progress is made between consecutive iterations.

6.3.2 Multichannel case

Let's define the set of channel indices as $\mathbb{M} = \{1,2,\ldots,M\}$. In order to explore the potential of the multichannel setting, we expand the definition of the algorithm (eq. 6.10) to a case in which we have M microphones:

$$\begin{aligned}\underset{\mathbf{H}_1\in\mathbb{C}^{(F-P)\times(P+1)}}{\operatorname{argmin}} \quad & \|\mathbf{H}_1 - \mathbf{V}_{\mathbb{M}/1}^{\odot -1} \odot \mathbf{X}_1 \odot \textstyle\sum_{m=2}^{M} \mathbf{H}_m \odot \mathbf{X}_m^* \|^2_{\mathbf{W}\odot\mathbf{V}_{\mathbb{M}/1}} \\ \text{such that} \quad & \operatorname{rank}(\mathbf{H}_1) \le K \\ & \mathbf{H}_1 \in \mathscr{T}_{F,P}.\end{aligned} \tag{6.11}$$

where $\mathbf{V}_{\mathbb{M}/1} = \sum_{m=2}^{M} |\mathbf{X}_m|^{\odot 2}$. The definition of $\mathbf{V}_{\mathbb{M}/m}$ where $m \in \mathbb{M}$ follows naturally. With an appropriate initialization with GCC-PHAT, the algorithm converges to the true filters $\mathbf{H}_m$ by alternatively solving the problem for all of the different m's.

Generalization of Cadzow denoising upgraded to multi-channel case: In a multichannel case we need to minimize a cross-relation objective function for all the possible pairs of M microphones. In order to avoid the central part of the equation, the global objective function will consist of the concatenation of all the cross-relation objective functions of all the microphone pairs:

$$\| [\mathbf{A} \mid \mathbf{B}] \|_F^2 = \|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2, \tag{6.12}$$

where | is used for matrix concatenation.

We will have one optimization problem to solve for every microphone. Therefore, the Cadzow formulation generalized for M microphone case gives the following objective function for each microphone (here we give the definition for the first channel and all the definition for all the other channels follows analogously):

$$\begin{aligned}O(m=1) = \sum_{m=2}^{M} \|\mathbf{H}_1\|^2_{\mathbf{W}\odot|\mathbf{X}_m|^{\odot 2}} - 2\,\Re\Big(\sum\sum \mathbf{H}_1^* \odot \mathbf{X}_1 \odot \mathbf{W} \odot \sum_{m=2}^{M} \mathbf{H}_m \oslash \mathbf{X}_m \odot |\mathbf{X}_m|^{\odot 2}\Big) + \\ + \sum_{m=2}^{M} \|\mathbf{H}_m \oslash \mathbf{X}_m \odot \mathbf{X}_1\|^2_{\mathbf{W}\odot|\mathbf{X}_m|^{\odot 2}}.\end{aligned} \tag{6.13}$$

We introduce the following sum: $\mathbf{V}_i = \sum_{j\in\mathbb{M}\setminus i} |\mathbf{X}_j|^{\odot 2}$. Finally, for $m \in \mathbb{M}$ we have:

$$\begin{aligned}O(m) = \|\mathbf{H}_m - \frac{\mathbf{X}_m}{\mathbf{V}_m} \sum_{i\in\mathbb{M}\setminus m} \mathbf{H}_i \oslash \mathbf{X}_i \odot |\mathbf{X}_i|^{\odot 2} \|^2_{\mathbf{W}\odot\mathbf{V}_m} = \\ = \|\mathbf{H}_m - \frac{\mathbf{X}_m}{\mathbf{V}_m} \odot \sum_{i\in\mathbb{M}\setminus m} \mathbf{H}_i \odot \mathbf{X}_i^* \|^2_{\mathbf{W}\odot\mathbf{V}_m}.\end{aligned} \tag{6.14}$$

Definition of the algorithm updates: As explained beforehand, we solve one optimization problem per filter $\mathbf{H}_m$, $m \in \mathbb{M}$. Unlike in the [39], we change the updates in our algorithm by taking into account the previous estimation of the filter $\mathbf{H}_m$:

$$\mathbf{H}_m^{(l+1)} = \mathscr{P}_{\mathscr{R}_K}\Big(\mathbf{S}_m^{(l)} + \gamma(\mathbf{H}_m^{(l)} - \mathbf{S}^{(l)}) - \mu \mathbf{V}_{\mathbb{M}/m} \odot \mathbf{W} \odot (\mathbf{H}_m^{(l)} - \tilde{\mathbf{H}}_m)\Big) \tag{6.15}$$

$$\mathbf{S}_m^{(l+1)} = \mathbf{S}_m^{(l)} - \mathbf{H}_m^{(l+1)} + \mathscr{P}_{\mathscr{T}}(2\mathbf{H}_m^{(l+1)} - \mathbf{S}_m^{(l)}) \tag{6.16}$$

where $\mathscr{P}_{\mathscr{R}_K}$ is a projection to the space of rank-K matrices, $\mathscr{P}_{\mathscr{T}}$ is a projection to the space of Toeplitz matrices, l is the iteration index, $\mathbf{H}_m$ are estimated filter Toeplitz matrices and $\mathbf{S}_m$ are helper matrices. The initial values are: $\mathbf{H}_m^{(1)}$ gets computed from the Generalized Cross Correlation with Phase Transform (GCC-PHAT) [85], $\tilde{\mathbf{H}}_m = \mathbf{V}_{\mathbb{M}/m}^{\odot -1} \odot \mathbf{X}_m \odot \sum_{j \in \mathbb{M}, j \neq m} \mathbf{H}_j \odot \mathbf{X}_j^*$ (the minimizer of eq. 6.11) and $\mathbf{S}_m^{(1)} = \tilde{\mathbf{H}}_m$. All the Toeplitz matrices are of width P.

To ensure the convexity of the optimization problem, inspired by the indications from authors of [39] that the convergence of denoising can be ensured by decreasing the size of γ and μ, we halve the gradient descent step size μ every time the objective function tends to increase.

6.4 An alternating projections algorithm: MUSHU

Although not a particular acronym, the name of the algorithm *MUSHU* is given according to the Disney movie character that accompanies Mulan on her journeys, since this is an extension of the Mulan algorithm.

For this implementation we have changed our initialization scheme to GCC-PHAT [85] for the estimation of the Direction of Arrival of the initial Dirac in all the channels. We have used the implementation of GCC-PHAT available in the *pyroomacoustics* software package [163].

Outer iterations are related to the indexing of the repetitions of the whole algorithm and the inner iterations are related to the iterations of the upgraded Cadzow algorithm with the algorithm updates defined in 6.15 and 6.16.

The global cost function of our problem is defined in the following way:

$$C(\mathbf{H}_1, ..., \mathbf{H}_M) = \sum_{m \in \mathbb{M}} \sum_{m \in \mathbb{M}, n \neq m} \|\mathbf{H}_n \odot \mathbf{X}_m - \mathbf{H}_m \odot \mathbf{X}_n\|_W^2. \tag{6.17}$$

Finally, for each one of the estimated and denoised matrices $\mathbf{H}_m$, $m \in \mathbb{M}$ we find a corresponding annihilating filter $\boldsymbol{a}_m$. All this will lead to the retrieval of the locations and weights of the Diracs as shown in the pseudocode of the Algorithm 5.

6.5 Discussion on results and data collection

Since the idea behind this algorithm was to extend the original *MULAN* algorithm and apply it to real data, we have started to explore available databases of room impulse responses with

Algorithm 5 MUSHU algorithm

Input: Frequency-domain multichannel measurements $\{\hat{\boldsymbol{x}}_{1:M}[f]; f \in \mathbb{F}\}$ computed via DFT (5.10); *max_iter_outer*; *max_iter_inner*; *conv_thresh.*

Output: Echo locations and weights $\{\boldsymbol{\tau}_m[k], \boldsymbol{c}_m[k]\}_{m,k=1}^{M,K}$.

```
iter_outer := 0;
for $m = 1 \to M$ do
    $\mathbf{H}_m^{(1)} :=$ gccphat($\hat{\boldsymbol{x}}_{1:M}$);
end for
repeat
    iter_outer := iter_outer + 1;
    for $m = 1 \to M$ do
        iter_inner := 0;
        repeat
            Update $\mathbf{H}_m^{(iter\_outer)}$ with Cadzow denoising upgraded (eq. 6.15) and (eq. 6.16)
        until iter_inner=max_iter_inner
    end for
    $C^{(iter\_outer)} = C(\mathbf{H}_1^{(iter\_outer)}, \dots, \mathbf{H}_M^{(iter\_outer)})$;
until iter_outer=max_iter_outer or $|C^{(iter\_outer)} - C^{(iter\_outer-1)}| <$ conv_thresh
for $m = 1 \to M$ do
    $\boldsymbol{a}_m :=$ min_eig_vec($\mathbf{H}_m$);
    $\boldsymbol{r}_m :=$ roots($P_{\boldsymbol{a}_m}$); $\boldsymbol{\tau}_m := -\arg(\boldsymbol{r}_m)/(2\pi\Delta_f)$; $\boldsymbol{c}_m :=$ abs($\mathbf{D}_m^{-1}\mathbf{V}(\boldsymbol{r}_m)^\dagger \hat{\boldsymbol{h}}_m$);   // Sec. 5.3.1
end for
return shifted and scaled $\{\boldsymbol{\tau}_m[k], \boldsymbol{c}_m[k]\}_{m,k=1}^{M,K}$;     // See Sec. 5.3.3
```

labelled echoes. This data exploration has resulted in a jupyter notebook[2].

Table 6.1 – Room impulse response databases with and without labels for early reflections.

Project name	Annotated	# rooms	# source pos	# mic pos	Link
Acoustic Echoes Reveal Room Shape	yes	3	1	5	link[3]
modo_db	yes	1	3	256	link[4]
3D Room Reconstruction with Sound	no	1	17	12	link[5]
FIT Reverb Database	no	9	X	31	link[6]

---

[2] https://github.com/epfl-lts2/early_echo_estimation/blob/master/visualize_measured_rirs.ipynb

[3] https://infoscience.epfl.ch/record/186657?ln=en

[4] https://github.com/Chutlhu/modo_db

[5] https://vgm.iit.it/tutorials/3d-room-reconstruction-with-sound

[6] https://speech.fit.vutbr.cz/software/but-speech-fit-reverb-database

6.6 Conclusion

The proposed solution can find applications in the following problems: dereverberation, acoustical scene analysis and room shape estimations, inside home assistants. In the spirit of reproducible research, we have made our implementation available on github[7]. Due to the lack of time, extensive experiments with the newly proposed method are left for future work. Initial anchor experiments have shown that the method has potential for off-grid retrieval of echoes in a noisy blind deconvolution setting. It has also shown potential in the case of underfitting (when the level of sparsity is underestimated). In this case the proposed methods detects *the highest K* peaks.

---

[7] https://github.com/epfl-lts2/early_echo_estimation

7 A Sparsity Measure for Echo Density Growth in General Environments[1]

Statistical parameters that characterize impulse responses in enclosures, such as the reverberation time, have been extensively studied in room acoustics [91], along with fairly standard estimation algorithms [58]. These parametric models provide insight into impulse responses and enable efficient, natural sounding artificial reverberation [186, 165] and efficient acoustical encoding [151] for interactive auralization. However, parameters characterizing enclosures are insufficient for convincing spatial audio rendering in augmented and virtual reality applications which increasingly feature a rich variety of spaces that are partially or fully outdoors [147, 150, 151], such as courtyards, forests, and urban street canyons.

We investigate how acoustic impulse responses in these transient spaces might differ from enclosures, whether obtained through measurement or simulation [172, 173, 111]. In particular, motivated by the common observation that outdoor scenes are sparsely reflecting [173], we study the temporal growth of echo density in the impulse response. Our goal is to characterize how this growth might differ - if at all - between indoor and outdoor acoustic impulse responses, using a parametric power-law model. To the best of our knowledge, such an investigation has not be done before. Prior techniques, compared in [103], study echo density primarily for classifying the first moment when the impulse response is sufficiently diffuse, called the *mixing time* [138]. This is in contrast to our goal, which is to quantify and analyze the detailed echo density evolution *before* the mixing time. Part of the impulse response after the mixing time is usually modeled as white Gaussian noise, since the diffuse behavior prevails.

Our main contribution is a *sorted density* (SD) measure of echo density that enables such an investigation. We show SD to be theoretically meaningful while being robust to complex 3D scenes. In contrast to simple scenes such as a cuboid (shoebox), echo density in complex scenes cannot be defined as number density of non-zero values in the impulse response. Firstly, surface details and irregularities cause wave scattering so that strong reflections do not appear as exact copies of the source pulse in the impulse response, but rather contain

---

[1] Work done as research intern at Microsoft Research, Redmond.

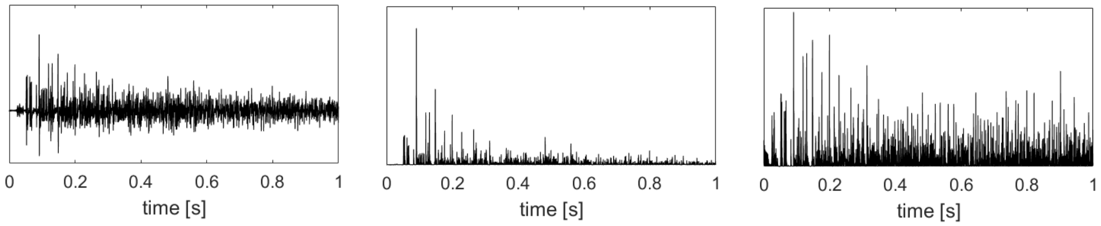


Figure 7.1 – The acoustic impulse response (left) is converted to an echogram (middle). A local energy normalization factors out the energy decay envelope (right).

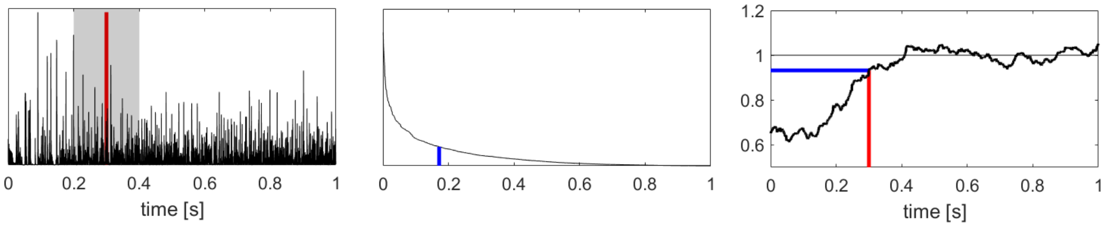


Figure 7.2 – Normalized echogram is analyzed (left) with a rectangular sliding window (shaded) centered at each sample (red line). The sorted density is computed, as a fraction of window width (middle, blue line). The processing for each sample and normalizing with expected value for Gaussian noise yields echo density (right).

substantial linear distortion. Secondly, the distorted strong arrivals are intermixed with numerous weak arrivals from diffuse scattering caused by geometric clutter. This makes it challenging to define and separate out "salient" peaks to measure their temporal density, such as in [47] to estimate mixing time, as compared in [103]. Our sorted density function (illustrated in Figures 7.1 and 7.2) is an aggregate measure that avoids peak separation or detection, obviating such difficulties.

We validate our SD measure against the theoretical notion of echo density on simple enclosures and observe good agreement. We then apply our technique to measured and simulated impulse responses on complex scenes and observe that the echo density growth with time which can be modeled well as t^n, where the growth power behaves like $n \approx 2$ indoors and $n \approx 1$ outdoors, with intermediate values in mixed cases. Based on these results, we observe that the growth power of echo density during early reflections is a promising new statistical parameter that discriminates indoor and outdoor acoustics.

7.1 Echo density measure

Given an input band-limited impulse response $h_i(t)$ we find the first-arrival delay of the direct sound, τ_0, when the acoustic impulse response is modeled like in (eq. 5.14). This can be estimated by manual inspection to locate the signal onset, or using a detection algorithm [151]. Direct sound is removed by setting: $h_i(t) = 0, t < \tau_0 + 10$ ms. This yields the input response to the echo density estimation, $h(t) \equiv h_i(t + \tau_0), t \geq 0$. Echo density is then computed using a two-pass procedure, illustrated in Figures 7.1 and 7.2 respectively, as will be formally discussed further.

7.1.1 Local energy normalization

The input response is converted to an echogram, $e(t) \equiv h^2(t)$. The first pass performs a local energy normalization which factors out the energy decay in the response thus ensuring that the number density estimates are not biased by the overall energy envelope of the response, making the measure fairly insensitive to the reverberation time. This ensures that the energy decay trend does not affect our results. We normalize each sample value with the local mean of surrounding samples weighted with a Tukey window w:

$$\tilde{e}(t) = \frac{e(t)}{\int e(t+\tau)\ w(\tau)\ d\tau}\ , \tag{7.1}$$

where the Tukey window (tapered cosine) is given by:

$$w(t) = \begin{cases} \frac{1}{2}\left\{1+\cos\left(\frac{2\pi}{r}[t-r/2]\right)\right\}, & 0 \leq t < \frac{r}{2} \\ 1, & \frac{r}{2} \leq t < 1-\frac{r}{2} \\ \frac{1}{2}\left\{1+\cos\left(\frac{2\pi}{r}[t-1+r/2]\right)\right\}, & 1-\frac{r}{2} \leq t \leq 1 \end{cases} \tag{7.2}$$

Window has length L and r is the ration of the cosine-tapered section length to full window's length, $r \in [0,1]$.

We have used a continuous time notation for brevity, the integrals are to be understood as discrete summation. The width of the window defines the temporal locality for normalization. A half-width of $T_n = 10$ ms corresponds to the interval of perceptual echo fusion [104] and was found to work well in practice. The Tukey window is normalized so that $\int w(\tau)\ d\tau = 1$. The symmetric cosine tapering segments have width of 5ms each with a 10ms long constant segment in the middle. As the example in Figure 7.1 shows, the resulting signal is much more amenable for sparsity analysis, emphasizing peaks without explicit detection. This is important for two reasons: if we explore simulated data, then we are usually dealing with perfect resoponses generated by image source model [10, 21] or with bandlimited simulation as a result of FDTD based approaches [147]; on the other hand, real measurements suffer from noise. This universality of application of our method emphasizes need for a robust solution.

The main advantages of the proposed peak enhancement technique is that it does not require any kind of assumption on the exponential decay of the amplitudes in the impulse response, and also there is no hard threshold for deciding if a certain sample is a peak or not.

7.1.2 Sorted Density (SD)

We employ a simple measure of sparsity in a discrete positive signal $\boldsymbol{s}$. Our main idea is to sort the signal to yield a monotonically decreasing signal $\hat{\boldsymbol{s}}$. The sparser $\boldsymbol{s}$ is, the faster $\hat{\boldsymbol{s}}$ will fall off as a function of number of samples. Any smooth measure of the width of $\hat{\boldsymbol{s}}$ normalized with number of samples should then yield a notion of fractional energy density in the signal. An example is shown in Figure 7.2. We assume that the highest level of sparsity happens when there is only one peak and the lowest level of sparsity is manifested in the part of the impulse response that exhibits reverberant behavior (here peaks usually observe trends of Gaussian noise).

A natural way to compute width is via first-moment of sample index i with $\hat{\boldsymbol{s}}$ serving as weight. This is the sorted density functional,

$$D(\boldsymbol{s}) \equiv \frac{1}{L} \frac{\sum_{i=1}^{L} i \, \hat{\boldsymbol{s}}[i]}{\sum_{i=1}^{L} \hat{\boldsymbol{s}}[i]} \tag{7.3}$$

where L is the number of samples in the observed window. The sorted density is a unitless measure with values ranging between 0 and 0.5 corresponding respectively to minimal echo density when $\boldsymbol{s}$ contains a single non-zero sample, to maximum when all values are non-zero and equal. Gaussian noise g has an intermediate (expected) value of $D(g) = 0.18$. This is ensured with the $\frac{1}{L}$ normalization.

We then estimate the echo density function for the input response, $h(t)$, by employing a sliding rectangular time window on the normalized echogram, $e_n(t)$ and computing the sorted density in each window:

$$N'_{sd}(t) = \frac{D(\tilde{e}(t \in (t - T_l, t + T_l))}{D(g)}, \tag{7.4}$$

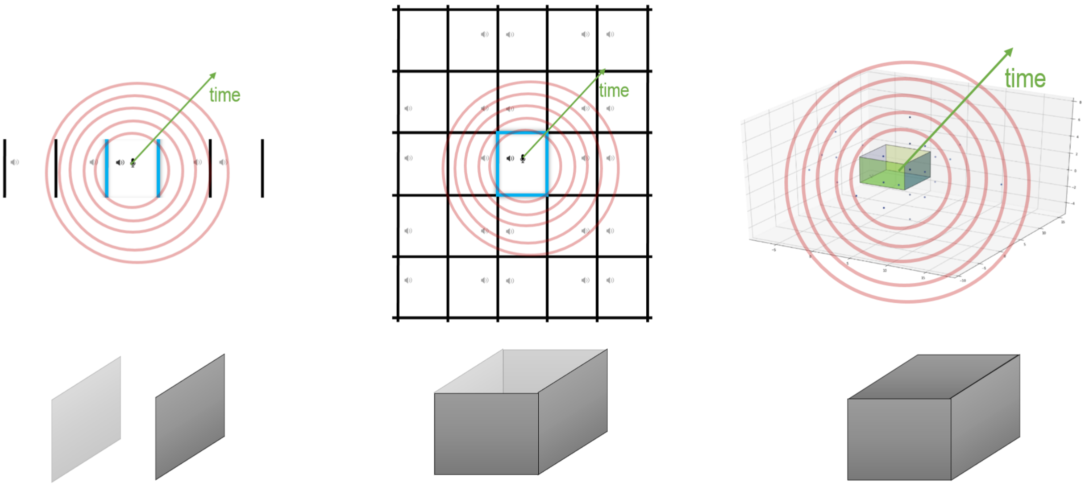
where any samples $e_n(t)$ for $t < 0$ are discarded from the analysis. Note the normalization with $D(g)$, so that an echo density of $N'_{sd} = 1$ indicates Gaussian noise. T_l is the half-width of the rectangular window and we empirically found $T_l = 100$ ms to work well. As shown in Figure 7.2 this yields an intuitive trend of echo density that initially increases and then settles near some maximum value (close to 1 indoors) as the response transitions to late reverberation.

7.2 Statistical model

We describe our general model for echo density growth, analytical motivation and fitting procedure.

7.2.1 Analytical motivation

For simple geometries such as a shoebox (rectangular) room where geometric acoustics is accurate the echo density may be defined rigorously by counting the number $N(t)$ of geometric paths that arrive at the listener within time t after the source emits an impulse. For any source location, the corresponding image sources form a periodic, discrete sampling of 3D space.



$$\frac{dN_r}{dt} = \frac{ct}{Ldt} = \frac{c}{L} \approx C \qquad \frac{dN_r}{dt} = 4\pi\frac{(ct)^2}{Sdt} = 8\pi\frac{c^2t}{S} \approx Ct \qquad \frac{dN_r}{dt} = \frac{4}{3}\pi\frac{(ct)^3}{Vdt} = 4\pi\frac{c^3t^2}{V} \approx Ct^2$$

Figure 7.3 – Echo density trends for various types of space. From left to right: in case of parallel walls, the echo density trend is constant and $n \approx 0$ (there is no echo build up), for a room without a ceiling we have $n \approx 1$ and for a room with all six walls we have $n \approx 2$.

Observing that the maximum propagation path length until time t is ct where c is the speed of sound, we have: $N(t) \propto (ct)^3$ by counting all image sources in the spherical ball with radius ct. Taking the time derivative to convert echo count to echo density, the full expression is [91, p. 110],

$$N'(t)_{indoor} = \alpha t^2, \tag{7.5}$$

where α is a geometry-dependent parameter, given by $\alpha = 4\pi c^3/V$ for room volume V. This result also holds under theoretically ideal diffuse field conditions. Note that this model describes the behaviour only up to the mixing time τ_{mix} where the impulse response approaches noise so that $N'(t)$ approaches a constant.

Removing the roof of the shoebox yields a courtyard-like geometry with 4 surrounding walls and a ground. This represents a reverberant outdoor scene where most reflectors surround the source and listener horizontally. Ignoring edge diffraction from the top wall edges where each point becomes a new source of a wave, the image sources occupy a periodic sampling of 2D (rather than 3D) space, so that number of echoes $N(t) \propto (ct)^2$ and the echo density, $N'(t)_{outdoor} \propto t$. All this can be observed in Figure 7.3. Based on these observations, we hypothesize the general model for any acoustical environment:

$$N'(t; N'_0, \alpha, n, \tau_{\text{mix}}) = \begin{cases} N'_0 + \alpha t^n, t < \tau_{\text{mix}} \\ N'_\infty, t \geq \tau_{\text{mix}} \end{cases}, \tag{7.6}$$

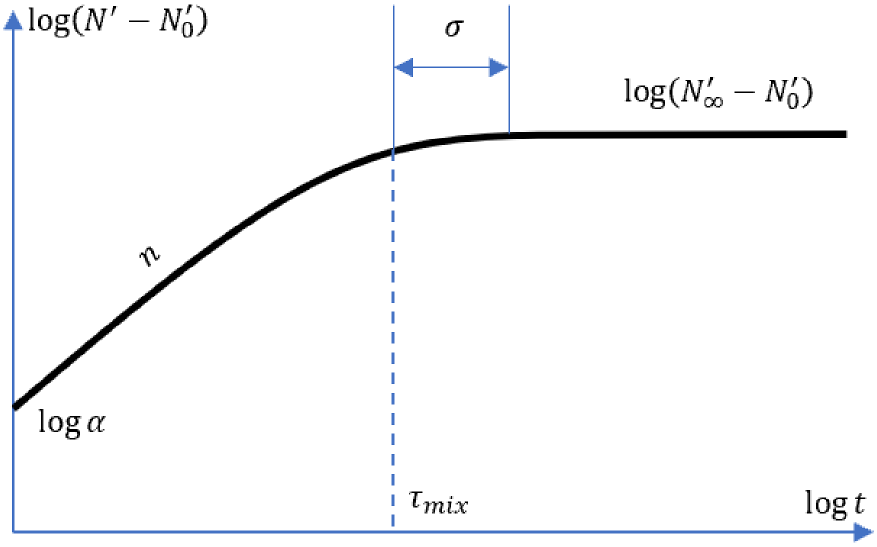


Figure 7.4 – Log-domain parametric model that is fitted to extracted echo density trend.

where $N'_\infty \equiv N'_0 + \alpha\tau_{\text{mix}}^n$ to ensure continuity, and $\{N'_0, \alpha, n, \tau_{\text{mix}}\}$ are the model parameters. The analytical results above do not apply near $t = 0$ or $t = \tau_{\text{mix}}$. Near $t = 0$ one must have some non-zero echo density, N'_0, due to initial reflections, followed by power-law growth that remains continuous and then stabilizes near some maximum value, N'_∞ at the mixing time, τ_{mix}. The continuous parameter n is the focus of our experiments, with the hypothesis that it should be ~ 1 outdoors and ~ 2 indoors based on analytical considerations above. Some geometric information about the scene size is also contained in α, although its interpretation has a dependence on n, whose study we leave for future work.

7.2.2 Model fitting

To robustly estimate the growth power n, we first separately estimate N'_0. We then perform fitting on $\log(N' - N'_0)$. As illustrated in Figure 7.4, this simplifies the model in Eq. 7.6 to two linear segments respectively that meet at $t = \tau_{\text{mix}}$: $\log(\alpha) + n\log(t)$ and $\log(N'_\infty - N'_0)$. In Figure 7.5 we can see example of types of spaces and their models in the logarithmic time domain. To reduce sensitivity in fitting due to non-smooth model at $t = \tau_{\text{mix}}$, we cross-fade between the two linear segments via a sigmoid window:

$$W(t; \tau_{\text{mix}}, \sigma) = \frac{1}{2}\left(1 - \tanh\left(\frac{t - \tau_{\text{mix}}}{\sigma}\right)\right). \tag{7.7}$$

The parameter σ controls width of the cross-fade, which we set to $\sigma = 20$ ms, as this length has empirically shown to give good results. The resulting smoothed parametric model is

$$\log(N'(t; \alpha, n, \tau_{\text{mix}}) - N'_0) = W \cdot (\log\alpha + n\log t) + (1 - W) \cdot \log(N'_\infty - N'_0). \tag{7.8}$$

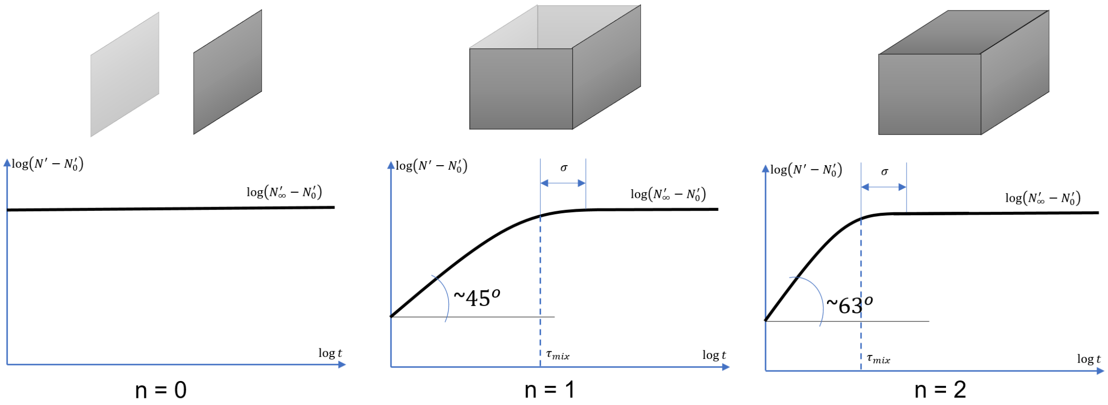


Figure 7.5 – Echo density model for various types of space. From left to right: in case of parallel walls, the echo density trend is constant and $n \approx 0$ (there is no echo build up), for a room without a ceiling we have $n \approx 1$ and for a room with all six walls we have $n \approx 2$.

Given the observed echo density profile N'_{sd} from Eq. 7.4, we estimate N'_0 as the minimum value of the echo density, $\min\{N'_{sd}(t)\}$ and then fit the above model to $\log(N'_{sd} - N'_0)$ using non-linear least squares. We constrain the search space to accelerate convergence. The search for α is unbounded, but for n is bounded by $[0,5]$ and for $(N'_\infty - N'_0)$ is bounded by $[0,2]$. With this choice of bounds we have avoided manual tuning in the fitting procedure, since the observations have implied that the sufficient upper bounds would be 2.5 and 0.5, respectively.

7.3 Results

Our experiments have two goals. First, we compare against theory on enclosures to validate our technique. Second, we compare the echo density growth power, n, between indoor and outdoor cases.

7.3.1 Experimental data

Experiments are performed on impulse responses acquired from both measurements and 3D wave simulations. Simulations allow tests with tightly controlled 3D geometry, but are necessarily band-limited due to computational cost restrictions. We use the time-domain spectral wave solver [148] inside the Triton simulator that was introduced earlier in Section 1.3.1. All simulations are band-limited to 1kHz with sampling frequency of 6kHz with the source and microphone placed close to the center of the room, but off the axes of symmetry and more than 1m apart. With these constraints, the results were not found to be sensitive to exact placement. Surface absorption coefficient was set to 0.05 for all frequencies in all simulations in order to have lightly damped conditions. While measured responses necessarily contain more noise, we have noticed that a higher sample rate improves the reliability of our technique, presumably because there is a larger number of samples within each analysis

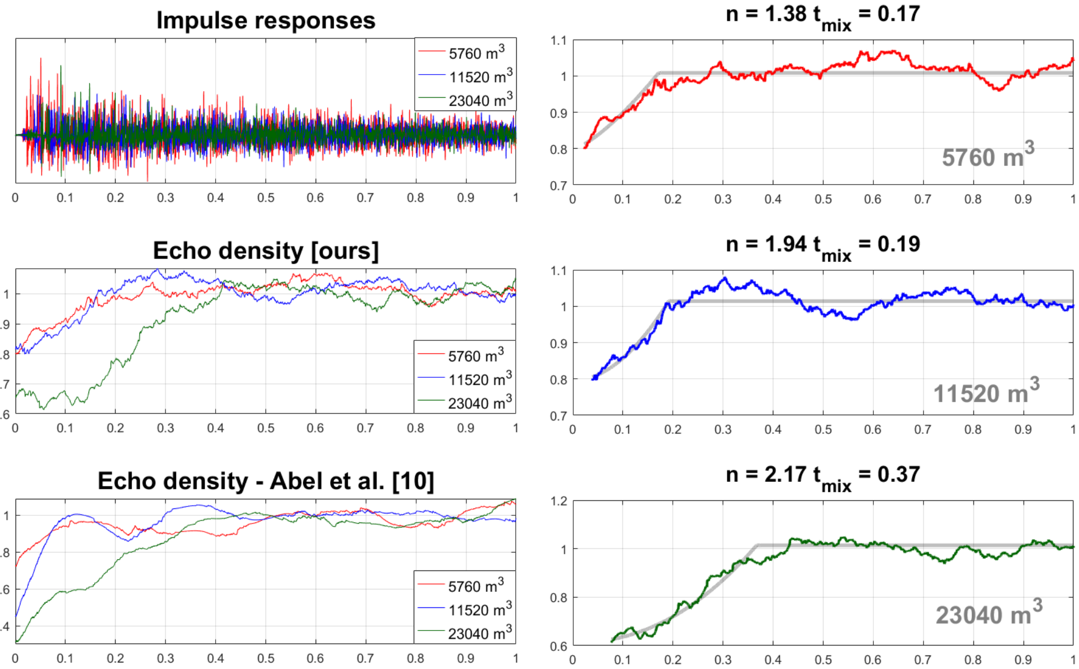


Figure 7.6 – Validation of method on shoebox scenes. Impulse responses are on left top. Three rooms are tested with volumes increasing by factor of two. Fitted models are plotted in grey color. Our echo density measure shows a growth power $n > 1$ as expected for indoors (right column).

window for statistical estimation.

7.3.2 Validation on simple enclosures

If our sorted density measure (Eq. 7.4) is a valid generalization of the theoretical notion of echo density (Eq. 7.5), we expect $n \approx 2$ on simple enclosures where geometric acoustics underlying Eq. 7.5 is reasonably accurate. We test this hypothesis with simulations on two types of such geometries: shoebox and convex polyhedron.

Figure 7.6 shows experiments on three shoebox rooms with volume increasing by a factor of two. Input responses are on left top. Here we compare our echo density measure (left middle) to [3] (left bottom), with the latter using the same window half-width T_l as our method. Both techniques are normalized so a value of 1 indicates late reverberation. Both techniques show an increasing trend, reaching around 1 at similar mixing times, τ_{mix}. However, our measure is designed to also model echo density growth before τ_{mix}, as shown on the right. This can be clearly confirmed by observing Figure 7.6. All cases show a growth power $n > 1$ as expected for indoors, with the two larger rooms agreeing well with theory with $n \approx 2$. For the smallest room however, n is smaller. We observe this systematic bias for smaller spaces with our technique. Echo density buildup is quick in small rooms, leaving a short span for model fitting. Our sorted density analysis window is also quite wide with $T_l = 100$ ms which is a contributing

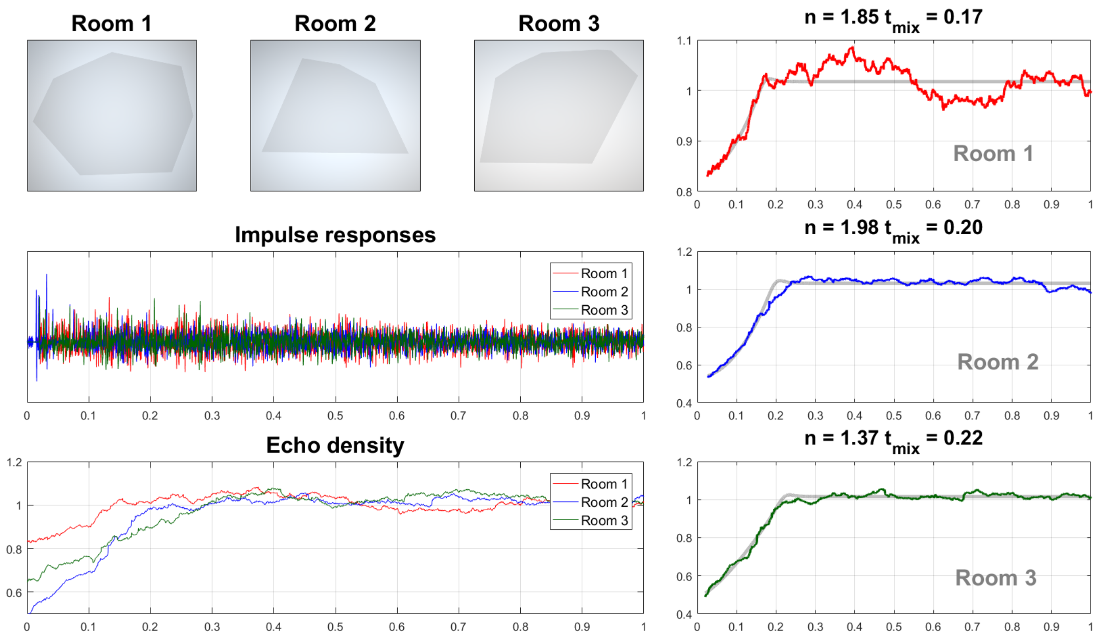


Figure 7.7 – Echo density on simulated convex polyhedral rooms with flat ground and ceiling.

factor, but we found this width necessary to build reliable statistics.

Figure 7.7 tests three general convex polyhedral room geometries with large flat reflectors. The polyboxes were randomly generated such that their volume is within $[10000, 20000]\mathrm{m}^3$. The echo density shows a close to quadratic growth in the first two cases with more irregular geometry, agreeing well with theory. In some cases, like "Room 3," we observe a decrease in n, perhaps because of flutter echoes between the two large near-parallel faces. Such periodicity in the response also motivated avoidance of symmetry axes in the shoebox tests.

7.3.3 Indoor to outdoor scene modification

As discussed in Section 7.2.1 if we remove the roof of a shoebox to turn it into a "courtyard", we theoretically expect $n = 1$, with some deviations caused by edge diffraction. We performed simulations in a shoebox room with a ceiling that gradually opens, as shown in Figure 7.8. This case reminds of a box for a domino game. As the roof is removed, the value of n smoothly decreases from near 2 towards 1, with intermediate values in the middle. This fits with theoretical expectations on the closed and open extremes, and also illustrates that the technique is resilient to mixed cases somewhere between indoors and outdoors.

7.3.4 Varying volume with fixed reverberation time

Figure 7.9 compares measured impulse responses on three enclosures with large variation in scene volume but differing absorption coefficient so that the reverberation times are similar. The three measurements were taken from the Reverb Challenge corpus ("Room 2", 106 m^3) [80],

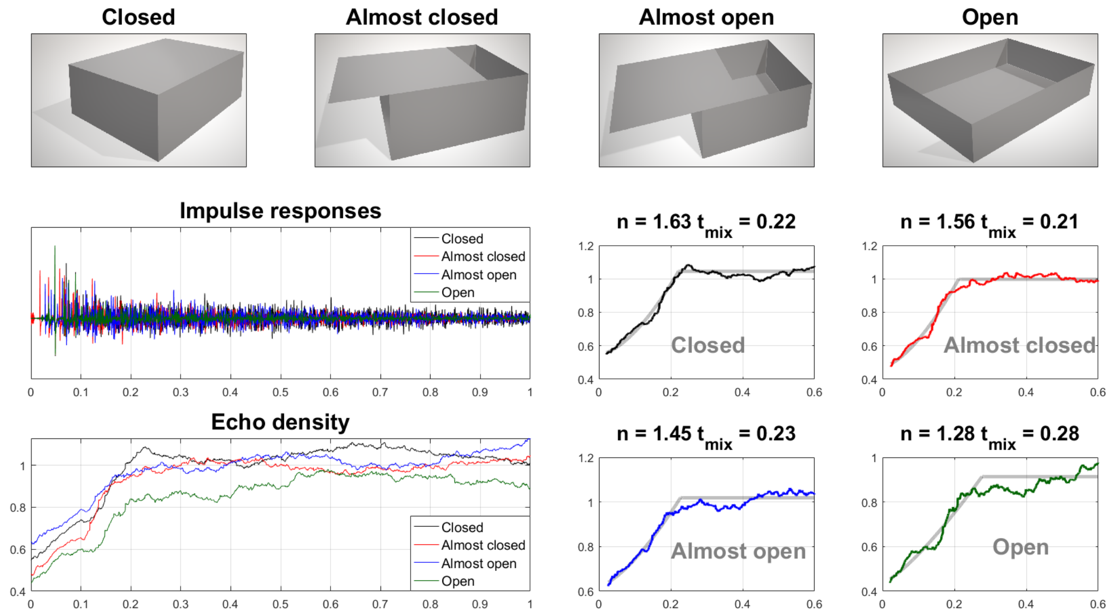


Figure 7.8 – A simulated shoebox room that gradually transforms to a courtyard (the domino game box setting). Echo density growth power, n, decreases smoothly as the scene progresses towards outdoors.

and from the Open AIR database ("Dixon Studio, York University Theatre", 1058 m^3, "Central Hall, York University", 8000 m^3) [115]. The energy decay curves are nearly identical (left column, middle). All of the measurements have a sampling frequency of 16kHz. In all cases the echo density trend is plausible, increasing and settling near 1. For the two larger rooms, we observe values of $n \approx 1.7$ and 2.4, corresponding well to indoors, with the smaller of the two rooms producing smaller value, a bias we noted earlier. Regression fails on the smallest room with volume similar to a small office (≈ 100m^3) indicating that our regression could be improved to handle small rooms better. This could be achieved in the following way: by decreasing the size of the analysis window and increasing the sampling frequency for rooms of smaller volume.

7.3.5 Indoor versus outdoor location in urban area

We measured impulse responses in urban office building at two locations inside and outside, shown on a 3D cutaway top view in Figure 7.10. Sampling frequency was 48kHz. We find values of n in good agreement with expectations, 1.80 indoors and 0.87 outdoors, showing a clear difference between indoor and outdoor acoustics in a highly complex scene.

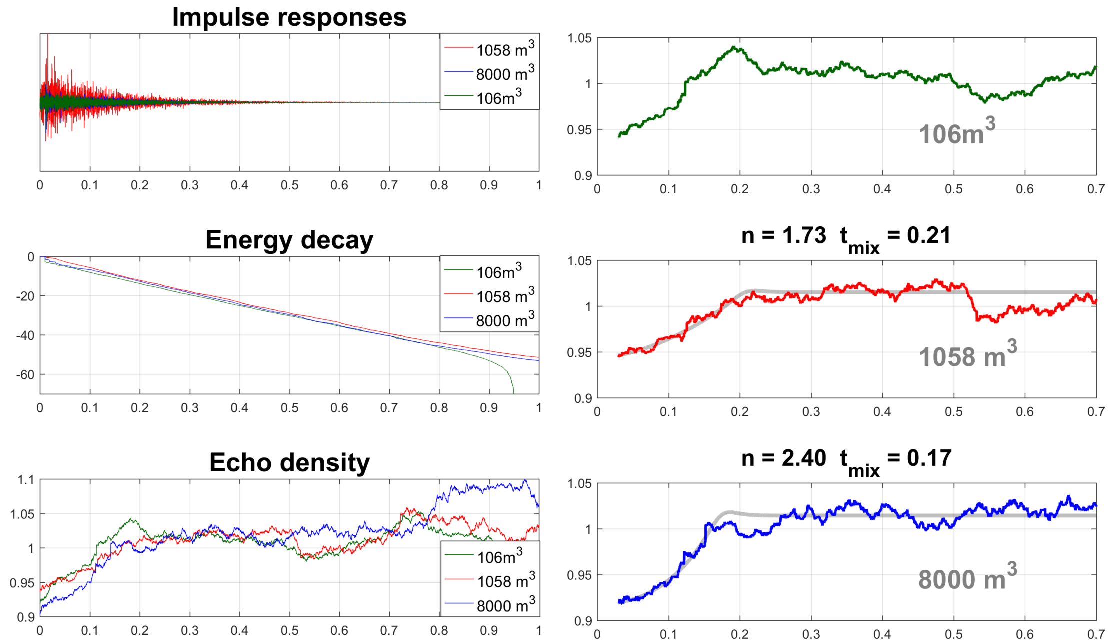


Figure 7.9 – Comparison on measured responses in three rooms with different volumes, but same reverberation time. For the two larger rooms, n is around 2, agreeing well with expectations.

7.4 Conclusion and future work

We study the detailed temporal evolution of echo density in impulse responses for applications in acoustic analysis and rendering on general environments. For this purpose, we propose a smooth *sorted density* measure that yields an intuitive trend of echo density growth with time. This is fitted with a general power-law model motivated from theoretical considerations.We validate the framework against theory on simple room geometries and present experiments on measured and numerically simulated impulse responses in complex scenes. The method is found to agree well with theory. Our results show that the growth power of echo density is a promising statistical parameter that shows noticeable, consistent differences between indoor and outdoor responses, meriting further study.

We wish to improve the robustness of the method in the future, especially for small rooms. The size parameter, α, and mixing time, τ_{mix}, contain geometric information about the scene. But in outdoor cases ($n \approx 1$) they no longer admit interpretation in terms of "room" volume. A study on the geometric interpretation of these parameters in general scenes could prove to be a fruitful future direction.

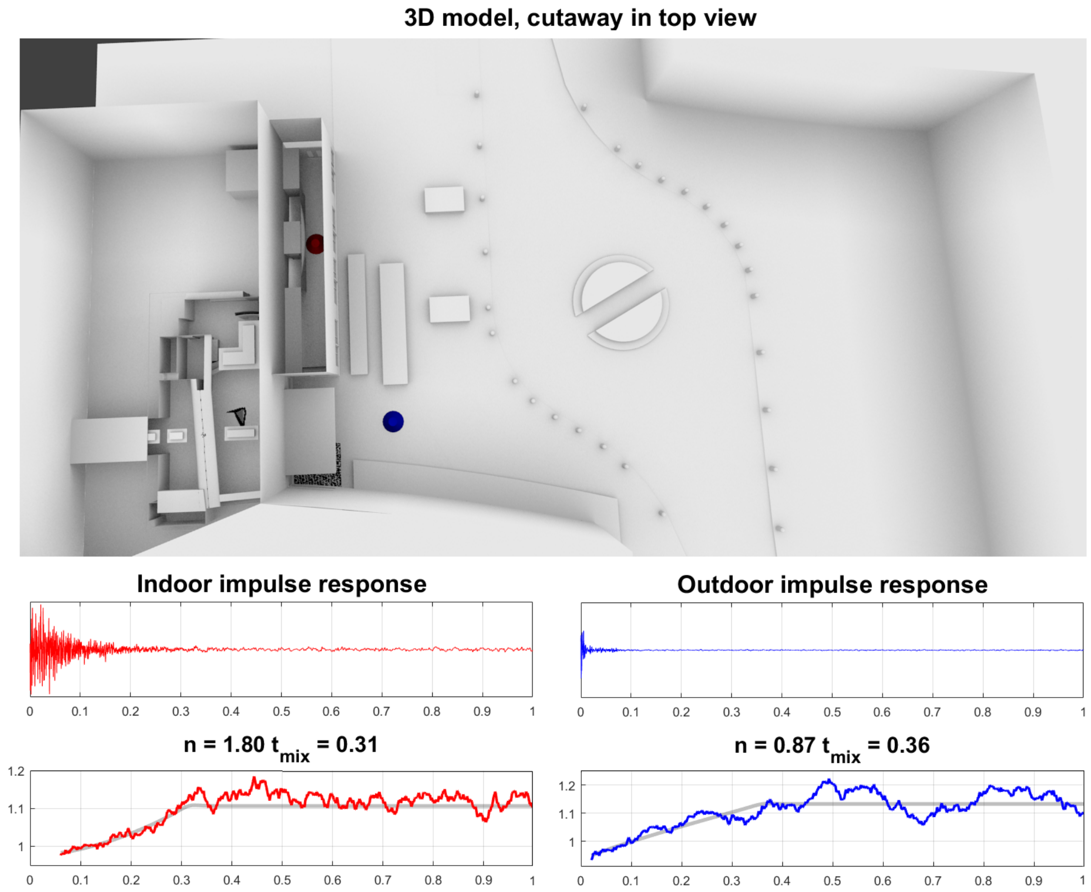


Figure 7.10 – Measurements were performed in the two locations shown in the 3D cutaway top view, indoors (red) and outdoors (blue). The two locations are clearly differentiated by $n = 1.8$ and 0.87 respectively.

7.5 Acknowledgements

Thanks to Jon Hanzelka and Pedro Urbina Escos from the Microsoft Synthetics team for the LIDAR scan and 3D model of the building used in our “urban area” tests.

Part V

Parametric models for audio classification

8 Audio Representations for Deep Learning

Since the recent trends in audio have been evolving around deep learning, we will give an introduction about the representations of audio within this framework and also give an overview of the recent trends in this domain. The origin of deep learning is still under question, although many authors such as Goodfellow and Bengio [65], Schmidhuber [166] and LeCun [94] tend to share the contribution to the launch of this exciting field.

Because the deep learning technique can be applied only when a sufficient size dataset is available for the training, we will slightly diverge from the main focus of the thesis, that is from *acoustics* to *audio*. Due to the fact that large scale room impulse response databases that could be used for these types of approaches are still unavailable[1] we will be making a slight shift from parametric data exploration for acoustics to parametric data exploration for audio. To have a better idea how hard it is to prepare a sufficient acoustic database for deep learning, [60] gives an example of making an acoustic dataset, augmentation and comments on the heterogeneity of the available recordings that is a result of lack of standardized procedure for recording and storing such data.

Regardless of the type of the data, the goal of deep learning and machine learning research in general is not to seek a universal learning algorithm, but to rather understand what kinds of distributions are relevant to the "real world". The aim is to build algorithms that will perform well on data generating distributions that are close to what can be manifested in the world around us. In most cases the solutions that are available are not easily generalizable and tend to be application or data specific. This has already been manifested in the classical signal processing: We know that wavelet functions are good for a sparse representation of images, and some audio signals have a sparse representation in the domain of Short Time Fourier Transform (STFT), for example speech.

Due to the fact that the initial deep neural networks were built for processing of images expecting a $2D$ input, in the early solutions audio would be converted into *spectrograms* and fed in that format to the network.

---

[1] most of the available audio/acoustic datasets are listed here: http://www.cs.tut.fi/~heittolt/datasets.html

8.1 Audio representations with fixed parameters

In audio signal processing, time-frequency representations such as spectrograms are central tools. They have an intuitive interpretation and reveal insightful information to the human expert. It is not a surprise that many deep learning approaches to audio signals use such representations as well [36, 142]. It is also convenient as most of the deep network architectures have been initially developed for image processing and require 2D arrays of values as inputs. The network learns to detect time-frequency patterns, similarly to what is done on images. Depending on the task, it may then output a classification of a sound [137, 159], a denoised signal [106] or separated sources [31]. The most recent trends include preprocessing the spectrograms [105], before they are fed to the network.

However, natural images and spectrograms do not possess the same properties and turning an audio file into an image has some limitations. Among them, spectrogram representations can be defined in many different ways, with different time window shapes and sizes or different frequency spacing. Also, images are $2D$ spatial representations and the spacing on the axis is usually the same, but when it comes to spectrograms we have axis of different nature whose spacing can be variable. [36] and [142] give a review of the different time-frequency representations used in deep learning. In addition, patterns in the time-frequency plane are different from those that can be found in images: the former are usually less complex, with smoother edges and limited textures. Furthermore, the axes are not equivalent in the spectrogram as frequency is different from time. For example a frequency-shifted pattern may result in a different sound classification [95], while a temporal shift does not (though similar problems can also emerge in the classification task of images). Moreover, the spectrogram is the magnitude of the Short-Time Fourier Transform (STFT) and the information contained in the phase is not taken into account. Lastly, computing a spectrogram, and possibly inverting it for synthesis, adds a computational burden which can be important for large audio datasets.

Next to spectrogram[2] as shown in Figure 8.1a, some other visual representations have emerged. These include: *pyknogram*[3], which is an audio representation from 1995 [139] as shown in Figure 8.1b that represents a modified version of STFT that emphasizes more the harmonic structure of speech (co-channel speech analysis), and also *rainbowgram*[4] which is an audio representation from 2017 [54] as shown in Figure 8.1c that puts emphasis on the derivative of the phase of the signal. The rainbowgram is a modified version of the Constant Q Transform (CQT) with the intensity of the lines proportional to the log magnitude of the power spectrum and the color given by the derivative of the phase. Although the spectrograms tend to have a general purpose, the pyknograms are mostly applied for speech enhancement and analysis, and the rainbowgrams are used for music synthesis [54, 46].

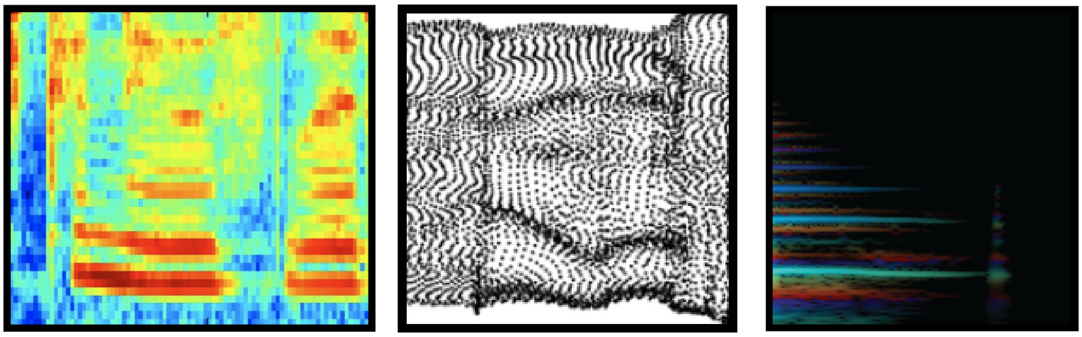
There are also approaches that combine multiple features and feed it to the network in a

---

[2] https://github.com/drammock/spectrogram-tutorial

[3] https://github.com/idnavid/pyknograms

[4] https://github.com/tarepan/rainbowgram



(a) Spectrogram (b) Pyknogram (c) Rainbowgram

Figure 8.1 – Audio representations with fixed parameters for deep learning.

multi-channel manner [171], for example by coupling four major groups of audio features: the Mel-Frequency Cepstral Coefficients (MFCC) [45], the Gammatone Frequency Cepstral Coefficients (GFCC), CQT and Chroma features. This results in an increase of the accuracy of the network, but does not contribute to potentially better understanding of network's behavior and also requires a large preprocessing footprint.

All of these representations have *fixed parameters*, which makes them rigid and invariant to application and data type. The same as in the case of images, not all sounds should be processed equally, so this imposes a requirement for an adaptable representation. Although the fixed representations can perform well for tasks such as audio classifications, they have only a moderate performance for speech separation [55] and are not an appropriate intermediate representation for a task of speech synthesis, for example. This suggests that audio representations have to have an adaptive form, since in an optimal case they are task dependent.

8.1.1 The most common parameters of fixed representations

The most common parameters that are needed for these types of fixed representations are: window size, window type, overlap (stride), frequency spacing and number of filters. The window size is usually around 25ms, but is tends to be application dependent and could go as low as 10ms in hearing aids application. Typical window types are shown in Table 8.1 where the time index n satisfies: $0 \leq n \leq N$ and $L = N + 1$.

Table 8.1 – Typical window types for fixed audio representations.

Window type	**expression**
Rectangular window	$w[n] = 1$
Hanning window	$w[n] = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi n}{L}\right)\right]$
Hamming window	$w[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right)$

The overlap is usually chosen from the set: $\{25\%, 50\%, 75\%\}$ and the filter number, that will define the height of the representational image, is usually 40. The frequency spacing is usually chosen according to the log-Mel perceptual scale.

8.2 Learning without an intermediate representation: End-to-end approaches in audio

In order to overcome the limitations of fixed representations, an alternative direction has been chosen consisting of taking an end-to-end approach where the raw audio file is the input of the network. The recent success of Wavenet [121, 123, 122] demonstrates the efficiency of this approach for audio synthesis of speech. Raw audio input is also beneficial for speech separation tasks - Tasnet [107] as well as Wave-U-Net [176] show better performances for speech separation and faster processing compared to spectrogram-masking approaches.

In end-to-end approaches, one-dimensional convolutions are applied to raw audio signals. However, kernel size needs to be much larger than the one used for image applications. Indeed, at a sampling rate of 44kHz, 44 samples represent 1 ms of audio signal. To capture audio patterns that have duration of 10, 100 ms or more, in particular low frequency patterns, either large kernels are needed or deeper convolutional architectures (to allow for combinations of kernels at many different positions in time). Both solutions lead to a large increase in the number of parameters to be learned and hence require more training time and more data. The "atrous" convolution have been introduced in Wavenet in order to increase the time length of the kernel without increasing the number of weights to learn. Finding alternative ways for unlocking the time-length limit is an important challenge for raw audio processing in deep learning.

8.3 Audio representations with learnable parameters

We propose and investigate the design of a new convolutional layer where kernels are parameterized functions, in order to provide an audio representation with *learnable parameters*. This layer is an input layer of a convolutional neural network for audio applications. The kernels within are defined as functions having a band-pass filter shape, with a limited number of trainable parameters. So we will be learning the sets of parameters of filters in a certain filter bank. This will enable us to learn only a few parameters instead of learning the full length filter that can be a few hundreds of parametrs long.

The concept of learning filters has been first introduced in three recent works by [168], [153] and [78]. The first one introduces Gaussian filters in the input layer. Parameters are the amplitude, the Gaussian width and the modulation frequency. An increase of the classification accuracy is reported with the learned parameters. However, the filter learning is seen as a fine-tuning of the network after the first training pass with fixed Gaussian parameters. The authors report and discuss the evolution of the filters' amplitude during the fine tuning. The filter frequencies tend to keep their initial values although they are learnable. The possible adaptation of the temporal width of the filters is not given.

Building such a type of layer is motivated by several end-to-end learning studies that investigate convolution kernels learned from the raw audio signal [48, 184, 64, 157]. They all show that the input kernel's focus in frequency is similar to the one of the Mel or auditory scale. The kernel shapes in the spectral domain are similar to band-pass filters, with more narrow-band kernels localized on the low frequency spectrum than in the high frequency. This behavior does not depend on the network architecture nor on the application such as speech recognition [69, 198] or audio tagging [48]. All of these results suggest that the logarithmic spacing of frequencies and bandwidth properties first established in the psycho-acoustics studies with the Mel/Bark scales are somewhat universal in audio analysis tasks. These works point out the tendency of the input convolution kernels to adopt band-pass filter shapes.

Hence, we hypothesize that designing kernels with a band-pass property results in an inductive bias that helps the network to converge more rapidly and possibly reduces overfitting. Our first motivation is to confirm this hypothesis. On the other hand, the studies cited above remain experimental without, yet, precise spectral and temporal properties of the kernels. In addition, most of them initialize the kernels as band-pass filters with a Mel scale frequency spacing. So the influence of the kernel initialization remains unclear. Our second motivation is to investigate more precisely these filters' properties.

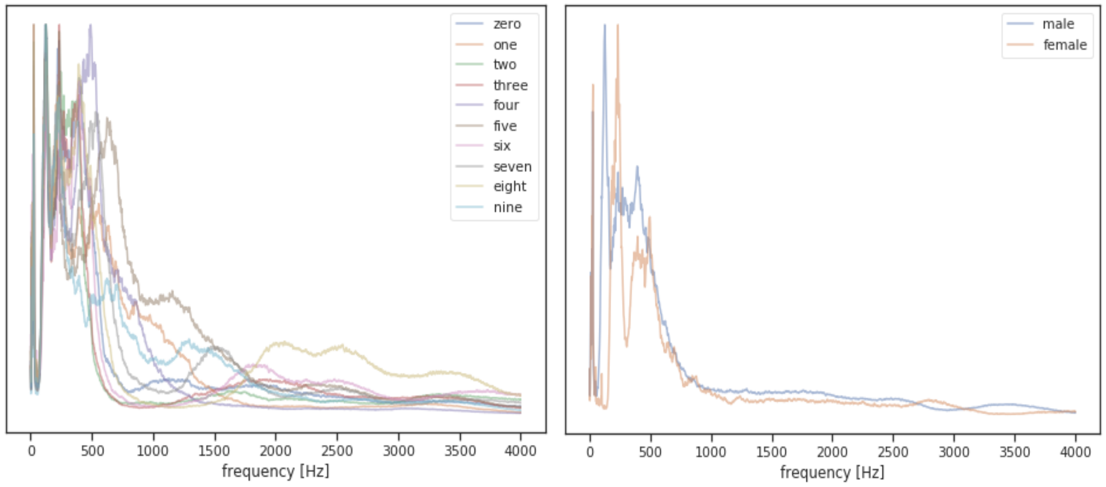
8.4 Audio datasets for audio classification tasks

In the audio domain, the distribution of energy in sound depends on the dataset, as can be seen in Figure 8.2. Here we observe the energy distribution for different classes over various datasets: *AudioMNIST* [15], *GoogleSpeechCommands v2* [192], *UrbanSound8K* [160] and *BirdVox* [158]. The *BirdVox* database differs from the other databases by a specific energy distribution having most of the energy in the high-frequency domain.

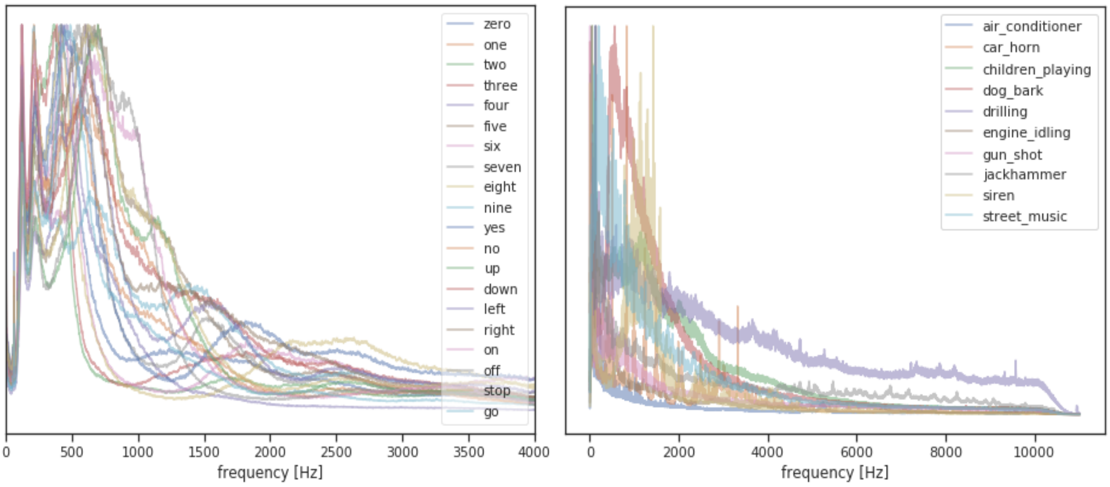
This thesis will focus on the problem of sound classification for the following databases: *AudioMNIST*, *GoogleSpeechCommands v2* and *UrbanSound8K*, whose statistics are given in Table 8.2.

Table 8.2 – Class statistics over different datasets.

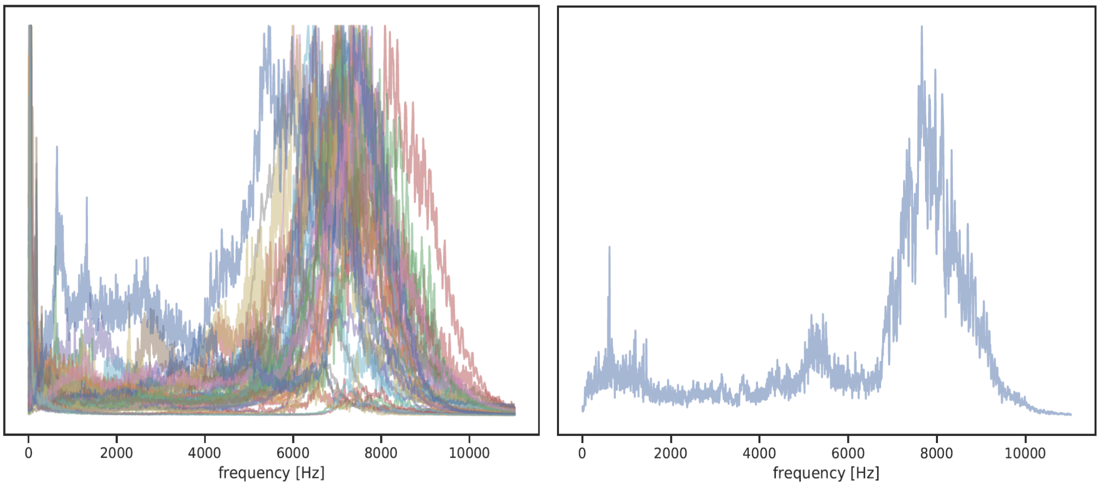
Database	**# of samples**	**# of classes**	**largest class size**	**smallest class size**
AudioMNIST	30000	10	3000	3000
GoogleSpeechCommands v2	105829	35	4052	1557
UrbanSound8K	9732	10	1000	374



(a) *AudioMNIST* - digit label (b) *AudioMNIST* - gender label



(c) *GoogleSpeechCommands v2* (d) *UrbanSound8K*



(e) *BirdVox* (f) *BirdVox* - one class example

Figure 8.2 – Dataset energy distribution per class and corresponding labels.

9 SpectroBank: A Filter-bank Convolutional Layer for CNN-based Audio Applications[1]

Adopting a hybrid approach, half way between the raw audio and the spectrogram, we propose to learn particular filters' shapes having a limited number of parameters that fully define them. These filters are the kernels of the first convolutional input layer of the network. This set of kernels may be seen as a filter bank. Consequently, the new input layer acts on the raw audio and outputs a learned time-frequency representation, adapted to the task. The functions we propose are modulated Gaussian windows, Gammatone and Gammachirp functions. Their performance will be compared with wavelets, that are present in literature.

The goal of these filters is two-fold. Firstly, it reduces the number of parameters to learn. Unlike in the end-to-end approach, it makes the size of the kernel independent of the number of weights to learn and enables the usage of large temporal inputs. We show that this approach speeds up the learning process and improves the accuracy on several audio classification tasks. In addition, our experiments show that the number of filters required to obtain the best results is small, around 20-30. We also demonstrate that the performances of different functions proposed in audio signal processing (modulated Gaussian, Gammatone and Gammachirp functions) give close results and are better than wavelets at classifying sounds. Secondly, this layer of parameterized functions helps understanding the filtering process done within the first layer of deep networks. This opens the way to a better interpretation of the neural networks and beyond, of the intricate relationship and the shape of audio patterns in the time-frequency space. In our experiments, a relationship between the central frequency of the filter and its temporal width emerges with the learning. This is in agreement with the Equivalent Rectangular Bandwidth (ERB) and Bark scales found in psycho-acoustic studies.

In our solution the filter layer is fully integrated in the learning process, the parameters are learned from the beginning. The filter amplitude is not a parameter in our case as the weights of the following layers enable weighted combination of filters. With our approach, the evolution of the frequency and width of the different filters is more visible. In [153], the authors introduce a layer, called SincNet, made of sine modulated functions that approximate band-

---

[1] Work done with Benjamin Ricaud and Nicolas Aspert at École polytechnique fédérale de Lausanne.

pass rectangular windows in the frequency domain. The learned parameters are the minimal and maximal cut-off frequencies of each band-pass filter. One of the main results is given by the cumulative frequency response of the SincNet filters. The network tends to focus more on particular regions of the frequency space, where formants are localized. This is interesting as it shows how the parameterized filters enable a precise interpretation of the learning and underline particular spectral properties of the data. The present work goes further in this direction. Eventually, [78] introduce wavelet filter banks learned for speech recognition. Each kernel is a wavelet defined by a single parameter, its scale. It shows evidences both of the efficiency of this approach and of the possibility to interpret the shape of the learned kernels. We compare the efficiency of the wavelet filters with several other modulated windows and show that the former under-performs on audio signals.

9.1 Learnable filter banks (SpectroBank)

We design a new convolutional neural network layer, called SpectroBank. In this layer the kernels are functions defined by a few parameters that are learned. We call these functions *filters*, making a parallel with filters in signal processing. Indeed, these functions have the property of being band-pass filters and are well known in audio signal processing. One of the trainable parameters of each filter is the central frequency of the band-pass filter. The second parameter is the bandwidth of the filter (or a quantity closely related to it). Hence this set of filters forms a filter bank where the frequency and bandwidth of the filters may be adapted to the data and to the learning task. Note that the learned filterbank may not cover the entire spectrum but should focus on important spectral regions that are the most discriminative for classification.

The input of the SpectroBank layer is a 1D audio signal and the output is a 2D representation. The output representation axes are time and filter number. Since each filter is associated to a particular frequency band, this 2D representation can be seen as a time-frequency one (or time-scale in the case of wavelets). Initializing the filters by increasing frequencies (or scales), we can influence the frequency ordering to follow the filter number. Filter functions and their parameters are recalled on Table 9.1. Their shape in time is illustrated on Fig. 9.1 and in frequency on Fig. 9.2 , with increasing oscillating frequency (or scale for wavelet) from blue to purple (starting from $f = 0$).

Table 9.1 – Description of the filter bank types and the parameters used during training. In most of our experiments, γ is fixed to 4.

Filter Type	# of parameters	Parameters
wavelet	1	s - scaling
Gaussian	2	f - frequency σ - width
Gammatone	3	f - frequency, b - bandwidth, γ - order
Gammachirp	3	f - frequency, b - bandwidth, c - chirp trend

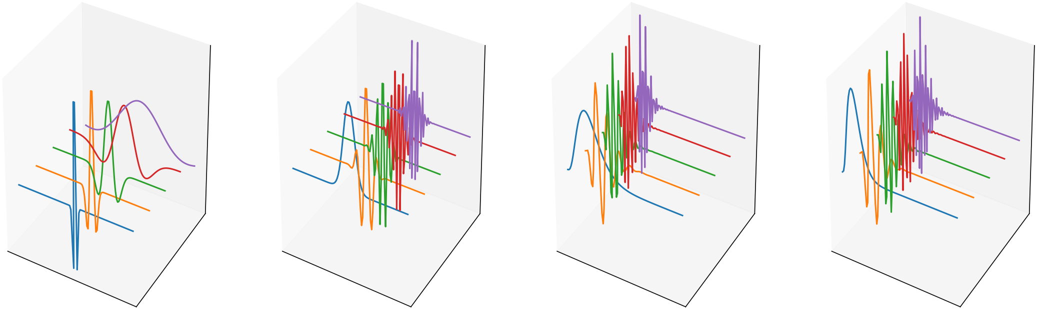


Figure 9.1 – Examples of filter banks in time domain. From left to right: wavelet filters, Gaussian filters (cosine modulation), Gammatone filters (envelope, cosine and sine modulations) and Gammachirp filters, for fixed bandwidth and different frequencies.

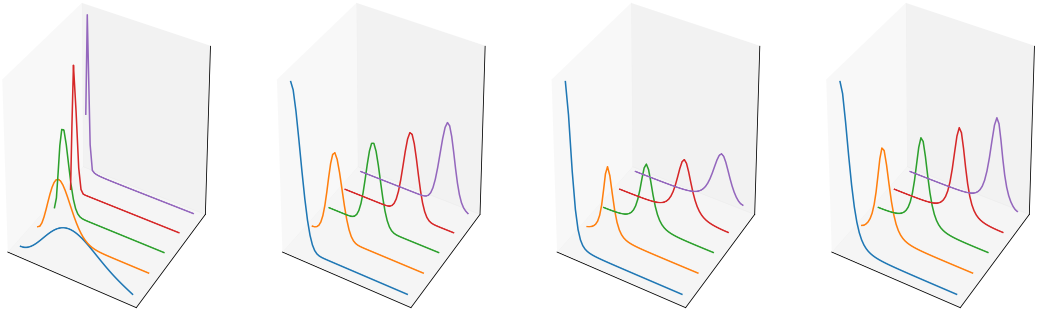


Figure 9.2 – Examples of filter banks in frequency domain. From left to right: wavelet filters, Gaussian filters (cosine modulation), Gammatone filters (envelope, cosine and sine modulations) and Gammachirp filters, for fixed bandwidth and different frequencies.

In all the definitions, N denotes the filter length and n is the variable (sample number). The time in seconds can be expressed using the sampling frequency f_s with $t = n/f_s$ and the frequency in Hertz with $f \times f_s$, where $f \in [0, 0.5]$ is the normalized frequency in the formulas.

Mexican hat wavelet. In order to compare to the state-of-the-art, we use the Mexican hat wavelet introduced in the paper by [78]:

$$\boldsymbol{w}[n] = \frac{2}{\pi^{1/4}\sqrt{3s}}\left(\frac{n^2}{s^2} - 1\right) e^{-\frac{n^2}{s^2}}, \tag{9.1}$$

with $n \in [-N/2, (N-1)/2]$ and $s > 0$ being the scale parameter.

Gaussian filter. Here, $n \in [-N/2, (N-1)/2]$. The Gaussian filter g is defined as follows:

$$\boldsymbol{g}[n] = \sqrt{\frac{2}{\sqrt{\pi}\sigma}} e^{-\frac{n^2}{2\sigma^2}} \left(\cos(2\pi f n) + j\sin(2\pi f n)\right). \tag{9.2}$$

The parameter $\sigma > 0$ is the variance of the Gaussian (temporal window width) and f is the oscillating frequency. It is a complex-valued function that we split into its real and imaginary parts. For each f and σ two kernels are created, one with the cosine modulation and one with the sine one.

Gammatone filter. The Gammatone filter [43, 129, 68] is another example of kernel. It is defined on the interval $n \in [0, N-1]$ as :

$$\boldsymbol{h}[n] = A(\gamma, b) n^{\gamma-1} e^{-2\pi b n} \left(\cos(2\pi f n) + j\sin(2\pi f n)\right), \tag{9.3}$$

where A is the normalization, $A(\gamma, b) = \sqrt{2(4\pi b)^{(2\gamma+1)}/\Gamma(2\gamma+1)}$. The parameter γ is the order of the Gammatone. It can be learned or fixed to 2 or 4. These two orders are the best suited ones for modeling the human hearing related filter bank [128]. In the experiments, we will fix $\gamma = 2$ or $\gamma = 4$. The other learnable parameters are b, related to the width of the function, and f the frequency. The symbol Γ denotes the Gamma function. The bandwidth B of h depends linearly on b and is given by the following formula [43]:

$$B(\gamma, b) = 2(2^{1/\gamma} - 1)^{1/2} b. \tag{9.4}$$

Gammachirp filter. This function is similar to the Gammatone family ones but possesses an oscillating frequency that may evolve with time. The Gammachirp function [73] is defined on the interval $n \in [0, N-1]$ as follows:

$$\boldsymbol{k}[n] = A(\gamma, b) n^{\gamma-1} e^{-2\pi b n} \left[\cos(2\pi f n + c\ln(n+\epsilon)) + j\sin(2\pi f n + c\ln(n+\epsilon))\right], \tag{9.5}$$

where A is defined above. In the present work, γ is fixed to $\gamma = 4$. This filter possesses 3 parameters, b related to the width of the window, f to the frequency and c to the chirp value. To avoid the logarithmic singularity at the origin, we add a small positive value $\epsilon = 10^{-4}$ to the expression.

The kernel shapes proposed in the present work are based on specific signal processing functions. They are used for performing short-time Fourier transforms or more generally for designing filterbanks. Modulated (truncated) Gaussian are emblematic examples. Gammatones and Gammachirp functions are used in cochlear models [161]. They provide interesting results when combined with deep learning models for speech enhancement [13].

Remark 1: All the functions are defined and normalized in the continuous domain. In our application, the filters are discretized and truncated in order to be implemented in the convolution layer. Since they all vanish away from zero, it remains a good approximation, provided that the function's width does not exceed the fixed filter length N.

Remark 2: The modulated window functions are defined with a cosine (real part) and a sine (imaginary part) term, relating them to the Fourier transform, the spectral domain and the standard definition of filters in signal processing. For the sake of simplicity, in our experiments, we have chosen to use only the cosine term. The absence of the sine term did not affect the accuracy of our classification results. The network is able to adapt and detect discriminative patterns with a shifted cosine modulation.

Remark 3: It is important to distinguish the filter length N from the filter temporal width σ or b (or s for the scale). The filter length is fixed, can not be learned and is the size of the vector on which the filter is defined. The temporal width is learned and specifies the spread of the function over the vector of size N. Therefore, the filter temporal width is always smaller than the filter's length.

9.2 Experiments and results

We apply SpectroBank to several classification tasks described in the following sections. We want to assess it on standard tasks found in the literature presented in the introduction. We have chosen 2 freely available speech datasets: *AudioMNIST* [15] and *GoogleSpeechCommands v2* [192]. Both datasets contain words pronounced by different speakers. These datasets are dedicated to limited-vocabulary speech recognition tasks and the goal is to train the network to correctly recognize the word present in each audio sequence. We also investigate the performances of SpectroBank on an environmental sound dataset in order to cover more diverse audio patterns. We have chosen the *UrbanSound8K* dataset [160]. This dataset has been used recently for end-to-end learning [42, 1]. The overall statistics for all the datasets used in the experiments is given in Table 8.2. In addition the distribution of spectral energy per class is provided on Fig. 8.2. Most of the speech energy is located in the 0-1.5kHz frequency band.

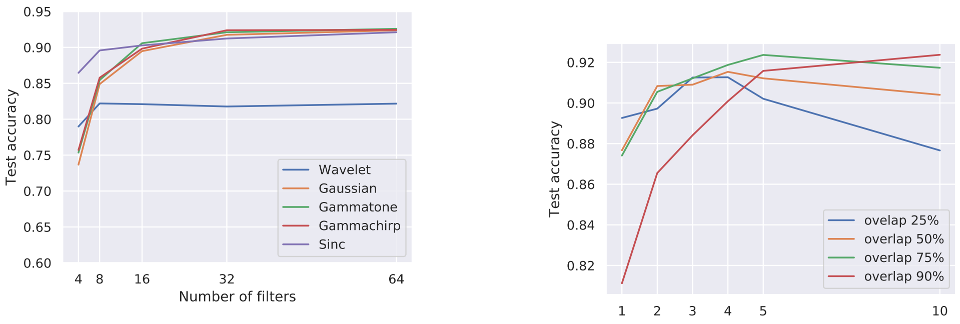
In order to compare the impact of the SpectroBank layer on the learning and classification results, we use existing network architectures and modify the first layer. For networks with raw audio input, the first convolutional layer (performing a standard 1D convolution) is replaced by our proposed parameterized convolution layer. Our layer is then followed by a non-linear ReLU activation function, $y = max(0, x)$ where x is the input and y is the output. A stride parameter is available allowing to define the overlap in time of consecutive convolutions.

Since our focus is on learning from audio, we decided to compare our approach only to similar techniques, despite the fact that image-based network achieve sometimes higher accuracy than the purely audio-based ones. All the models used for the experiments were implemented using the Keras framework [37]. Detailed architectures of all networks can be found in Appendix B. Training was performed using a NVidia GTX1080Ti having 11 GB of RAM.

Input layer initialization. When initializing a filter bank for learning, most of the available solutions start from a filter bank with a Mel-scale (or log-scale) frequency spacing [156, 198, 153, 78]. This frequency distribution is supposed to be optimal for audio processing and learning. However, in the present work, we want to check this assumption. Hence in all the experiments (except when comparing with SincNet where we retain the mel-based initialization from [153]), we use linearly spaced frequencies (or scales), distributed over the entire spectrum and a constant initialization value for the bandwidth (σ or b).

9.2.1 The impact of SpectroBank parameters

The learnable parameters of the SpectroBank filters are not the only values that may influence the network accuracy. The choice of the filter type is important as well as the filter length and the layer stride (filter overlap). We have tested different configurations and the results are shown on Fig. 9.3. On the left, the accuracy increases with the number of filters up to around 30. Beyond this, no improvement is reported. This number is hence a good compromise between accuracy and network complexity. When it comes to observing accuracy on the test set, a similar trend holds for all filter types. These results highlight the better performance of the modulated windows compared to the wavelets. On Fig. 9.3b, the impact of the filter overlap (or kernel stride) is shown, exhibiting two different behaviors. First, for large windows (beyond 5ms), a large stride lead to a drop in accuracy. Indeed, the filter width (spread



(a) Accuracy for different filter types and numbers of filters. The modulated windows filters achieve similar performances and reach a better accuracy than the wavelets.

(b) Accuracy of Gammatone filters for different lengths and overlap ratios.

Figure 9.3 – Influence of several SpectroBank layer properties on the network accuracy. (Dataset: *GoogleSpeechCommands*)

of the modulated window or wavelet) may be much smaller than the filter total length N. Nevertheless, the overlap is measured on the total length. Narrow windows may not overlap at all and information is lost during the convolution process. Secondly, short kernels (less than 4ms) with large overlap (or small stride), can render the network short-sighted in time. In that case, long temporal patterns require the combination of a large amount of successive output values. The convolutional layers following the SpectroBank layer, deeper inside the network, may not be able to capture these long patterns. This results as well in a drop of the accuracy observed on Fig. 9.3b.

9.2.2 *AudioMNIST* Results

The original *AudioMNIST* paper [15] performs digit classification using raw audio as input to a network called AudioNet. The code[2] supplied with the paper has been re-used to perform 5-fold validation on the data. AudioNet is made of six convolutional layers, each convolution being followed by a max-pooling layer, and two dense layers, connected to an output layer. In all tests performed using this dataset, the models were trained using the Adam optimizer with default parameters during 50 epochs. Batch size used was set to 256 and loss function used was the categorical cross-entropy. Test accuracy was then computed after this training phase and the same process was repeated for each fold.

On the *AudioMNIST* dataset sampled at 8 kHz, AudioNet has ca. 17 million trainable parameters. The original paper from [15] claims an accuracy of 92.53% ± 2.04%, whereas our implementation of AudioNet using Keras and Adam optimizer (instead of SGD in the original paper, since using the Adam optimizer gave better results) yields an average accuracy of 94.9% ± 1.54%, which is already a significant improvement. We performed the same 5-fold validation using a modified version of AudioNet where the first convolutional layer is replaced by a *SpectroBank* layer. This layer consists in 32 4th-order Gammatone filters of length 80 (corresponding to 10 ms at 8 kHz). The stride has been set such that the overlap between two consecutive convolution steps is equal to 75%. In this modified network, the number of trainable parameters drops to ca. 3.5 million trainable parameters, i.e. a reduction in size by a factor 5. Using the SpectroBank-enabled AudioNet the average accuracy increases to 96.8% ± 1.22%.

Another SpectroBank-enabled network was used to perform the classification task on *AudioMNIST*. The architecture, loosely adapted from the one used in the paper by [1], is described in Appendix (Table B.1). Despite its much smaller number of trainable parameters (ca. 300'000), its average accuracy improves to 98.0% ± 0.41%. For the sake of completeness, we also trained this network, replacing the Gammatone filters by the learned wavelets as in [78], and the learned SincNet filters from [153]. A summary of all results achieved using AudioMNIST can be found in Table 9.2.

Although that we have shown a reduced number of learnable parameters, number of param-

---

[2] https://github.com/soerenab/AudioMNIST

eters does not necessarily reflect on the complexity of learning. The gradients for all of the learnable parameters in our Spectrobank layer can be computed similar to [78]. All the filter types are differentiable over their parameters that the network learns.

Table 9.2 – *AudioMNIST* mean test accuracy

Network	# Trainable parameters	Avg. accuracy
AudioNet	17 M	94.9% ± 1.54%
SpectroBank-AudioNet	3.5 M	96.8% ± 1.22%
SpectroBank-custom (Gammatone)	**300 k**	**98.0% ± 0.41%**
SpectroBank-custom (SincNet)	300 k	97.2% ± 1.0%
SpectroBank-custom (wavelet)	300 k	89.9% ± 1.18%

9.2.3 *GoogleSpeechCommands* Results

The *GoogleSpeechCommands* dataset provides similar data to the *AudioMNIST* one, with a larger number of classes (35) to distinguish. In the original setting, the goal was to classify 15 unwanted words together as *unknown*. However, in the experiments we performed, we classify each word independently. This dataset does not have pre-defined folds, but train, test and validation data are specified explicitly. We focus on the "*Basic*" network of the *SampleCNN* group described in [79]. Using an input signal resampled to 22.05 kHz, the Basic network has 8 identical blocks, each block being made of a 1D convolution (size 3), followed by a batch normalization, ReLU activation and max pooling. In our experiments, we adapted the proposed setting in order to keep the original 16 kHz sampling of the dataset and ended up with a 7 blocks (vs. 8) network in order to avoid empty dimensions. The code[3] provided by [79] was used as basis for our experiments. Reducing the number of blocks to 7 and keeping the original 16 kHz sampling rate yields networks having similar number of trainable parameters (ca. 2.3 million vs ca. 2.5 million respectively for 7 blocks/16 kHz and 8 blocks/22.05 kHz).

Given that *GoogleSpeechcCommands* does not possess pre-defined folds for n-fold validation, the experiments were repeated 5 times in order to compute the mean accuracy. The original results from [79] give an average accuracy of 92.5% ± 0.7% (averaged over 3 training runs). When reproducing a similar experiment (training performed with SGD optimizer, with early stopping), with the simplified SampleCNN using 16 kHz data, we found the average accuracy to be 93.34% ± 1.26%.

We created a SpectroBank-enabled version of SampleCNN, replacing the first block by a spectrobank layer and modifying the other basic blocks introduced by [79], as described in Appendix B, Table B.5. The SpectroBank layer is made of 80 4th order Gammatone filters, overlapping by 80% and having a length representing 10 ms. As our initial layer contains less

---

[3] https://github.com/tae-jun/sampleaudio

filters than the initial implementation (80 vs. 128), the basic block modifications allow to keep non-empty sizes when the number of basic blocks increases. The number of basic blocks is identical (7), reducing the number of trainable parameters to 1 million. Unlike the original paper, this network was trained using the Adam optimizer, while keeping the same learning rate reduction strategy. The early stopping is usually activated after less than 20 epochs. The mean accuracy achieved using this network improves slightly to 93.45% ± 1.35%. All these results are summarized in Table 9.3.

Table 9.3 – *GoogleSpeechCommands* mean test accuracy

Network	# Trainable parameters	Avg. accuracy
SampleCNN-8Blocks (@22.05kHz)	2.5 M	92.5% ± 0.7%
SampleCNN-7Blocks (@16kHz)	2.3 M	93.34% ± 1.26%
SpectroBank-SampleCNN (@16kHz)	**1 M**	**93.45% ± 1.35%**

9.2.4 *UrbanSound8K* Results

One of the main interests of this dataset resides in the fact that the environmental sounds exhibit spectral characteristics that are quite different from speech datasets studied in the previous sections. It is however a more challenging dataset, firstly because its size is almost an order of magnitude smaller, and secondly because of the longer input data (each sample being 4 seconds long).

We base our experiments on the works from [42] and more recently [1], that also perform classification task using convolutional networks on raw audio input. [42] define several network architectures, with numbers of trainable parameters ranging from 200'000 to 4 millions. We will focus on the two smallest networks, referred to as M3 and M5 in the original paper. Despite dataset being split into 10 folds for training and validation, only one test (using the 10th fold for validation) has been done in [42]. We tested those networks using an existing Keras implementation[4] and performed 10-fold validation to get the mean accuracy over all folds, using data resampled to 8 kHz. The average accuracy for M3 was found to be 58.94% ± 3.83% (vs. 56.12% in the original paper) and the one for M5 66.98% ± 6.37% (vs. 63.4% initially).

SpectroBank-enabled versions of M3 and M5 have been created for comparison. The first layer consists in 24 4th-order Gammatone filters, overlapping by 75% and having a length representing 10ms. All networks were trained for 100 epochs using the Adam optimizer, reducing the learning rate by a factor of 2 after 10 epochs without improvement of the validation loss. Type of the optimizer was chosen according to the setting that has given best performance. SpectroBank-enabled M3 accuracy is very close to the one achieved with initial M3, namely 59.17% ± 5.33%. However, the SpectroBank-M3 has ca. 22'000 parameters, i.e. close to ten

---

[4] https://github.com/philipperemy/very-deep-convnets-raw-waveforms

times less than initial M3. In the case of SpectroBank-M5, mean accuracy is improved to 67.45% ± 5.48% (with a number of trainable parameters very close to initial M5, i.e. slightly more than 500'000). We also tested the SpectroBank-SampleCNN architecture described in section 9.2.3, and achieved a mean accuracy of 69.16% ± 5.95%. When comparing more specifically the 10^{th} fold best accuracy achieved by [42] is 71% using M18 model (3.7 million parameters), while our approach reaches an accuracy of 75.8%. Higher accuracy has been achieved on this dataset using raw audio [98]. However, they resort to data augmentation, which was not used in our experiments. All results are summarized in Table 9.4.

Table 9.4 – *UrbanSound8K* mean test accuracy

Network	# Trainable parameters	Avg. accuracy
M3	222 k	58.94% ± 3.83%
SpectroBank-M3	**22.5 k**	**59.17% ± 5.33%**
M5	561 k	66.98% ± 6.37%
SpectroBank-M5	**513 k**	**67.45% ± 5.48%**
SpectroBank-SampleCNN	**1 M**	**69.16% ± 5.95%**

The approach taken by [1] is to perform classification on overlapping splits of initial audio data (usually having a length of 1 second). They also compare to a network taking a single block of data (having a length of ca. 3 seconds). While the code was supposed to be made available after final publication, the repository[5] was still empty at the date of submission of the thesis. The model was then reimplemented and trained according to the description found in the paper, using all 4 seconds of input data instead of trimming it to 3 seconds. Instead of the mean accuracy claimed (83% ± 1.3%, from Table 2 in original paper), our tests only achieved 63.8% ± 5.68%, which is a significant difference. We have been unable so far to explain this discrepancy.

9.2.5 Properties of learned filters

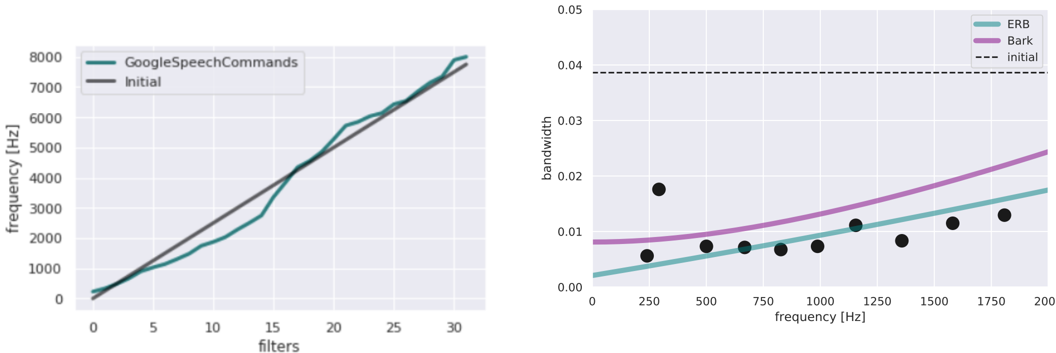
The learned parameters of the SpectroBank filters can reveal insights about the data and the learning process. As stated in the introduction, several studies have shown a tendency governing the spacing in frequency of their learned kernels, approximations of band-pass filters. The spacing becomes exponentially large as the frequency increases, following what is called a Mel scale. This is in agreement with psycho-acoustics tests on the human cochlear system. In order to go further in this direction, we investigate 1) the frequency spacing and 2) we test the relationship between the temporal width of the filters and their central frequency. Indeed, psycho-acoustic models (the Equivalent Rectangular Bandwidth (ERB) model [62] and the Bark model [200]) provide such a relationship. This is made possible by our approach

---

[5] https://github.com/sajabdoli/Environmental_sound_classification, last accessed Oct 27^{th} 2019

where the temporal width as well as the filter central frequency are well defined for each filter.

Frequency spacing. The SpectroBank layer is initialized with a linear frequency spacing from 0 to the Nyquist frequency. After the learning phase, the filter frequencies have evolved and moved away from their initial value as can be seen on Fig. 9.4a. The frequency distribution is not exponential but we can point out several interesting facts. Firstly, the final curve is flatter than the initialization in the range 0-2kHz (more filters in this range). It shows that the network tends to favour filters with a band-pass in this range for its discriminative process. Secondly, beyond 4kHz, the filters stay close to their original value. This suggests that there is not enough meaningful information in this frequency range for a correct learning. This is indeed the case for speech where the main information resides below 4kHz (see Figure 8.2).



(a) Frequency distribution of the filters before (straight line) and after training (green curve)

(b) Bandwidth and frequency of the learned filters (black dots) over the range 0-2kHz. The curves are the psycho-acoustical relationships given by the ERB and Bark scales. Black dashed line: initial bandwidth value for all filters.

Figure 9.4 – Bandwidth and frequency of the learned Gammatone filters (B of Eq. (9.4) and f parameters) using the *GoogleSpeechCommands* dataset

Bandwidth and frequency. The learned filter banks can be compared to filter banks modeling the human auditory system. Two main models can be found in the literature, the ERB model [62], and the Bark model [200]. In these models the bandwidth B of the filter is related to its central frequency f by explicit formulas given in Appendix C. The ERB and Bark curves are plotted on Fig. 9.4b, together with the learned parameters of the Gammatone filters (black dots). We observe a very good agreement between the ERB curve and the learned filters for frequencies below 2kHz. Ravanelli et al. [153] show that for a neural network applied to a speech dataset, the focus of the learning is situated around the pitch frequency located at 130Hz (male) and 230Hz (female), and the first and second formants, which are around 500Hz and 1kHz respectively. A formant is a concentration of acoustic energy around a particular frequency in a speech wave. This is exactly the frequency region where our learned filters match the ERB scale. Although that the patterns are apparent, we can also notice some apparent outliers in the low frequency domain. The interpretation can be twofold: either the amount of data in certain frequency ranges is insignificant for a successful training of the network or

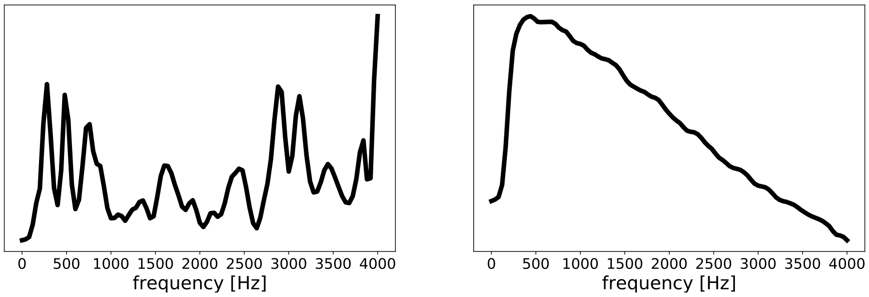


Figure 9.5 – Cumulative frequency energy distribution for learned filters on AudioMNIST dataset, SpectroBank-XS network trained with Gammatone (order 4) and SincNet first layer, with 32 filters.

there are some additional features, beyond the ERB scale, that would complete the picture (such as in [171] that combines multiple fixed audio representations for increased precision of classification).

Cumulative distribution. In Figure 9.5 we can see the cumulative energy distribution, in the frequency domain, of the learned filters for Gammatone and Sinc filters. We have used 32 filters during the training. From the Gammatone distribution, we can observe that filters focus on at least some of the frequencies relevant for speech, as discussed earlier, in the range 100Hz - 1kHz. The sinc distribution has the same global shape as in [153], but is less conclusive about the formants. We also note a difference in the low-frequency region below 100Hz, where the distribution drops in our case. We point out that both our dataset and classification task are different, which could explain the discrepancies. It still shows the high focus of the filters on the range 100Hz - 1kHz, where the distribution curve is the highest.

9.3 Interpretation of network's decision making

On the other hand, researchers have been interested in tackling the questions of what contributes to network's decision and how does information propagate over the layers of the network? In [15] authors observe the relevance propagation inside spectrogram and raw audio representations for an audio classification task. As can be seen in the paper, the highest weights are given to some sort of local extremes. In order to illustrate the behavior of the SpectroBank layer, in Figure 9.6 we have decided to visualize the evolution of the f parameter for a case with synthetic data where the key frequencies are at the red dashed lines, i.e. $\{0.1, 0.2, 0.4\} \times f_s$. Although the frequencies were initialized on a uniform regular grid, throughout the epochs they converge towards the frequencies where the key information is contained. In the case when there are too many filters, some of the filters stay where they were initialized because all the required information was covered by the rest of the filters from the filter bank.

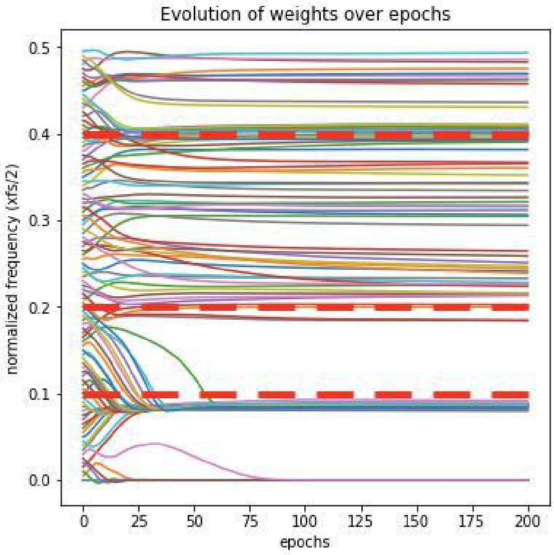


Figure 9.6 – Filter convergence illustration on synthetic data

9.4 Conclusion

Decades of research in audio signal processing have brought important knowledge about sounds, speech and audio information. This knowledge may be inserted within neural networks as a priori information and turned into efficient inductive biases. This is what we show with the example of the SpectroBank layer. This is a layer of parameterized filters adapted to the extraction of audio information. Moreover, the trained network possesses properties than can, in turn, bring new insights about audio data back to the audio signal processing community.

We show that networks having such an input layer can achieve state-of-the-art accuracy on several audio classification tasks. This approach, while reducing the number of weights to be trained along with network training time, enables larger kernel sizes, an advantage for audio applications. Furthermore, the learned filters bring additional interpretability and a better understanding of the data properties exploited by the network.

Future work in this direction and further developments of convolutions with parameterized functions may lead to important progress both in deep learning and audio signal processing. The reduction of the number of trainable parameters decreases the network complexity, along with the training time. It also enables a better interpretation of the network adaptation to the data. The future work will also include the expansion of our method for the application to source separation [127, 49] problem and also to the problem of voice synthesis. The goal of our learnable representation is to have a simple model that would give perceptually accurate results.

On the other hand, the most recent solutions available are able to do speech-to-speech translation without any intermediate text representation, as presented by the solution called Translatotron [76]. Such approaches might also affect the future steps for SpectroBank with further optimizations in the direction of more efficient learning procedures with smaller data and training footprints.

All the results shown in this chapter are reproducible. The code can be found on github[6].

---

[6] https://github.com/epfl-lts2/spectrobank

Part VI

Conclusion and Future Work

Conclusion

All the topics that were discussed within the scope of this thesis can be divided into two parts. The first part addresses problems in acoustics with underlying sparsity. We cover the low-frequency and the mid-high-frequency range processing with the application of the on-grid and off-grid parsimonious methods. The second part of the thesis focuses on the recent trends in machine learning and discusses the potential of learning parametric audio representations.

At the starting point of this thesis, the compressed sensing techniques have been applied to various types of problems: principal component analysis on graphs [170], redundant dictionary design [152], variable density compressive sampling [145], Fourier imaging [144], hyperspectral Imaging [63], image reconstruction from multiview measurements [143], magnetic resonance fingerprinting [44] and many more. However, applications to problems in acoustics were quite coarse.

Over the course of this PhD thesis compressed sensing was used for the estimation of the sound pressure in a room from a limited number of microphones. It has also been used for the estimation of the room geometry as well as for sound source localization inside of the room.
The takeaway message: we can apply a robust technique for the modal characterization of the acoustic behavior of the room. The identified parameters of modes can be further used for the sound source localization of a wideband source inside of the room.

On the other hand, the theory of Finite Rate of Innovation was used to estimate the properties of Diracs within a Room Impulse Response in an off-grid manner for the problem of blind deconvolution. Some initial exploration was done for the case of expansion of this algorithm to the application in a real world setting.
The takeaway message: The locations of early reflections within a Room Impulse Response can be estimated in an off-grid manner which enables maintaining higher accuracy even with cheaper sensors, that sample data at low rates.

We have also built a new statistical echo density measure that characterizes the type of the Acoustic Impulse Response that can be used for better encoding of audio in the domain of Virtual, Augmented or Mixed Reality.
The takeaway message: By using a simple statistical measure that relies on the first moment, we can determine the type of the acoustic environment where the Acoustic Impulse Response

was recorded. This solution can find an application in audio rendering for games or for Virtual reality, in cases where the user is in spaces that are not fully enclosed or fully open, such as: caves, outdoor corridors, courtyards etc.

In the last part of the thesis we have improved the classification of audio for the cases of speech and environmental sounds. We propose a new type of parametric layer for deep neural networks that can be incorporated into different types of deep learning frameworks and can also potentially find an application in the sound synthesis.
The takeaway message: The parametric learning for audio classification has shown tendencies to rely on well established perceptual models, but at the same time has shown slight discrepancies which indicated that there is a need for multi-feature exploration. These types of solutions can be applied in classification of sounds for audio surveillance.

By observing the mathematical background and the structure of acoustical data, we can easily expand the application of proposed methods to other scientific domains. For example, echo detection and estimation has an application not only in acoustics [22, 188] for room shape estimation [50, 41, 40] or beamforming [51, 164, 140], but also in submarine navigation in sonars [84], in seismology [162], in ultrasound imaging [4] and in radioastronomy [125].

A Proof: Binaural weighted norm as an objective function

Here we will prove the equivalence between (eq. 6.7) and (eq. 6.10). We observe the Frobenius norm of the sum of two matrices:

$$\begin{aligned}\|\mathbf{A}+\mathbf{B}\|_F^2 = \|\mathbf{A}\|_F^2 + 2\,\Re(\operatorname{trace}(\mathbf{A}^*\mathbf{B})) + \|\mathbf{B}\|_F^2 = \\ \|\mathbf{A}\|_F^2 + 2\,\Re\Big(\sum_i\sum_j(\mathbf{A}^*\odot\mathbf{B})[i,j]\Big) + \|\mathbf{B}\|_F^2,\end{aligned} \tag{A.1}$$

where $\Re$ denotes the real part of a complex number. This equivalence will be used on parts of proofs with $\overset{ms}{=}$ (ms - matrix sum).

In the following M will denote the number of microphones (channels).

Case M = 2: Starting from the cross-relation $\|\mathbf{H}_1\odot\mathbf{X}_2 - \mathbf{H}_2\odot\mathbf{X}_1\|_F^2$ (eq. 6.7) and going to the Cadzow upgraded weighted matrix norm algorithm from (eq. 6.7) as in [39], we have:

$$\begin{aligned}\|\mathbf{H}_1 - \mathbf{H}_2\odot\mathbf{X}_1\oslash\mathbf{X}_2\|^2_{\mathbf{W}\odot|\mathbf{X}_2|^{\odot 2}} \overset{ms}{=} \|\mathbf{H}_1\|^2_{\mathbf{W}\odot|\mathbf{X}_2|^{\odot 2}} - \\ -2\Re\Big(\sum_i\sum_j(\mathbf{H}_1^*\odot\mathbf{H}_2\odot\mathbf{X}_1\oslash\mathbf{X}_2\odot\mathbf{W}\odot|\mathbf{X}_2|^{\odot 2})[i,j]\Big) + \\ +\|\mathbf{H}_2\odot\mathbf{X}_1\oslash\mathbf{X}_2\|^2_{\mathbf{W}\odot|\mathbf{X}_2|^{\odot 2}}.\end{aligned} \tag{A.2}$$

So if we optimize over $\mathbf{H}_1$, we can neglect the last part of the sum. This is finally equivalent to (eq. 6.10):

$$\|\mathbf{H}_1 - \mathbf{H}_2\odot\mathbf{X}_1\oslash\mathbf{X}_2\|^2_{\mathbf{W}\odot|\mathbf{X}_2|^{\odot 2}}. \tag{A.3}$$

Case M > 2: In a multi-channel setting, we need to include all the available measurements in the retrieval of information. Therefore, for every $m \in \mathbb{M}$ we want to minimize:

$$\|\mathbf{H}_1 - \mathbf{H}_m\odot\mathbf{X}_1\oslash\mathbf{X}_m\|^2_{\mathbf{W}\odot|\mathbf{X}_m|^{\odot 2}}. \tag{A.4}$$

This we will denote as: $C(\mathbf{H}_1,\mathbf{H}_m)$. When solving the multi-channel optimization problem over $\mathbf{H}_1$ as the variable, we have:

$$\begin{aligned} O(m=1) = \sum_{m \in \mathbb{M} \setminus 1} C(\mathbf{H}_1, \mathbf{H}_m) = \sum_{m \in \mathbb{M} \setminus 1} \|\mathbf{H}_1 - \mathbf{H}_m \odot \mathbf{X}_1 \oslash \mathbf{X}_m\|^2_{\mathbf{W} \odot |\mathbf{X}_m|^{\odot 2}} = \\ = \sum_{m \in \mathbb{M} \setminus 1} \left(\|\mathbf{H}_1\|^2_{\mathbf{W} \odot |\mathbf{X}_m|^{\odot 2}} - 2\Re\left(\sum\sum (\mathbf{H}_1^* \odot \mathbf{H}_m \odot \mathbf{X}_1 \oslash \mathbf{X}_m \odot \mathbf{W} \odot |\mathbf{X}_m|^{\odot 2}) \right) + \|\mathbf{H}_m \odot \mathbf{X}_1 \oslash \mathbf{X}_m\|^2_{\mathbf{W} \odot |\mathbf{X}_m|^{\odot 2}} \right) = \\ = \|\mathbf{H}_1\|^2_{\mathbf{W} \odot \sum_{m \in \mathbb{M} \setminus 1} |\mathbf{X}_m|^{\odot 2}} - 2\Re\left(\sum\sum (\mathbf{W} \odot \mathbf{H}_1^* \odot \mathbf{X}_1 \odot \sum_{m \in \mathbb{M} \setminus 1} \mathbf{H}_m \odot \mathbf{X}_m^*) \right) + \sum_{m \in \mathbb{M} \setminus 1} \|\mathbf{H}_m \odot \mathbf{X}_1 \oslash \mathbf{X}_m\|^2_{\mathbf{W} \odot |\mathbf{X}_m|^{\odot 2}}. \end{aligned} \tag{A.5}$$

After introducing $\mathbf{V}_i = \sum_{j \in \mathbb{M} \setminus i} |\mathbf{X}_j|^{\odot 2}$, this is finally equivalent to (eq. 6.11):

$$\|\mathbf{H}_1 - \mathbf{V}_{\mathbb{M}/1}^{\odot -1} \odot \mathbf{X}_1 \odot \sum_{m=2}^{M} \mathbf{H}_m \odot \mathbf{X}_m^* \|^2_{\mathbf{W} \odot \mathbf{V}_{\mathbb{M}/1}} \tag{A.6}$$

B Network architectures

Detailed architecture for SpectroBank-enabled networks used in the experiments are given in this section. All convolutional and dense layers use ReLU activation, except for the output layer using softmax.

Table B.1 – SpectroBank custom architecture for *AudioMNIST*.

Layer	**Output size**
Input	8000 × 1
SpectroBank (32 filters, size 80, stride 20)	400 × 32
Convolution (32 filters, size 32, stride 2)	200 × 32
MaxPooling (stride 4)	50 × 32
Convolution (64 filters, size 16, stride 2)	25 × 64
Convolution (128 filters, size 8, stride 2)	13 × 128
Convolution (256 filters, size 4, stride 2)	7 × 256
MaxPooling (stride 4)	1 × 256
Dense (128)	128
Dropout 0.5	128
Dense (64)	64
Dropout 0.5	64
Dense 10	10

Table B.2 – SpectroBank-XS custom architecture for AudioMNIST.

Layer	Output size
Input	8000 × 1
SpectroBank (32 filters, size 80, stride 20)	400 × 32
MaxPooling (stride 4)	100 × 32
Dense (16)	16
Dropout 0.5	16
Dense 10	10

Table B.3 – M3-SpectroBank custom architecture for *UrbanSound8K*.

Layer	Output size
Input	32000 × 1
SpectroBank (24 filters, size 80, stride 20)	1600 × 24
Batch Normalization	1600 × 24
MaxPooling (stride 4)	400 × 24
Convolution (256 filters, size 3, stride 1)	400 × 256
MaxPooling (stride 4)	100 × 256
Global Average Pooling	256
Dense 10	10

Table B.4 – M5-SpectroBank custom architecture for *UrbanSound8K*.

Layer	**Output size**
Input	32000 × 1
SpectroBank (24 filters, size 80, stride 20)	1600 × 24
Batch Normalization	1600 × 24
MaxPooling (stride 4)	400 × 24
Convolution (128 filters, size 3, stride 1)	400 × 128
Batch Normalization	400 × 128
MaxPooling (stride 4)	100 × 128
Convolution (256 filters, size 3, stride 1)	100 × 256
Batch Normalization	100 × 256
MaxPooling (stride 4)	25 × 256
Convolution (512 filters, size 3, stride 1)	25 × 512
Batch Normalization	25 × 512
MaxPooling (stride 4)	6 × 512
Global Average Pooling	512
Dense 10	10

Table B.5 – SampleCNN-SpectroBank basic block. Choice of k is detailed in [79]

Layer	**Output size**
Input	$N \times d$
Convolution (k filters, size 4, stride 1)	$N \times k$
Batch Normalization	$N \times k$
MaxPooling (stride 2)	$\frac{N}{2} \times k$

Table B.6 – SampleCNN-SpectroBank architecture for *GoogleSpeechCommands* ($n = 35$) or *Urbansound8K* ($n = 10$). *BB* stands for 'Basic Block', and *GMP* for 'Global Max Pooling'

Layer	**Output size**
Input	16000×1
SpectroBank (80 filters, size 160, stride 40)	200×80
Batch Normalization	200×80
BB 0 ($k = 80$)	100×80
BB 1 ($k = 80$)	50×80
BB 2 ($k = 160$)	25×160
BB 3 ($k = 160$)	12×160
BB 4 ($k = 160$)	6×160
BB 5 ($k = 160$)	3×160
BB 6 ($k = 320$)	1×320
Concatenate (GMP(BB 4), GMP(BB 5), GMP(BB 6))	640
Dense	640
Batch Normalization	640
Dropout (0.25)	640
Dense n	n

C Perceptually motivated models: ERB and Bark scales

Two main models of auditory filter bank system provide the expression of a filter bandwidth B with respect to its frequency f. In the Bark model [200] the expression is the following:

$$B_b(f) = 25 + 75[1 + 1.4\left(\frac{f}{1000}\right)^2]^{0.69}, \tag{C.1}$$

and in the ERB (Equivalent Rectangular Bandwidth) scale [62]:

$$B_{ERB}(f) = 24.7(4.37f + 1). \tag{C.2}$$

These expression are the ones used in the present work.

In addition, these auditory models provide expressions for the frequency spacing between consecutive filters, that follow a logarithmic law. For a given filter number k in the set of filters, its frequency can be obtained by using the following formula: $f = 228.846\left(e^{k_{ERB}/9.265} - 1\right)$. This relationship is more often expressed in terms of k as a function of the frequency:

$$k_{ERB} = 9.265 \log\left(1 + \frac{f}{228.846}\right), \tag{C.3}$$

The Bark model has a similar expression:

$$k_b = 13 \arctan\left(0.76\frac{f}{1000}\right) + 3.5 \arctan\left(\frac{f}{7500}\right)^2. \tag{C.4}$$

One can also compare with the Mel-scale. Sampling linearly on the Mel-scale m leads to logarithmic frequency sampling:

$$m = 1127 \ln\left(1 + \frac{f}{700}\right). \tag{C.5}$$

Bibliography

[1] ABDOLI, S., CARDINAL, P., AND KOERICH, A. L. End-to-end environmental sound classification using a 1D convolutional neural network. *Expert Systems with Applications* (2019).

[2] ABED-MERAIM, K., LOUBATON, P., AND MOULINES, E. A subspace algorithm for certain blind identification problems. *IEEE transactions on information theory 43*, 2 (1997), 499–511.

[3] ABEL, J. S., AND HUANG, P. A simple, robust measure of reverberation echo density. In *Audio Engineering Society Convention 121* (Oct. 2006).

[4] ACHIM, A., BUXTON, B., TZAGKARAKIS, G., AND TSAKALIDES, P. Compressive sensing for ultrasound RF echoes using a-stable distributions. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE* (2010), IEEE, pp. 4304–4307.

[5] ADELSON, E. H., AND BERGEN, J. R. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing* (1991), MIT Press, pp. 3–20.

[6] AGHASI, A., AHMED, A., HAND, P., AND JOSHI, B. A convex program for bilinear inversion of sparse vectors. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 8548–8558.

[7] AHMED, A., AGHASI, A., AND HAND, P. Blind deconvolutional phase retrieval via convex programming. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 10030–10040.

[8] AISSA-EL-BEY, A., AND ABED-MERAIM, K. Blind SIMO channel identification using a sparsity criterion. In *IEEE 9th International Workshop on Signal Processing Advances in Wireless Communications(SPAWC)* (2008), IEEE, pp. 271–275.

[9] AJDLER, T., SBAIZ, L., AND VETTERLI, M. The plenacoustic function and its sampling. *IEEE Transactions on Signal Processing 54*, 10 (Oct. 2006), 3790–3804.

[10] 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 (1979), 943–950.

[11] Antonello, N., Sena, E. D., Moonen, M., Naylor, P. A., and van Waterschoot, T. Room impulse response interpolation using a sparse spatio-temporal representation of the sound field. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 25*, 10 (Oct. 2017), 1929–1941.

[12] Antonello, N., van Waterschoot, T., Moonen, M., and Naylor, P. A. Identification of surface acoustic impedances in a reverberant room using the FDTD method. In *14th International Workshop on Acoustic Signal Enhancement (IWAENC)* (2014), IEEE, pp. 114–118.

[13] Baby, D., and Verhulst, S. Machines hear better when they have ears. *arXiv preprint arXiv:1806.01145 abs/1806.01145* (6 2018).

[14] Baraniuk, R. G. Compressive sensing [lecture notes]. *IEEE Signal Processing Magazine 24*, 4 (July 2007), 118–121.

[15] Becker, S., Ackermann, M., Lapuschkin, S., Müller, K., and Samek, W. Interpreting and explaining deep neural networks for classification of audio signals. *CoRR abs/1807.03418* (2018).

[16] Bertin, N., Daudet, L., Emiya, V., and Gribonval, R. *Compressive Sensing in Acoustic Imaging.* Springer International Publishing, Cham, 2015, pp. 169–192.

[17] Bertin, N., Kitić, S., and Gribonval, R. Joint estimation of sound source location and boundary impedance with physics-driven cosparse regularization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2016), IEEE, pp. 6340–6344.

[18] Bhaskar, B. N., Tang, G., and Recht, B. Atomic norm denoising with applications to line spectral estimation. *CoRR abs/1204.0562* (2012).

[19] Blu, T., Dragotti, P. L., Vetterli, M., Marziliano, P., and Coulot, L. Sparse sampling of signal innovations. *IEEE Signal Processing Magazine 25*, 2 (Mar. 2008), 31–40.

[20] Boche, H., Calderbank, R., Kutyniok, G., and Vybral, J. *Compressed Sensing and Its Applications: MATHEON Workshop 2013*, 1st ed. Birkhäuser Basel, 2015.

[21] Borish, J. Extension of the image model to arbitrary polyhedra. *The Journal of the Acoustical Society of America 75*, 6 (1984), 1827–1836.

[22] Boulandet, R., Falourd, X., Rossi, M., and Lissek, H. Localisation des premières réflexions dans une salle par chrono-goniométrie acoustique. *Proceeding of the 10th Congrès Français d'Acoustique* (2010).

[23] BOULANDET, R., MOSIG, J., AND LISSEK, H. *Tunable Electroacoustic Resonators Through Active Impedance Control of Loudspeakers.* PhD thesis, École Polytechnique Fédérale de Lausanne, 2012.

[24] BOYD, S., AND VANDENBERGHE, L. *Convex Optimization.* Cambridge University Press, New York, NY, USA, 2004.

[25] CADZOW, J. A. Signal enhancement-a composite property mapping algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing 36,* 1 (Jan. 1988), 49–62.

[26] CAIAFA, C. F., AND CICHOCKI, A. Multidimensional compressed sensing and their applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 3,* 6 (2013), 355–380.

[27] CANDÈS, E. J., ROMBERG, J., AND TAO, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theor. 52,* 2 (Feb. 2006), 489–509.

[28] CANDÈS, E. J., AND WAKIN, M. B. An introduction to compressive sampling. *IEEE signal processing magazine 25,* 2 (2008), 21–30.

[29] CANDÈS, E. J., WAKIN, M. B., AND BOYD, S. P. Enhancing sparsity by reweighted l 1 minimization. *Journal of Fourier Analysis and Applications 14,* 5 (2008), 877–905.

[30] CANDÈS, E. J., ELDAR, Y. C., AND NEEDELL, D. Compressed sensing with coherent and redundant dictionaries. *CoRR abs/1005.2613* (2010).

[31] CHANDNA, P., MIRON, M., JANER, J., AND GÓMEZ, E. Monoaural audio source separation using deep convolutional neural networks. In *International conference on latent variable analysis and signal separation* (2017), Springer, pp. 258–266.

[32] CHANDRASEKARAN, V., RECHT, B., PARRILO, P., AND WILLSKY, A. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics 12* (12 2010).

[33] CHI, Y. Guaranteed blind sparse spikes deconvolution via lifting and convex optimization. *IEEE Journal of Selected Topics in Signal Processing 10,* 4 (2016), 782–794.

[34] CHI, Y., PEZESHKI, A., SCHARF, L., AND CALDERBANK, R. Sensitivity to basis mismatch in compressed sensing. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing* (Mar. 2010), pp. 3930–3933.

[35] CHITANONT, N., YATABE, K., ISHIKAWA, K., AND OIKAWA, Y. Spatio-temporal filter bank for visualizing audible sound field by schlieren method. *Applied Acoustics 115* (1 2017), 109–120.

[36] CHOI, K., FAZEKAS, G., CHO, K., AND SANDLER, M. A tutorial on deep learning for music information retrieval. *arXiv preprint arXiv:1709.04396* (2017).

[37] CHOLLET, F., ET AL. Keras. https://keras.io, 2015.

[38] CHOUDHARY, S., AND MITRA, U. On the properties of the rank-two null space of nonsparse and canonical-sparse blind deconvolution. *IEEE Transactions on Signal Processing 66*, 14 (2018), 3696–3709.

[39] CONDAT, L., AND HIRABAYASHI, A. Cadzow denoising upgraded: A new projection method for the recovery of dirac pulses from noisy linear measurements. *Sampling Theory in Signal and Image Processing 14*, 1 (2014), 17–47.

[40] CROCCO, M., AND DEL BUE, A. Estimation of TDOA for room reflections by iterative weighted L1 constraint. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2016), IEEE, pp. 3201–3205.

[41] CROCCO, M., TRUCCO, A., MURINO, V., AND DEL BUE, A. Towards fully uncalibrated room reconstruction with sound. In *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)* (2014), IEEE, pp. 910–914.

[42] DAI, W., DAI, C., QU, S., LI, J., AND DAS, S. Very deep convolutional neural networks for raw waveforms. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), vol. abs/1610.00087, IEEE, pp. 421–425.

[43] DARLING, A. Properties and implementation of the gammatone filter: a tutorial. *Speech Hearing and Language, Work in Progress, University College London, Department of Phonetics and Linguistics* (1991), 43–61.

[44] DAVIES, M., PUY, G., VANDERGHEYNST, P., AND WIAUX, Y. A compressed sensing framework for magnetic resonance fingerprinting. *SIAM Journal on Imaging Sciences 7* (12 2013).

[45] DAVIS, S. B., AND MERMELSTEIN, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *ACOUSTICS, SPEECH AND SIGNAL PROCESSING, IEEE TRANSACTIONS ON* (1980), 357–366.

[46] DEFOSSEZ, A., ZEGHIDOUR, N., USUNIER, N., BOTTOU, L., AND BACH, F. Sing: Symbol-to-instrument neural generator. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 9041–9051.

[47] DEFRANCE, G., DAUDET, L., AND POLACK, J.-D. Using matching pursuit for estimating mixing time within room impulse responses. *Acta Acustica united with Acustica 95*, 6 (2009), 1071–1081.

[48] DIELEMAN, S., AND SCHRAUWEN, B. End-to-end learning for music audio. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2014), IEEE, pp. 6964–6968.

[49] DITTER, D., AND GERKMANN, T. A multi-phase gammatone filterbank for speech separation via tasnet, 2019.

[50] DOKMANIĆ, I., PARHIZKAR, R., WALTHER, A., LU, Y. M., AND VETTERLI, M. Acoustic echoes reveal room shape. *Proceedings of the National Academy of Sciences 110*, 30 (2013), 12186–12191.

[51] DOKMANIĆ, I., SCHEIBLER, R., AND VETTERLI, M. Raking the cocktail party. *IEEE Journal of Selected Topics in Signal Processing 9*, 5 (2015), 825–836.

[52] DONOHO, D. L. Compressed sensing. *IEEE Trans. Information Theory 52*, 4 (2006), 1289–1306.

[53] ELAD, M. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[54] ENGEL, J., RESNICK, C., ROBERTS, A., DIELEMAN, S., ECK, D., SIMONYAN, K., AND NOROUZI, M. Neural audio synthesis of musical notes with wavenet autoencoders, 2017.

[55] ERDOGAN, H., HERSHEY, J. R., WATANABE, S., AND ROUX, J. L. Deep recurrent networks for separation and recognition of single-channel speech in nonstationary background audio. In *New Era for Robust Speech Recognition, Exploiting Deep Learning* (2017).

[56] FARLOW, S. J. *Partial differential equations for scientists and engineers.* Dover Books on Mathematics. Dover, New York, NY, 1993.

[57] FOUCART, S., AND RAUHUT, H. *A Mathematical Introduction to Compressive Sensing.* Birkh&#228;user Basel, 2013.

[58] GADE, A. Acoustics in Halls for Speech and Music. In *Springer Handbook of Acoustics*, T. Rossing, Ed., 2007 ed. Springer, May 2007, ch. 9.

[59] GAROFOLO, J. S., LAMEL, L. F., FISHER, W. M., FISCUS, J. G., AND PALLETT, D. S. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon technical report n 93* (1993).

[60] GENOVESE, A., GAMPER, H., PULKKI, V., RAGHUVANSHI, N., AND TASHEV, I. Blind room volume estimation from single-channel noisy speech. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (May 2019), IEEE.

[61] GERSTOFT, P., MECKLENBRÄUKER, C. F., SEONG, W., AND BIANCO, M. Introduction to compressive sensing in acoustics. *The Journal of the Acoustical Society of America 143*, 6 (2018), 3731–3736.

[62] GLASBERG, B. R., AND MOORE, B. C. Derivation of auditory filter shapes from notched-noise data. *Hearing research 47*, 1-2 (1990), 103–138.

[63] GOLBABAEE, M., ARBERET, S., AND VANDERGHEYNST, P. Compressive source separation: Theory and methods for hyperspectral imaging. *IEEE Transactions on Image Processing 22*, 12 (Dec. 2013), 5096–5110.

[64] GOLIK, P., TÜSKE, Z., SCHLÜTER, R., AND NEY, H. Convolutional neural networks for acoustic modeling of raw time signal in lvcsr. In *Sixteenth annual conference of the international speech communication association* (2015).

[65] GOODFELLOW, I., BENGIO, Y., COURVILLE, A., AND BENGIO, Y. *Deep learning*, vol. 1. MIT Press, 2016.

[66] HAMILTON, B., BILBAO, S., HAMILTON, B., AND BILBAO, S. Fdtd methods for 3-d room acoustics simulation with high-order accuracy in space and time. *IEEE/ACM Trans. Audio, Speech and Lang. Proc. 25*, 11 (Nov. 2017), 2112–2124.

[67] HECKEL, R., AND SOLTANOLKOTABI, M. Generalized line spectral estimation via convex optimization. *IEEE Transactions on Information Theory 64*, 6 (June 2018), 4001–4023.

[68] HOHMANN, V. Frequency analysis and synthesis using a gammatone filterbank. *Acta Acustica united with Acustica 88*, 3 (5 2002), 433–442.

[69] HOSHEN, Y., WEISS, R. J., AND WILSON, K. W. Speech acoustic modeling from raw multichannel waveforms. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015), IEEE, pp. 4624–4628.

[70] HUA, Y. Estimating two-dimensional frequencies by matrix enhancement and matrix pencil. *IEEE Transactions on Signal Processing 40*, 9 (Sept. 1992), 2267–2280.

[71] HUA, Y. Fast maximum likelihood for blind identification of multiple FIR channels. *IEEE transactions on Signal Processing 44*, 3 (1996), 661–672.

[72] IHLENBURG, F., CIORANESCU, D., LLOYD, G., AND PAULIN, J. *Finite Element Analysis of Acoustic Scattering*. Applied Mathematical Sciences. Springer, 1998.

[73] IRINO, T., AND PATTERSON, R. D. A time-domain, level-dependent auditory filter: The gammachirp. *The Journal of the Acoustical Society of America 101*, 1 (1997), 412–419.

[74] JAIN, P., TEWARI, A., AND DHILLON, I. S. Orthogonal matching pursuit with replacement. In *Proceedings of the 24th International Conference on Neural Information Processing Systems* (USA, 2011), NIPS'11, Curran Associates Inc., pp. 1215–1223.

[75] JASPAN, O., FLEYSHER, R., AND LIPTON, M. Compressed sensing mri: A review of the clinical literature. *The British journal of radiology 88* (09 2015), 20150487.

[76] JIA, Y., WEISS, R. J., BIADSY, F., MACHEREY, W., JOHNSON, M., CHEN, Z., AND WU, Y. Direct speech-to-speech translation with a sequence-to-sequence model. *CoRR abs/1904.06037* (2019).

[77] Kammoun, A., El Bey, A. A., Abed-Meraim, K., and Affes, S. Robustness of blind subspace based techniques using Lp quasi-norms. In *IEEE 11th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)* (2010), IEEE, pp. 1–5.

[78] Khan, H., and Yener, B. Learning filter widths of spectral decompositions with wavelets. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 4601–4612.

[79] Kim, T., Lee, J., and Nam, J. Comparison and analysis of samplecnn architectures for audio classification. *IEEE Journal of Selected Topics in Signal Processing 13*, 2 (May 2019), 285–297.

[80] Kinoshita, K., Delcroix, M., Yoshioka, T., Nakatani, T., Sehr, A., Kellermann, W., and Maas, R. The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on* (2013), IEEE, pp. 1–4.

[81] Kitic, S., Albera, L., Bertin, N., and Gribonval, R. Physics-driven inverse problems made tractable with cosparse regularization. *IEEE Trans. Signal Processing 64*, 2 (2016), 335–348.

[82] Kitić, S., Albera, L., Bertin, N., and Gribonval, R. Physics-driven inverse problems made tractable with cosparse regularization. *IEEE Transactions on Signal Processing 64*, 2 (Jan. 2016), 335–348.

[83] Kitić, S., Bertin, N., and Gribonval, R. Hearing behind walls: Localizing sources in the room next door with cosparsity. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (May 2014), pp. 3087–3091.

[84] Kleeman, L., and Kuc, R. Sonar sensing. In *Springer handbook of robotics*. Springer, 2016, pp. 753–782.

[85] Knapp, C., and Carter, G. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing 24*, 4 (Aug. 1976), 320–327.

[86] Kodrasi, I., Jukić, A., and Doclo, S. Robust sparsity-promoting acoustic multichannel equalization for speech dereverberation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Mar. 2016), pp. 166–170.

[87] Kowalczyk, K., Habets, E. A., Kellermann, W., and Naylor, P. A. Blind system identification using sparse learning for tdoa estimation of room reflections. *IEEE Signal Processing Letters 20*, 7 (2013), 653–656.

[88] Koyano, Y., Yatabe, K., Ikeda, Y., and Oikawa, Y. Physical-model based efficient data representation for many-channel microphone array. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Mar. 2016), pp. 370–374.

[89] Kreković, M., Dokmanić, I., and Vetterli, M. Omnidirectional bats, point-to-plane distances, and the price of uniqueness. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Mar. 2017), pp. 3261–3265.

[90] Krokstad, A., Strom, S., and Sørsdal, S. Calculating the acoustical room response by the use of a ray tracing technique. *Journal of Sound Vibration 8*, 1 (July 1968), 118–125.

[91] Kuttruff, H. *Room Acoustics*, 4 ed. Taylor & Francis, 2000.

[92] Laitinen, M.-V., Pihlajamäki, T., Erkut, C., and Pulkki, V. Parametric time-frequency representation of spatial sound in virtual worlds. *ACM Trans. Appl. Percept. 9*, 2 (June 2012), 8:1–8:20.

[93] Larson, M. G., and Bengzon, F. *The Finite Element Method: Theory, Implementation, and Applications.* Springer Publishing Company, Incorporated, 2013.

[94] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature 521* (5 2015), 436–44.

[95] Lee, J., and Nam, J. Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging. *IEEE signal processing letters 24*, 8 (2017), 1208–1212.

[96] Lee, K., Tian, N., and Romberg, J. Fast and guaranteed blind multichannel deconvolution under a bilinear system model. *IEEE Transactions on Information Theory 64*, 7 (2018), 4792–4818.

[97] Lewy, H., Friedrichs, K., and Courant, R. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen 100* (1928), 32–74.

[98] Li, S., Yao, Y., Hu, J., Liu, G., Yao, X., and Hu, J. An Ensemble Stacked Convolutional Neural Network Model for Environmental Event Sound Recognition. *Applied Sciences 8*, 7 (July 2018), 1152.

[99] Li, X., Gannot, S., Girin, L., and Horaud, R. Multichannel identification and non-negative equalization for dereverberation and noise reduction based on convolutive transfer function. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 26*, 10 (2018), 1755–1768.

[100] Li, X., Girin, L., Horaud, R., and Gannot, S. Binaural sound source localization based on direct-path relative transfer function. *ArXiv abs/1509.03205* (2015).

[101] Li, Y., and Bresler, Y. Global geometry of multichannel sparse blind deconvolution on the sphere. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 1132–1143.

[102] LIN, Y., CHEN, J., KIM, Y., AND LEE, D. D. Blind channel identification for speech dereverberation using l1-norm sparse learning. In *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2008, pp. 921–928.

[103] LINDAU, A., KOSANKE, L., AND WEINZIERL, S. Perceptual evaluation of physical predictors of the mixing time in binaural room impulse responses. In *Audio Engineering Society Convention 128* (May 2010).

[104] LITOVSKY, R. Y., COLBURN, S. H., YOST, W. A., AND GUZMAN, S. J. The precedence effect. *The Journal of the Acoustical Society of America 106*, 4 (1999), 1633–1654.

[105] LOSTANLEN, V., SALAMON, J., CARTWRIGHT, M., MCFEE, B., FARNSWORTH, A., KELLING, S., AND BELLO, J. P. Per-channel energy normalization: Why and how. *IEEE Signal Processing Letters 26*, 1 (Jan. 2019), 39–43.

[106] LU, X., TSAO, Y., MATSUDA, S., AND HORI, C. Speech enhancement based on deep denoising autoencoder. In *Interspeech* (2013), pp. 436–440.

[107] LUO, Y., AND MESGARANI, N. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 696–700.

[108] MALLAT, S. G., AND ZHANG, Z. Matching pursuits with time-frequency dictionaries. *Trans. Sig. Proc. 41*, 12 (Dec. 1993), 3397–3415.

[109] MARMAROLI, P., CARMONA, M., ODOBEZ, J. M., FALOURD, X., AND LISSEK, H. Observation of vehicle axles through pass-by noise: A strategy of microphone array design. *IEEE Transactions on Intelligent Transportation Systems 14*, 4 (Dec. 2013), 1654–1664.

[110] MARMAROLI, P., ODOBEZ, J. M., FALOURD, X., AND LISSEK, H. A bimodal sound source model for vehicle tracking in traffic monitoring. In *2011 19th European Signal Processing Conference* (Aug. 2011), pp. 1327–1331.

[111] MEHRA, R., RAGHUVANSHI, N., CHANDAK, A., ALBERT, D. G., KEITH WILSON, D., AND MANOCHA, D. Acoustic pulse propagation in an urban environment using a three-dimensional numerical simulation. *The Journal of the Acoustical Society of America 135*, 6 (2014), 3231–3242.

[112] MIGNOT, R., CHARDON, G., AND DAUDET, L. Low frequency interpolation of room impulse responses using compressed sensing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 22*, 1 (Jan. 2014), 205–216.

[113] MOIOLA, A., HIPTMAIR, R., AND PERUGIA, I. Plane wave approximation of homogeneous helmholtz solutions. *Zeitschrift für angewandte Mathematik und Physik 62*, 5 (July 2011), 809.

[114] MORSE, P., AND INGARD, K. *Theoretical Acoustics.* International series in pure and applied physics. McGraw-Hill, 1971.

[115] MURPHY, D. T., AND SHELLEY, S. Openair: An interactive auralization web resource and database. In *Audio Engineering Society Convention 129* (2010), Audio Engineering Society.

[116] MÜLLER, S., AND MASSARANI, P. Transfer-function measurement with sweeps. *Journal of the Audio Engineering Society 49* (6 2001).

[117] NAM, S., DAVIES, M. E., ELAD, M., AND GRIBONVAL, R. The Cosparse Analysis Model and Algorithms. *Applied and Computational Harmonic Analysis* (2012). Preprint available on arXiv since 24 Jun 2011.

[118] NAM, S., AND GRIBONVAL, R. Physics-driven structured cosparse modeling for source localization. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Mar. 2012), pp. 5397–5400.

[119] NEEDELL, D., AND TROPP, J. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis 26*, 3 (2009), 301–321.

[120] NYQUIST, H. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers 47*, 2 (Apr. 1928), 617–644.

[121] 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. *arXiv preprint arXiv:1609.03499* (2016).

[122] OORD, A. V. D., LI, Y., BABUSCHKIN, I., SIMONYAN, K., VINYALS, O., KAVUKCUOGLU, K., DRIESSCHE, G. V. D., LOCKHART, E., COBO, L. C., STIMBERG, F., ET AL. Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433* (2017).

[123] PAINE, T. L., KHORRAMI, P., CHANG, S., ZHANG, Y., RAMACHANDRAN, P., HASEGAWA-JOHNSON, M. A., AND HUANG, T. S. Fast wavenet generation algorithm. *arXiv preprint arXiv:1611.09482* (2016).

[124] PAL, D. K., AND MENGSHOEL, O. J. Stochastic cosamp: Randomizing greedy pursuit for sparse signal recovery. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I* (2016), pp. 761–776.

[125] PAN, H., BLU, T., AND VETTERLI, M. Towards generalized fri sampling with an application to source resolution in radioastronomy. *IEEE Transactions on Signal Processing 65*, 4 (Feb. 2017), 821–835.

[126] PAN, H., SCHEIBLER, R., BEZZAM, E. F., DOKMANIC, I., AND VETTERLI, M. Frida: Fri-based doa estimation for arbitrary array layouts. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), 5. 3186–3190.

[127] PARIENTE, M., CORNELL, S., DELEFORGE, A., AND VINCENT, E. Filterbank design for end-to-end speech separation, 10 2019.

[128] PATTERSON, R. D., NIMMO-SMITH, I., HOLDSWORTH, J., AND RICE, P. An efficient auditory filterbank based on the gammatone function. *a meeting of the IOC Speech Group on Auditory Modelling at RSRE 2*, 7 (1987).

[129] PATTERSON, R. D., ROBINSON, K., HOLDSWORTH, J., MCKEOWN, D., ZHANG, C., AND ALLERHAND, M. Complex sounds and auditory images. In *Auditory physiology and perception*. Elsevier, 1992, pp. 429–446.

[130] PEEL, T., EMIYA, V., RALAIVOLA, L., AND ANTHOINE, S. Matching pursuit with stochastic selection. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)* (Aug. 2012), pp. 879–883.

[131] PEIC TUKULJAC, H., DELEFORGE, A., AND GRIBONVAL, R. Mulan: A blind and off-grid method for multichannel echo retrieval. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 2182–2192.

[132] PEIC TUKULJAC, H., DOKMANIC, I., AND RANIERI, J. Time-varying fri theory for sound source localization. Tech. rep., École Polytechnique Fédérale de Lausanne, 2015.

[133] PEIC TUKULJAC, H., LISSEK, H., AND VANDERGHEYNST, P. Localization of sound sources in a room with one microphone. *Wavelets And Sparsity Xvii 10394* (2017), 13. 1039401.

[134] PEIC TUKULJAC, H., PULKKI, V., GAMPER, H., GODIN, K., TASHEV, I. J., AND RAGHUVANSHI, N. A sparsity measure for echo density growth in general environments. *2019 Ieee International Conference On Acoustics, Speech And Signal Processing (Icassp)* (2019), 226–230.

[135] PEIC TUKULJAC, H., RICAUD, B., ASPERT, N., AND VANDERGHEYNST, P. Spectrobank: A filter-bank convolutional layer for {cnn}-based audio applications. In *Submitted to International Conference on Learning Representations* (2020). under review.

[136] PEIC TUKULJAC, H., VU, T. P., LISSEK, H., AND VANDERGHEYNST, P. Joint estimation of the room geometry and modes with compressed sensing. *Proceedings of the 2018 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP)* (2018), 6882–6886.

[137] PICZAK, K. J. Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)* (2015), IEEE, pp. 1–6.

[138] POLACK, J.-D. Playing billiards in the concert hall: The mathematical foundations of geometrical room acoustics. *Applied Acoustics 38,* 2 (1993), 235–244.

[139] POTAMIANOS, A., AND MARAGOS, P. Speech formant frequency and bandwidth tracking using multiband energy demodulation. *1995 International Conference on Acoustics, Speech, and Signal Processing 1* (1995), 784–787 vol.1.

[140] PRICE, R., AND GREEN, P. E. A communication technique for multipath channels. *Proceedings of the IRE 46,* 3 (1958), 555–570.

[141] PRONY, B. G. R. Essai expérimental et analytique sur les lois de la dilatabilité des fluides élastiques et sur celles de la force expansive de la vapeur de l'eau et la vapeur de l'alkool, à différentes températures, 1795.

[142] PURWINS, H., LI, B., VIRTANEN, T., SCHLÜTER, J., CHANG, S.-Y., AND SAINATH, T. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing 13,* 2 (2019), 206–219.

[143] PUY, G., AND VANDERGHEYNST, P. Robust image reconstruction from multiview measurements. *SIAM Journal on Imaging Sciences 7* (12 2012).

[144] PUY, G., VANDERGHEYNST, P., GRIBONVAL, R., AND WIAUX, Y. Universal and efficient compressed sensing by spread spectrum and application to realistic fourier imaging techniques. *EURASIP Journal on Advances in Signal Processing 2012,* 6 (2012).

[145] PUY, G., VANDERGHEYNST, P., AND WIAUX, Y. On variable density compressive sampling. *IEEE Signal Processing Letters 18,* 10 (Oct. 2011), 595–598.

[146] RAGHUVANSHI, N., MEHRA, R., MANOCHA, D., AND LIN, M. Adaptive rectangular decomposition: A spectral, domain-decomposition approach for fast wave solution on complex scenes. *The Journal of the Acoustical Society of America 132* (9 2012), 1890.

[147] RAGHUVANSHI, N., NARAIN, R., AND LIN, M. C. Efficient and Accurate Sound Propagation Using Adaptive Rectangular Decomposition. *IEEE Transactions on Visualization and Computer Graphics 15,* 5 (2009), 789–801.

[148] RAGHUVANSHI, N., NARAIN, R., AND LIN, M. C. Efficient and accurate sound propagation using adaptive rectangular decomposition. *IEEE Transactions on Visualization and Computer Graphics 15,* 5 (Sept. 2009), 789–801.

[149] RAGHUVANSHI, N., AND SNYDER, J. Parametric wave field coding for precomputed sound propagation. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2014 33* (July 2014).

[150] RAGHUVANSHI, N., AND SNYDER, J. Parametric Wave Field Coding for Precomputed Sound Propagation. *ACM Trans. Graph. 33,* 4 (July 2014).

[151] RAGHUVANSHI, N., AND SNYDER, J. Parametric directional coding for precomputed sound propagation. *ACM Trans. Graph. 37*, 4 (July 2018), 108:1–108:14.

[152] RAUHUT, H., SCHNASS, K., AND VANDERGHEYNST, P. Compressed sensing and redundant dictionaries. *IEEE Trans. Inf. Theor. 54*, 5 (May 2008), 2210–2219.

[153] RAVANELLI, M., AND BENGIO, Y. Speaker recognition from raw waveform with sincnet. *2018 IEEE Spoken Language Technology Workshop (SLT)* (2018), 1021–1028.

[154] RICHARDSON, M. H., AND FORMENTI, D. L. Global curve fitting of frequency response measurements using the rational fraction polynomial method, 1985.

[155] RIVET, E. T. J. L. *Room Modal Equalisation with Electroacoustic Absorbers.* PhD thesis, École Polytechnique Fédérale de Lausanne, 2016.

[156] SAINATH, T. N., KINGSBURY, B., MOHAMED, A.-R., AND RAMABHADRAN, B. Learning filter banks within a deep neural network framework. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding* (2013), IEEE, pp. 297–302.

[157] SAINATH, T. N., WEISS, R. J., SENIOR, A., WILSON, K. W., AND VINYALS, O. Learning the speech front-end with raw waveform cldnns. In *Sixteenth Annual Conference of the International Speech Communication Association* (2015).

[158] SALAMON, J., BELLO, J., FARNSWORTH, A., ROBBINS, M., KEEN, S., KLINCK, H., AND KELLING, S. Towards the automatic classification of avian flight calls for bioacoustic monitoring. *PLoS ONE 11* (11 2016), e0166866.

[159] SALAMON, J., AND BELLO, J. P. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters 24*, 3 (2017), 279–283.

[160] SALAMON, J., JACOBY, C., AND BELLO, J. P. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia* (2014), ACM, pp. 1041–1044.

[161] SAREMI, A., BEUTELMANN, R., DIETZ, M., ASHIDA, G., KRETZBERG, J., AND VERHULST, S. A comparative study of seven human cochlear filter models. *The Journal of the Acoustical Society of America 140*, 3 (2016), 1618–1634.

[162] SATO, H., FEHLER, M. C., AND MAEDA, T. *Seismic wave propagation and scattering in the heterogeneous earth*, vol. 496. Springer, 2012.

[163] SCHEIBLER, R., BEZZAM, E., AND DOKMANIC, I. Pyroomacoustics: A python package for audio room simulations and array processing algorithms. *CoRR abs/1710.04196* (2017).

[164] SCHEIBLER, R., DI CARLO, D., DELEFORGE, A., AND DOKMANIĆ, I. Separake: Source separation with a little help from echoes. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE.

[165] SCHLECHT, S. J., AND HABETS, E. A. P. Feedback delay networks: Echo density and mixing time. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 25*, 2 (Feb. 2017), 374–383.

[166] SCHMIDHUBER, J. Deep learning in neural networks: An overview. *CoRR abs/1404.7828* (2014).

[167] SCHNASS, K., AND VANDERGHEYNST, P. Dictionary preconditioning for greedy algorithms. *IEEE Transactions on Signal Processing 56*, 5 (May 2008), 1994–2002.

[168] SEKI, H., YAMAMOTO, K., AND NAKAGAWA, S. A deep neural network integrated with filterbank learning for speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Mar. 2017), pp. 5480–5484.

[169] SEMECHKO, A. Suite of functions to perform uniform sampling of a sphere v 1.3, online. https://ch.mathworks.com/matlabcentral/fileexchange/37004-suite-of-functions-to-perform-uniform-sampling-of-a-sphere, 2015. [Online; accessed 05-October-2017].

[170] SHAHID, N., PERRAUDIN, N., PUY, G., AND VANDERGHEYNST, P. Compressive PCA on graphs. *CoRR abs/1602.02070* (2016).

[171] SHARMA, J., GRANMO, O.-C., AND GOODWIN, M. Environment sound classification using multiple feature channels and deep convolutional neural networks, 8 2019.

[172] SPRATT, K., AND ABEL, J. A digital reverberator modeled after the scattering of acoustic waves by trees in a forest. *Audio Engineering Society - 125th Audio Engineering Society Convention 2008 2* (1 2008), 1284–1293.

[173] STEVENS, F., MURPHY, D. T., SAVIOJA, L., AND VÄLIMÄKI, V. Modeling sparsely reflecting outdoor acoustic scenes using the waveguide web. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 25*, 8 (Aug. 2017), 1566–1578.

[174] STOICA, P. List of references on spectral line analysis. *Signal Processing 31* (1993), 329–340.

[175] STOICA, P., AND MOSES, R. L. *Introduction to spectral analysis.* Prentice Hall Upper Saddle River, NJ, 1997.

[176] STOLLER, D., EWERT, S., AND DIXON, S. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv preprint arXiv:1806.03185* (2018).

[177] TANG, G., BHASKAR, B. N., AND RECHT, B. Near minimax line spectral estimation. In *2013 47th Annual Conference on Information Sciences and Systems (CISS)* (Mar. 2013), pp. 1–6.

[178] TANG, G., BHASKAR, B. N., SHAH, P., AND RECHT, B. Compressed sensing off the grid. *IEEE Transactions on Information Theory 59*, 11 (Nov. 2013), 7465–7490.

[179] TIBSHIRANI, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), 267–288.

[180] TROPP, J. A. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory 52*, 3 (Mar. 2006), 1030–1051.

[181] TROPP, J. A. On the conditioning of random subdictionaries. *Applied and Computational Harmonic Analysis 25*, 1 (2008), 1–24.

[182] TROPP, J. A., GILBERT, A. C., AND STRAUSS, M. J. Algorithms for simultaneous sparse approximation: Part i: Greedy pursuit. *Signal Process. 86*, 3 (Mar. 2006), 572–588.

[183] TROPP, J. A., AND WRIGHT, S. J. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE 98*, 6 (June 2010), 948–958.

[184] TÜSKE, Z., GOLIK, P., SCHLÜTER, R., AND NEY, H. Acoustic modeling with deep neural networks using raw time signal for lvcsr. In *Fifteenth annual conference of the international speech communication association* (2014).

[185] VAIRETTI, G., SENA, E. D., CATRYSSE, M., JENSEN, S. H., MOONEN, M., AND VAN WATERSCHOOT, T. A scalable algorithm for physically motivated and sparse approximation of room impulse responses with orthonormal basis functions. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 25*, 7 (July 2017), 1547–1561.

[186] VALIMAKI, V., PARKER, J. D., SAVIOJA, L., SMITH, J. O., AND ABEL, J. S. Fifty years of artificial reverberation. *IEEE Transactions on Audio, Speech, and Language Processing 20*, 5 (July 2012), 1421–1448.

[187] VAN DENBOOMGAARD, R., AND VAN DERWEIJ, R. Gaussian convolutions numerical approximations based on interpolation. In *Scale-Space and Morphology in Computer Vision: Third International Conference, Scale-Space 2001 Vancouver, Canada, July 7–8, 2001 Proceedings 3* (2001), Springer, pp. 205–214.

[188] VAN LANCKER, E. *Acoustic goniometry a spatio-temporal approach.* PhD thesis, École Polytechnique Fédérale de Lausanne, Lausanne, 2002.

[189] VANDENBERGHE, L., AND BOYD, S. Semidefinite programming. *SIAM Rev. 38*, 1 (Mar. 1996), 49–95.

[190] VETTERLI, M., KOVACEVIC, J., AND GOYAL, V. K. *Foundations of Signal Processing.* Cambridge Univ. Press, Cambridge, 2014.

[191] VETTERLI, M., MARZILIANO, P., AND BLU, T. Sampling signals with finite rate of innovation. *IEEE transactions on Signal Processing 50*, 6 (2002), 1417–1428.

[192] WARDEN, P. Speech commands: A dataset for limited-vocabulary speech recognition. *CoRR abs/1804.03209* (2018).

[193] X. Falourd, L. Rohr, M. R., and Lissek, H. Spatial echogram analysis of a small auditorium with observations on the dispersion of early reflections. *Inter-Noise 2010 - noise and sustainability* (June 2010).

[194] Xenaki, A., and Gerstoft, P. Grid-free compressive beamforming. *The Journal of the Acoustical Society of America 137*, 4 (2015), 1923–1935.

[195] Xenaki, A., Gerstoft, P., and Mosegaard, K. Compressive beamforming. *The Journal of the Acoustical Society of America 136*, 1 (2014), 260–271.

[196] Xu, G., Liu, H., Tong, L., and Kailath, T. A least-squares approach to blind channel identification. *IEEE Transactions on signal processing 43*, 12 (1995), 2982–2993.

[197] Ye, J. C., Kim, J. M., Jin, K. H., and Lee, K. Compressive sampling using annihilating filter-based low-rank interpolation. *IEEE Transactions on Information Theory 63*, 2 (Feb. 2017), 777–801.

[198] Zeghidour, N., Usunier, N., Kokkinos, I., Schaiz, T., Synnaeve, G., and Dupoux, E. Learning filterbanks from raw speech for phone recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 5509–5513.

[199] Zhang, Y., Kuo, H.-w., and Wright, J. Structured local minima in sparse blind deconvolution. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 2322–2331.

[200] Zwicker, E., and Terhardt, E. Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. *The Journal of the Acoustical Society of America 68*, 5 (1980), 1523–1525.

[201] İnan, B., Cernak, M., Grabner, H., Tukuljac, H. P., Pena, R. C., and Ricaud, B. Evaluating Audiovisual Source Separation in the Context of Video Conferencing. In *Proc. Interspeech 2019* (2019), pp. 4579–4583.

MSc Helena Peić Tukuljac

Personal Information

Skype:	helena.peic.tukuljac	Date of birth:	July 29, 1991
Nodes in the networks:	LinkedIn R arXiv.org		
E-mail address:	helena.peic.tukuljac@gmail.com	Nationality:	Croatian, Serbian

Education

September 2015 – December 2019: PhD - École Polytechnique Fédérale de Lausanne, Switzerland
Field: Digital signal processing, inverse problems, optimization and acoustics

1. LCAV – Audiovisual Communication Laboratory – supervised by Prof. Martin Vetterli (project)
2. LTS2 – Signal Processing Laboratory – supervised by Prof. Pierre Vandergheynst

Relevant courses: Mathematical foundations of signal processing, Convex optimization and applications, Distributed information systems and Audio engineering

Thesis topic: **Sparse and Parametric Modeling with Application to Acoustics and Audio**

Thesis description:

1. Time-series analysis in the domain of signal processing and optimization with sparse priors
2. Parametric modeling of room acoustics
3. Deep learning applied to classification problems of time-series

September 2014 – July 2015: MSc - Faculty of Technical Sciences, Novi Sad, Serbia
Grade Point Average (GPA): 10.00 (5.00-10.00 scale)
Thesis topic: *A Recommender System for Hybrid Digital Television*

September 2010 – July 2014: BSc - Faculty of Technical Sciences, Novi Sad, Serbia
Grade Point Average (GPA): 9.95 (5.00-10.00 scale)
Thesis topic: *An Implementation of an Acquisition System for DVB (Digital Video Broadcasting) Data*

Internship Experience

June – September 2018:
Microsoft Research, Redmond, WA, United States
Supervised by: Nikunj Raghuvanshi (Interactive Media Group, Triton project, blog post)
Collaborators: Prof. Ville Pulkki (Aalto University, Finland)
Ivan Tashev (Audio and Acoustics Research Group)

October – December 2017:
Inria Research Center, Rennes, France
PANAMA (Parsimony and New Algorithms for Audio and Signal Modeling)
Research in the domain of inverse problems in acoustics – supervised by Prof. Rémi Gribonval

August 2014:
Telecom Seeds for the Future
Huawei Technologies, Shenzhen, China

RT-RK – Research and development company and national research institute, Novi Sad, Serbia
July 2013 – July 2014:
Various projects in the domain of embedded systems
July 2012:
Android summer school

Work Experience

September 2015 – December 2019:
École Polytechnique Fédérale de Lausanne, Switzerland
Job responsibilities: Researcher, Teaching assistant

August 2014 – August 2015:
RT-RK – Research and development company and national research institute, Novi Sad, Serbia
Job responsibilities: Software developer, Teaching assistant, Researcher – paper publication

Teaching Experience

2019: Signals and Systems II
2018: Signals and Systems I and II
2016: Analyse I

2015: DSP Algorithm and Structure Fundamentals 2
2014: Television Set and Image Processing Software

Supervision and Mentorship

Master theses (with logitech):

- Berkay Inan – *Evaluating Audiovisual Source Separation in the Context of Video Conferencing* (github)
- Pollet Vincent – *Neural Network Based Audio Blind Source Separation for Noise Suppression* (github)

Master student projects (inside LTS laboratory):

- Belbaraka Ali – *Alexa, Let's Play Hide and Seek!* (github)
- Janjar Youssef – *Direction of Arrival Estimation in Enclosures with Deep Learning* (github)
- Weber Justine Jeanne Gaëlle – *End-to-End Learning for Music Audio Exploration* (github)

CHIC (China Hardware Innovation Camp) – supervising AKANE project

Awards and scholarships

2011-2014: Rotary club student scholarship recipient

Mileva Maric Einstein award for the best computer science student for the school year 2014/2015

Accelerate Your Code Competition in 2012 (algorithms and parallel programming challenge) – award for great software optimization skills (C++ and parallel programming applied to DNA data analysis)

IEEE 2014 ICCE Berlin Best Paper Award

womENcourage 2017 Google travel grant recipient

National competitions in mathematics: II prize in 2009 and III prize in 2007

Selected publications

BSc

S. Pejić, **H. Peić Tukuljac**, M. Knežević, I. Papp, *One implementation of extendable application for collecting EPG data from internet sources*, The 4th IEEE International Conference on Consumer Electronics - Berlin (IEEE 2014 ICCE-Berlin), vol., no., pp. 230-232

MSc

H. Peić Tukuljac, S. Majstorović, M. Knežević, T. Maruna, *A Service for Metadata Enrichment for Video on Demand Systems*, The 5th IEEE International Conference on Consumer Electronics - Berlin (IEEE 2015 ICCE-Berlin), vol., no., pp. 445-448

PhD

H. Peić Tukuljac, H. Lissek, P. Vandergheynst, *Localization of sound sources in a room by one microphone*, Wavelets and Sparsity XVII, SPIE, San Diego, 2017; code; featured on Nuit Blanche blog; presentation recording

H. Peić Tukuljac, V. T. Pham, H. Lissek, P. Vandergheynst, *Joint estimation of room geometry and modes using compressed sensing*, ICASSP 2018, Calgary, Alberta, Canada; code; presentation slides

H. Peić Tukuljac, A. Deleforge, R. Gribonval, *MULAN: A Blind and Off-Grid Method for Multichannel Echo Retrieval*, NIPS 2018, Montreal, Canada; code

H. Peić Tukuljac, V. Pulkki, H. Gamper, K. Godin, I. Tashev, N. Raghuvanshi, *A sparsity measure for echo density growth in general environments*, ICASSP 2019, Brighton, UK

B. Inan, M. Cernak, H. Grabner, **H. Peić Tukuljac**, R. Pena and B. Ricaud, *Evaluating Audiovisual Source Separation in the Context of Video Conferencing*, Interspeech 2019, Graz, Austria

H. Peić Tukuljac, B. Ricaud, N. Aspert, P. Vandergheynst, *SpectroBank: A Filter-bank Convolutional Layer for CNN-Based Audio Applications*, ICLR 2020, Addis Ababa, Ethiopia, Submitted

H. Peić Tukuljac, A. Deleforge, *Estimating Early Acoustic Echoes from Noisy Speech with Multichannel Structured Low-Rank Optimization*, In preparation

Other activities

2016 ICCE Berlin Technical Program Committee, ICASSP 2018 Reviewer, NeurIPS 2019 Reviewer

Serbian Intellectual Property Office - "Patent register" - num. 2014/0640
"System for recommending popular content in digital television"

2009: Petnica Science Center in Valjevo, Serbia – research related activities in the field of mathematics

TransformTECH program – educator in domain of *Demystifying Artificial Intelligence* (Jan and Sep 2019)

Skills

Proficient user of programming languages: python (array processing, shallow and deep learning), matlab, Java, C, C++

Tools for software versioning: git and Subversion (SVN)

Operating systems: Windows, Linux + Android

Language proficiency

Serbian (mother tongue), English (proficient, C2) , French (intermediate, B1), German (intermediate, B1)

Other: Membership in Mensa International