BRNO UNIVERSITY OF TECHNOLOGY

Faculty of Electrical Engineering
and Communication

DOCTORAL THESIS

Brno, 2016 Ing. Václav Mach

BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF TELECOMMUNICATIONS
ÚSTAV TELEKOMUNIKACÍ

INPAINTING OF MISSING AUDIO SIGNAL SAMPLES
DOPLŇOVÁNÍ CHYBĚJÍCÍCH VZORKŮ V AUDIO SIGNÁLU

DOCTORAL THESIS
DIZERTAČNÍ PRÁCE

AUTHOR — **Ing. Václav Mach**
AUTOR PRÁCE

SUPERVISOR — **doc. Mgr. Pavel Rajmic, Ph.D.**
ŠKOLITEL

BRNO 2016

ABSTRACT

Recently, sparse representations of signals became very popular in the field of signal processing. Sparse representation mean that the signal is represented exactly or very well approximated by a linear combination of only a few vectors from the specific representation system.

This thesis deals with the utilization of sparse representations of signals for the process of audio restoration, either historical or recent records. Primarily old audio recordings suffer from defects like crackles or noise. Until now, short gaps in audio signals were repaired by interpolation techniques, especially autoregressive modeling. Few years ago, an algorithm termed the Audio Inpainting was introduced. This algorithm solves the missing audio signal samples inpainting using sparse representations through the greedy algorithm for sparse approximation.

This thesis aims to compare the state-of-the-art interpolation methods with the Audio Inpainting. Besides this, ℓ_1-relaxation methods are utilized for sparse approximation, while both analysis and synthesis models are incorporated. Algorithms used for the sparse approximation are called the proximal algorithms. These algorithms treat the coefficients either separately or with relations to their neighbourhood (structured sparsity). Further, structured sparsity is used for audio denoising.

In the experimental part of the thesis, parameters of each algorithm are evaluated in terms of optimal restoration efficiency vs. processing time efficiency. All of the algorithms described in the thesis are compared using objective evaluation methods Signal-to-Noise ratio (SNR) and PEMO-Q. Finally, the overall conclusion and discussion on the restoration results is presented.

KEYWORDS

Sparse Representations, Audio Inpainting, Proximal Algorithms, Audio Restoration, Denoising.

ABSTRAKT

V oblasti zpracování signálů se v současné době čím dál více využívají tzv. řídké reprezentace signálů, tzn. že daný signál je možné vyjádřit přesně či velmi dobře aproximovat lineární kombinací velmi malého počtu vektorů ze zvoleného reprezentačního systému.
Tato práce se zabývá využitím řídkých reprezentací pro rekonstrukci poškozených zvukových záznamů, ať už historických nebo nově vzniklých. Především historické zvukové nahrávky trpí zarušením jako praskání nebo šum. Krátkodobé poškození zvukových nahrávek bylo doposud řešeno interpolačními technikami, zejména pomocí autoregresního modelování. V nedávné době byl představen algoritmus s názvem Audio Inpainting, který řeší doplňování chybějících vzorků ve zvukovém signálu pomocí řídkých reprezentací. Zmíněný algoritmus využívá tzv. hladové algoritmy pro řešení optimalizačních úloh.
Cílem této práce je porovnání dosavadních interpolačních metod s technikou Audio Inpaintingu. Navíc, k řešení optimalizačních úloh jsou využívaný algoritmy založené na ℓ_1-relaxaci, a to jak ve formě analyzujícího, tak i syntetizujícího modelu. Především se jedná o proximální algoritmy. Tyto algoritmy pracují jak s jednotlivými koeficienty samostatně, tak s koeficienty v závislosti na jejich okolí, tzv. strukturovaná řídkost. Strukturovaná řídkost je dále využita taky pro odšumování zvukových nahrávek.
Jednotlivé algoritmy jsou v praktické části zhodnoceny z hlediska nastavení parametrů pro optimální poměr rekonstrukce vs. výpočetní čas. Všechny algoritmy popsané v práci jsou na praktických příkladech porovnány pomocí objektivních metod odstupu signálu od šumu (SNR) a PEMO-Q. Na závěr je úspěšnost rekonstrukce poškozených zvukových signálů vyhodnocena.

KLÍČOVÁ SLOVA

Řídké reprezentace, interpolace signálů, proximální algoritmy, restaurace zvuku, odšumování.

MACH, Václav *Inpainting of Missing Audio Signal Samples*: doctoral thesis. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications, 2016. 125 p. Supervised by doc. Mgr. Pavel Rajmic, Ph.D.

DECLARATION

I declare that I have written my doctoral thesis on the theme of "Inpainting of Missing Audio Signal Samples" independently, under the guidance of the doctoral thesis supervisor and using the technical literature and other sources of information which are all quoted in the thesis and detailed in the list of literature at the end of the thesis.

As the author of the doctoral thesis I furthermore declare that, as regards the creation of this doctoral thesis, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone's personal and/or ownership rights and I am fully aware of the consequences in the case of breaking Regulation § 11 and the following of the Copyright Act No 121/2000 Sb., and of the rights related to intellectual property right and changes in some Acts (Intellectual Property Act) and formulated in later regulations, inclusive of the possible consequences resulting from the provisions of Criminal Act No 40/2009 Sb., Section 2, Head VI, Part 4.

Brno ....

(author's signature)

ACKNOWLEDGEMENT

During the years at the Department of Telecommunications of the Brno University of Technology I had the possibility to meet many great people and learn a lot of things either connected with the technical part of signal processing or for enjoyment.

The biggest thanks goes to my supervisor Pavel Rajmic for his huge support during my studies, for a lot of ideas he gave me in a great environment and relaxed atmosphere. He is a wonderful teacher, mentor and model of an outstanding scientist and person generally. Thank you also for providing rights to use some of your figures in my thesis.

During my studies I had a great chance to visit several conferences and travel abroad to internships in Vienna for several times. Here, I met very good friends and colleagues who helped me to survive in a new place and gave me professional advice and care in the research topic. First of all I have to name Christoph Wiesmeyer, then Nicki Holighaus, Thibaud Necciari, Nathanael Perraudin and Gino Angelo Velasco. Thank also to their supervisors and project leaders Hans Feichtinger, Monika Doerfler and Peter Balazs.

At the Brno University of Technology I was lucky to meet and work with Zdenek Prusa and Marie Dankova and my bachelor and master students, especially Roman Ozdobinski. Thank you to Jirka, Peca, Kamil, Kristian, David, Honza and all the people from BUT who kept challenging and inspiring me.

I had a great opportunity to use my professional sound processing knowledge together with my biggest hobby, folklore music. Thank you to Jarmila Prochazkova and Lucie Uhlikova from The Institute of Ethnology of the Academy of Sciences of the Czech Republic, v.v.i. for a lot of experience and inspiration and providing me with archive records.

The most important motivation was from my family. Thanks you my parents Marta and Stanislav, I would never have started and finished my Ph.D. thesis without your encouragement. Concluding with my biggest THANKS to my loving wife Barbora and our little daughter Alžběta whose love and enthusiasm makes every day brighter.

Brno ....

(author's signature)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

ACKNOWLEDGEMENT

Research described in this doctoral thesis has been implemented in the laboratories supported byt the SIX project; reg. no. CZ.1.05/2.1.00/03.0072, operational program Výzkum a vývoj pro inovace.

Brno . . . . . .

. . . . . .

(author's signature)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

CONTENTS

List of symbols, physical constants and abbreviations **15**

1 Introduction **16**
1.1 Types of damages on archive recordings 18
1.1.1 Scratches 18
1.1.2 Fissures and welds 18
1.1.3 Scratching off the groove 18
1.1.4 Mould 19
1.1.5 Pitch fluctuation 19
1.2 Consequent audio signal artifacts 19
1.2.1 Clicks 19
1.2.2 Crackles 19
1.2.3 Noise 20
1.2.4 Hum and rumbling 21
1.2.5 Clipping 21
1.3 Thesis roadmap 21

2 State of the Art **23**
2.1 Samples repetition 23
2.2 Autoregression modelling 24
2.2.1 Linear prediction 25
2.2.2 AR modeling of signal samples 26
2.2.3 AR modelling of signal parameters 29
2.3 Chapter summary 35

3 Preliminary knowledge **36**
3.1 Notation 36
3.2 Vector norms 36
3.3 Matrix norms 37
3.4 Pseudoinverse 37
3.5 Vector spaces and basis vectors 38
3.5.1 Orthogonal and orthonormal basis 38
3.6 Frames 38
3.6.1 General definition 38
3.6.2 Frame operator, dual frames 39
3.6.3 Gabor frames 40
3.6.4 Partition of Unity 41

3.7 Chapter summary . . . 41

4 Sparse representations 42
4.1 Properties of sparse representations . . . 42
4.2 Audio Inpainting . . . 43
4.2.1 Audio Declipping . . . 44
4.3 Dictionaries . . . 45
4.3.1 Most frequent dictionaries . . . 45
4.3.2 Dictionary learning . . . 48
4.3.3 State-of-the-art . . . 49
4.3.4 K-SVD . . . 50
4.4 Structured sparsity . . . 50
4.5 Chapter summary . . . 51

5 Algorithms for sparse approximations 52
5.1 Greedy algorithms . . . 52
5.1.1 Orthogonal Matching Pursuit . . . 52
5.2 Relaxation algorithms . . . 53
5.2.1 Basis Pursuit . . . 53
5.2.2 Proximal algorithms . . . 54
5.2.3 Proximity operator . . . 56
5.2.4 Weighting of atoms . . . 56
5.3 Structured sparsity . . . 57
5.4 Hybrid algorithms . . . 59
5.5 Chapter summary . . . 59

6 Goals of the thesis 60

7 Evaluation of audio restoration 61
7.1 Objective evaluation . . . 61
7.1.1 Signal-to-Noise Ratio . . . 61
7.1.2 PEMO-Q . . . 61
7.2 Experimental results description . . . 62
7.2.1 Inpainting toolbox . . . 62
7.2.2 Audio examples . . . 63
7.3 Interpolation methods . . . 64
7.3.1 Samples repetition . . . 64
7.3.2 AR modeling of signal samples . . . 67
7.3.3 Sinusoidal modeling . . . 71
7.4 Greedy algorithm . . . 75

7.5 Dictionary learning 81
7.6 ℓ_1-relaxation algorithm 83
7.6.1 Dictionary redundancy 83
7.6.2 Weighting of atoms 84
7.6.3 Analysis vs. Synthesis model 85
7.6.4 Non-harmonic signals 91
7.6.5 Processing time 91
7.6.6 Structured sparsity 100
7.7 Final evaluation and chapter summary 108

8 Conclusion **110**

Bibliography of the author **113**

References **115**

LIST OF FIGURES

1.1 Wax Cylinder and its box 17
1.2 Gramophone record click 20
1.3 Periodical noise in digitized wax cylinder recordings 21
1.4 Audio signal clipping 22
2.1 Crossfading using the raised-cosine function. 25
2.2 Lattice-based structure of order 1. 27
2.3 Set of sinusoids with figured gap. 31
2.4 Block scheme of partials linking decision algorithm. 32
2.5 A spectrogram of piece of the harmonic signal (guitar chord) with a gap: the original signal (a), tonal parts of the signal (b) and a residuum (c). 34
3.1 Illustration of unit balls (a) B_0^2, (b) $B_{0,5}^2$, (c) B_1^2, (d) B_2^2 a (e) B_∞^2. . . . 37
3.2 Mercedes-Benz frame. 39
4.1 Graphical illustration of the sparse representation 43
4.2 An example of audio declipping. 45
4.3 Regular sampling in STFT segments. 47
4.4 Hann window function (blue), modulated, translated and windowed cosine functions (green, black). 48
4.5 Irregular sampling in STFT segments. 49
5.1 Window not affected(a) and affected (b) by gap. 57
7.1 SNR of interpolation by samples repetition for various neighbourhood length. 65
7.2 Interpolation by the samples repetition algorithm with various neighbourhood size q_u. 65
7.3 Signal-to-Noise Ratio (SNR) of interpolation by the samples repetition algorithm with various model order and gap length. 66
7.4 Processing time of interpolation by the samples repetition algorithm with various model order and gap length. 67
7.5 SNR of Various order of Least-Squares Residual Predictor interpolation. 68
7.6 SNR of Various order of Weighted Forward-Backward Predictor interpolation. 69
7.7 SNR of interpolation by autoregressive (AR) modeling of the samples (LSRI) with various model order and gap length. 69
7.8 SNR of interpolation by AR modeling of the samples (WFBI) with various model order and gap length. 70
7.9 Comparison of interpolation by the two algorithms for AR signal modeling. 71

7.10 Interpolation results of various gap length and frequency threshold of sinusoidal modeling. 72
7.11 Interpolation results of various gap length and mean amplitude of sinusoidal modeling. 73
7.12 Processing time of audio interpolation by sinusoidal modeling with various gap length and mean amplitude vector length. 74
7.13 SNR of audio interpolation by sinusoidal modeling with various gap length and AR model order. 74
7.14 Processing time of audio interpolation by sinusoidal modeling with various gap length and AR model order. 76
7.15 Audio Inpainting using Orthogonal Matching Pursuit (OMP) of various neighbourhood size and number of coefficients obtained in each iteration in terms of (a) SNR and (b) processing time. 77
7.16 SNR of audio inpainting by OMP with various neighbourhood size and dictionary redundancy. 78
7.17 Processing time of audio inpainting by OMP with various neighbourhood size and dictionary redundancy. 78
7.18 SNR of audio inpainting by OMP with various neighbourhood size according to gap length. 79
7.19 SNR of audio inpainting by OMP with various neighbourhood size according to gap length with highlighted best results by green points. 80
7.20 Processing time of audio inpainting by OMP with various neighbourhood size according to gap length. 80
7.21 Signal reconstruction of a guitar audio sample using the K-Means Singular Value Decomposition (K-SVD) algorithm and greedy solver. 82
7.22 SNR of inpainting with various redundancy and window length in analysis model. 84
7.23 SNR of inpainting with various redundancy and window length in synthesis model. 85
7.24 Mean SNR of inpainting with various gap length and window length in the analysis model. 86
7.25 Mean SNR of inpainting with various gap length and window length in the synthesis model. 87
7.26 Average SNR of inpainting with various gap length and window length in the analysis model. Dashed line with green points indicates the maximum values for each gap length. 88
7.27 Mean SNR of inpainting with various gap length and window length in the synthesis model. Dashed line with green points indicates the maximum values for each gap length. 89

7.28 Standard deviation of the SNR of inpainting with various gap length and window length in the analysis model. 89
7.29 Standard deviation of the SNR of inpainting with various gap length and window length in the synthesis model. 90
7.30 Mean SNR of inpainting with various gap length and window length in the analysis model (music07_16kHz.wav). 92
7.31 Mean SNR of inpainting with various gap length and window length in the synthesis model (music07_16kHz.wav). 92
7.32 Average inpainting processing time of file music11_16khz with gap size of 160 samples and various window length. 94
7.33 Average inpainting processing time of file music11_16khz with gap size of 800 samples and various window length. 95
7.34 Average inpainting processing time of file music11_16khz with gap size of 1600 samples and various window length. 95
7.35 Ratio of synthesis/analysis average inpainting processing time of file music11_16kHz. 96
7.36 Ratio of synthesis/analysis average proximal algorithm iterations of file music11_16kHz. 97
7.37 Average inpainting iterations number of file music11_16khz with gap size of 160 samples and various window length. 97
7.38 Average inpainting iterations number of file music11_16khz with gap size of 800 samples and various window length. 98
7.39 Average inpainting iterations number of file music11_16khz with gap size of 1600 samples and various window length. 98
7.40 Number of iterations of the proximal algorithm with various gap and window length in the analysis model. Dashed line with green points indicates the maximum values for each gap length. 99
7.41 Number of iterations of the proximal algorithm with various gap and window length in the synthesis model. Dashed line with green points indicates the maximum values for each gap length. 100
7.42 Comparison of the relative SNR and number of iterations for file music11_16kHz, gap length of 960 samples, window lenght of 4104 samples. 101
7.43 Comparison of the signal reconstruction using analysis and synthesis model. 102
7.44 SNR of inpainting by structured sparsity with various λ and exponent parameters. 103
7.45 Processing time of inpainting by structured sparsity with various λ and exponent parameters. 103

7.46 SNR of inpainting by structured sparsity with various coefficients neighbourhood size and gap length. 104
7.47 Processing time of inpainting by structured sparsity with various coefficients neighbourhood size and gap length. 104
7.48 Time and spectral representation of inpainting using the structured sparsity. 105
7.49 Spectrogram of the original instrumental recording 106
7.50 Reconstruction by the structured sparsity 107
7.51 Reconstruction by the professional software 107

LIST OF TABLES

7.1 List of experimental audio samples. 64
7.2 Values of AR model order and according neighbourhood size in the test. 75
7.3 Values of the gap length in the test 85
7.4 Best SNR results of `music11_16kHz.wav` with its standard deviation 88
7.5 Parameters of points with the worst standard deviation of the SNR results of `music11_16kHz.wav` 90
7.6 Best SNR results of `music07_16kHz.wav` with its standard deviation 91
7.7 Results of inpainting experiments over all of the sound files (analysis model, sorted by the average SNR) 93
7.8 Results of inpainting experiments over all of the sound files (synthesis model, sorted by the average SNR) 93
7.9 Structured sparsity parameters 108
7.10 Complex evaluation of inpainting/interpolation algorithms 109

LIST OF SYMBOLS, PHYSICAL CONSTANTS AND ABBREVIATIONS

ADMM	Alternating-Direction Method of Multipliers
AR	autoregressive
BP	Basis Pursuit
DCT	Discrete Cosine Transform
ERB	Equivalent Rectangular Bandwidth
FISTA	Fast Iterative Shrinkage/Thresholding Algorithm
FOCUSS	Focal Underdetermined System Solver
FFT	Fast Fourier Transform
IIR	Infinite Impulse Response
ISTA	Iterative Shrinkage/Thresholding Algorithm
K-SVD	K-Means Singular Value Decomposition
LASSO	Least Absolute Shrinkage and Selection Operator
MNS	Musical Noise Supression
MOD	Method of Optimal Directions
MP	Matching Pursuit
NSGT	Nonstationary Gabor Transform
OMP	Orthogonal Matching Pursuit
PSM	Perceptual Similarity Measure
PU	Partition of Unity
RMSE	Root Mean Square Error
SNR	Signal-to-Noise Ratio
STFT	Short Time Fourier Transform
SVD	Singular Value Decomposition
TV	Total Variation
WMDCT	Windowed Modified Discrete Cosine Transform

1 INTRODUCTION

Historical sound recordings usually suffer from imperfections. The history of sound recordings started at the end of nineteenth century with wax cylinders, followed by shellac, acetate, vinyl disc recordings, stainless steel wire recordings and magnetic tapes. Typical distortions like a hiss, impulse noise, crackle, wow and flutter, background noise or power line hum are a natural part of such archive audio sources. The high quality digital audio brought by Digital Audio Tape (DAT) or Compact Disc (CD) caused an enormous increase of sound quality demands. Meanwhile, the interest in nostalgic and historical material was still retained. Therefore, requirement for the audio restoration of degraded recordings grew.

Audio restoration is a generalized term for the process of removing faultiness from sound recordings. The perfect restoration would reconstruct the original audio exactly as captured by the transducer (e.g. microphone, reproducer, horn). Original pure quality can, of course, never be reached in practice. However, there are methods which can come close according to some suitable error criterion ideally based on the perceptual characteristics of the human hearing.

First historical recordings were recorded using phonograph on the wax cylinders. In the area of today's Czech Republic and Slovakia, the first phonogram recordings were captured at the end of 19th century by the famous music composer Leoš Janáček and his collaborators. Recently, after more than one hundred years, the wax cylinders were re-recorded[1] by the digital systems, digital restoration was performed and they are available together with detailed study for a large audience in English [2] and also in Czech [3]. For illustration see photo of the wax cylinder in Fig. 1.1.

The theoretical interest in the process of audio restoration was enforced by cooperation with The Institute of Ethnology of the Academy of Sciences of the Czech Republic, v.v.i.[2] and the National Institute of Folk Culture[3] which led in joint research interests in audio signal processing and the ethnological research. The restoration of ancient audio recordings was performed in a sensitive way to the music genre. The experience of treating a lot of sound impairments using a commercial software resulted in experiments using novel restoration methods based on the sparse representations.

Another historical recordings are captured on gramophone records. Currently, a lot of algorithms and software are able to clean up the unwanted artifacts from the gramophone records reliably, e.g. [79],[43],[51]. Some archivists claim that vinyl is the best material for long term analogue audio signal preservation.

[1] Re-recording means digitization of the original wax cylinders by modern digital systems.

[2] `http://eu.avcr.cz/index.html`

[3] `http://www.nulk.cz/`

Fig. 1.1: Wax Cylinder and its box

Magnetic tapes that were used in the fifties are currently not in a bad condition in the sense of damage. However, the magnetic layer of the tape is falling off the tape and causes signal drop-outs. Those tapes, which are not digitized yet will have to be re-recorded into the digital systems in the foreseeable future [57].

Quite surprising can be the need for restoration of modern digital media, like CDs, MiniDisc, etc. The first medias which are more than 25 years old are reaching their life cycle and the stored information can be lost despite of error correction codes [21].

Audio restoration is not only the matter of historical recordings. Nowadays audio systems may cause an error during audio samples processing either with or without human influence. Professional audio/video production has human and technical resources that are mostly aware of audio processing failures. However, people who are not experienced in working with recording equipment can cause signal degradation which is usually discovered in the post-processing or mastering stage. A typical example is overloading the signal level resulting in signal saturation called clipping. Recording shots for a documentary movie with faulty audio settings could be a disaster for the producer and the authenticity of the scene could be never repeated.

Another example is the packet loss during VoIP transmission. It depends on a preference of the system with VoIP media transmission. Pure low quality IP telephony could be enhanced by filling in the gaps caused by lost packets. This process is called the Packet Loss Concealment [82]. In contrast, there are cases

where such an artificial gap infilling is absolutely undesirable, e.g. Air Traffic Control (ATC) voice transmission [10].

Last example of current field where audio restoration could be utilized is an audio encoding. Errors which appear during audio encoding into lossy formats without the possibility of re-encoding could be restored by audio restoration algorithms.

The main massage of this section should be the purpose of the audio restoration process. All of the algorithms and software tools are a great assistants for audio mixing, post-processing, mastering and publishing. However, they should never be used for audio archiving, since the original information is lost during the restoration process.

1.1 Types of damages on archive recordings

Condition of the material is clearly affected by the age of the sound information carrier. The description of common types of damages is based on a detailed study of the collection of wax cylinders [9] [8].

1.1.1 Scratches

Scratches are well-known artifacts for a lot of listeners from the gramophone records. However, cylinders are made of rather weak material compared to vinyl and it results in clicks with much more signal amplitude.

1.1.2 Fissures and welds

Cylinders contain periodic interferences similar to rustling mostly caused by wider cracks (fissures). Some of the cylinders were broken into a few pieces which were put together by welding. Welds arisen during mechanical restoration of cylinders cause very similar disturbance like cracks.

1.1.3 Scratching off the groove

The signal quality decreased with every replaying of the cylinder and especially in the case of inappropriate seating of the point[4], which could scratch off some of the material containing the desired information[5]

[4] If we speak about phonograph the *point* is a name for the stylus known from gramophones.

[5] The composer Leoš Janáček himself spoke about this issue in his lecture delivered in Brno in 1922: "The phonograph is an unfortunate means of listening to folk songs! It wears down. I play it five times and there is nothing left any more. When the mass is scratched off by the needle, the song is gone!"[2]

1.1.4 Mould

Mould is generated by the organic composition of the cylinder material. The mould formed a white coat on the brown cylinder and its spread is directly proportional to the signal interference.

1.1.5 Pitch fluctuation

Another problem of wax cylinders and acetate discs[6] is variable speed of the cylinder rotation which causes intonation instability. This kind of signal degradation is called *wow* or *flutter*[43]. Several stages of the recording or re-recording process caused the pitch fluctuation. The first one was the recording in 1912 where the phonograph was not in a good condition. The same machine was used for re-recording of of the cylinders on the acetate discs, therefore the original fluctuation was preserved and a new one arose.

Re-recording of the cylinders to the *acetate disc* in the sixties helped to preserve the recordings since during next fifty years until the process of digitization the cylinders degraded too much. Acetate discs suffer from most of the damages described above. Moreover, the electroacoustics chain of the recording studio influenced the signal especially in the sense of low frequency rumbling.

1.2 Consequent audio signal artifacts

1.2.1 Clicks

The level of clicks which appeared in the broken and physically conserved cylinders was very high. But regarding their progress they are easy to detect and remove. After removing the clicks (especially the strongest ones) there still remains a low frequency interference. Removing of such an interference can be quite complicated. Usually a simple high-pass filter is not feasible because the highest frequency reached by these low frequency artifacts are overlapping with the lowest male singing formant. For illustration of a common gramophone click see Fig. 1.2

1.2.2 Crackles

The progress of the specific crackles on wax cylinders recordings can not be detected in the form of clicks or regular crackles The proportion of their length in relation to the length of the signal without the interference ranges from $1:5$ to $1:1$. The most damaged cylinders contain alternating short sections, with lengths

[6] The ancestor of gramophone record.

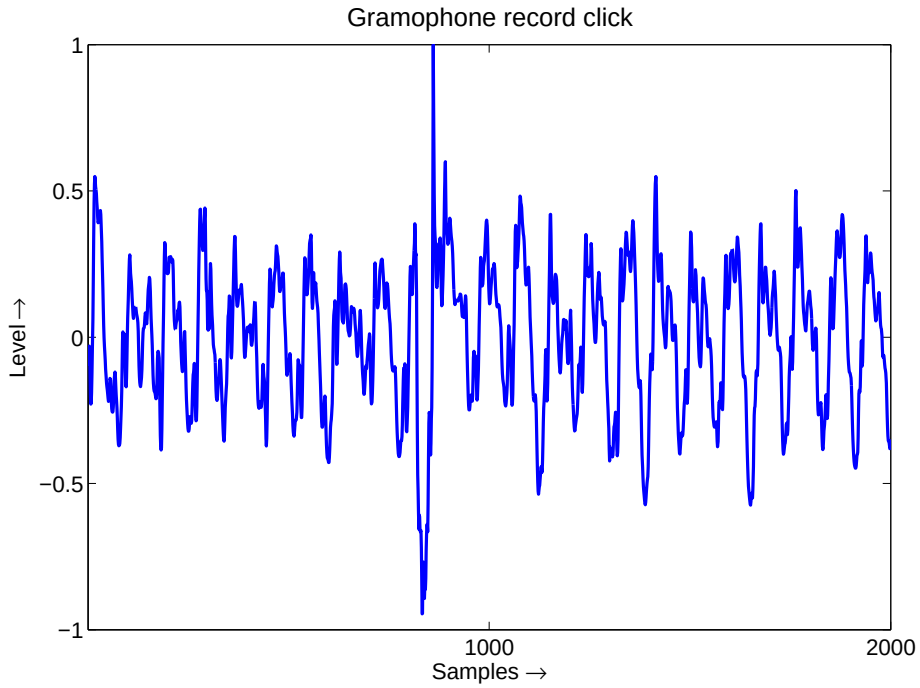


Fig. 1.2: Gramophone record click

between 50 and 250 ms, of more or less noise-contaminated signal (see Fig. 1.3), and the given recording does not offer enough information to find a suitable pattern of spectral reconstruction. In addition, the application of the standard denoising tool results in much more musical noise, and therefore the level of intervention within the recordings is based on the number of occurrences of these artifacts.

1.2.3 Noise

The noise coming from digitized wax cylinders is spread across the whole audible frequency spectrum. Considering restoration of large set of recordings, the restoration process has to be very careful to preserve relatively equal noise levels within each recording. Automatic dynamic signal compression is impossible when such recordings are processed, because any increase or decrease of the signal level is audible, and constant changes of loudness levels (especially noise levels) worsen the subjective impression of the listening. An experienced listener might hear the so-called *musical noise*[7] resulting from denoising of the recording. In the case of cylinders, however, the unpleasant noise already originates from the digitized cylinders and represents no artifact resulting from restoration of the original sound signal.

[7]Musical noise denotes non-stationary short-term harmonic signals resulting from the process of denoising. They resemble "twittering". For more about this issue see [31]

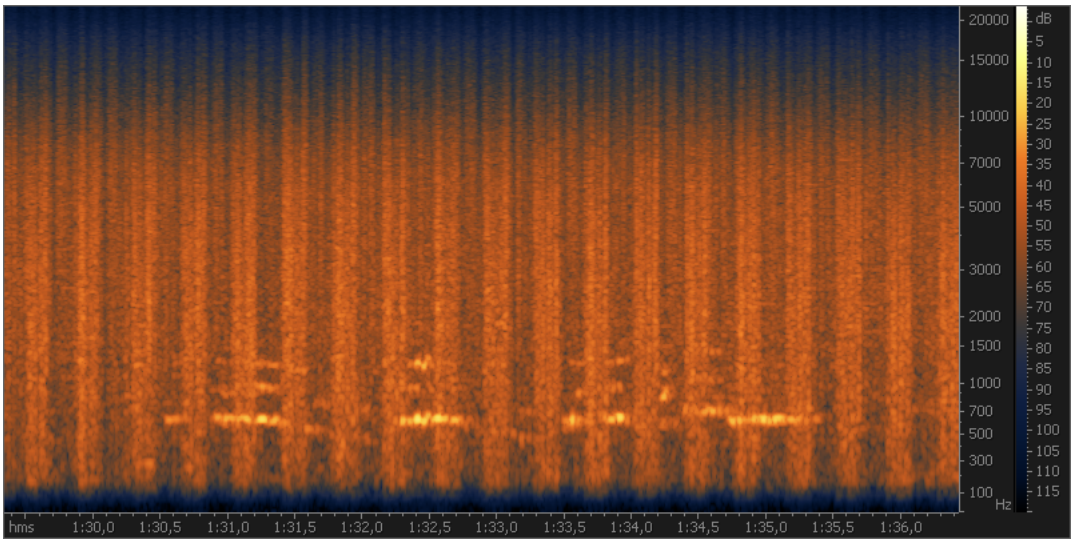


Fig. 1.3: Periodical noise in digitized wax cylinder recordings

1.2.4 Hum and rumbling

Low frequencies within the range of circa 150 to 300 Hz include "rumbles", bangs and other disrupting sounds caused by the cylinder rotation. From the point of view of the filtration process this is a relatively easily remediable problem, for example with the high-pass filter. This, however, creates an issue in the case of low frequency instruments (bass guitar, double bass) or male singing or spoken word, where the lowest harmonic partial or formant reaches into this area, and therefore the high-pass filter must be used to filter the signal carefully.

1.2.5 Clipping

If the signal level exceeds the threshold it is called clipping, see Fig. 1.4. It occurs when the input signal level (absolute value) is higher then the maximum range of the converter and causes in a non-linear distortion (many high frequency harmonics). There are two kinds of clipping: hard and soft. If the information in the clipped peaks is completely eliminated we speak about the hard clipping. It cannot be restored anymore. On the other hand, soft clipping could be restored if no portion of the signal is completely eliminated.

1.3 Thesis roadmap

The thesis is organized as follows. In Sec. 2 the State-of-the-Art methods for audio interpolation are presented such as samples repetition and AR modeling of signal samples and parameters. Sec. 3 describes the preliminary knowledge for the theory

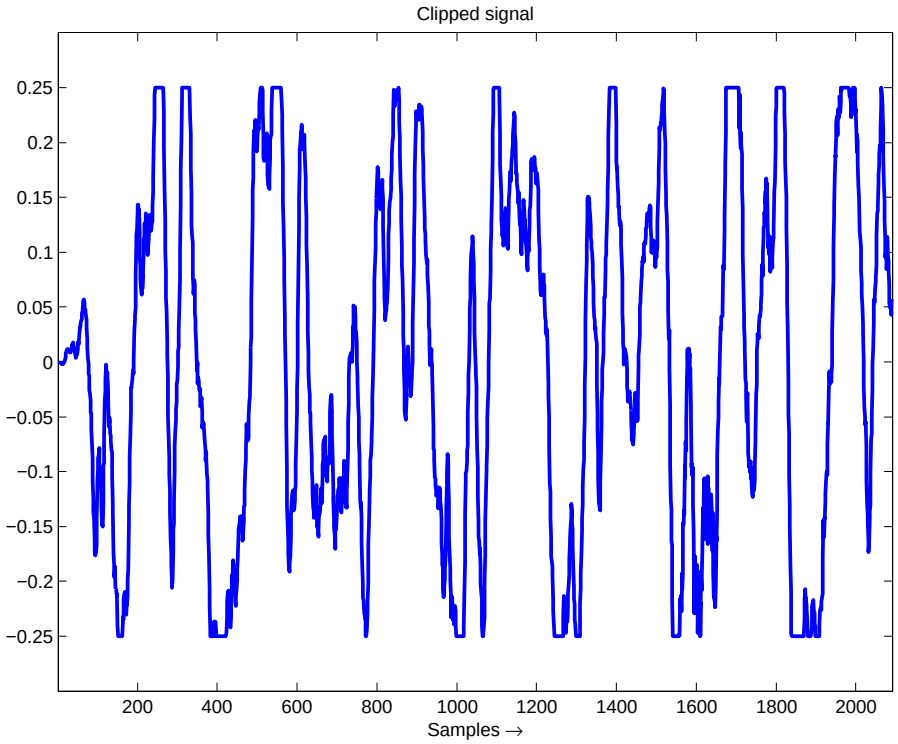


Fig. 1.4: Audio signal clipping

of sparse representations presented in Sec. 4. Solving the problem of audio inpainting using sparse representations need appropriate solvers for sparse approximation. These are described in Sec. 5. Preliminary to the main chapter with experiments main goals of the thesis are summed in Sec. 6. The largest chapter Sec. 7 shows results of complex experiments of all methods together with the evaluation and analysis of the results. Finally the thesis is concluded in Sec. 8 followed by the list of references, information about the author of the thesis and CV.

2 STATE OF THE ART

The early methods of interpolation of the missing samples portions performed an *extrapolation*, i.e. expanding the signal from only one side of the gap. Extrapolation algorithms were based on a signal periodicity and samples repetition supposed to fill in reliably only a stationary signal. Various methods focused on searching for the exact signal period and repeating the very last samples prior to the missing segment [39], [44], [90]. A simple method based on the signal repetition is described in Sec. 2.1.

The knowledge of the reliable samples from only one side of the gap (e.g. in the real-time applications) is always limiting. The extrapolation from both sides of the gap is more advantageous since there is less restriction on the signal stationarity. *Audio restoration* is the typical example of such a situation when there is a reliable signal from both sides and real-time processing is not necessary. Filling in the signal based on knowledge of the samples from both sides of the gap is called an *interpolation*.

The signal interpolation is based on modeling of an autoregressive process that improves the results of interpolation in contrast with samples repetition. Interpolation based on the autoregressive modeling of the signal samples is described in Sec. 2.2.2. A more advantageous approach based on interpolation of the signal parameters (amplitude, frequency) by the autoregressive modeling is described in Sec. 2.2.3.

2.1 Samples repetition

The estimation of the missing samples is performed using the fundamental period of the reliable signal around the gap. The periodicity is useful in cases where the signal under investigation behaves periodically e.g. vowels in the speech signal. This method is sometimes referred to as the waveform substitution.

In the following text the method for filling in missing samples is called *The Weighted Repetitive Substitution*. The core of this algorithm is replacing of the missing piece of data by a linear combination of two blocks of known samples. One of simpler methods of the interpolation is to repeat the most recent M samples

$$\{x[l-M], x[l-M+1], \ldots, x[l-1]\}, \tag{2.1}$$

where x is a signal vector and l is an index of the first missing sample. However, this simple method is often suffering from the mismatch at one or both ends of the gap.

If the pitch period of the reliable signal is determinated, then the number of samples in the period is denoted as q_{L} (left side) and q_{R} (right side), respectively. The left side estimate is obtained as

$$\widehat{x}_{\mathrm{L}}[i] = x[i - K_{\mathrm{L}} \cdot q_{\mathrm{L}}], \qquad l \leq i < l + M, \tag{2.2}$$

where

$$K_{\mathrm{L}} = \left\lceil \frac{M}{q_{\mathrm{L}}} \right\rceil \tag{2.3}$$

and the right side estimate is obtained as

$$\widehat{x}_{\mathrm{R}}[i] = x[i + K_{\mathrm{R}} \cdot q_{\mathrm{R}}], \qquad l \leq i < l + M, \tag{2.4}$$

where

$$K_{\mathrm{R}} = \left\lceil \frac{M}{q_{\mathrm{R}}} \right\rceil. \tag{2.5}$$

The ceiling operation $\lceil x \rceil$ performs rounding-up.

Having an estimate from both sides of the gap, the resulting signal is a linear weighted combination of both periodicity-based substitutions.

$$\hat{x}_{\mathrm{LR}}[i] = w_{\mathrm{L,M}}[i - l]\hat{x}_{\mathrm{L}}[i] + w_{\mathrm{R,M}}[i - l]\hat{x}_{\mathrm{R}}[i] \tag{2.6}$$

for $l \leq i < l + M$, where w is weighting function for the left (L) and right (R) side, respectively. Vector $\hat{x}$ represents the estimated samples from the appropriate side. Weighting is usually performed using the raised-cosine function, which for both sides forms a couple of weighting sequences [39]. See Fig. 2.1 for an example of such crossfading.

This method often suffers from introducing the artifacts when the restoration length is roughly similar as the stationarity duration of the signal. It is useful to use only the group of samples equivalent the the most recent pitch period. Such portion of samples is then repeated all over the signal gap. This operation is described as

$$\widehat{x}_{\mathrm{L}}[i] = x[i - q_{\mathrm{L}} + \mathrm{mod}(i - l, q_{\mathrm{L}})], \qquad l \leq i < l + M \tag{2.7}$$

and

$$\widehat{x}_{\mathrm{R}}[i] = x[\mathrm{mod}(K_{\mathrm{R}} \cdot q_{\mathrm{R}} - M + i - l, q_{\mathrm{R}}) + M], \qquad l \leq i < l + M. \tag{2.8}$$

2.2 Autoregression modelling

Bringing an adaptability of restoration of the missing signal samples using an AR process was first described in [52]. The restriction on this approach was the knowledge of the position of unknown samples and sufficiently large neighborhood of the

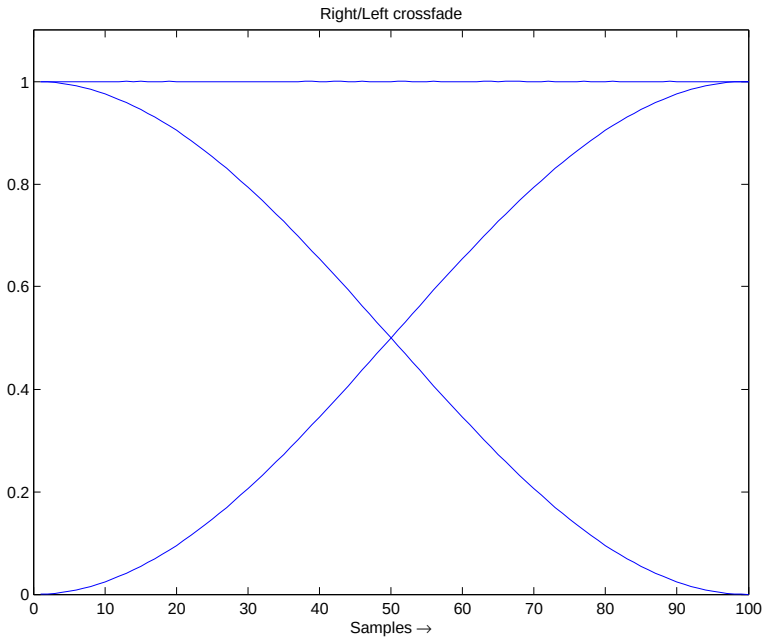


Fig. 2.1: Crossfading using the raised-cosine function.

reliable samples around the gap. Recovering of the missing samples is performed by minimizing the sum of squares of the residual errors which result in the estimates of the autoregressive parameters. Results of this interpolation (as described in the paper) are satisfying for gaps of length 0.36 ms ($f_s = 44100$ Hz) for audio signals and 12.5 ms ($f_s = 8000$ Hz) for speech signals which does not show too much robustness of this method. Moreover, this approach assumes only the autoregressive forward predictor (restoration based only on the previous samples).

Missing signal can be estimated from previous samples (forward prediction) or forward samples (backward prediction) or from both sides. The interpolation based on autoregressive modeling from both sides of the gap is introduced in [39] and will be described in the following text.

2.2.1 Linear prediction

Pure extrapolation describes a process of extending known portions of samples $x[i]$ with predicted values $\hat{x}[i]$. The prediction is performed by the determined model from the known signal samples. The quality of the prediction is a result of the precision of the model. Previously computed values $(x[i-1], \ldots, x[i-p])$ weighted by the autoregressive coefficients a together with the current sample value $x[i]$ provide the input of the autoregressive model described as

$$x[i+1] = \frac{1}{a_0}(a_1 x[i-1] + \cdots + a_p x[i-p]) + x[i] = \frac{1}{a_0}\left(\sum_{k=1}^{p} a_k x[i-p]\right) + x[i], \quad (2.9)$$

where p is an order of the model. After the $\mathcal{Z}$-transform the corresponding transfer function is

$$H(z) = \frac{1}{\sum_{k=0}^{p} a_k z^{-k}} \tag{2.10}$$

which shows that the extrapolation can be implemented as a single recursive Infinite Impulse Response (IIR) filter [41].

In terms of linear prediction the predicted sample $\hat{x}[i]$ is computed as

$$x[i] = \sum_{k=1}^{k} a_k x[i-k] + u[i], \tag{2.11}$$

where $u[i]$ is a term called the forward prediction error describing the accuracy of the predictor

$$u[i] = x[i] - \hat{x}[i]. \tag{2.12}$$

The coefficients a_k should be chosen to get the lowest prediction error $\sum_{i=1}^{p} u[i]^2$ [41]. Among different ways of the linear predictor implementation, the block-based method will be used in the following text.

2.2.2 AR modeling of signal samples

Autoregressive modeling is based on the *linear prediction*. Signal interpolation based on autoregressive modeling reduces the drawbacks of samples repetition model described in Sec. 2.1. Recalculation of missing samples in both sides of the gap is utilized according to the linear prediction from Eq. 2.11.

Among several block-based methods for calculating the autoregressive parameters two of them were chosen: Yule-Walker and Burg method.

Yule-Walker method

A method based on the autocorrelation is operating only on a currently analyzed block of length I where the total energy of the prediction error is computed as

$$U = \sum_{i=0}^{I+p-1} u^2[i]. \tag{2.13}$$

In the Yule-Walker method the parameters are calculated using the autocorrelation coefficients $r[i]$ in the Toeplitz matrix as

$$\begin{bmatrix} r[0] & r[1] & \dots & r[p-1] \\ r[1] & r[0] & \dots & r[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r[p-1] & r[p-2] & \dots & r[0] \end{bmatrix} \cdot \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = \begin{bmatrix} -r[2] \\ -r[3] \\ \vdots \\ -r[p+1] \end{bmatrix}. \tag{2.14}$$

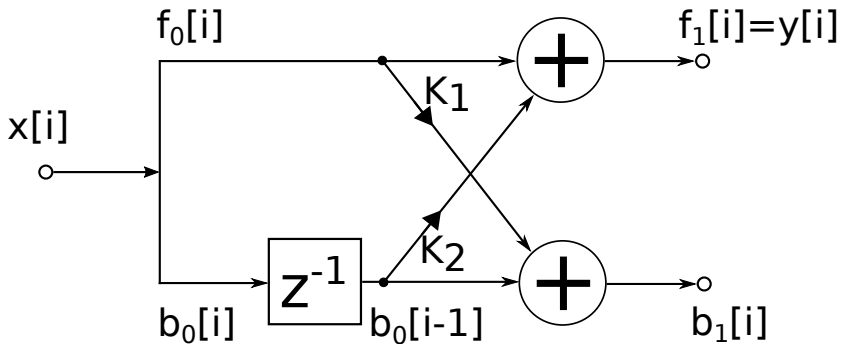


Fig. 2.2: Lattice-based structure of order 1.

The autocorrelation coefficients $r[n]$ are computed as

$$r[n] = \frac{1}{I} \sum_{l=n}^{I-1} s[l]s[l-n], \tag{2.15}$$

where $s[l]$ is a windowed block of data determined as

$$s[l] = w[l]x[l], \tag{2.16}$$

where $w[l]$ represents the weighting function, e.g. the raised-cosine function from Fig. 2.1. The system from Eq. 2.15 could be efficiently solved by the Levinson-Durbin algorithm [39][41].

Burg method

This block-wise method fits autoregressive model parameters to the input data by least-squares minimization performed on the sum of forward and backward error energy over all lattice stages. See the basic lattice-based structure of order 1 in Fig. 2.2. Such autoregressive parameters are supposed to satisfy the Levinson-Durbin recursion. The minimized error energy is computed as

$$J_n = \sum_{i=n}^{I-1} (f_n^2[i] + b_n^2[i]), \tag{2.17}$$

where i indicates the number of signal sample and n is the current order of the reflection coefficient k_n (comparable to the filter coefficient a_n of the direct form filter). The derivative of J_n with respect to k_n is set to zero

$$\frac{\mathrm{d}J_n}{\mathrm{d}k_n} = 0 \tag{2.18}$$

and solved for k_n by the so-called "Burg formula" as

$$k_n = \frac{2 \sum_{i=n}^{I-1} (f_{n-1}[i] b_{n-1}[i-1])}{\sum_{i=n}^{I-1} (f_{n-1}^2[i] b_{n-1}^2[i-1])}. \tag{2.19}$$

Recursive computation of Eq. 2.19 is used for computing the reflection coefficients k_n of the lattice for all $n = 1, \ldots, p$. Every time the new coefficient k_n is computed, all the lattice stages have to be updated according to

$$\begin{aligned} f_n[i] &= f_{n-1}[i] - k_n b_{n-1}[i-1] \\ b_n[i] &= f_{n-1}[i-1] - k_n f_{n-1}[i], \qquad n = 1, \ldots, p. \end{aligned} \tag{2.20}$$

This method is suitable for signal extrapolation using only a few reliable samples [41].

An implementation of a Burg algorithm starts with loading the initial values of f_0 and b_0. The autoregressive coefficient a_0 value is set to 1. A zero vector of length p for the reflection coefficients values is initialized. The maximum value of the order p must be smaller (at least by 1) than the length of the signal. In each iteration n the temporary vector is created to store the forward and backward error values from the last step. Then, the Burg formula (Eq. 2.19) is computed followed by recalculation of the new prediction errors f_p and b_p with current coefficient k_n. The last step is to convert the reflection coefficient vector k to the corresponding linear prediction coefficients as

$$\begin{gathered} a_p[0] = 1 \quad \text{for} \quad p = 1, 2, \ldots, I-1, \\ a_p[p] = k_p, \\ a_p[n] = a_{p-1}[n] + k_p a_{p-n}[p-n] \\ \text{for} \quad 1 \leq n \leq p-1, \quad p = 2, \ldots, I-1. \end{gathered} \tag{2.21}$$

Least-Squares Residual Predictor

Furthermore, with autoregressive coefficients already computed the completion of missing samples has to be performed. Among several algorithms for this computation, two of them seem to perform the best results. The first one is Least Squares-Residual, which assumes a certain stationarity of the signal around the signal gap [39].

Two autoregressive parameter vectors are obtained from the left and the right side of the gap. Due to the sake of brevity, only the core steps of this algorithm are explained. To obtain the estimate of samples inside the gap, the linear system of equations have to be computed:

$$\mathbf{D} \cdot \hat{\mathbf{x}} = \mathbf{y}, \tag{2.22}$$

where

$$\mathbf{D} = \mathbf{A}^{\mathrm{T}}\mathbf{A} + \mathbf{B}^{\mathrm{T}}\mathbf{B} \tag{2.23}$$

and

$$\mathbf{y} = -\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{a} - \mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{b}. \tag{2.24}$$

The algorithm is compiled of following steps:

1. Determination of Toeplitz matrices **A** and **B** from the left- and right-sided autoregressive parameters estimates,
2. calculation of the symmetric system matrix **D** (Eq. 2.23),
3. calculation of the vector **y** (Eq. 2.24),
4. solving a linear system of equations (Eq. 2.22).

The matrix **L** of size $M \times (K+1)$ belonging to the left-sided computation is

$$\mathbf{L} = \begin{bmatrix} 0 & x(l-1) & x(l-2) & \dots & x(l-K) \\ 0 & 0 & x(l-1) & \dots & x(l-K+1) \\ 0 & 0 & 0 & \dots & x(l-K+2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & x(l-1) \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \tag{2.25}$$

where M is a size of the gap and K is order of the autoregressive model. The matrix **R** is assembled in a similar way. For the detailed steps of the computation see [39].

Weighted Forward-Backward Predictor

The second method based on the AR model is the Weighted Forward-Backward Predictor which in contrast to previous method does not compute complex matrix inversion in Eq. 2.22. Instead, it is looking for suboptimal solutions through weighting of forward and backward prediction as in Eq. 2.6. Detailed descriptions of these algorithms are in [39].

2.2.3 AR modelling of signal parameters

Pure time domain interpolation methods presented in previous section often fail if the length of the signal gap is longer than 10 ms. Bringing into account a two-dimensional time-frequency signal structure promises more space to the process of restoration of degraded recordings [64]. This section describes a method of multi-band interpolation.

Sinusoidal modelling

Sinusoidal model of the signal interprets the audio signal as a sum of harmonic and non-harmonic components usually called the *partials*. These components are

described by an amplitude, frequency and phase in time. Using these parameters the resulting signal is obtained as

$$y(t) = \sum_{p=1}^{P} A_p(t) \cos(\phi_p(t)), \tag{2.26}$$

where P is the number of partials, A_p is an instantaneous amplitude and ϕ_p is an instantaneous phase of the pth partial. The phase is defined as

$$\phi_p(t) = \phi_p(0) + 2\pi \int_0^t f_p(u)\mathrm{d}u, \tag{2.27}$$

where f_p is the instantaneous frequency. This group of three parameters (f_p, A_p, ϕ_p) of the additive model represents particular samples of partials. However, to achieve a satisfying reconstruction the observed signal has to be stationary pseudo-periodically [61].

Decomposition of the signal to partials is mostly performed by McAulay-Quartieri method [67] in the state-of-the-art contributions. This method is based on the Fourier transform. Three main steps of the algorithm are following:

1. Time-Frequency analysis using the Short Time Fourier Transform (STFT),
2. extraction of local maxims (peaks) to get the triplet of parameters,
3. assembling of obtained parameters in time.

Because of various reasons some of parameters of the peaks can be faulty or even missing. In the second step, the peaks extraction, these corrupted peaks could be thrown away, therefore, the affected spectral component will never be able to be reconstructed. Backward reconstruction of the signal from sinusoidal partials is rendered by simple sinusoid oscillators, with f_p, A_p, ϕ_p as input parameters. Utilization of this modelling for missing signal reconstruction is in section 7.3.3.

Prediction and connecting of partials

Set $\boldsymbol{B}$ represents sinusoids before the gap with the last sample position of n_1. Set $\boldsymbol{A}$ represents sinusoids after the gap with the first sample position of n_2. The partials of the $\boldsymbol{B}$ and $\boldsymbol{A}$ sets are identified as P_i and P_j respectively, such that

$$P_i = \{P_i(n), n = n_1 - l_i + 1, \ldots, n_1\}, \tag{2.28}$$

$$P_j = \{P_j(n), n = n_2, \ldots, n_2 + l_j - 1\}, \tag{2.29}$$

$$P_k(n) = (f_k(n), A_k(n), \phi_k(n)), \text{ for all } k, \tag{2.30}$$

where l_i and l_j represent the size of P_i and P_j respectively. See Fig. 2.3 for graphical interpretation. The triplet of instantaneous parameters at the sample n is contained in $P_k(n)$.

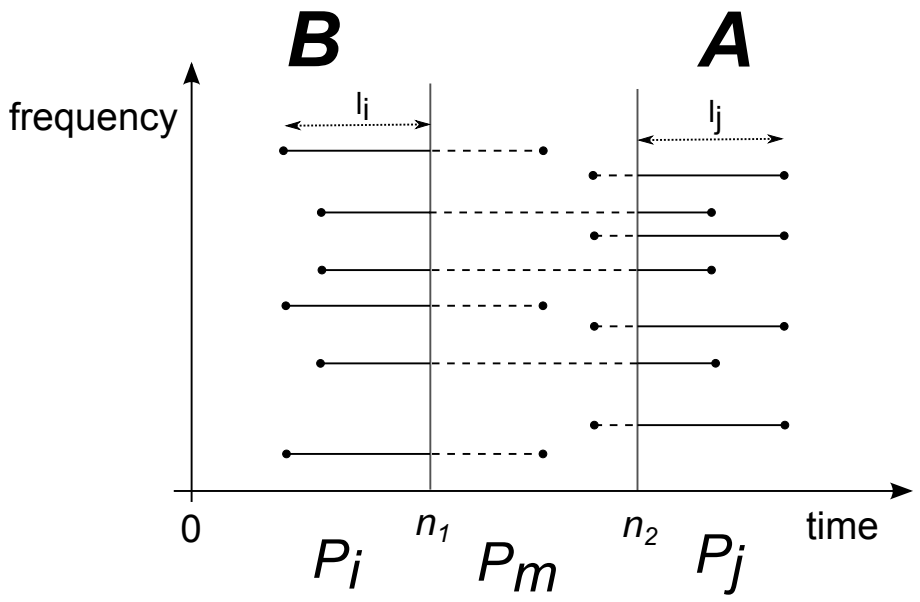


Fig. 2.3: Set of sinusoids with figured gap.

Considering $\hat{P}_i$ the predicted information from the left side of the gap and $\hat{P}_j$ from the right side, respectively, such that

$$\hat{P}_i = \{\hat{P}_i(n_1 + k), k = 1, \ldots, n_2 - n_1 - 1\}, \tag{2.31}$$

$$\hat{P}_j = \{\hat{P}_j(n_2 - k'), k' = 1, \ldots, n_2 - n_1 - 1\}, \tag{2.32}$$

$$\hat{P}_k(n) = (\hat{f}_k(n), \hat{A}_k(n)), \text{ for all } k. \tag{2.33}$$

$\hat{P}_k(n)$ encapsulates the couple of predicted frequency and amplitude because the phase is not being predicted, but deduced from the frequency.

The missing signal portion is interpolated by a resulting partial $\hat{P}_m$ starting at $n_1 + 1$ and finishing at $n_2 - 1$. $\hat{P}_m$ is represented as a combination of $\hat{P}_i$ and $\hat{P}_j$ if certain circumstances (described in [61]) are satisfied. A couple of $\hat{P}_i$ and $\hat{P}_j$ could be merged together if these two partials are matching. If not, unmatched pairs are extrapolated.

The predicted parameters $\hat{P}_i$ and $\hat{P}_j$ are estimated using the linear prediction coefficients obtained by the Burg method. The number of observed samples must be at least twice the model order. Linking the partials is decided using a criterion of the maximum distance of the last frequency of P_i and the first frequency of P_j. If the distance is below the threshold Δ_f, such that

$$|f_i(n_1) - f_j(n_2)| < \Delta_f, \tag{2.34}$$

the two partials are considered as a couple. This process is illustrated in a block scheme in Fig. 2.4.

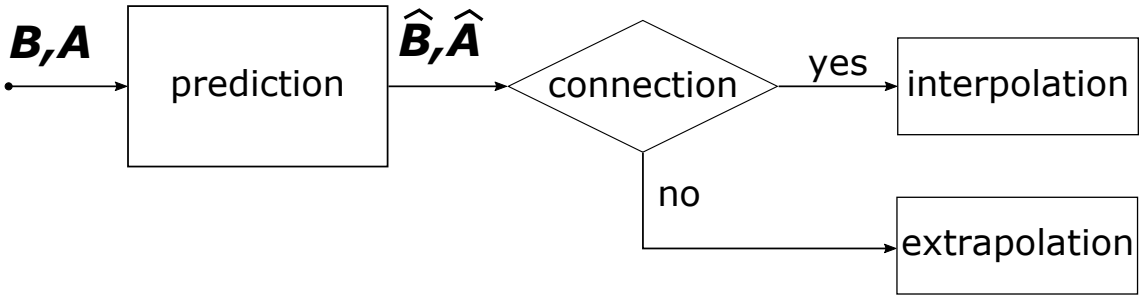


Fig. 2.4: Block scheme of partials linking decision algorithm.

Interpolation of the frequency $\hat{f}_m$ from the predicted frequencies $\hat{f}_i$ and $\hat{f}_j$ is computed by crossfading using a window function w as

$$\hat{f}_m(n) = w\left(\frac{n-n_1}{n_2-n_1}\right)\hat{f}_i(n) + \left[1 - w\left(\frac{n-n_1}{n_2-n_1}\right)\right]\hat{f}_j(n). \tag{2.35}$$

If the predicted frequencies $\hat{f}_i$ and $\hat{f}_j$ are of the same length then the symmetric cosine window w is computed as

$$c(t) = \frac{1+\cos[\pi(1+t)]}{2}. \tag{2.36}$$

In the case that the two partials are not the same length an asymmetric crossfading has to be performed [61].

Usually, the amplitude of a partial is modulated more than the frequency. Starting with $\hat{A}_i(n_2)$ equal to the mean amplitude of the partial P_j from sample n_2 to $\min(n_2+M, n_2+l_j-1)$, where M should be chosen to set an individual length of the considered partial length, the increment of the predicted amplitude $\hat{A}_i$ is defined as

$$\delta_i(n) = \frac{n-n_1}{n_2-n_1}\left[\frac{\sum_{p=0}^{\min(M,l_j-1)} A_j(n_2+p)}{\min(M,l_j-1)+1} - \hat{A}_i(n_2)\right]. \tag{2.37}$$

Likewise, the according approach is applied on the amplitude estimate from the right side $\hat{A}_j$ by adding an increment $\delta_j(n)$ computed as

$$\delta_j(n) = \frac{n_2-n}{n_2-n_1}\left[\frac{\sum_{p=0}^{\min(M,l_i-1)} A_i(n_1-p)}{\min(M,l_i-1)+1} - \hat{A}_j(n_1)\right]. \tag{2.38}$$

Finally, an asymmetrical crossfading of the corrected amplitudes is performed as

$$\hat{A}_m(n) = w\left(\frac{n-n_1}{n_2-n_1}\right)[\hat{A}_i(n)+\delta_i(n)] + \left[1 - w\left(\frac{n-n_1}{n_2-n_1}\right)\right][\hat{A}_j(n)+\delta_j(n)]. \tag{2.39}$$

Interpolation of phase in the missing segment is computed as

$$\tilde{\varphi}(n_1+1) = \phi_m(n_1) + \pi T[f(n_1) + \hat{f}_m(n_1+1)], \tag{2.40}$$

$$\tilde{\varphi}(n) = \tilde{\varphi}(n_1+1) + \pi T \sum_{p=n_1+1}^{n} [\hat{f}_m(p-1) + \hat{f}_m(p)], \tag{2.41}$$

where $\varphi(n)$ is an unwrapped phase at sample $n \in [n_1+2, n_2]$ and T is a hop size in seconds. After this approximation a non-smooth phase discontinuity could arise at the end of the missing segment. Therefore, an error of the phase is computed and spread through the missing segment, for details see [61].

In case when the condition 2.34 is not satisfied, the partials are unmatched and are going to be extrapolated. Considering partials starting in $\boldsymbol{B}$ the sinusoid decays in the missing region. On the other hand, partials finishing in $\boldsymbol{A}$ probably start in the gap. Predicted parameters $\hat{P}_i$ and $\hat{P}_j$ are the starting point of the extrapolation.

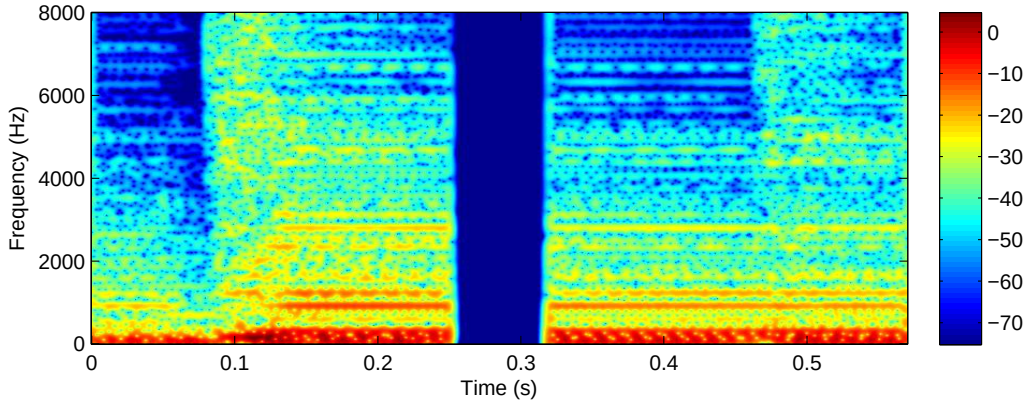
Consider the maximum length of extrapolation $l_{\boldsymbol{B}}$ and $l_{\boldsymbol{A}}$, respectively. Frequencies $\hat{f}_i$ and $\hat{f}_j$ are predicted as described in previous paragraphs together with the phase. The amplitude of the partial from left side has to be faded out according to $l_{\boldsymbol{B}} \leq n_2 - n_1 - 1$. On the contrary, generating an onset of a non-connected partial of part $\boldsymbol{A}$ is not that easy. Affected partials onsets of all sinusoids should be synchronized: started on the same index $n_2 - l_{\boldsymbol{A}}$ and faded.

Interpolation of residuum background noise

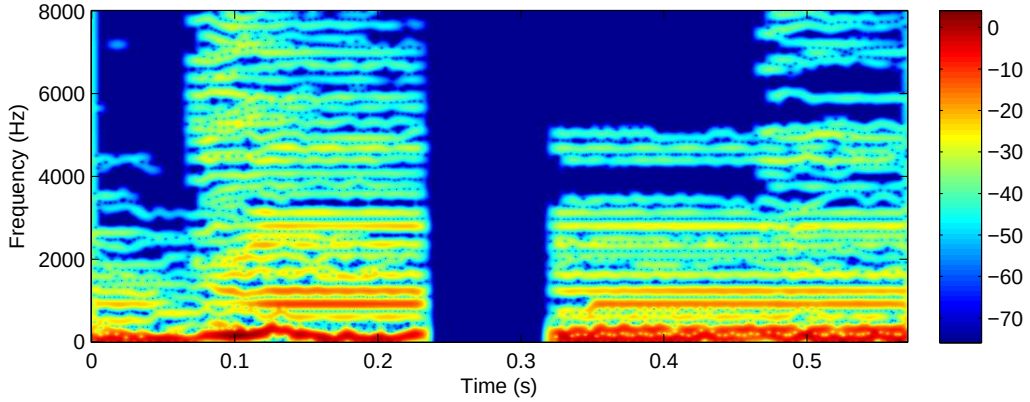
The proposed method in [61] models only the tonal part of the signal while the noisy part is not considered at all. An extension was brought in [64] adding the interpolation of the noisy residual signal and minor harmonics. Since most of real world signals contain noise with significant amount of energy, the residuum should not be avoided.

The proposed approach begins with subtracting the synthesized partials from the original audio signal for $\boldsymbol{B}$ and $\boldsymbol{A}$ side separately. The interpolation of the residuum is performed for all frequency bands using the same algorithm as for the sinusoidal model with coefficients obtained by the Burg method. Furthermore, the interpolated residuals from the left and the right side are cross-faded using a proper window. After the interpolation of the residuum, the synthesized sinusoidal components are summed together with the residuum to get the full interpolation of the gap.

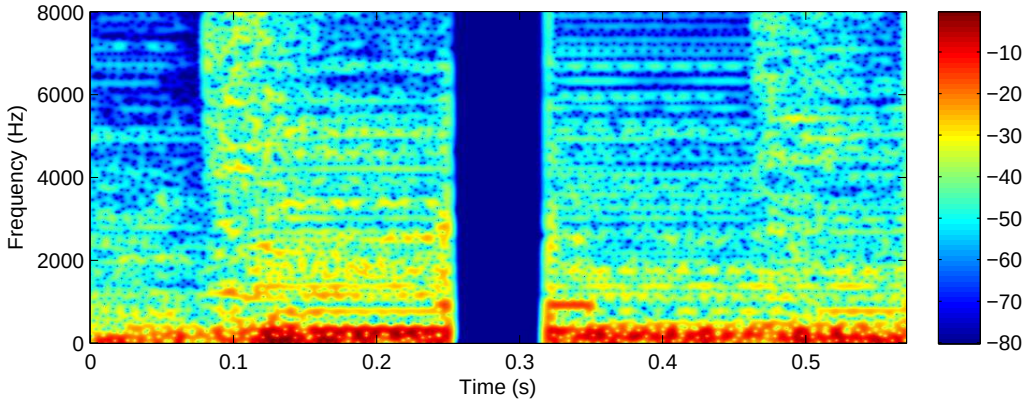
An example of the tonal and noisy part separation by sinusoidal modeling is on Fig. 2.5. The source signal is a recording of a guitar playing one chord repeatedly.



(a)



(b)



(c)

Fig. 2.5: A spectrogram of piece of the harmonic signal (guitar chord) with a gap: the original signal (a), tonal parts of the signal (b) and a residuum (c).

2.3 Chapter summary

Presented state-of-the-art methods are supposed to be comparative approach to the novel methods based on underdetermined systems of linear equations which will be described in the next chapter. Regarding the results in papers utilized as the references in this chapter, the sinusoidal modeling should be the most promising state-of-the-art method for audio signal interpolation. Experiments of an audio interpolation using these methods will be described in Sec. 7.3.

3 PRELIMINARY KNOWLEDGE

3.1 Notation

Hermitian transpose $\mathbf{A}^*$ is a complex conjugate of a complex matrix $\mathbf{A}$ defined as

$$\mathbf{A}^* = (\bar{\mathbf{A}})^T. \tag{3.1}$$

A support $\text{supp}(\mathbf{x}) = \{i \,|\, x_i \neq 0\}$ is a group of indices where a vector $\mathbf{x}$ has nonzero values .

A k-sparse vector $\mathbf{x} \in \mathbb{C}^N$ which has at most k nonzero values is defined as

$$\|\mathbf{x}\|_0 \leq k. \tag{3.2}$$

Real signals are usually not sparse in a strict sense and instead of zero values there are very small non-zero values [47], [74], [33].

3.2 Vector norms

In the area of mathematics, norm is a function that assigns non-negative length (size) to each vector. Zero vector has a length of zero. Definition of a ℓ_p-norm of a vector $\mathbf{x} \in \mathbb{C}$ is

$$\begin{aligned}
\|\mathbf{x}\|_p &:= \left(\sum_{i=1}^{N} |x_i|^p \right)^{1/p} \quad \text{for} \quad 1 \leq p < \infty, \\
\|\mathbf{x}\|_p &:= \sum_{i=1}^{N} |x_i|^p \quad \text{for} \quad 0 < p < 1, \\
\|\mathbf{x}\|_\infty &:= \max_i |x_i| \,, \\
\|\mathbf{x}\|_0 &:= |\text{supp}(\mathbf{x})| \,.
\end{aligned} \tag{3.3}$$

In a strict sense the norm is considered just in $1 \leq p \leq \infty$ case, however, for thesis purposes p will be used as a general notation of ℓ_p-norm. Among all possible p values the most common are:

- $\|\cdot\|_0$ which represents a number of nonzero elements in a vector $\|\mathbf{x}\|_0 = |\text{supp}(\mathbf{x})|$,
- $\|\cdot\|_1$ which represents a sum of absolute values of a vector $\|\mathbf{x}\|_1 = \sum_i |x_i|$,
- $\|\cdot\|_2$ which represents an euclidian norm of a vector $\|\mathbf{x}\|_2 = \sqrt{\sum_i |x_i|^2}$. The notation is usually simplified as $\|\cdot\|$.

Unit balls of the most common norms are illustrated in Fig. 3.1 [47], [33].

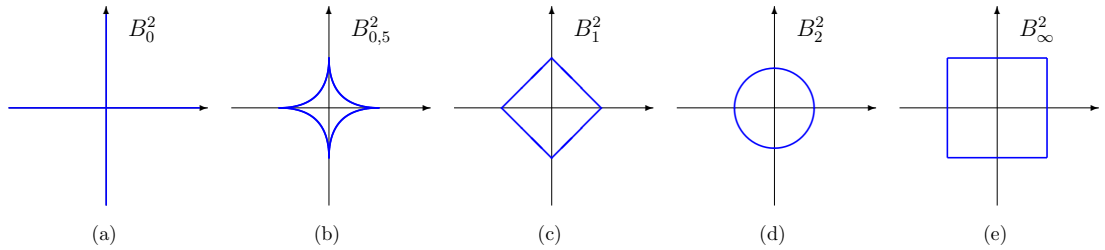


Fig. 3.1: Illustration of unit balls (a) B_0^2, (b) $B_{0,5}^2$, (c) B_1^2, (d) B_2^2 a (e) B_∞^2.

3.3 Matrix norms

Considering matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ where m is a number of rows and n is a number of columns several common norms are usually utilized e.g. Frobenius norm, Nuclear norm or Total Variation (TV) norm (this one is usable also for vectors) etc. In this thesis we will be dealing with (p, q)-mixed norms defined as

$$\|\mathbf{X}\|_{p,q} = \left\| \left[\|x_{1,1}, \ldots, x_{1,m}\|_p, \|x_{2,1}, \ldots, x_{2,m}\|_p, \ldots, \|x_{n,1}, \ldots, x_{n,m}\|_p \right] \right\|_q, \tag{3.4}$$

where ℓ_p-norm is applied on each row and ℓ_q-norm is applied on a resulting vector. This could be rewritten as

$$\|\mathbf{X}\|_{p,q} = \left(\sum_{g=1}^{G} \left(\sum_{m=1}^{M} |x_{g,m}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}}, \tag{3.5}$$

where g and m represent indices of the group and member, respectively (see Sec. 5.3). This thesis and signal processing generally is dealing with $\|\mathbf{X}\|_{1,2}$ or $\|\mathbf{X}\|_{2,1}$ [59].

3.4 Pseudoinverse

Solving an underdetermined linear system (more columns than rows) $\mathbf{Ax} = \mathbf{b}$ we have infinite number of solutions. There is just a single solution with the smallest energy $\mathbf{A}^+\mathbf{b}$. Pseudoinverse $\mathbf{A}^+$ of a matrix $\mathbf{A}$ is a generalization of the inverse matrix for non-regular matrices satisfying all of the following criteria:

1. $\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}$,
2. $\mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+$,
3. $(\mathbf{A}\mathbf{A}^+)^* = \mathbf{A}\mathbf{A}^+$,
4. $(\mathbf{A}^+\mathbf{A})^* = \mathbf{A}^+\mathbf{A}$.

If matrix $\mathbf{A}$ has got full row rank, $\mathbf{A}\mathbf{A}^*$ is invertible, $\mathbf{A}^+$ is computed as

$$\mathbf{A}^+ = \mathbf{A}^*(\mathbf{A}\mathbf{A}^*)^{-1} \tag{3.6}$$

and it results in the right inverse $\mathbf{A}\mathbf{A}^* = \mathbf{I}$ [48], [24], [91], [15].

3.5 Vector spaces and basis vectors

A vector space $\mathbb{V}$ of finite dimension $n \in \mathbb{N}$ is generated by a system of generators $\mathbf{E} \subset \mathbb{V}$. Each vector $\mathbf{x} \in \mathbb{V}$ is then built up by a finite linear combination of generators. Minimal system of linearly independent generators is called the *basis* of a vector space and any vector from vector space $\mathbb{V}$ can be uniquely constructed using these generators. By placing each basis vectors $\{\mathbf{b}_1, \dots \mathbf{b}_n\}$ as columns a basis matrix $\mathbf{B}$ is created. Therefore, any vector $\mathbf{x} \in \mathbb{V}$ is

$$\mathbf{x} = \sum_{i=1}^{n} c_i \mathbf{b}_i = \mathbf{B} \cdot \mathbf{c}, \tag{3.7}$$

where c_i are scalars called the coordinates of signal $\mathbf{x}$ in $\mathbf{B}$. An index i can represent various indexing dependent on a type of dictionary. In a case of frequency (Fourier) dictionary it means a frequency, in a case of time-scale dictionary it means a dilation and a scale, in a case of time-frequency dictionary it means a translation and a modulation [78], [74].

3.5.1 Orthogonal and orthonormal basis

It is very advantageous using of the orthogonal and orthonormal basis. The definition of orthogonal basis is that for every random pair of vectors from basis $\mathbf{B} = \{\mathbf{b}_1, \dots \mathbf{b}_n\}$ the following condition is fulfilled

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0, \quad \langle \mathbf{b}_i, \mathbf{b}_i \rangle \neq 0. \tag{3.8}$$

In the other words, all possible pairs of a basis vectors are perpendicular to each other. Moreover, the orthonormal basis is defined such that $\|\mathbf{b}_i\| = 1$ or $\mathbf{B}^* = \mathbf{B}^{-1}$ [74].

3.6 Frames

3.6.1 General definition

The representation of the signal $\mathbf{x}$ is not unique considering the number of generators of a vector space $\mathbb{V}$ is greater than the dimension n of the space. The signal vector can be represented using various linear combinations. This property is called the *underdetermination* and the group of linearly dependent basis vectors is called the *frame*. Frames are generally less constrained than the basis and are widely utilized because of their flexibility. According to a mathematical definition a frame is formed by a countable set of vectors $\{\mathbf{\Phi}_k\}_{k \in J}$ in a vector space $\mathbb{V}$ if there are two positive

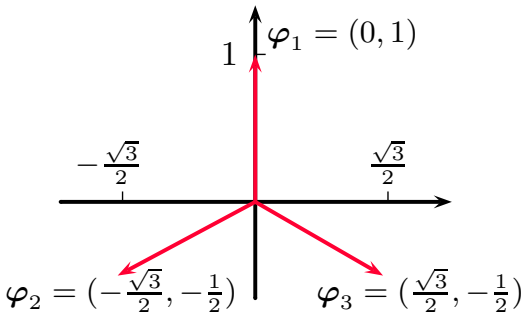


Fig. 3.2: Mercedes-Benz frame.

constants $0 < A \leq B < \infty$ such that

$$A\|\mathbf{x}\|^2 \leq \sum_{k \in J} |\langle \mathbf{x}, \mathbf{\Phi}_k \rangle|^2 \leq B\|\mathbf{x}\|^2, \ \forall \mathbf{x} \in \mathbb{V}, \tag{3.9}$$

where constants A, B are representing *the frame bounds*. Each element (column) of a frame $\mathbf{\Phi}_k$ is called the *atom* [74], [25].

3.6.2 Frame operator, dual frames

The *frame operator* $\mathbf{S}$ is defined as

$$\mathbf{S}\mathbf{x} = \sum_k \langle \mathbf{x}, \mathbf{\Phi}_k \rangle \mathbf{\Phi}_k. \tag{3.10}$$

In a finite dimension the frame operator is represented as a matrix such that

$$\mathbf{S} = \mathbf{\Phi}\mathbf{\Phi}^*. \tag{3.11}$$

The upper bound A is equal to the smallest eigenvalue of a frame operator. Likewise, lower frame bound B is equal to the biggest eigenvalue. If $A = B$ the frame is called *tight*. Moreover, if $A = B = 1$ the frame is called *normalized* or *Parseval tight*. The most basic example of a tight frame is the Mercedes-Benz frame illustrated in Fig. 3.2 [78].

Dual frame is an important element in searching for coordinates c_k for a vector representation as $\mathbf{x} = \sum_k c_k \mathbf{e}_k$ using frame $\mathbf{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_m\}$ in vector space $\mathbb{V}$ of dimension $n < m$. Frame $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_m\}$ is called the dual frame of $\mathbf{E}$ if each $\mathbf{x} \in \mathbb{V}$ is defined as

$$\mathbf{x} = \sum_k \langle \mathbf{x}, \mathbf{f}_k \rangle \, \mathbf{e}_k = \sum_k \langle \mathbf{x}, \mathbf{e}_k \rangle \, \mathbf{f}_k. \tag{3.12}$$

A matrix representation of the same equation as (3.12) is

$$\mathbf{x} = \mathbf{E}\mathbf{F}^*\mathbf{x} = \mathbf{F}\mathbf{E}^*\mathbf{x}. \tag{3.13}$$

Searching for coordinates c_k of vector $\mathbf{x}$ in a primary frame is performed using the dual frame by equation $c_k = \langle \mathbf{x}, \mathbf{f}_k \rangle$. Matrix representation of the same problem is $\mathbf{c} = \mathbf{F}^*\mathbf{x}$. This operation is called the *analysis*. A backwards reconstruction of the signal by superposition of atoms is called the *synthesis*. Detailed description of these two operations in the sense of searching for sparse representations will be discussed in Sec. 5.2.2 [74].

Each primary frame has got an infinite number of dual frames in general. However, usually the task is to find a *canonical* dual frame. This frame results in a set of coefficients $\{c_k\}_{k \in J}$ with minimal energy (norm). Such frame is defined as

$$\mathbf{f}_k = \mathbf{S}^{-1}\mathbf{e}_k \tag{3.14}$$

or in matrix form, respectively as

$$\mathbf{F} = \mathbf{S}^{-1}\mathbf{E}. \tag{3.15}$$

If matrix $\mathbf{E}$ has a full row rank, using the Moore-Penrose pseudoinverse (3.4) a canonical dual frame can be obtained by

$$\mathbf{E}^+ = \mathbf{E}^*(\mathbf{E}\mathbf{E}^*)^{-1}. \tag{3.16}$$

Because of a large size and an insufficient conditionality of matrices searching for a dual frame or inverse of $\mathbf{S}$, respectively, can result in a computationally hard task. A tight frame whose frame operator $\mathbf{S}$ is a diagonal matrix, such as

$$\mathbf{S} = A\mathbf{I} \tag{3.17}$$

can be advantageous in this case. Dual frame is computed just by rescaling of the primary frame as

$$\mathbf{f}_k = \mathbf{e}_k / A. \tag{3.18}$$

Therefore, an important attention is focused on obtaining a tight frame [78], [74].

3.6.3 Gabor frames

A special family of frames are the *Gabor frames* [74], [16]. The construction is based on the translation and modulation operators. The signal $\mathbf{x}$ is represented as a superposition if translated and modulated version of the basic function $g \in L^2(\mathbb{R})$ which are generated as

$$g_{\tau,\omega}(t) = g(t - \tau)\mathrm{e}^{2\pi i t\omega}. \tag{3.19}$$

The question is which basic function $g \in L^2(\mathbb{R})$ and translation and modulation parameters τ, ω choose such that the resulting Gabor system creates a frame in the $L^2(\mathbb{R})$ space [24]. Function g is called the window function or generator and the first case in the history was to use the Gaussian function with infinite support vector

$$g(x) = \exp(-x^2/2). \tag{3.20}$$

3.6.4 Partition of Unity

Considering tight frames, windows as a basic building blocks of atoms should fulfill a condition called the Partition of Unity (PU) in their fundamental definition (without modulation or translation). PU means that the sum of all windows samples is equal to 1 such as

$$\sum_{n \in N} g_n(k) = 1 \quad \text{for} \quad k \in \mathbb{Z}, \tag{3.21}$$

where N is a number of atoms of the frame [75]. If a particular window is a square root of a window satisfying PU then the window generates a tight Gabor frame (and also the Wilson/Windowed Modified Discrete Cosine Transform (WMDCT) basis) if the number of frequency channels (modulation steps) is smaller than the window length [76].

3.7 Chapter summary

In this section the basic knowledge needed for understanding the following sparse representations theory was introduced. A lot of consequence knowledge about frames theory is in detail described in referenced books and papers, however, deeper look into the details is not necessary for this thesis. Following chapters will mostly deal with frames, therefore, a back reference into this chapter may be notable.

4 SPARSE REPRESENTATIONS

During last decade, the attention of researchers in the field of signal processing increasingly focused on mathematical methods searching so-called *sparse representations*. Mathematical fields such as linear algebra, functional analysis, convex optimization or statistics provide a basis for finding sparse solutions of systems of linear equations which is usable in various fields. This thesis focuses on the use of sparse representations in the field of signal theory and systems, which were one of the first areas of sparse solutions applications.

The basic task is to find a solution of a linear system

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{y} = \mathbf{D}\mathbf{x}, \tag{4.1}$$

where $\mathbf{D}$ is a transformation matrix called the *dictionary*, $\mathbf{x}$ is a coefficient vector, which we are looking for and $\mathbf{y}$ is a vector of input signal samples. Eq. 4.1 represents the synthesis version of the problem which was introduced first from a historical point of view. Solution is called sparse if the resultant vector of the coefficients $\mathbf{x}$ contains only a few non-zero elements in comparison with the size of the system of linear equations. Because of uncertainty of the solution there is a possibility of adaptivity of the solution, which is amongst others the most appropriate for the particular purpose. Graphical demonstration of the problem is in Fig. 4.1

4.1 Properties of sparse representations

Advantages of sparse representations

Sparse solutions offer adaptation of the dictionary for a particular purpose which benefits in terms of information compression, analysis, interpretation and numerical stability. Their advantage is the ability to represent the signal with a few important coefficients. Additionally, there are approaches how get much higher resolution of sparsely representable objects [92].

Disadvantages of sparse representations

Because the selected dictionaries are overdetermined (the matrix contains more columns than rows), searching for such solutions can be computatively intensive, depending on the choice of algorithm for signal decomposition (see Sec. 5). Moreover, improperly assembled dictionaries can lead to system instability. Problems with stability can be avoided in advance during building of the dictionary.

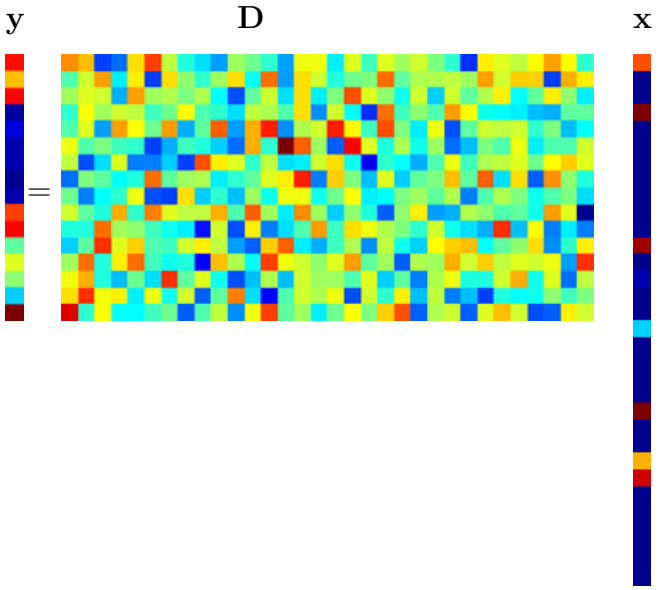


Fig. 4.1: Graphical illustration of the sparse representation

4.2 Audio Inpainting

Sparse signal representations for inpainting problems were first used in image signal processing [35] and a few years later the Audio Inpainting algorithm was introduced in [12].

The Audio Inpainting is based on the approximation of missing or distorted information with a linear combination of atoms $\{\mathbf{d}_j\}$ from the dictionary. Input signal samples are classified into two parts: reliable samples with support vector $\mathcal{I}^{\mathrm{r}}$ and distorted samples with support vector $\mathcal{I}^{\mathrm{m}}$. The process of separation of missing/distorted samples is done either manually or using some error detection algorithm. Longer signals (typically audio) can be segmented into portions of defined length equal to the atom length N and defined overlap, while information about signal support is preserved. Segments are formed in a matrix $\mathbf{Y}$ as well as their support vectors form the measurement matrix $\mathbf{M}^{\mathrm{r}}$ with reliable samples support and $\mathbf{M}^{\mathrm{m}}$ with missing signals support. For those segments where wrong samples are detected an individual inpainting process is performed. Wrong samples are identified in a measurement matrix $\mathbf{M}^{\mathrm{m}} \in \{0, 1\}$ as ones as well as reliable samples in matrix $\mathbf{M}^{\mathrm{r}} \in \{0, 1\}$ with values complementing the $\mathbf{M}^{\mathrm{m}}$ matrix. Using the measurement matrix reliable data are obtained as

$$\mathbf{y}^{\mathrm{r}} = \mathbf{M}^{\mathrm{r}}\mathbf{y} \tag{4.2}$$

and the main goal is to reconstruct the non-reliable samples $\mathbf{y}(\mathcal{I}^{\mathrm{m}})$, where $\mathcal{I}^{\mathrm{m}} = \{1, 2, \ldots, L\}$ is a vector of missing signal support from the whole input signal of length L. Regarding sparse representations of signals, each segment of reliable samples can be obtained by the input signal approximation through dictionary atoms

and appropriate coefficients

$$\mathbf{y}_i^r = \mathbf{M}_i^{\mathrm{r}}\mathbf{D}\mathbf{x}_i. \tag{4.3}$$

Coefficients $\mathbf{x}_i$ are computed from reliable samples using any greedy or relaxation algorithm. The process of analysis is performed by the dictionary $\mathbf{D}$ with atom length restricted to the length of the reliable samples of input signal. Recovering unknown samples $\widehat{\mathbf{y}}(\mathcal{I}^{\mathrm{m}})$ can be performed by estimating as $\widehat{\mathbf{x}}_i$ a sparse vector of each segment

$$\widehat{\mathbf{y}}_i(\mathcal{I}_i^{\mathrm{m}}) = \mathbf{M}_i^{\mathrm{m}}\mathbf{D}\widehat{\mathbf{x}}_i. \tag{4.4}$$

For reconstruction of missing samples with obtained coefficients $\mathbf{x}_i$ we use the original dictionary $\mathbf{D}$ with original atoms length. Only samples at missing positions are replaced by the reconstructed samples

$$\widehat{\mathbf{y}} = \mathbf{y}(\mathcal{I}^{\mathrm{r}}) + \widehat{\mathbf{y}}(\mathcal{I}^{\mathrm{m}}). \tag{4.5}$$

4.2.1 Audio Declipping

When the absolute value of the signal level is higher than the maximum input range of a digital acquisition system the audio waveform is truncated. This phenomenon is called *audio clipping*. An audio restoration task to remove this kind of distortion is called *audio declipping*.

Solving the problem by sparse representations results in an inverse problem close to the audio inpainting with additional constraint incorporated. Suppose we know the position of positive (resp. negative) clipped samples $\mathbf{M}^{\mathrm{c}+}$ and $\mathbf{M}^{\mathrm{c}-}$, level of the clipping $\hat{\theta}_{\mathrm{c}}$ and $-\hat{\theta}_{\mathrm{c}}$ and the maximum level of the signal $\hat{\theta}_{\max}$ and $-\hat{\theta}_{\max}$ such that

$$\mathbf{M}^{\mathrm{c}+}x \geq \hat{\theta}_{\mathrm{c}} \tag{4.6}$$

$$\mathbf{M}^{\mathrm{c}-}x \leq -\hat{\theta}_{\mathrm{c}} \tag{4.7}$$

and

$$\mathbf{M}^{\mathrm{c}+}x \leq \hat{\theta}_{\max} \tag{4.8}$$

$$\mathbf{M}^{\mathrm{c}-}x \geq -\hat{\theta}_{\max}. \tag{4.9}$$

Solving the problem with this additional knowledge is advantageous compared to the regular audio inpainting and according to the recent contributions it brings better results [11][86][56]. The amplitude of clipped portions of signal was originally higher than the upper clipping level or lower than the under clipping level, respectively (see Fig. 4.2). Therefore, the possible amplitude of the signal under reconstruction is constrained and the solution will more likely to be more accurate. This was just a small remark about the extension of the general audio inpainting problem, in the rest of the thesis audio declipping is not discussed anymore.

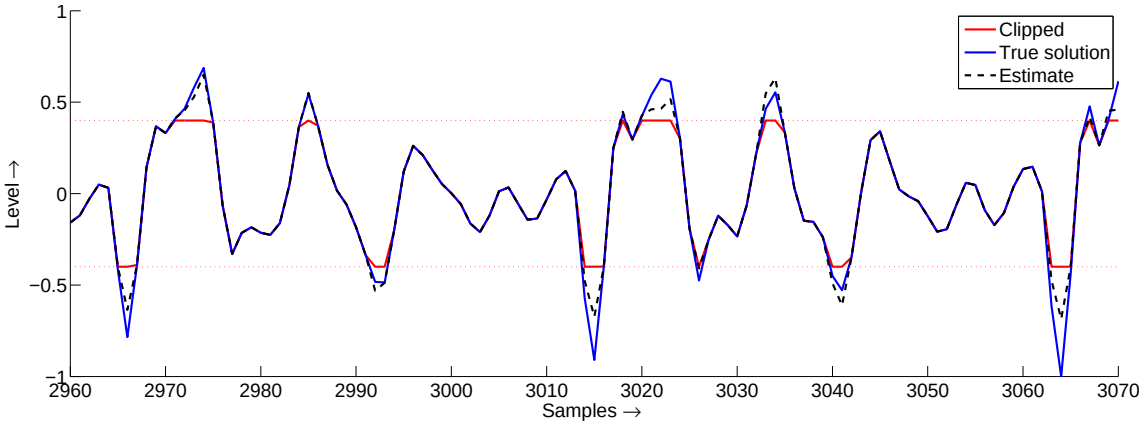


Fig. 4.2: An example of audio declipping.

4.3 Dictionaries

4.3.1 Most frequent dictionaries

Trivial dictionaries

The easiest way to create a dictionary is set of Dirac impulses. Each atom is a vector of zeros except the only value $\Phi_i[n] = 1_{\{n=i\}}$ for $i \in \{0, 1, \ldots, N-1\}$. This dictionary is an orthonormal basis in $\mathbb{R}^N$.

Another example of such a trivial dictionary is a Heaviside dictionary whose atoms are step functions with a specific point $\Phi_i[n] = 1_{\{n \geq i\}}$ for $i \in \{0, 1, \ldots, N-1\}$. However, atoms in this dictionary are not orthogonal.

Frequency dictionaries

Definitely the most popular dictionary from this group is the Fourier dictionary as a collection of sine and cosine functions. Cosine dictionary is generally defined as

$$\Phi_k = \cos(\omega_k n), \tag{4.10}$$

where ω_k corresponds to frequencies $\omega_k = \pi k/N$, $k = 0, \ldots, N$ and $n \in \{0, 1, \ldots, N\}$. A dictionary which contains N frequencies (atoms) is naturally orthogonal a produces a basis in $\mathbb{R}^N$. An overcomplete Fourier dictionary is obtained by denser frequency sampling. In the case of cosine dictionary redundancy particular frequencies are defined as $\omega_k = \pi k/(lN)$ for $k = 0, \ldots, N$. Redundancy is reached by $l > 1$ [23].

There are eight types of cosine dictionaries. Above all DCT-IV is the most popular type utilized in audio signal processing. The others are more common in

image signal processing. DCT-IV is defined as

$$X_k = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right] \tag{4.11}$$

for $k = 0, \ldots, N-1$ [80].

Time-scale dictionaries

This group of dictionaries is represented by the wavelet transform in particular [23], [50]. The fundamental component is a waveform called the *mother wavelet* ψ. The dictionary is built up using translations and dilations of this mother wavelet. Indices $k = (a, b)$ define the scale $a \in (0, \infty)$ and the dilation $b \in [0, \infty]$, respectively. The wavelet itself is defined as

$$\Phi_{(a,b)} = \psi(a(t-b)) \cdot \sqrt{a}. \tag{4.12}$$

If we want to build an orthonormal basis in $L^2(\mathbb{R})$ the dictionary must contain n atoms with scaling parameter specified by the diadic scaling

$$a_j = 2^j/n \quad \text{for} \quad j = j_0, \ldots, \log_2(n) - 1. \tag{4.13}$$

Further, the dialation have to satisfy the condition of an integer multiple of the scale

$$b_{j,k} = k \cdot a_j \quad \text{for} \quad k = 0, \ldots, 2^j - 1. \tag{4.14}$$

Likewise the frequency dictionary, an overcomplete case is reached by denser sampling of the dilation parameter [30]. Considering wavelet frames for sparse signal representations is more complex issue. However, wavelets are not utilized for audio inpainting in this thesis.

Time-frequency dictionaries

Traditional Fourier analysis works with harmonic functions that have global range. It means that one important coefficient influences the signal across its entire length. Nevertheless, the natural perception of human hearing considers the frequency structure as time dependent. This is the motivation for extension of the Fourier analysis into time a variant system.

Time-frequency analysis is a projection of signal $\mathbf{y}$ onto the atoms $\mathbf{g}_{\tau,\omega}$. In fact, we are dealing with a redundant STFT known as the *Gabor analysis* [40]. Gabor frames are described in detail in Sec. 3.6.3. The distribution of sampling points in the time-frequency plane and the corresponding frame construction is defined as

$$\mathbf{G}(g, a, b) = \mathbf{M}_{bm}\mathbf{T}_{an}g, \tag{4.15}$$

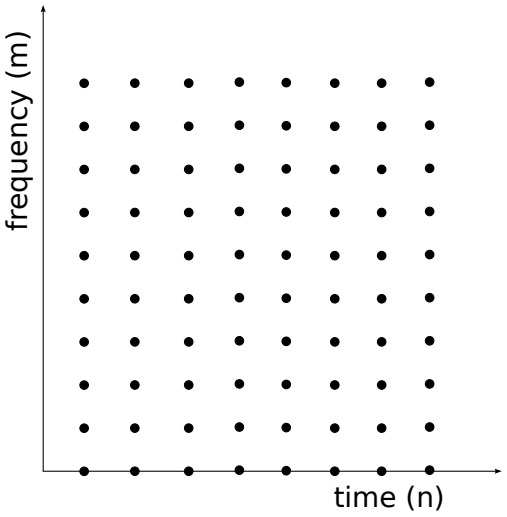


Fig. 4.3: Regular sampling in STFT segments.

where $\mathbf{M}_{bm}$ is a modulation operator, $\mathbf{T}_{an}$ is a translation operator and $a, b, m, n \in \mathbb{Z}$ are sampling parameters of the STFT. An example of such sampling of the time-frequency plane is in Fig. 4.3. The basic question is how to choose parameters a, b and a function $g \in L^2(\mathbb{R})$ to form a frame in space $L^2(\mathbb{R})$. Limits of the time-frequency resolution are described by the *Heisenberg's principle of uncertainty* saying that there is no signal well concentrated in both time and frequency [32]. For more contribution in the field of optimal parameters selection see e.g. [40].

Function g is called the *window function*. Optimal window function is smooth, symmetric and the spectral envelope is decreasing fast enough. The easiest choice of the window function could be the rectangular window. However, the frequency domain of such window is a sinc function with non-optimal spectral decay. In theory the Gaussian function $g(x) = \exp(-x^2/2)$ is the only function with optimal time-frequency concentration according to the Heisenberg's principle [74]. In praxis, the disadvantage is its support of infinite length. Therefore, more feasible window functions are utilized such as Hamming's, Hann's, Nuttall's etc. An example of modulated and translated Hann's window is in Fig. 4.4.

Necessary condition for forming a Gabor frame is that the *lattice* has to be dense enough. This two dimensional array of Gabor coefficients could be visualised as a spectrogram (see an example of the spectrogram in Fig. 1.3 or 7.49). If the Gabor system forms a frame, then the reconstruction of the signal from time-frequency coefficients could be performed.

Traditional Gabor analysis has a regular time-frequency sampling which is not very suitable for real audio samples analysis. Such sounds are naturally not stationary. Recently, the invertible Nonstationary Gabor Transform (NSGT) was presented

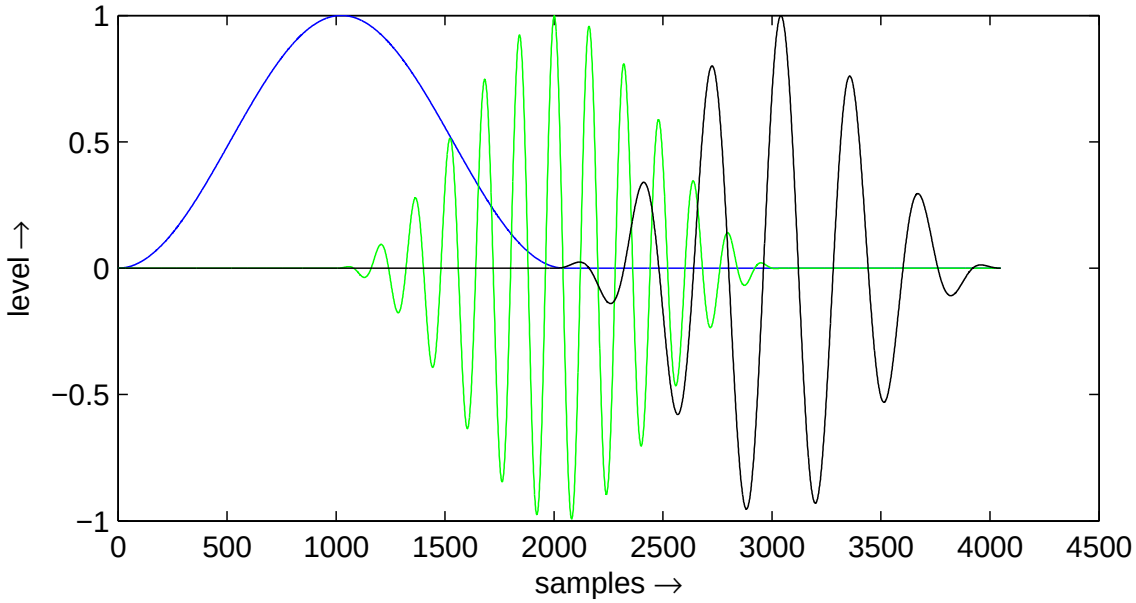


Fig. 4.4: Hann window function (blue), modulated, translated and windowed cosine functions (green, black).

in [16]. This transform allows adaptivity of the analysis window and therefore sampling positions in the time-frequency plane based on the onset detection. During the history of signal processing there were several attempts for implementing of such an adaptive transform. However, this is the first one which allows backward signal reconstruction without an error. Moreover, an implementation is effectively done using Fast Fourier Transform (FFT). An example of an irregular sampling is in Fig. 4.5.

Even more recent papers are describing an application of NSGT for invertible Constant-Q transform [89] and Equivalent Rectangular Bandwidth (ERB) lets [70], which are more convenient for audio signal representations because of their natural approximation of human hearing.

4.3.2 Dictionary learning

Stationary and a-priori specified dictionaries (like DCT or Gabor) are efficient because of their fast transformation process. Such dictionaries are mostly designed for a specific group of signals. One possible way how to represent the signal more sparsely is to adapt the dictionary to a specific signal.

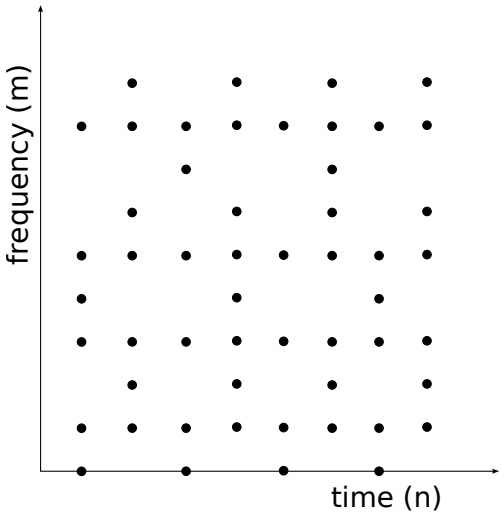


Fig. 4.5: Irregular sampling in STFT segments.

Principle of dictionary learning

The process of adaptation of the dictionary to the specific signal comprises two main steps: the first step is to isolate a group of training samples from the reliable part of the signal. This means that we can not train dictionary from noisy or missing sections of the signal, which will later be reconstructed. The range and number of samples, from which the dictionary will be *learned*, specifies the user when setting parameters for the selected algorithm. In the second step the adaptation itself will be processed on the dictionary data gathered in the first step.

Preparing such adapted dictionary brings a more computationally demanding task, which certainly prolong the total time required for data reconstruction. However, output may result in sparser representation and thus more accurate reconstruction of missing data [17].

4.3.3 State-of-the-art

The first idea of adaptation of the dictionary was published in 1996 by prof. Ölshausenem using the maximum likelihood method [71]. In practice, significantly more successful was a Method of Optimal Directions (MOD) presented in 2000 [37]. The same author, prof. Engan, presented even one year earlier dictionary learning method using the maximum a-posteriori probability [38]. A similar approach was published in 2001 [68]. Comparably important achievement was a few years later algorithm called the Union of Orthonormal Bases [62]. Recently, there is quite an increase of papers dealing with dictionary learning methods, nevertheless, the most popular algorithm is the K-SVD and will be described in detail in the following

subsection.

4.3.4 K-SVD

This algorithm was introduced in 2006 by the team of prof. Elad in [13]. The name is derived from the K-means algorithm [42] which is used for the vector quantisation. The process of vector quantisation consists of assigning of the training data to the most correlated atom from a given set of templates. The correlation is computed using ℓ_2 norm. Each vector from the training data is then represented by a single coefficient. Second part of the name is the Singular Value Decomposition (SVD).

There is an obvious relation between the sparse representations and vector quantisation because vector quantisation is an extreme case of sparse signal representations. Only one coefficient is utilized for the signal decomposition and this coefficient must be binary (0 or 1).

The predecessor of the K-SVD algorithm is the aforementioned algorithm MOD, which updates the entire dictionary in each iteration of a training process. The advantage of K-SVD is that within one iteration only one atom is updated. Thus, the result accelerates the convergence of the system.

K-SVD is a non-convex algorithm and each iteration comprises of two steps. In the first step the dictionary is fixed and the algorithm computes the coefficients. The second step is the dictionary update. The goal of the algorithm is to adapt the dictionary $\mathbf{D}$ in order to achieve a higher degree of sparsity in the representation of input signal $\mathbf{y}_k$, using any algorithm which approximates the optimization problem

$$\widehat{\mathbf{c}}_k = \arg\min_{\mathbf{x}_k} \|\mathbf{y}_k - \mathbf{D}\mathbf{x}_k\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}_k\|_0 \leq S_0. \tag{4.16}$$

K-SVD algorithm is described in details in algorithm 1 or [33].

4.4 Structured sparsity

Description of the sparse representations in previous sections implicitly assumed independence between the synthesis coefficients. However, it is clear especially in natural audio signals exists a relationships between the synthesis coefficients. An example of the relationship could be seen in every spectrogram where the musical signal has some persistence in time and related harmonic components in frequency. Considering structures between the coefficients could be advantageous in audio signals processing in a sparse way as described in Sec. 5.3.

4.5 Chapter summary

The basic introduction to sparse representations has been provided in this chapter. General topic was focused on the problem of Audio Inpainting and corresponding dictionaries for audio signal processing were introduced. Wide set of dictionaries will be restricted in rest of the thesis to frequency and mostly time-frequency dictionaries since their properties are suitable for applying the frames theory and overdetermined systems.

Algorithm 1 K-SVD

Initialization: $\mathbf{D}^{(0)} \in \mathbb{R}^{n \times K}$,

$J = 1$,

$S_{\max}$... max. sparsity of vectors $\mathbf{x}_i$

Repeat until convergence (stopping rule):

Sparse coding

1: Solve using any pursuit algorithm

$\min_{\mathbf{x}_k}\{\|\mathbf{y}_k - \mathbf{D}\mathbf{x}_k\|_2^2\}$ s.t. $\|\mathbf{x}_k\|_0 \leq S_{\max}$

Dictionary update

For each atom $k = 1, 2, \ldots, K$ in $\mathbf{D}^{J-1}$ update by:

2: Set the group of indices using updated atom

$\omega_k = \{i | 1 \leq i \leq N, \mathbf{x}_T^k(i) \neq 0\}$

3: Compute the error matrix $\mathbf{E}_k$ by

$\mathbf{E}_k = \mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}_T^j$

4: Restrict $\mathbf{E}_k$ choosing only columns corresponding to ω_k to obtain $\mathbf{E}_k^R$

5: Apply SVD decomposition $\mathbf{E}_k^R = \mathbf{U}\Delta\mathbf{V}^T$

6: Update dictionary atom $\tilde{\mathbf{d}}_k$ (first column of $\mathbf{U}$)

7: Update coefficient vector $\mathbf{x}_R^k$ (first column of $\mathbf{V}$ multiplied by $\Delta(1,1)$)

8: Set $J = J + 1$

5 ALGORITHMS FOR SPARSE APPROXIMATIONS

Traditional methods of synthesis use transformation matrix Φ which is orthogonal (i.e. that the dimensions are $n \times n$) and forms the basis. The analysis coefficients are obtained by simply inverting the operator Φ. However, since the tasks that we deal with use overdetermined dictionaries that produce infinitely many solutions it is necessary to apply the algorithms that find optimal solutions in some way. Searching for sparse solutions is a problem of ℓ_0 norm minimization. Unfortunately, this norm is not a convex function and it is not possible to use any currently existing algorithm solving convex optimization. It constitutes an NP-hard problem and the exact solution can not be found in polynomial time [22].

5.1 Greedy algorithms

Greedy algorithms are selecting one (or more) of the most important atoms in each iteration. An important feature is that once the coefficient of the respective atom is found it does not change during the calculation of other coefficients nevermore. Algorithms like Matching Pursuit (MP) [66] or the currently popular modification called OMP [73] have relatively low complexity, unfortunately, achieving the global optimum is not guaranteed.

5.1.1 Orthogonal Matching Pursuit

The algorithm performs the following steps in each iteration: first calculates the correlation of all atoms of the dictionary with the current input signal segment. The highest coefficient which is found in this step is saved, backward synthesis of the coefficients and the dictionary is performed and the signal is then compared with the original input signal. The difference between these two signals is stored as a residue. The reconstruction error for the residual is computed as the square of ℓ_2 norm and every following iteration correlates atoms (except those already used) with the current residuum. Therefore, new coefficients are being obtained repeatedly. The reconstruction error decreases with increasing number of coefficients.

OMP algorithm iterates over and over again until one or more stopping criterion is fulfilled. The criterion to end the cycle can be a maximum sparsity i.e. that the algorithm stops after a predetermined number of iterations (the number of iterations = degree of sparsity = total number of coefficients). Another option is to set minimum error which the algorithm must achieve to terminate the run. In this case the number of iterations can not be estimated in advance. It is possible to also set

these two conditions together. This may be advantageous if the error condition is set too small that can not be achieved in a reasonable time, therefore, the cycle ends at the maximum number of iterations [33].

Detailed steps of OMP are described in algorithm 2.

Algorithm 2 Orthogonal matching pursuit

Input: $\mathbf{y}$... signal segment,
$\mathbf{D}$... dictionary,
s... level of sparsity,
ϵ... approximation error

Initialization: $i = 0$,
$\mathbf{c}$ = zero vector of length M
$\Omega_0 = \emptyset$... support set

Individual Steps:

Compute pseudo-inverse of dictionary $\mathbf{D}^+$
$\mathbf{r} = \mathbf{y}$
while $\|\mathbf{r}\|_2^2 > \epsilon$ and $i \leq s$ **do**
    $i = i + 1$
    Choose index j with maximal absolute value in $\mathbf{D}^+\mathbf{r}$
    Update support $\Omega_i = \Omega_{i-1} \cup j$
    Add the j-th entry of $\mathbf{D}^+_{\Omega_i} y$ to the j-th entry of $\mathbf{c}_k$
    $\mathbf{r} = \mathbf{y} - \mathbf{D}_{\Omega_i}\mathbf{c}_k$
end while

Output: $\mathbf{c}$... sparse coefficients approximating $\mathbf{y}$

5.2 Relaxation algorithms

The closest convex norm that can be utilized for approximation of sparse solutions is ℓ_1. In most cases results of minimization of the coefficients of ℓ_0 and ℓ_1 norm coincide. This group of algorithms assumes that under certain conditions we get to accurate or at least approximate solution. The algorithms are based on ℓ_1 relaxation.

5.2.1 Basis Pursuit

Basis Pursuit is an optimization problem which decomposes the signal into a superposition of atoms in an optimal way. Optimality is reached by having the smallest ℓ_1 norm of coefficients

$$\min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{subject to} \quad \mathbf{Dc} = \mathbf{y}, \tag{5.1}$$

among all considered decompositions where $\mathbf{c}$ is the coefficient vector, $\mathbf{D}$ is the dictionary and $\mathbf{y}$ is the resulting signal. Thanks to the non-differentiability of the ℓ_1 norm the decomposition can be much sparser than the older methods like Matching Pursuit or Method of Frames. Moreover, the Basis Pursuit is closely connected with linear programming and the optimization problem can be solved in almost linear time [23]. Pure Basis Pursuit algorithm finds an exact representation of the signal in a given domain. For the purpose of the Audio Inpainting some variation of the reconstructed signal is acceptable.

Another representative of the group of relaxation algorithms is a method called Focal Underdetermined System Solver (FOCUSS) [45].

5.2.2 Proximal algorithms

Proximal algorithms are methods from the optimization theory including relaxation tasks with ℓ_1 norm. Proximal algorithms are splitting the problem of sparse regression into separate problems as

$$\arg\min_{\mathbf{x}\in\mathbb{R}^N} \left(f_1(\mathbf{x}) + f_2(\mathbf{x})\right), \tag{5.2}$$

where $\mathbf{x}$ is the input (observed) signal, which are solved iteratively whereas the conditions of convergence of the algorithm are known. These algorithms are not very fast, however, the flexibility is advantageous [27].

Constrained version

The constrained form of the optimization problem (not corresponding to the task 5.2) is

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \qquad \text{subject to} \qquad \|\mathbf{D}\mathbf{x} - \mathbf{y}\|_2 \leq \delta, \tag{5.3}$$

where δ is an allowed error from the true solution. It can be transformed to the unconstrained form using the indicator function.

Let C be a nonempty subset $\mathbb{R}^n$. Then the indicator function of set C is defined as

$$\iota_C : \mathbf{x} \mapsto \begin{cases} 0 & \text{for } \mathbf{x} \in C, \\ \infty & \text{otherwise.} \end{cases} \tag{5.4}$$

Then, instead of Eq. 5.3 a new problem is solved

$$\arg\min_{\mathbf{x}} \ \|\mathbf{x}\|_1 + \iota_{\{\mathbf{x}:\,\mathbf{D}\mathbf{x}=\mathbf{y}\}}, \tag{5.5}$$

which is already in a requested form and the second addend enforces inclusion of the solution to the admissable set.

Unconstrained version

The task which is going to be solved is called the Least Absolute Shrinkage and Selection Operator (LASSO)[1]

$$\widehat{\mathbf{y}} = \mathbf{D} \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\min} \frac{1}{2} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 \tag{5.6}$$

where $\lambda \|\mathbf{x}\|_1$ is a regularization term which penalizes certain types of solutions and λ is a weighting coefficient controlling strength of the term. The higher the λ the more penalized the non-sparse solutions are. However, too high λ can cause inappropriate deviation from the data [88]. This parameter has to be chosen very cautiously. There is not any general rule how to choose the right value.

Analysis model

Solution of the problem when underdetermined system is utilized is the same for analysis and synthesis model. However, in an overcomplete case the solution is different. Analysis model is referred as a co-sparse analysis [69]. The point is to enforce sparsity of the analysis coefficients $\mathbf{A}\mathbf{y}$ instead of synthesis coefficients $\mathbf{c}$. The definition of solving and audio inpainting problem in an analysis sense is

$$\widehat{\mathbf{y}} = \underset{\mathbf{y}\in\mathbb{R}^N}{\arg\min} \|\mathbf{M}^{\mathrm{r}}\mathbf{x} - \mathbf{M}^{\mathrm{r}}\mathbf{y}\|_2^2 + \lambda \|\mathbf{A}\mathbf{y}\|_1 \tag{5.7}$$

where $\mathbf{x}$ is the observed signal, $\mathbf{y}$ is the unknown signal and $\mathbf{A}$ is an analysis operator. In an overcomplete case this approach is easier to solve compared to the synthesis model. In the ℓ_1 overcomplete case for every analysis operator $\mathbf{A}$ of a full matrix rank exists the equivalent dictionary $\mathbf{D}$. This statement does not work the other way round [34].

Synthesis model

The definition of solving and audio inpainting problem in an synthesis sense is

$$\widehat{\mathbf{y}}^{\mathrm{m}} = \mathbf{M}^{\mathrm{m}}\mathbf{D} \cdot \underset{\mathbf{c}\in\mathbb{R}^N}{\arg\min} \left(\frac{1}{2} \|\mathbf{M}^{\mathrm{r}}\mathbf{D}\mathbf{c} - \mathbf{M}^{\mathrm{r}}\mathbf{x}\|_2^2 + \lambda \|\mathbf{c}\|_1 \right). \tag{5.8}$$

If both of the functions f_1 and f_2 from Eq. 5.2 are non-smooth, we are not able to compute the gradient (first derivative). Feasible method to solve the problem under these circumstances is *Douglas-Rachford* algorithm [26].

On the other hand, if at least one of the functions f_1, f_2 is smooth (eg. ℓ_2 norm), the gradient for this function is defined and we can utilize the $Forward-Backward$ algorithm to solve the sparse regression problem.

[1] Shrinkage operator is a method that makes decision about keeping or discarding the coefficient.

Minimization of the function is performed using the *proximity operator* which minimizes the function without getting too far from the initialization point. The proximity operator is a generalization of the projection [28].

Some of methods known from other areas of signal processing which can be formulated as proximal are (F)ISTA (Fast Iterative Schrinkage/Thresholding Algorithm) [19] or Alternating-Direction Method of Multipliers (ADMM) [20].

5.2.3 Proximity operator

Proximity operator of a convex function is an extension of the projection operator on a convex set which searches for a closest point to the initialization point in a convex set. For every $\mathbf{x} \in \mathbb{R}^n$ the minimization problem

$$\arg\min_{\mathbf{y}\in\mathbb{R}^N} \frac{1}{2}\left\|\mathbf{x}-\mathbf{y}\right\|_2^2 + f(\mathbf{y}), \tag{5.9}$$

admits a unique solution denoted by $\mathrm{prox}_f\,\mathbf{x}$ and this is the proximity operator of function f. Proximity operators are well suited for iterative minimization algorithms and from the signal processing point of view they have a natural interpretation in terms of denoising. Wide list of proximity operators is defined in [27].

Audio Inpainting via Proximal Splitting

The synthesis model for solving an Audio Inpainting problem defined in 5.8 is solved by proximal splitting. The general inpainting problem is then reformulated as

$$\widehat{\mathbf{y}}^{\mathrm{m}} = \mathbf{M}^{\mathrm{m}}\mathbf{D}\cdot\arg\min_{\mathbf{x}\in\mathbb{C}^N}\left(\underbrace{\frac{1}{2}\left\|\mathbf{M}^{\mathrm{r}}\mathbf{D}\mathbf{c}-\mathbf{M}^{\mathrm{r}}\mathbf{x}\right\|_2^2}_{f_2} + \underbrace{\lambda\left\|\mathbf{c}\right\|_p}_{f_1}\right). \tag{5.10}$$

Gradient of f_2 is defined as

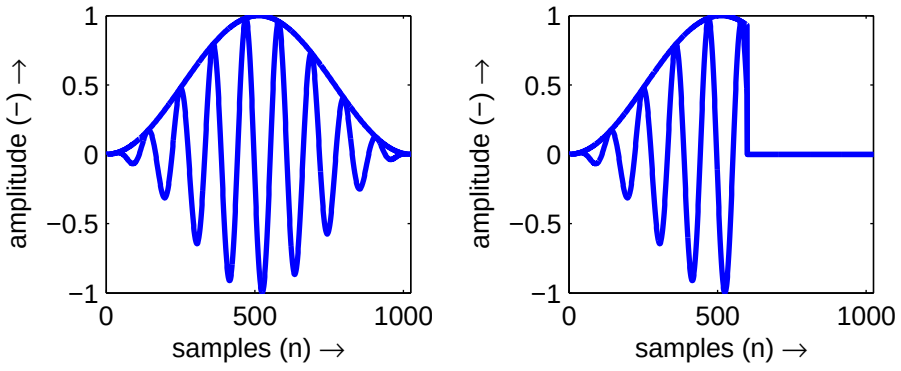
$$\frac{\mathrm{d}f_2(\mathbf{c})}{\mathrm{d}\mathbf{c}} = \frac{\mathrm{d}\frac{1}{2}\left\|\mathbf{D}^{\mathrm{r}}\mathbf{c}-\mathbf{x}^{\mathrm{r}}\right\|_2^2}{\mathrm{d}\mathbf{c}} = (\mathbf{D}^{\mathrm{r}})^{\intercal}(\mathbf{D}^{\mathrm{r}}\mathbf{c}-\mathbf{x}^{\mathrm{r}}). \tag{5.11}$$

For the sake of brevity matrix multiplication by the binary mask $\mathbf{M}^{\mathrm{r}}\mathbf{D}$ is simplified as $\mathbf{D}^{\mathrm{r}}$.

5.2.4 Weighting of atoms

Signal atoms which are affected by the signal gap do not fulfill the condition of $\left\|d_j\right\|_2 = 1$. An illustration of such situation is in Fig. 5.1. As reffered in [12], weights are utilized as

$$\widetilde{\mathbf{D}} = \mathbf{M}^{\mathrm{r}}\mathbf{D}\mathbf{W}, \tag{5.12}$$



(a) (b)

Fig. 5.1: Window not affected(a) and affected (b) by gap.

where $\mathbf{W}_{ij} = 0$ for $i \neq j$ and $\mathbf{W}_{ij} = \|\mathbf{M}^{\mathrm{r}}\mathbf{d}_j\|_2^{-1}$. Using the computational tools described in Sec. 7.6.2 the computation of weights was reformulated as

$$\mathbf{w}_i = \frac{\|\mathbf{g}\|_2}{\sqrt{\|\mathbf{g}\|_2^2 - |\mathbf{m}^{\mathrm{m}} * \mathbf{g}^2|_i}}, \tag{5.13}$$

where $\mathbf{m}^{\mathrm{m}}$ is a missing mask of a corresponding atom of values $\{0, 1\}$. The value 0 represents a reliable sample, 1 represents a missing sample. The reason of such redefinition of weighting is that the representation of frames in software toolbox was not able to handle the weighting in a matrix form such as Eq. 5.12. Therefore, the computation applied on each coefficient c individually has to be performed.

5.3 Structured sparsity

Previous methods of ℓ_1-relaxation treated the coefficients independently regarding no correspondence with the neighbourhood. Nevertheless, every typical spectrogram of a musical signal is naturally structured. Considering this fact, the algorithm for sparse signal modelling incorporating information about a structure (evaluation of the coefficient on the strength of its neighbourhood) in an analysis stage of processing would be an advantage compared to the regular sparse modelling where coefficients are treated independently. While ℓ_1-norm performs individually on each coefficient, mixed norms (described in 3.3) can substitute this norm to perform independently on a group of coefficients [60]. Keeping or discarding particular coefficient under consideration is decided up to certain neighbourhood of the coefficient. Further improvement called the *Social Sparsity* means weighting of the coefficients in the neighbourhood [58].

The neighbourhood should be chosen according to the specific signal class under investigation, e.g. focused on tonal/transient part. According to this, structured

shrinkage operators representing a neighbourhood system have to be defined. The convex optimization problem for Audio Inpainting with mixed norms is reformulated as

$$\widehat{\mathbf{y}}^{\mathrm{m}} = \mathbf{M}^{\mathrm{m}}\mathbf{D} \cdot \arg\min_{\mathbf{c}\in\mathbb{C}^N} \left(\frac{1}{2} \|\mathbf{M}^{\mathrm{r}}\mathbf{y} - \mathbf{M}^{\mathrm{r}}\mathbf{D}\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_{p,q} \right) \tag{5.14}$$

where p represents a within-group penalty and q is an across-group penalty.

Due to the non-stationarity of sound signals, windowing with overlapping and weighting is incorporated. Generally, there are two set of mixed norms which are widely used in literature:

1. *Windowed-Group-Lasso*: Keeping a coefficient if the energy of its neighbourhood is large enough (positive correlation). $p = 2$, $q = 1$,
2. *Persistent-Elitist-Lasso*: Coefficient will be kept if its neighbourhood is energetic enough compared to the others. $p = 1$, $q = 2$ and one more index for sophisticated treatment are utilized.

Moreover, each of these two kinds of mixed norms have to satisfy an individual shrinkage operator. The shrinkage performs independently on a group of coefficients and evaluation of the coefficients relative to the strength of its neighbourhood sounds more natural. Regarding Eq. 3.5, the Group-LASSO shrinkage operator for all g, m is

$$\hat{x}_{g,m} = \overline{y}_{g,m} \left(1 - \frac{\lambda}{\left\| \overline{\mathbf{y}}_g \right\|_2} \right)^+ . \tag{5.15}$$

In this case, the whole 1D group of the coefficients is either kept or discarded and all the coefficients are thresholded by the same threshold. Pure Group-LASSO is useful together with a-priori knowledge of the structure of (sub)groups.

While this preposition is not satisfied in most cases, incorporation of the simultaneously active neighbourhood of the coefficient is advantageous. This is called the Windowed Group-LASSO (WGL) and the shrinkage operator is specified as

$$\hat{x}_{g,m} = \overline{y}_{g,m} \left(1 - \left[\frac{\lambda}{\|\overline{\mathbf{y}}_k\|_{\ell_2(\omega_{(k)})}} \right]^{\alpha} \right)^+ , \tag{5.16}$$

where $\omega_{(k)}$ specifies close indices around each coefficient of index $k = (g, m)$ and $\alpha = 1$ for classic WGL case. If exponent $\alpha = 2$, it encompasses the *empirical Wiener* introduced as an advantageous feature for audio denoising in [84] and [85].

On the other hand, different soft-thresholding is applied by Elitist-LASSO on particular coefficient where most of the group members are thresholded and only a few (at least one) coefficients remain. This operator enforces sparsity across groups. Obtaining of the coefficients $\overline{\mathbf{y}}$ and definition of $M_g(\lambda)$ are performed before each

shrinkage. The shrinkage operation is defined as

$$\hat{x}_{g,m} = \begin{cases} \operatorname{sgn}(\overline{y}_{g,m})(\left|\overline{y}_{g,m}\right| - \tau_g) & \left|\overline{y}_{g,m}\right| \geq \tau_g \\ 0 & \text{otherwise} \end{cases} \tag{5.17}$$

where the group independent threshold is

$$\tau_g = \frac{\lambda}{1 + \lambda M_g(\lambda)} \left\|\left| \overline{\mathbf{y}}_g \right|\right\|. \tag{5.18}$$

$\left\|\left| \overline{\mathbf{y}}_g \right|\right\|$ is an expression of a sum of M_g highest values of the vector $\overline{\mathbf{y}}_g$. For details see [59].

An equivalent of WGL was mirrored into Elitist-LASSO to promote persistence in the retained coefficients. Consider $\omega(g)$ a family of neighbours associated to group index g then the shrinkage operator is reformulated as

$$\tau_g{}' = \frac{\lambda}{1 + \lambda \left|\omega(g)\right|} \left\| \overline{\mathbf{y}}_{\omega(g)} \right\|_1 \tag{5.19}$$

and is called the Persistent Elitist-LASSO (PEL) [58].

Current applications of the structured sparsity are focused on denoising [87] or audio declipping [86]. Probably the most comprehensive work about structured sparsity was digested by dr. Kereliuk in his Ph.D. thesis [54].

5.4 Hybrid algorithms

Combining various algorithms can result in so-called *hybrid algorithm*, such as A*OMP, which utilizes an A* algorithm for information tree searching [53]. Another option may be based on thresholding algorithms [33].

Comparing various algorithms was carried out in several publications, eg. [48] or [46].

5.5 Chapter summary

This chapter described the algorithms which perform the sparse approximation. There are two main groups of algorithms: greedy and relaxation. In the experimental part of the thesis, each group of approximation algorithms will be represented by at least one delegate. Important properties of the algorithm, consequent computational efficiency and restoration quality will be examined and evaluated since these are the main points of this thesis. Theory described in this chapter will be referenced from the following sections.

6 GOALS OF THE THESIS

As mentioned in the introduction of the thesis, the main aim is a comprehensive study of both state-of-the-art interpolation methods and novel audio inpainting algorithms.

- The contribution for each method under investigation is focused on:
  - experimental research of parameters that most influence the reconstruction efficiency,
  - finding optimal values of the most important parameters.
- Current audio inpainting methods will be extended by methods based on the relaxation algorithms.
- All of the implemented methods will be compared with each other in terms of
  - restoration efficiency,
  - processing time.
- The analysis model of the LASSO problem will be experimentally proven for the audio inpainting, since there is no reference in the previous contributions.
- Methods will be compared in terms of the objective criterion SNR and the best results will be compared using the PEMO-Q algorithm.

7 EVALUATION OF AUDIO RESTORATION

7.1 Objective evaluation

7.1.1 Signal-to-Noise Ratio

A most common objective evaluation method of signal restoration is SNR. However, there are legitimate doubts about correctness of using this method. The term SNR will be used as an evaluation criterion, however, here the meaning is slightly different. The objection could be caused due to name of the evaluation criterion. In fact, there is no additional noise to be suppressed and compared to the original signal. The point of these experiments is to compare the original signal (which is known in our experiments) in the gap with the reconstructed (interpolated) samples of the same length. The SNR is defined as

$$\mathrm{SNR}(\mathbf{y}, \widehat{\mathbf{y}}) = 10 \log \frac{\|\mathbf{y}(\mathbf{I}^{\mathrm{m}})\|_2^2}{\|\mathbf{y}(\mathbf{I}^{\mathrm{m}}) - \widehat{\mathbf{y}}(\mathbf{I}^{\mathrm{m}})\|_2^2}, \tag{7.1}$$

where $\mathbf{y}$ is the original signal, $\widehat{\mathbf{y}}$ is the reconstructed signal and $\mathbf{I}^{\mathrm{m}}$ is the vector of indices of the corrupted signal.

7.1.2 PEMO-Q

PEMO-Q[1] evaluates the audio quality of a given distorted signal in a relation to the corresponding high-quality reference signal. The auditory model is employed to compute so-called internal representations adjusted to the cognitive aspects. These internal representations are projected on a Perceptual Similarity Measure (PSM), which is influenced by the signal type.

While PSM represents an overall perceived audio quality, the instantaneous objective audio quality PSMt is computed by successive cross correlation of 10 ms segments (in the original paper [49] called *frames*). Afterwards, PSMt is weighted by the moving average of the internal representation of the test signal.

Resulting evaluation is restricted to the interval $[-1, 1]$, where 1 denotes identity to the original signal and the smaller values the larger the deviations of the observed signal from the reference signal is.

On the producer's website[2] there is a demo version of the full software available with the restriction on the maximum length of evaluated file of 4 seconds.

[1] The acronym "PEMO" means the naming of the auditory quality model and "Q" means a quality assessment.

[2] Homepage of PEMO-Q demo version 1.3: `http://www.hoertech.de/cgi-bin/wPermission.cgi?file=/web_en/produkte/downloads.shtml`

The evaluation algorithm itself starts with pre-processing. In this stage, three steps are performed consisting time alignment, level alignment and deleting silent signal intervals. Next step is the auditory processing where the signal is split up into 35 critical bands whose center frequencies are equally spaced on an ERB scale with one filter per ERB. The adaptation stage contrasts signal amplitude fluctuations. Stationary parts are compressed and rapid changes are emphasized. Finally, the signal envelope is analyzed by a linear modulation bank of filters.

In post-processing stage the auditory perception model is applied such that the subjective evaluation of the perceived (dis)similarity of the two audio signals is predicted. For each channel, the linear cross correlation coefficient of internal representation of the reference and the test signal is calculated. For further details see [49].

7.2 Experimental results description

This chapter is focused on practical experiments of several interpolation/inpainting methods. In most cases the comparison and evaluation is done by the Signal-to-Noise Ratio.

All of the experiments have got following common input values:

- Input signal samples,
- index of the first missing sample,
- length (number of samples) of the signal gap.

Since this thesis deals with original audio samples without corruptions, the first step of the processing in the toolbox is an artificial creation of the gap. This is accomplished by setting the relevant samples to zero.

The second step before the interpolation itself is to trim the signal only to the necessary length around the gap. Processing of the whole input audio signal from the beginning to the end takes a long time (e.g. sinusoidal modeling or sparse representation). Therefore, each approach to audio interpolation uses individual method to trim the input signal according the the transformation/modeling parameters. These methods will be described for each interpolation algorithm separately.

7.2.1 Inpainting toolbox

An inpainting toolbox including all of the presented methods was developed and all the following experiments were performed using this MATLAB toolbox. The toolbox is called the *Brno-Wien Inpainting Toolbox* and is available at the enclosed CD of this thesis. The maintainer of the toolbox development was the author of this

thesis and other collaborators were people cooperating on the joint project of Brno University of Technology, University of Vienna and Austrian Academy of Sciences.

Inner computations of inpainting algorithms in the toolbox are dependent on some other toolboxes. Inpainting based on sinusoidal modeling is based on the McAulay-Quartieri method implemented in code called the *Sinewave and Sinusoid+Noise Analysis/Synthesis in Matlab*[3] [36].

Experiments of inpainting by greedy algorithms for sparse signal representations, particularly OMP are performed by our own implementation of the Orthogonal Matching Pursuit. Greedy algorithms are also utilized for dictionary learning methods (K-SVD) whose core algorithms are provided by OMP-Box v10 and KSVD-Box v13 [4].

Relaxation algorithms for solving the underdetermined systems and convex optimization are provided by the *UNLocBoX* v. 1.6.3[5]. Frames representations, spectrograms plotting and some other miscellaneous functions are provided by the *LTFAT* (The Large Time-Frequency Analysis Toolbox) v. 2.1.1[6].

7.2.2 Audio examples

Example audio files are obtained from the SMALLbox [7], a framework for processing signals using adaptive sparse structured representations. This toolbox was the first who provided audio inpainting examples based on the sparse representations. The list of audio files is in the Tab. 7.1 and the choice represents a fundamental set of different and most frequent kinds of audio files.

All of the files are single channel (mono) with the sampling frequency of 16 kHz and 16 bit depth with the time duration of 4 seconds.

Following inpainting/interpolation experiments are mostly batch tests of multiple parameters, however, the performance of each method is illustrated by time plot of a single experiment. The position of the gap and its length is always the same: music file *music11_16kHz.wav*, gap starts at 20000$^{\text{th}}$ sample and lasts 320 samples (20 ms).

[3] `http://www.ee.columbia.edu/~dpwe/resources/matlab/sinemodel/`
[4] `http://www.cs.technion.ac.il/~ronrubin/software.html`
[5] `https://lts2.epfl.ch/unlocbox/download/index.php`
[6] `http://ltfat.sourceforge.net/`
[7] `http://www.small-project.eu/software-data/smallbox/`

Tab. 7.1: List of experimental audio samples.

Filename	Character
music02_16kHz	harmonic, double bass
music03_16kHz	harmonic, guitar
music04_16kHz	harmonic, woman singing
music07_16kHz	non-harmonic, drums
music08_16kHz	harmonic, pop music
music09_16kHz	speech, rap
music10_16kHz	harmonic, orchestra
music11_16kHz	harmonic, guitar
music12_16kHz	speech, DJ show

7.3 Interpolation methods

7.3.1 Samples repetition

Method from Sec. 2.1 called *The Weighted Repetitive Substitution* was implemented and evaluated as the oldest comparable method for audio signal interpolation. As expected, the algorithm is very dependent on the perfect estimation of the signal period. There is only one input parameter q_{u} specifying the area surrounding the gap supposed to be the model for signal estimation. Better results would be expected using separate neighbourhood length for left and right side q_{L} and q_{R}, however, additional algorithm for signal period detection should be incorporated. Since the period detection is out of the scope of this thesis, period detection is not implemented. Moreover, high-quality results are not expected.

Regarding the example gap position and length, the best SNR = 1.28 dB was reached for $q_{\mathrm{u}} = 260$ samples. See Fig. 7.1 for results of the first set of experiments. Observing the neighbourhood of the gap the period of the signal was measured (by hand peak-to-peak distance measurement) with resulting value of 192 samples. However, the SNR measured using $q_{\mathrm{u}} = 192$ equals a worse result of -1.94 dB. In both cases of q_{u} the subjective listening of the result is not satisfying and the listener can clearly recognise the gap position in the sound. A time plot of the results is in Fig. 7.2 where the green dashed line indicates borders of the signal gap.

The batch experiment of samples repetition method was performed with following parameters. Gap size was selected from the range of $\{10, 20, \ldots, 100\}$ ms which corresponds to $\{160, 320, \ldots, 1600\}$ samples. Model order (neighbourhood size in other words) was selected as a ratio of the gap size figured in % in the range of

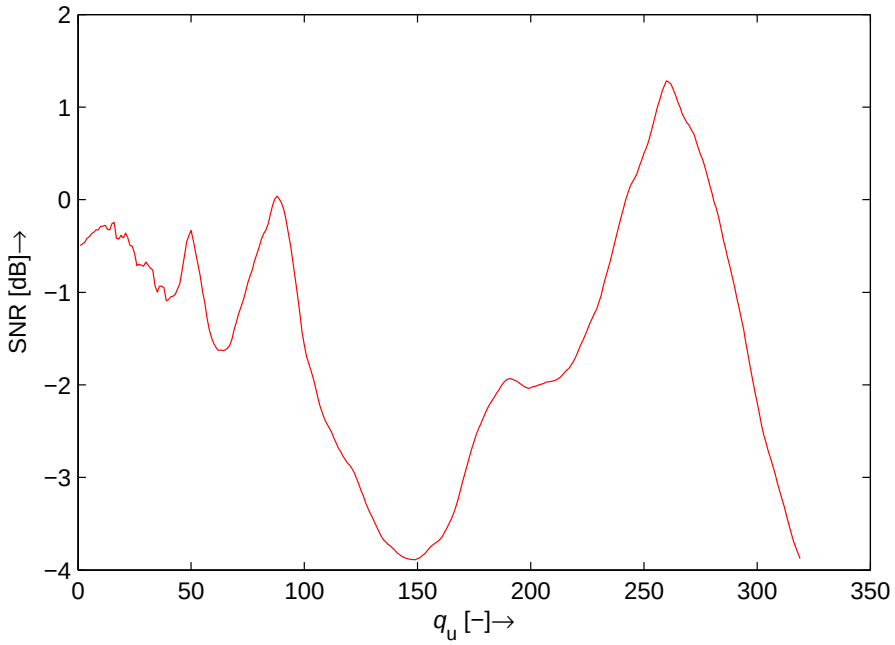


Fig. 7.1: SNR of interpolation by samples repetition for various neighbourhood length.

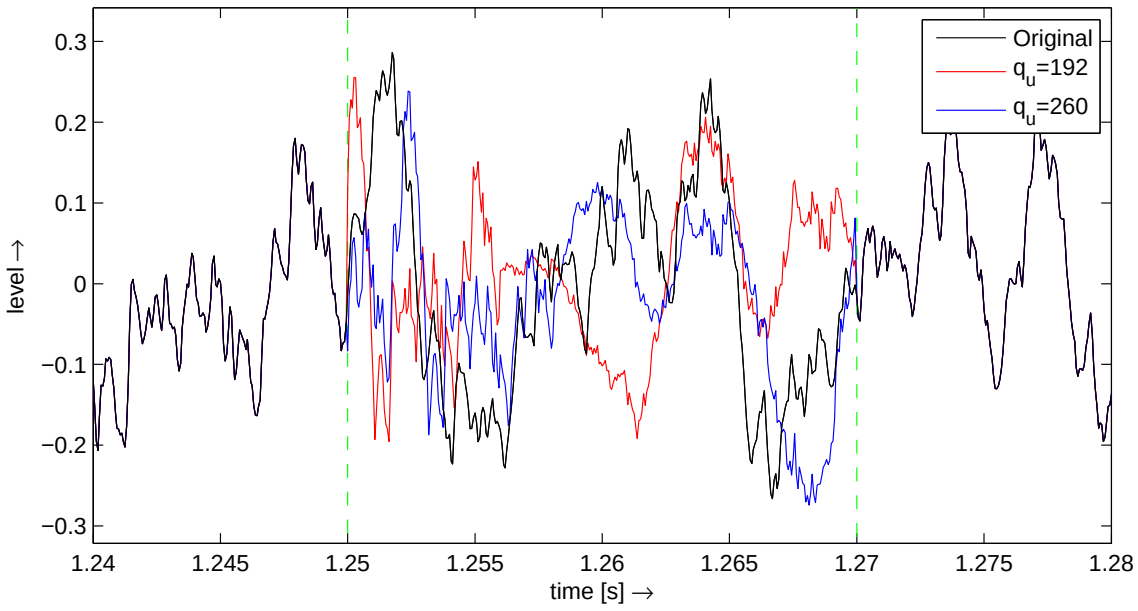


Fig. 7.2: Interpolation by the samples repetition algorithm with various neighbourhood size q_u.

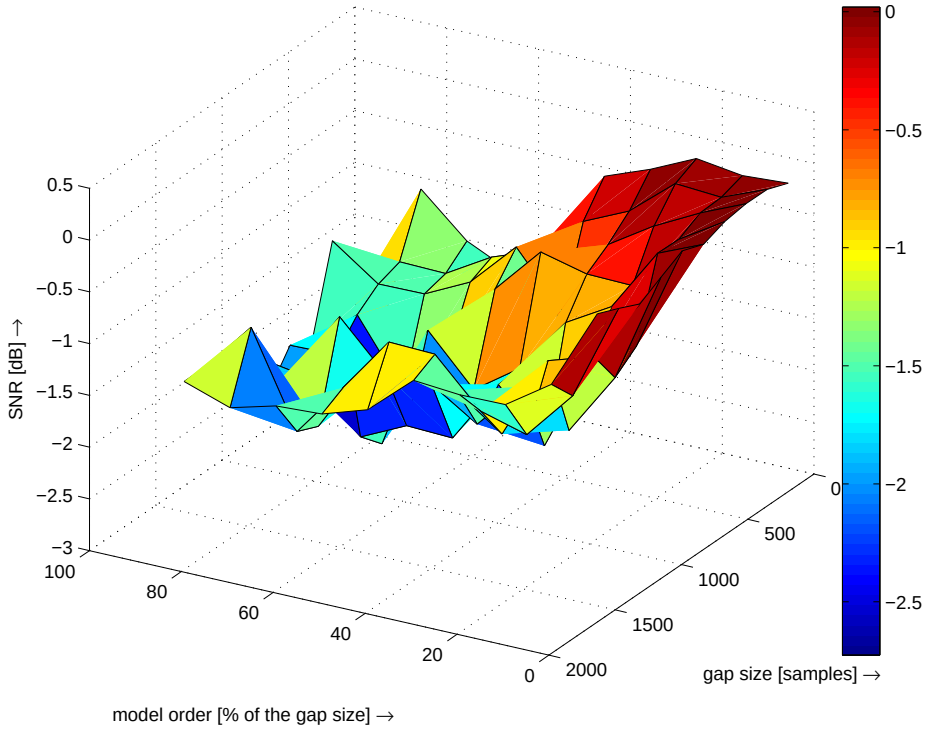


Fig. 7.3: SNR of interpolation by the samples repetition algorithm with various model order and gap length.

$\{1, 10, 20, \ldots, 90\}$ %. Experiment for each combination of model order and gap size was performed ten times on different gap positions. The figured results are mean values of these ten experiments.

In Fig. 7.3 are illustrated results of interpolation in the terms of SNR. Quite surprising is that the best results are reached for the shortest model order with the best value of SNR $= (0.020 \pm 0.004)$ dB for gap size of 30 ms and model order of 1 % of the gap size. All the average SNR results are in the range of $\langle -2.735, 0.020 \rangle$ dB which shows unsatisfying results overall.

A naturally arising question is whether the this kind of interpolation method is really useful. Listening to other interpolation experiments of another music files using this algorithm provides quite similar results and often the corrupted signal with values set to zero in the gap sounds more naturally even with signal degradation by the gap.

Regarding the speed of computations, this method is one of the fastest with the time duration of less than a second. The speed is definitely shorter than the gap length, therefore the algorithm is feasible also for real-time applications. In Fig. 7.4 is the visualisation of the average processing time which grows linearly with increasing gap size and is not dependent on the model order.

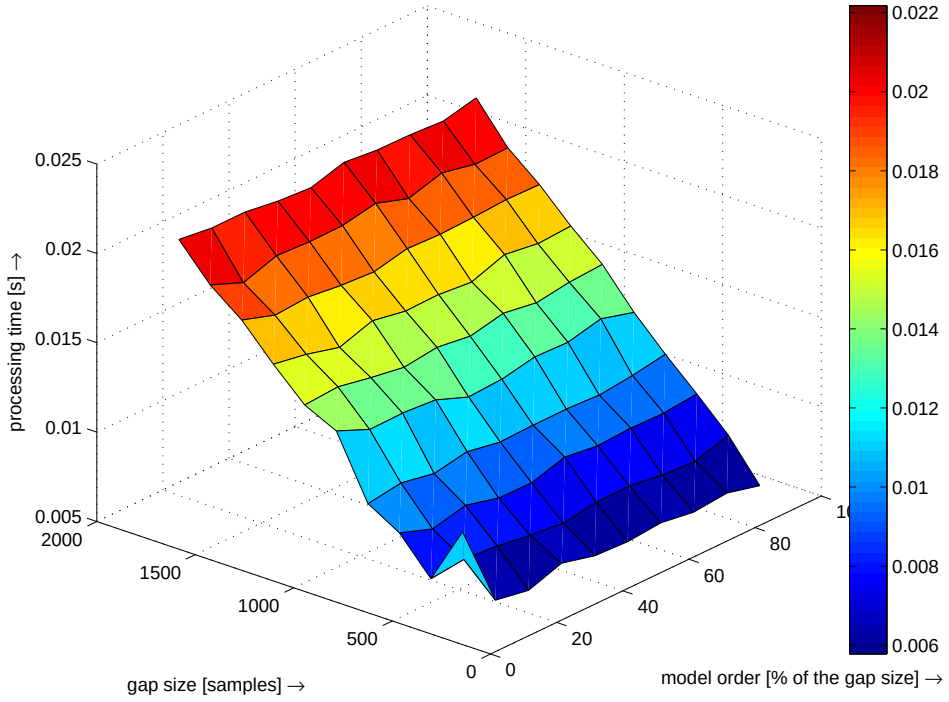


Fig. 7.4: Processing time of interpolation by the samples repetition algorithm with various model order and gap length.

7.3.2 AR modeling of signal samples

Interpolation methods based on AR modeling of signal samples are going to be examined in this section. Both of them are theoretically described in Sec. 2.2.2. First method, the *Least-Squares Residual Predictor* consists of four core steps. AR coefficients are computed by Yule-Walker method because of the simplicity and speed. In Fig. 7.5 there is a comparison of different AR model order selection. The process of interpolation in terms of the resulting SNR was performed on the example file and gap position with AR model order number of 1 up to the length of the gap minus 1.

From the figure it is obvious that the best SNR = 0.67 dB is reached with the maximum AR model order. In this case it is the value of 319 samples.

However, there are two important peaks of the SNR plot before the maximum. The first local peak is around the order of 195 which is close to the signal period around the gap. The second local maximum is around the value of 270. Both of them are rather close to important values of the neighbourhood size observed in the previous experiment using samples repetition method. To remind the important values, 195 is close to the signal period which is 192 and 270 is close to the neighbourhood size with maximum SNR value of the previous experiment.

Second method dealing with autoregressive modeling of signal samples is ca-

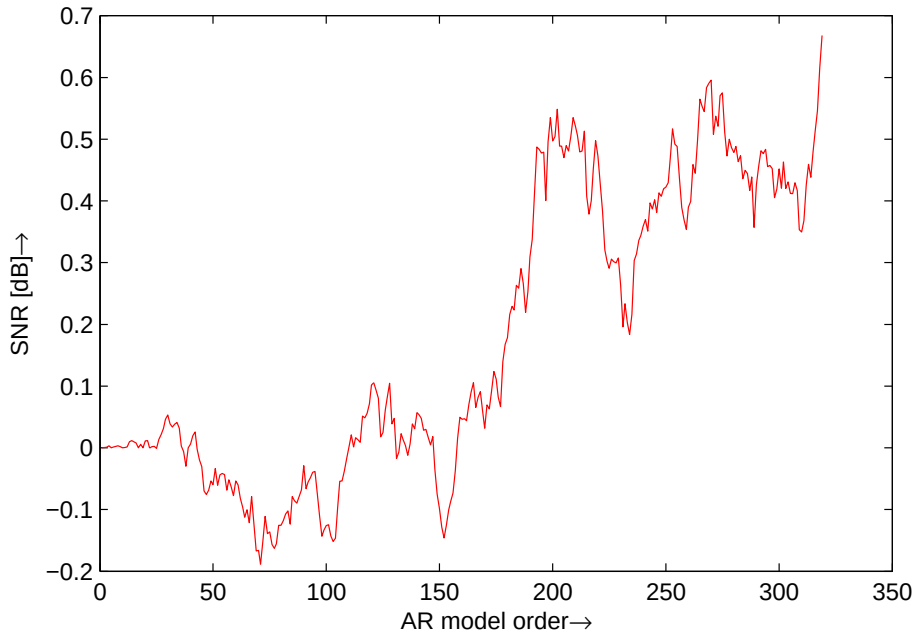


Fig. 7.5: SNR of Various order of Least-Squares Residual Predictor interpolation.

lled the *Weighted Forward-Backward Predictor*. The same experiment with example signal gap position and length was performed and the result is presented in Fig. 7.6. Here, the maximum SNR = 0.84 dB was reached at AR model order of 270. The model order of 270 is quite close to the important values of neighbourhood size/model order from previous experiments close to 260. Likewise, another peak in Fig. 7.6 at model order of 193 is similar with the period of the signal equal to 192 samples.

Results of batch testing of the first method, the *Least-Squares Residual Predictor* is illustrated in Fig. 7.7. The experiment was performed for gap size of $\{10, 20, \ldots, 100\}$ ms which corresponds to $\{160, 320, \ldots, 1600\}$ samples. Since the previous experiments for model order smaller than the gap length resulted in poor SNR the AR model order was selected from the range of $\{0.5, 1, 2, \ldots, 5, 6, 8, \ldots, 12\}$ times of the gap size. Every combination of gap size and AR model order was used ten times for different gap positions in music file *music11_16kHz.wav* and the resulting SNR was computed as an average of these experiments. There is a large area of SNR = 0 dB which identifies experiments that were not performed at all because of the very long processing time (> 1000 s for single interpolation experiment). The best SNR = 5.78 dB was reached for gap size of 360 samples and AR model order of 10 times the gap length. However, the variance of 14 dB makes the results very unstable. Therefore, there is no general recommendation for interpolation parameters selection.

Furthermore, the same experiment was made with the *Weighted Forward-Back-*

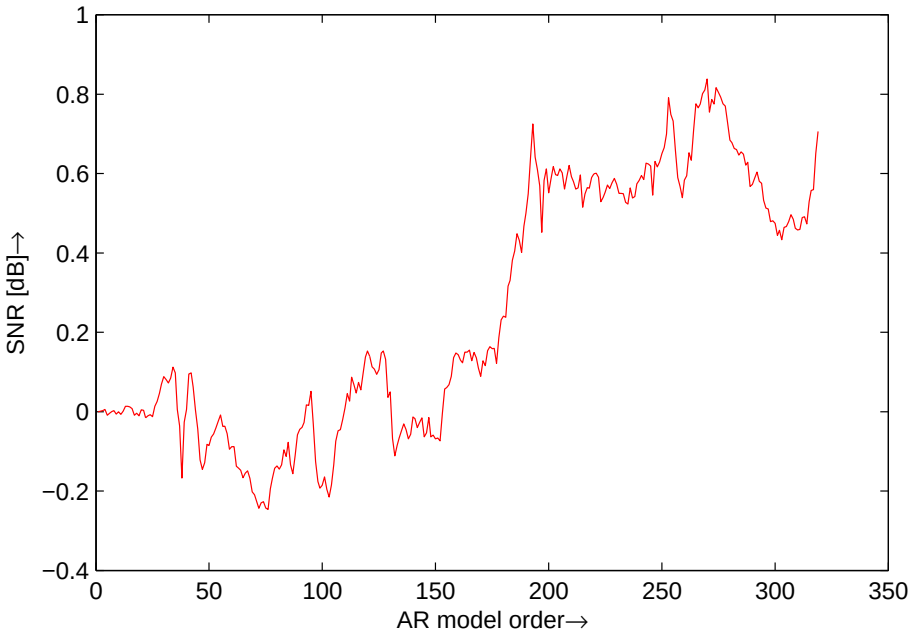


Fig. 7.6: SNR of Various order of Weighted Forward-Backward Predictor interpolation.

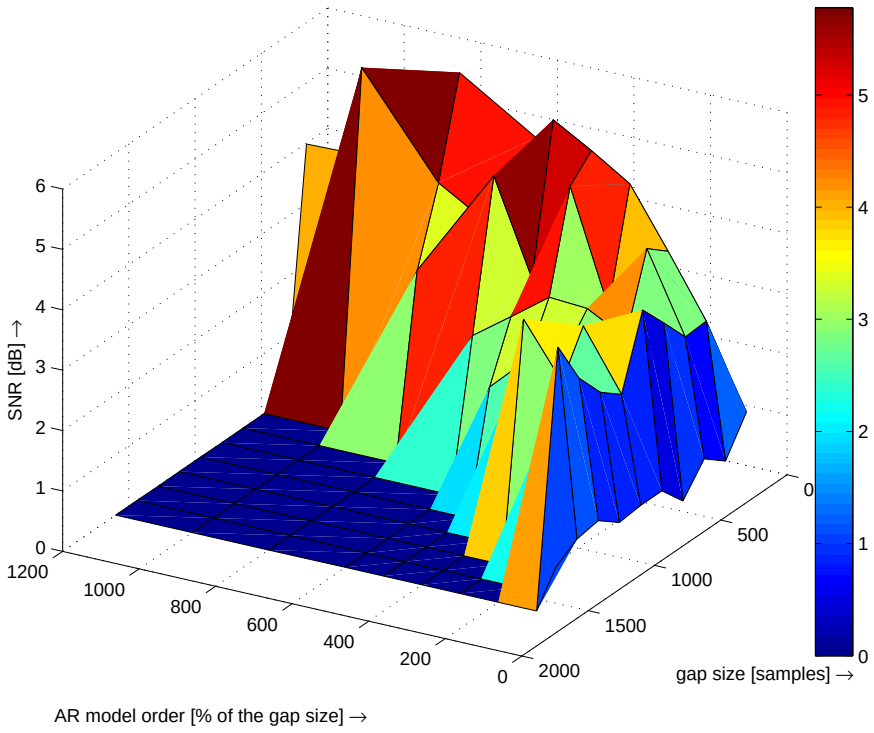


Fig. 7.7: SNR of interpolation by AR modeling of the samples (LSRI) with various model order and gap length.

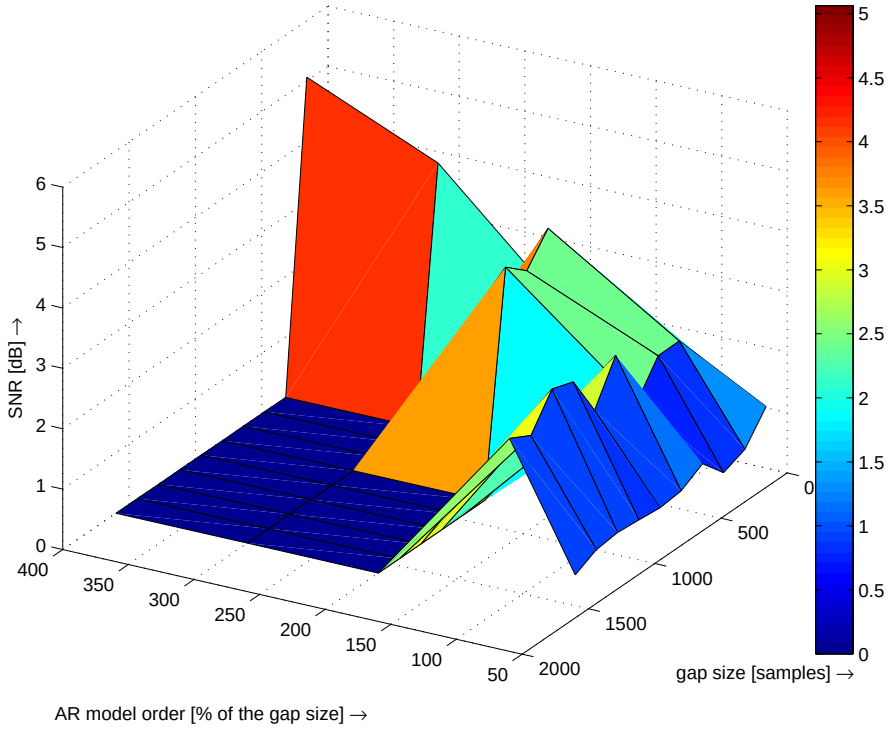


Fig. 7.8: SNR of interpolation by AR modeling of the samples (WFBI) with various model order and gap length.

ward Predictor. Since the computational load of this method is higher than the previous method, batch experiment was performed with much more restricted range of parameters. The AR model order was selected from the range of $\{0.5, 1, 2, 3, 4\}$ times the gap size. In Fig. 7.8 results of the interpolation are illustrated. Again, the plotted values are an average of ten experiments and values equal to zero mean that the experiment was not performed because of long computational time.

The best result (SNR = 5.06 dB) was reached for gap size of 160 samples and model order of 400 samples. Compared to LSRI method, the variance of the results was much lower (4.65 dB). However, the computational time of larger model orders makes this method unusable in real experiments.

The example gap is interpolated using the two methods in the test. AR model order of WFBI algorithm was set to three times the length of the gap. Same parameter for LSRI was set to nine times of the gap length. Both of the parameter values were extracted from the batch experiments results as the best parameters for gap length of 20 ms.

In Fig 7.9 there is the time plot of interpolation results of the two methods. There is an important decrease of signal energy near the borders of the gap, however, the energy of the reconstructed signal is growing while getting closer to the middle of the gap. The same phenomenon, a little decrease of energy, can be observed during

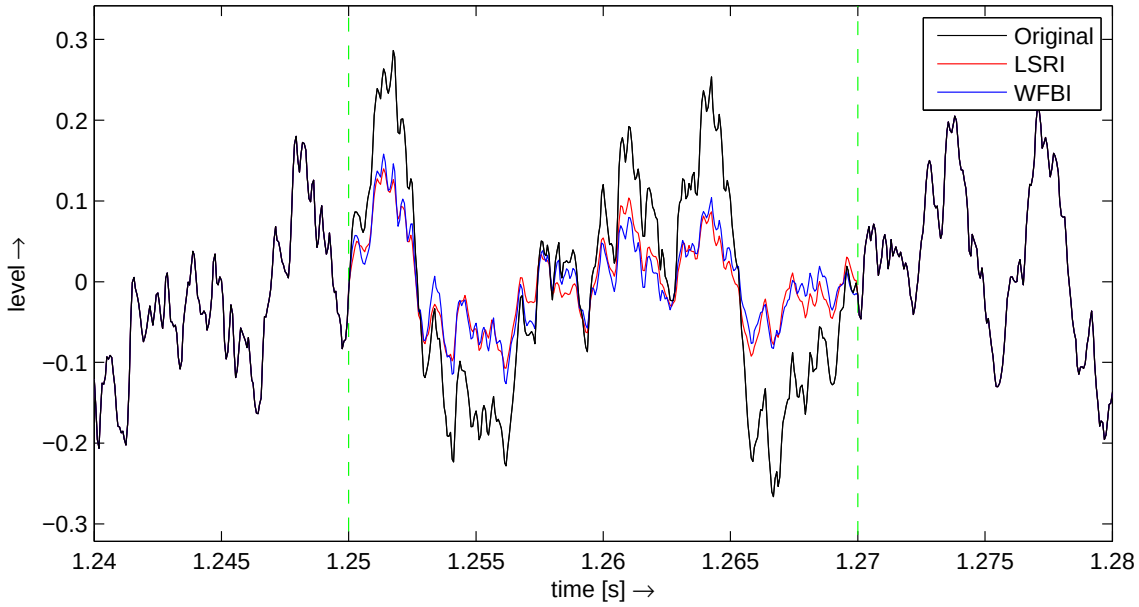


Fig. 7.9: Comparison of interpolation by the two algorithms for AR signal modeling.

the listening of the reconstruction (on the attached CD).

7.3.3 Sinusoidal modeling

This method expands the reliable signal around the gap into separate sinusoidal components called the *partials* and a residual noise. Each sinusoidal component is described by instantaneous frequency, amplitude and phase, which are modeled as an AR process and utilized for interpolation of samples in the gap. For determination of coefficients of the AR model the Burg method is used as described in Sec. 2.2.2.

Previous methods needed only one parameter to be set up: the model order or neighbourhood size. Here, four parameters influence the power of the algorithm:

- Frequency difference threshold,
- amplitude difference threshold,
- length of the vector for amplitude mean value computation,
- order of AR model.

For purpose of the experimental example gap position and length interpolation, these four parameters were examined for optimal values.

Frequency difference threshold is the maximum distance of two frequencies from the left and right side of the gap to be considered as matching as in Eq. 2.34. The higher the frequency threshold, the more likely interpolation of not matching partials is performed. On the other hand, if the signal is modulated more (vibrato) it is convenient to set the threshold higher. Examining the value of the frequency

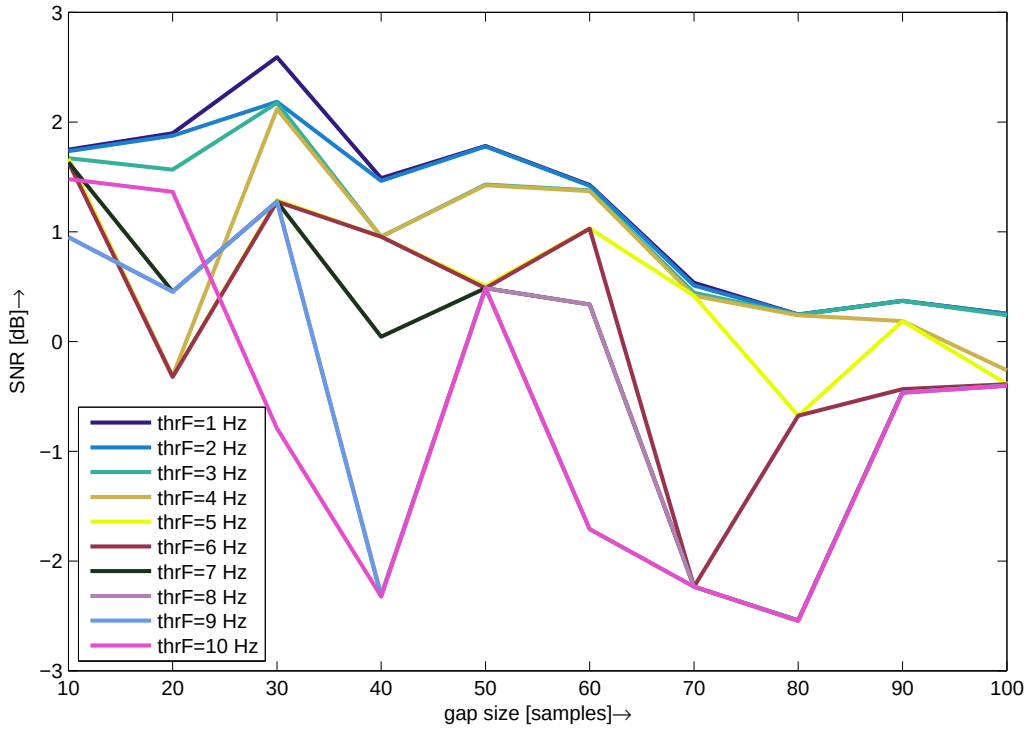


Fig. 7.10: Interpolation results of various gap length and frequency threshold of sinusoidal modeling.

threshold over the range from 1 Hz to 10 Hz in music file `music11_16kHz.wav` with gap start at sample no. 20000 and gap length range from 10 ms to 100 ms there is an obvious decay of SNR with increasing frequency threshold value (see Fig. 7.10). As a consequence frequency threshold in the following experiments will be set to value of 3 while this value should keep SNR higher and bring a little variability in the matching decision.

Amplitude difference threshold is not very varied like the frequency threshold, therefore the value should be set to a lower level. Moreover, to connect two partials from the left and right side of the gap both of the threshold limits (amplitude and frequency) must not be exceeded. Setting of the amplitude threshold to a higher value could raise the chance of connection of the non-corresponding partials. Assessing an experiment of interpolation with various amplitude threshold, the resulting SNR values were exactly the same for all amplitude thresholds from the range of $\langle 0.2; 3.6 \rangle$. Therefore, the following experiments will be performed with amplitude difference threshold set to $\mathrm{thrA} = 0.5$.

The third parameter is length of the vector for amplitude mean value computation (M). Regarding Eq. 2.37 and 2.38 the purpose of this parameter is to get an appropriate energy estimation of the partial on the opposite site of the gap. Batch testing for the optimal settings are figured in Fig. 7.11 shows that there are very

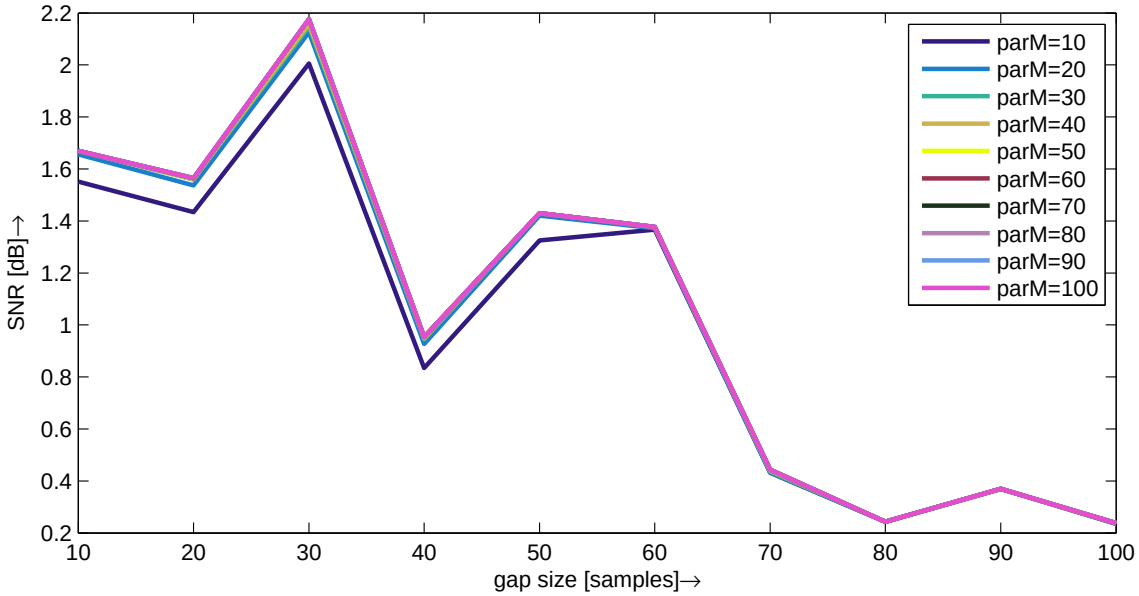


Fig. 7.11: Interpolation results of various gap length and mean amplitude of sinusoidal modeling.

little differences between the mean amplitude range of $\langle 10; 100 \rangle$. Mean amplitude of $M \geq 60$ results in completely same results of interpolation in terms of SNR and lower M produces SNR slightly lower. Since the difference is neglectable, following experiments will be performed with $M = 60$. Another reason for this particular value is the processing time of the interpolation algorithm. In Fig. 7.12 you see that the differences in computation times are smaller than 0.8 s.

Fourth parameter, the most important, is the model order. Number of AR coefficients of frequency of a single partial is set for both left and right side of the gap. Considering the length of a single partial from the left side l_i and an a-priori model order k_0, the resulting order k is computed as $k = \min(l_i, k_0)$. AR model order of each partial from both left and right side is computed separately. Note that single vector value of a partial represents result of STFT of the signal with fixed segment length of 256 samples and overlap of 128 samples.

Interpolation of the residual noise is done by AR modeling of samples, the model order k_t is computed as $k_t = 128 \cdot (k_0 + 1)$. In both cases (partials, noise), the limit of the AR model order is the beginning or end of the signal. The signal going to be interpolated is cut at the borders of the useful area around the gap according to the model order. This trimming causes an important acceleration of computations.

Batch testing results of file *music11_16kHz.wav* of various model order and gap length are illustrated in Fig. 7.13 and Fig. 7.14. The model order corresponds to a particular area according to the STFT window and overlap length around the

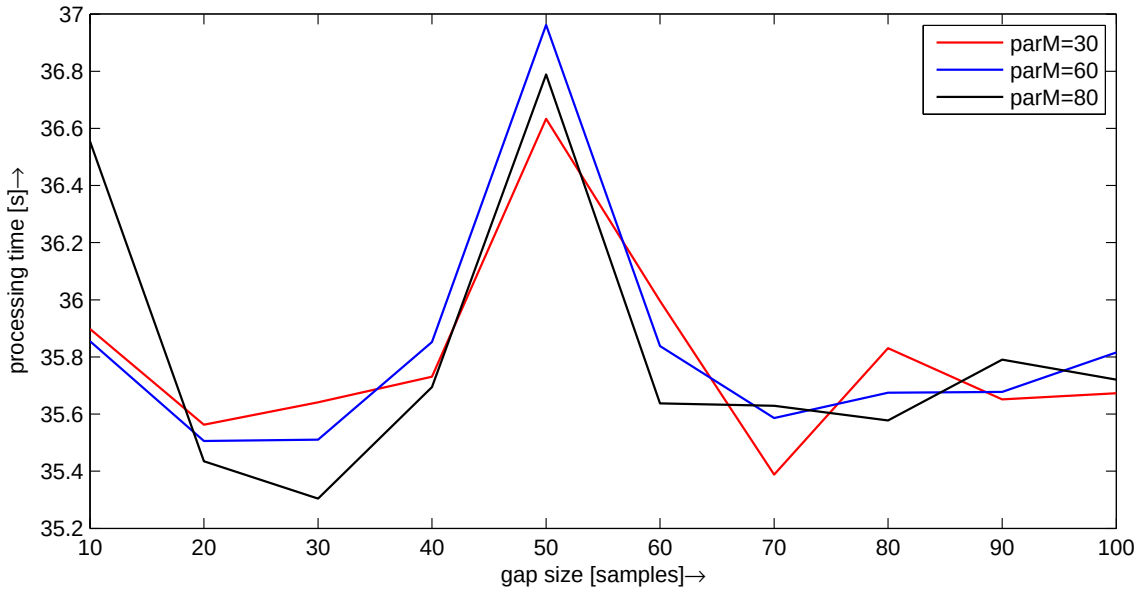


Fig. 7.12: Processing time of audio interpolation by sinusoidal modeling with various gap length and mean amplitude vector length.

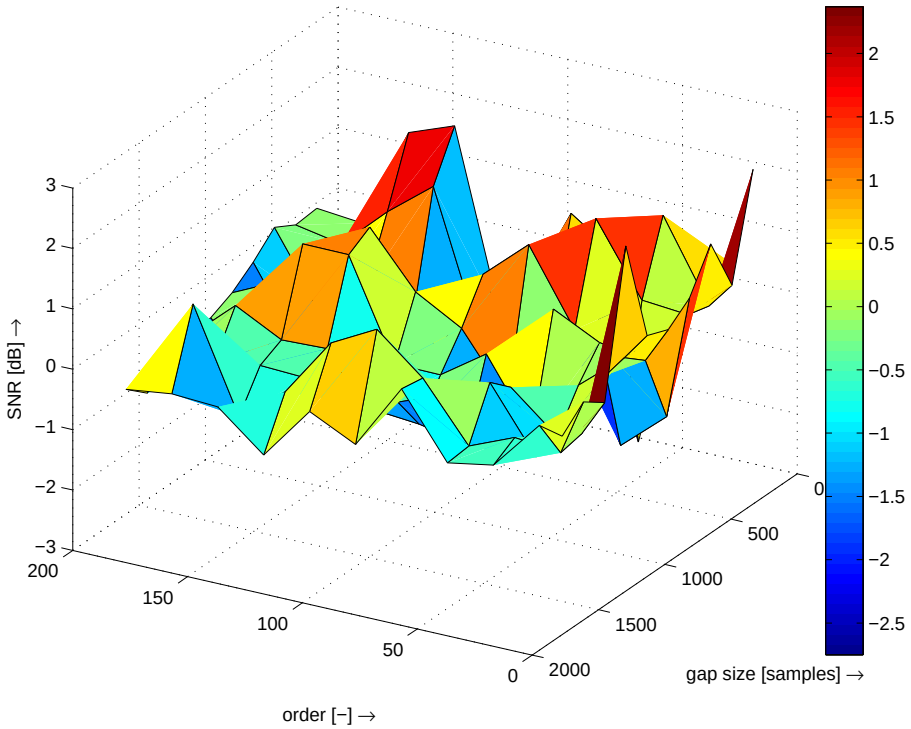


Fig. 7.13: SNR of audio interpolation by sinusoidal modeling with various gap length and AR model order.

Tab. 7.2: Values of AR model order and according neighbourhood size in the test.

order[-]	10	20	40	60	80	100
neighbourhood [samples]	1290	2580	5160	7740	10320	12900
order [-]	120	140	160	180	200	-
neighbourhood [samples]	15480	18060	20640	23220	25800	-

gap and the conversion is demonstrated in Tab. 7.2. For each combination of gap length and model order ten experiments with random gap position in the signal under investigation were performed. The mean values of SNR and processing time of these ten experiments were figured in the following surface graphs.

It is obvious that there is no conspicuous evolution of the SNR with respect to gap length in Fig. 7.13. A few peaks of SNR are located in the range of lower values of AR model order. Therefore, any general recommendations for the selection of the model order according to the gap length can not be defined. The interpolation algorithm has to be tuned for each gap size and position individually to reach the best results.

Regarding the speed of computations, from Fig. 7.14 it is obvious that the time consumption of the interpolation algorithm increases with the increasing model order value.

7.4 Greedy algorithm

The core algorithm for solving the sparse approximation by greedy methods is the OMP (Orthogonal Matching Pursuit - see Sec. 5.1) . This algorithm was implemented by our team and is a part of the inpainting toolbox. The dictionary (frame) consists of modulated atoms weighted by the Discrete Cosine Transform (DCT)-IV function and the frame is represented as a matrix where the columns are the particular atoms. The overcomplete dictionary is generated by function *odctdict()* from the *KSVDBox*[8] and the redundancy is caused by denser frequency sampling.

Following experiments are, as usual, performed on the sample signal, gap position and length as described in the beginning of this chapter. The signal around the gap is trimmed into a segment which is used for inpainting afterwards. The size of the segment (neighbourhood size) S is computed from the gap length M and neighbourhood multiplier N such that

$$S = N \times M. \tag{7.2}$$

[8] `http://www.cs.technion.ac.il/~ronrubin/software.html`

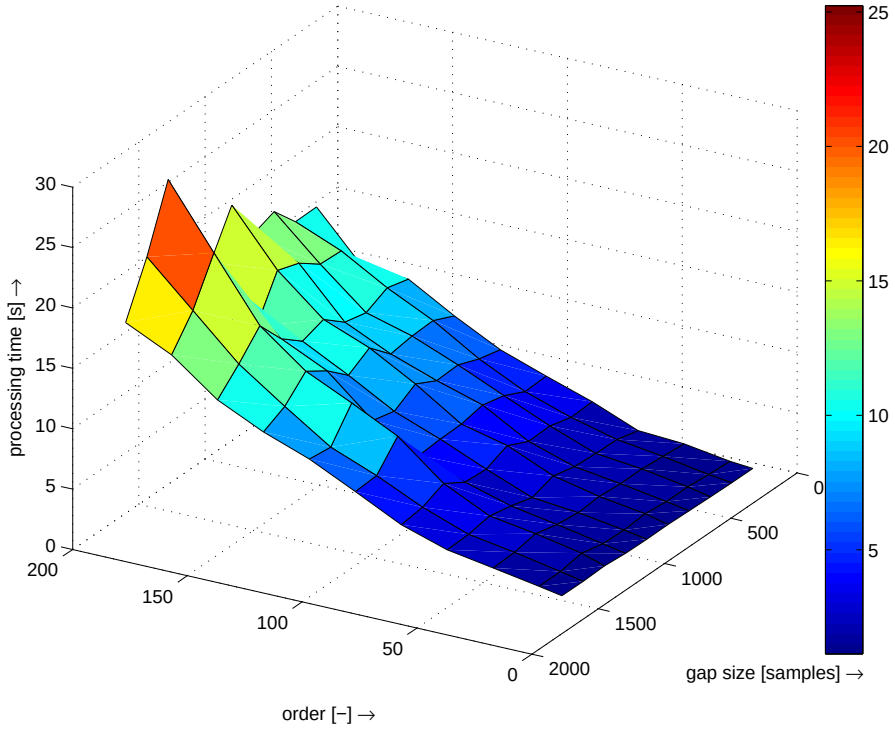


Fig. 7.14: Processing time of audio interpolation by sinusoidal modeling with various gap length and AR model order.

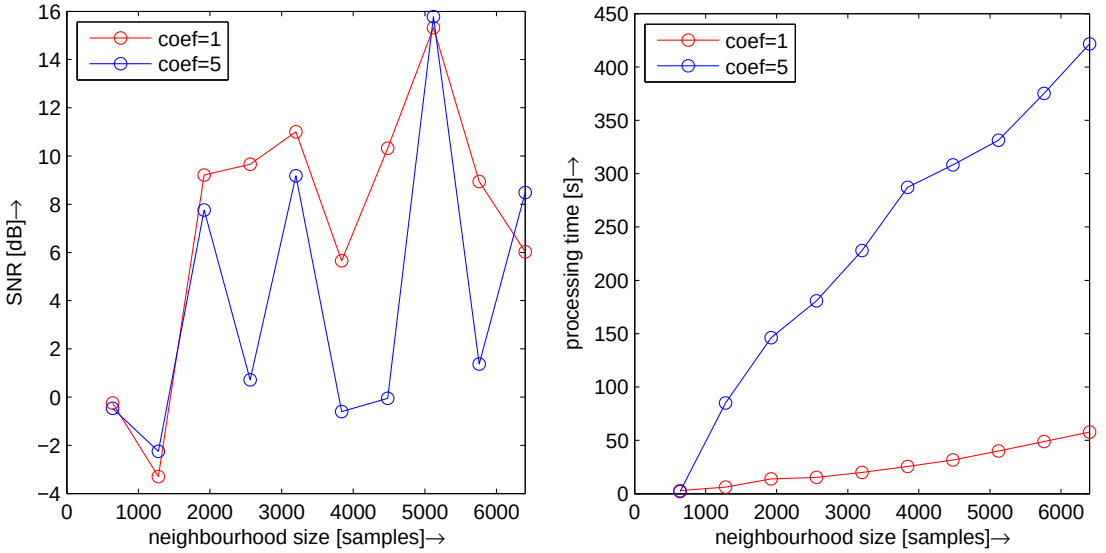
The size of the segment includes the size of the gap and the segment is centred to the middle of the gap. The whole segment size is then used as the length of one dictionary atom, i.e. no additional segmentation of the signal is incorporated.

There are several parameters of the OMP algorithm which influence the result of audio inpainting:

- no. of coefficients obtained in each iteration
- maximum number of iterations,
- maximum error,
- dictionary redundancy,
- neighbourhood size for inpainting.

Number of coefficients obtained in each iteration is the first parameter under investigation. Natural preset according to the OMP principle is only one coefficient obtained per iteration (see Alg. 2 for details). However, the algorithm offers the possibility to choose more than one coefficient at the same time.

In Fig. 7.15 there is a comparison of 1 and 5 coefficients simultaneously chosen in each iteration. Analyzing the figured results, better SNR was reached obtaining only 1 coefficient in a single iteration. In contrast, the processing time increases enormously when 5 coefficients are obtained simultaneously. Therefore, for the following experiments only 1 coefficient per an iteration will be obtained.



(a) (b)

Fig. 7.15: Audio Inpainting using OMP of various neighbourhood size and number of coefficients obtained in each iteration in terms of (a) SNR and (b) processing time.

Another parameter is the redundancy of the dictionary. Here, the redundancy of 1 means, that the number of modulation frequencies is equal to the segment length. The parameter under investigation is the multiplier of the number of modulation frequencies. In this experiment the redundancy factor *red* is set up in the range of $red = \{1, 2, \ldots, 10\}$.

Results in Fig. 7.16 show that redundancy of 1 and 2 produce poor results in terms of SNR, sometimes with SNR below zero. Higher values of redundancy factor produce better results with the best reached value of SNR = 15.64 dB for red = 9 and neighbourhood size of 1280 samples. However, from Fig. 7.17 it is obvious that the processing time of higher values of the redundancy factor is not efficient either for experimental or practical purpose. Therefore, for the following experiments redundancy factor of 3 will be used since it provides a good trade-off between the interpolation results and the processing time.

The most extensive batch testing was performed on music file *music*11_16-*kHz.wav*. The objective of the experiment was to find an optimal neighbourhood size according to the various gap length. The size of the gap was from the range of $\{10, 20, \ldots, 100\}$ ms which corresponds to $\{160, 320, \ldots, 1600\}$ samples. Neighbourhood size factor is a multiplier of the gap size resulting in the full neighbourhood size and was selected from the range of $\{2, 4, \ldots, 20\}$. Note that the full neighbourhood size includes the size of the gap with the centre of neighbourhood situated to the

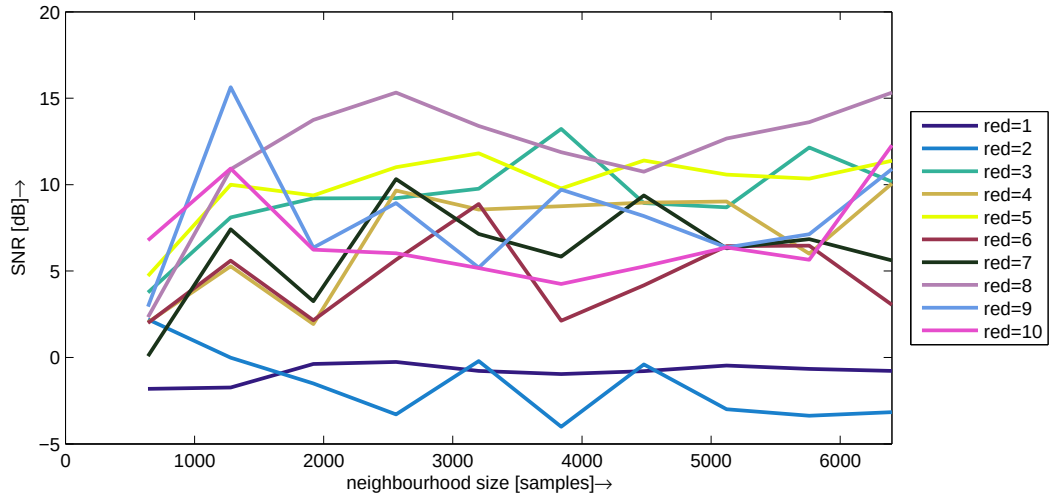


Fig. 7.16: SNR of audio inpainting by OMP with various neighbourhood size and dictionary redundancy.

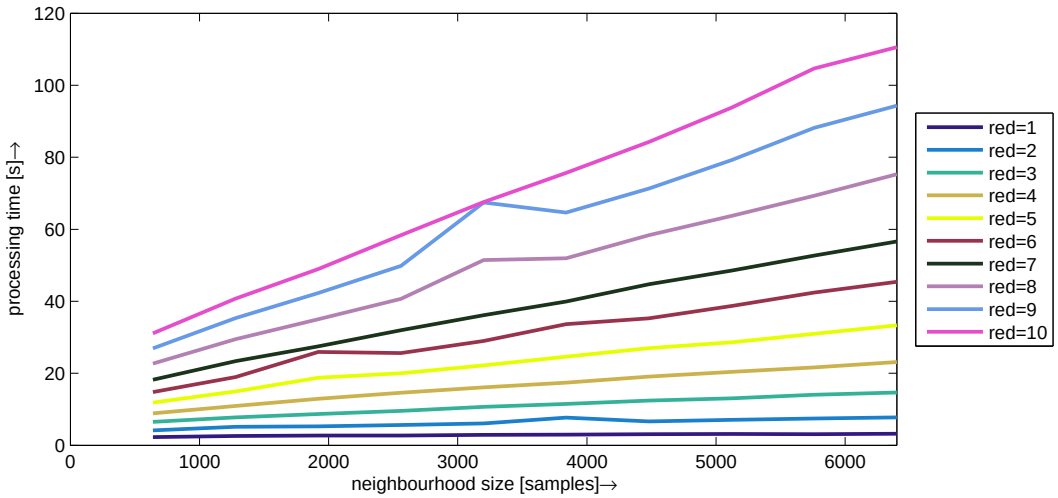


Fig. 7.17: Processing time of audio inpainting by OMP with various neighbourhood size and dictionary redundancy.

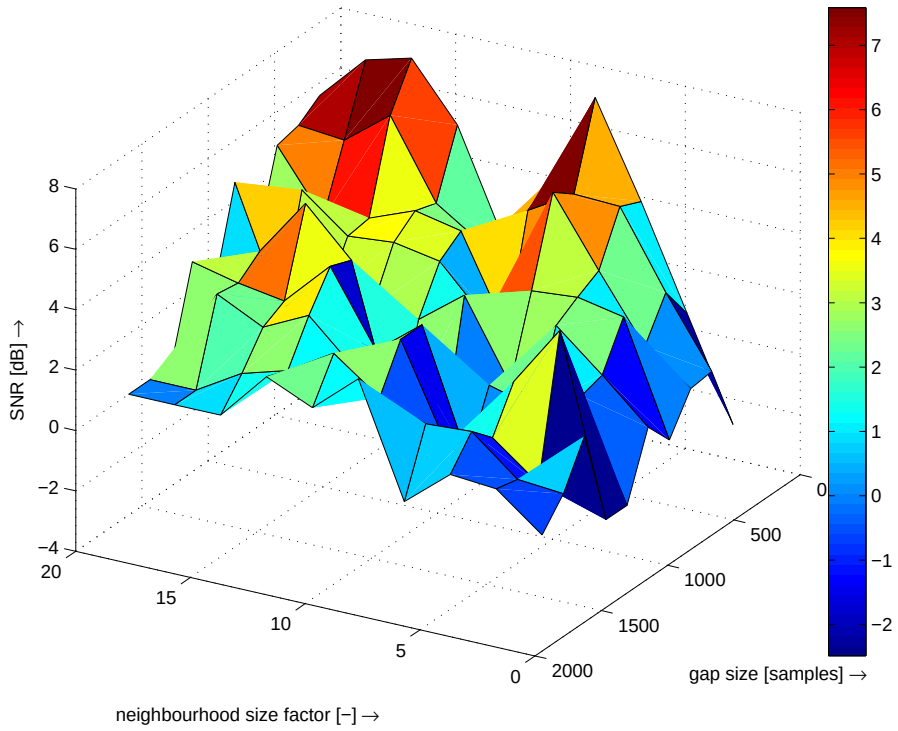


Fig. 7.18: SNR of audio inpainting by OMP with various neighbourhood size according to gap length.

middle of the gap. Each combination of the gap length and neighbourhood size factor was examined ten times with random gap position and the final results presented in the following figures are the average values from the ten experiments. Experiments of combination of the highest neighbourhood size factors equal to $16, 18, 20$ and gap size of $1280, 1440, 1600$ were skipped because of very high processing time demands. Details will be described in the following paragraphs.

In Fig. 7.18 an average SNR of this experiment is demonstrated. The best results were reached for the shortest gap size and are illustrated by the highest peaks in Fig. 7.18. The highest average SNR $= 7.59\,\mathrm{dB}$ was obtained for gap size of $10\,\mathrm{ms}$ and neighbourhood size factor of 8 which results in the neighbourhood size of $1280\,\mathrm{samples}$ ($80\,\mathrm{ms}$).

Better overview of the best results of this experiment is in Fig. 7.19 where the green points indicate the best SNR reached for every gap size. It is obvious that larger neighbourhood size produces better inpainting results in the terms of SNR. However, the processing time increases dramatically with increasing neighbourhood and gap size (see Fig. 7.20).

As a conclusion of this evaluation, the optimal trade-off between the speed of computation and the resulting inpainting performance using OMP algorithm should be the neighbourhood size factor of at least 4.

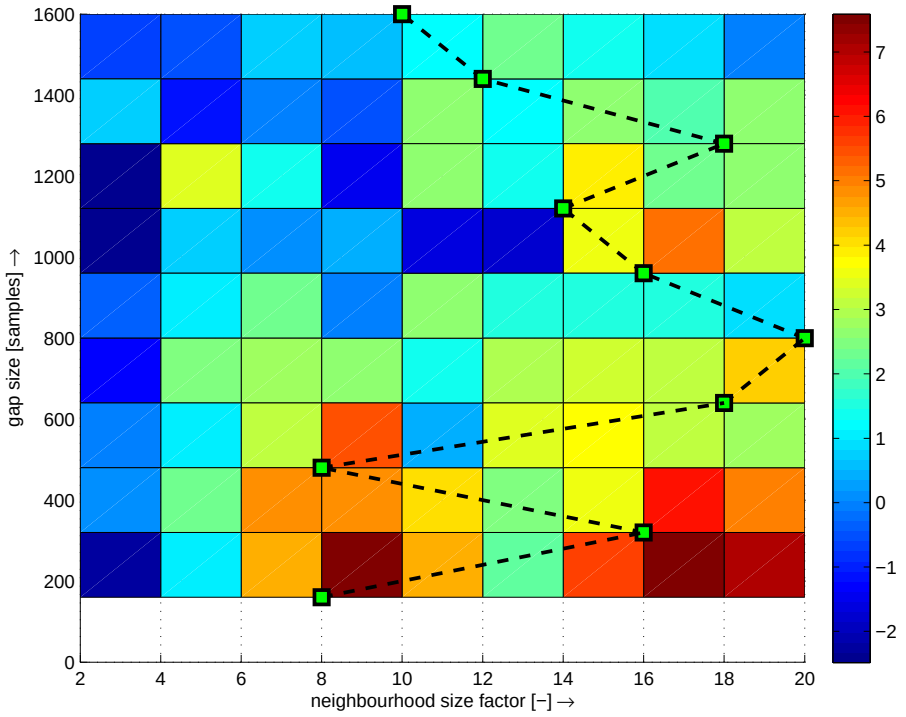


Fig. 7.19: SNR of audio inpainting by OMP with various neighbourhood size according to gap length with highlighted best results by green points.

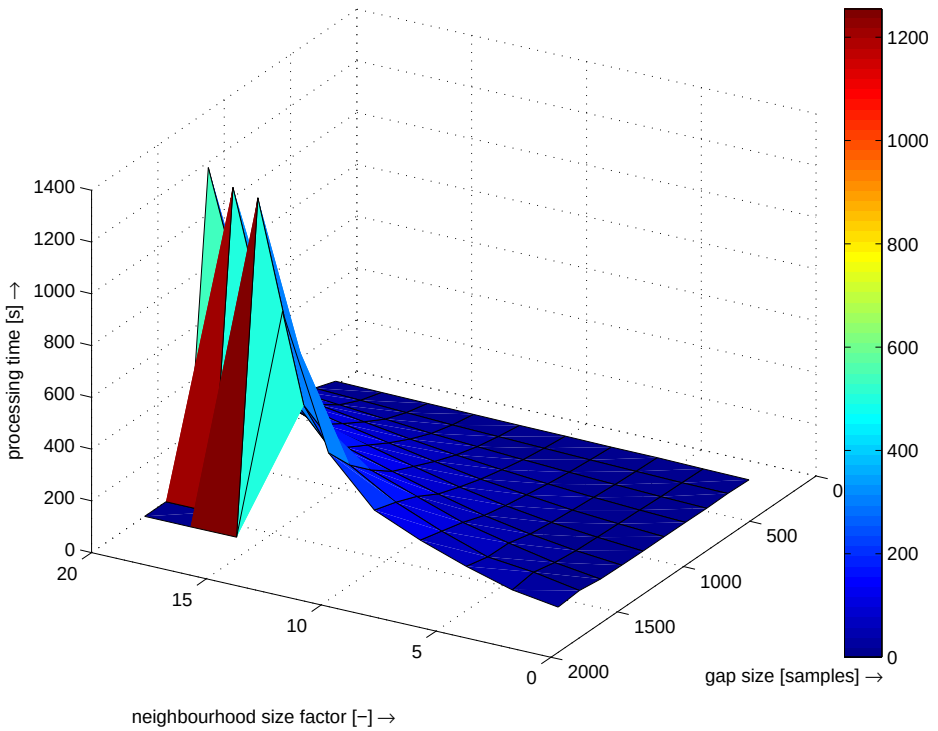


Fig. 7.20: Processing time of audio inpainting by OMP with various neighbourhood size according to gap length.

One more remark of this batch experiment is the processing time of large values of the gap size and neighbourhood size factor. In most cases the processing time of experiments with neighbourhood size factor > 16 together with gap size > 80 ms reached more than three hours for one experiment. Therefore, these experiments were omitted and in figures the value of the processing time and SNR is set to zero.

7.5 Dictionary learning

Dictionary learning described in theory in Sec. 4.3.2 was utilized for the audio inpainting together with the greedy algorithm OMP as a problem solver. Among several dictionary learning algorithms the K-SVD method implemented within the SMALLbox[9] was chosen for experiments. Several parameters were examined during the tests. According to user definition several iterations of the process of dictionary learning are performed. After each iteration of the dictionary learning the Root Mean Square Error (RMSE) is computed such that

$$\mathrm{RMSE} = \sqrt{\frac{\|\mathbf{Y} - \mathbf{D}_{\mathrm{opt}}\mathbf{X}\|_F^2}{N \times M}}, \tag{7.3}$$

where $\mathbf{X}$ is the matrix with segmented vector of original signal, $\mathbf{D}_{\mathrm{opt}}$ is the optimized dictionary of size $M \times N$ and $\mathbf{X}$ is a segmented vector of coefficients. Setting the number of iterations to 4 reaches a satisfying value of RMSE and after 10 iterations the RMSE is stabilized at its minimum.

Other experiments were focused on minimizing RMSE according to space between segments obtained from reliable samples to get the training data. Having a short audio file with not enough training segments of the signal it has to be decided between smaller segment shift for more training data and larger segment shift for less training data. However, decreasing the segment shift is nothing but artificial enlarging the amount of training data and the samples are repeated in training segments. Using the audio file of length 80 000 samples, the segment length of 256 samples and redundancy factor of 3, the dictionary $\mathbf{D}$ has got a size of 256×768 samples. With these parameters the shift of segments could be set to a value from interval $\{1, 2, \ldots, 100\}$. Increasing the segment shift value results in smaller RMSE during the dictionary learning process.

Another parameter of the dictionary learning explored further was the maximum number of non-zero coefficients $S_{\max}$. For $S_{\max} \in \{1, 2, 3, 4, 5\}$ dictionary learning experiments were performed with focus on the lowest RMSE depending on different $S_{\max}$ and therefore reaching the minimal error. After six iterations the minimal RMSE was reached by $S_{\max} = 3$ and remains minimal with very little change.

[9] `http://small-project.eu/software-data/smallbox/`

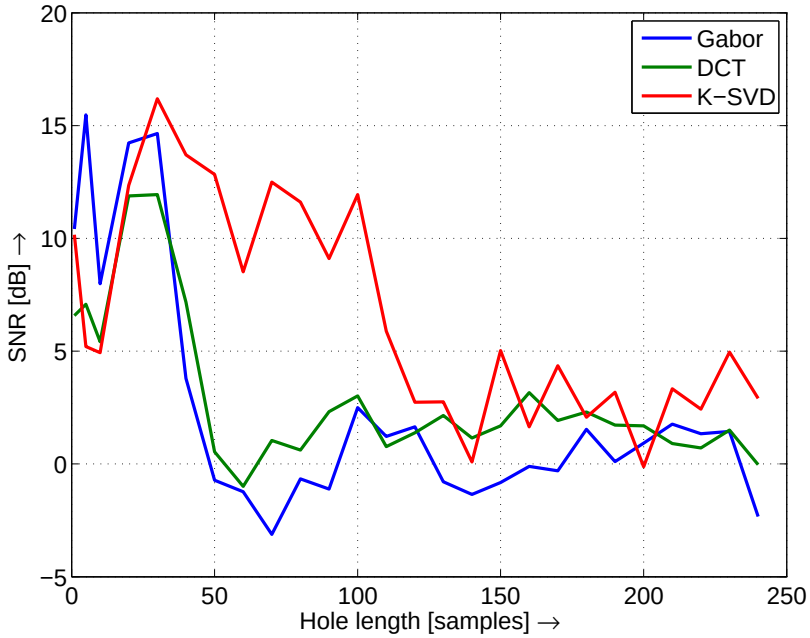


Fig. 7.21: Signal reconstruction of a guitar audio sample using the K-SVD algorithm and greedy solver.

The number of iterations has to be set up deliberately. The number can be small and RMSE will remain high (the dictionary is not adapted as much as it can be) or the number can be too high and after reaching the minimum RMSE the algorithm can waste the time with new iterations or worse the RMSE can raise up. That is why another experiment observing RMSE was performed with the best parameters obtained above. As a result, satisfying RMSE can be obtained with three or four iterations. This test was performed for number of iterations from interval $\{1, 2, \ldots, 200\}$.

An experiment was performed on the example music file *music11_16kHz.wav*. Figure 7.21 shows that for gap length of 40 to 110 samples dictionary trained by the K-SVD algorithm overcomes static dictionaries by about 10 dB. Results of reconstruction by static dictionaries (Gabor and DCT) produce almost the same results. Presented results were published in [6].

Further optimization of the K-SVD algorithm was expected from the method called the Incoherent K-SVD. The level of coherence between two atoms could be high using the natural K-SVD algorithm. In [65] the decorrelation algorithm for diversifying of the atoms was presented.

An effort for improvement of the dictionary learning parameters process was performed in the master thesis [72] supervised by the author of this thesis. Results say that decreasing incoherence to the values of 0.1 and 0.2 improves inpainting results

in terms of SNR. Another conclusion was that selecting the training atoms from areas similar to the area around the gap also makes an improvement of the results. An algorithm for segmentation of stationary parts of the signal was presented in [81]. The Incoherent K-SVD algorithm was examined for the reconstruction of various musical genres. As a result, most of typical genres (including country, hip-hop, pop and reggae) signal gaps were inpainted with better SNR than the static dictionaries.

The drawback of these methods is the computational time. While the time needed for training of the K-SVD dictionary of size 1024×1024 is about 10 minutes, and incoherent dictionary training of the same size lasts for several hours. Decreasing the number of training data reduces the computational time. On the other hand, lower values of coherence (which are in fact appreciated) extend the processing time of the dictionary learning.

However, there are doubts about correctness of the algorithm for searching for stationary objects. Further, the resulting dictionary trained by the Incoherent K-SVD algorithm is not overcomplete, therefore, it looses the advantage of frames and sparse representations. Because of these handicaps, this method was not examined further.

7.6 ℓ_1-relaxation algorithm

Most of the contribution was focused on the implementation of the analysis and the synthesis model for their comparison. Since all of the papers dealing with the audio inpainting use the synthesis model ([12],[55],[18]) there was a natural interest in an implementation of the analysis model and its evaluation. All of the methods are theoretically described in Sec. 5.2.

For all of the following experiments, the signal support is restricted around the missing gap position. Therefore, the inpainting algorithm is accelerated. The time-frequency transform is built up from time and frequency localized windows and their translations. Only the coefficients which contribute to the gap are going to be recovered. Consequent atoms with non-zero overlap with the gap are relevant. See [77] for more details.

7.6.1 Dictionary redundancy

Experiments testing various redundancy in time, i.e. parameter a in Eq. 4.15 were performed on a single wav file (*music08_16kHz.wav*). There were two varying parameters during these tests: redundancy value from the ranges of $\{2, 3, \ldots, 8\}$ and the length of the window of $\{513, 1026, 2052, 3078, 4104, 5130, 6156, 7182, 8208, 9234, 10260, 11286\}$ samples. As long as the window lengths have to be divisible by the

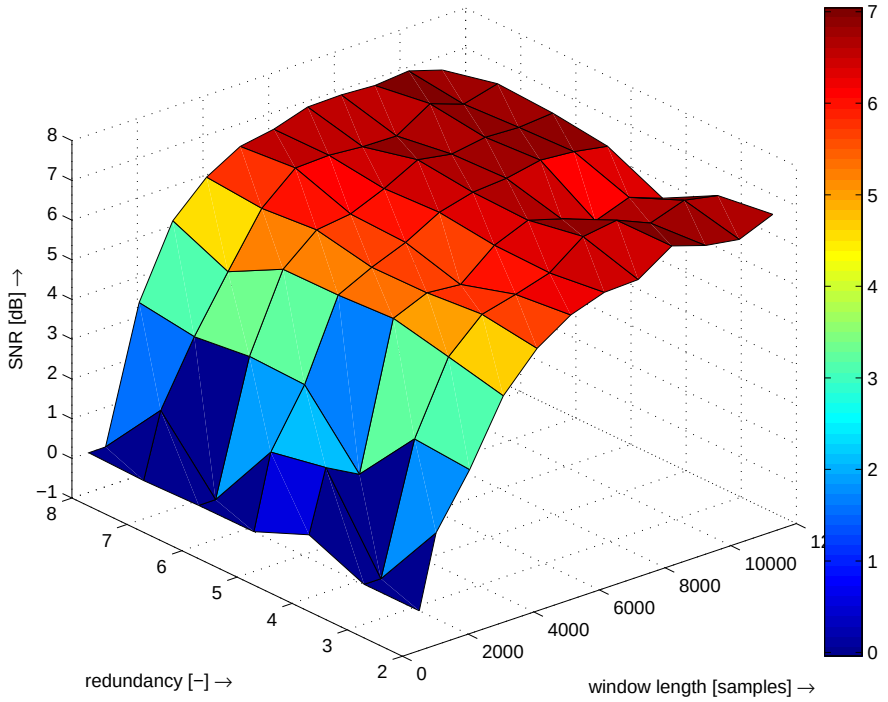


Fig. 7.22: SNR of inpainting with various redundancy and window length in analysis model.

redundancy number, the presented windows lengths were adjusted to the closest divisible value. These experiments were performed on a single gap with fixed position and the length of 320 samples (20 ms).

As illustrated in Fig. 7.22 and 7.23, the evolution of SNR for an increasing window length seems to be more or less the same for all redundancy values. It seems that in the synthesis model the SNR values are more fuzzy than in the analysis model. From this time the redundancy will be fixed to the value of 3 for all of the following experiments. Obtaining comparable results with all the redundancy values, the computational complexity is smaller with lower redundancy. Another reason for this value is that the Partition of Unity and a tightness of the frame is fulfilled with redundancy $= 3$ (window overlap is $2/3$) using Hann window [83].

7.6.2 Weighting of atoms

In [12] dictionary atoms afflicted by the signal gap are weighted. Weighting is performed as such that the ℓ_2 norm of the atom is equal to 1. In the LTFAT toolbox frames are defined by the function and it is very computationally inefficient to interpret frames as matrices (resulting in Out of memory error in Matlab). Therefore, the weights of the atoms presented in Sec. 5.2.4 are applied on resulting coefficients. This calculation is also challenging because of the required memory space, conse-

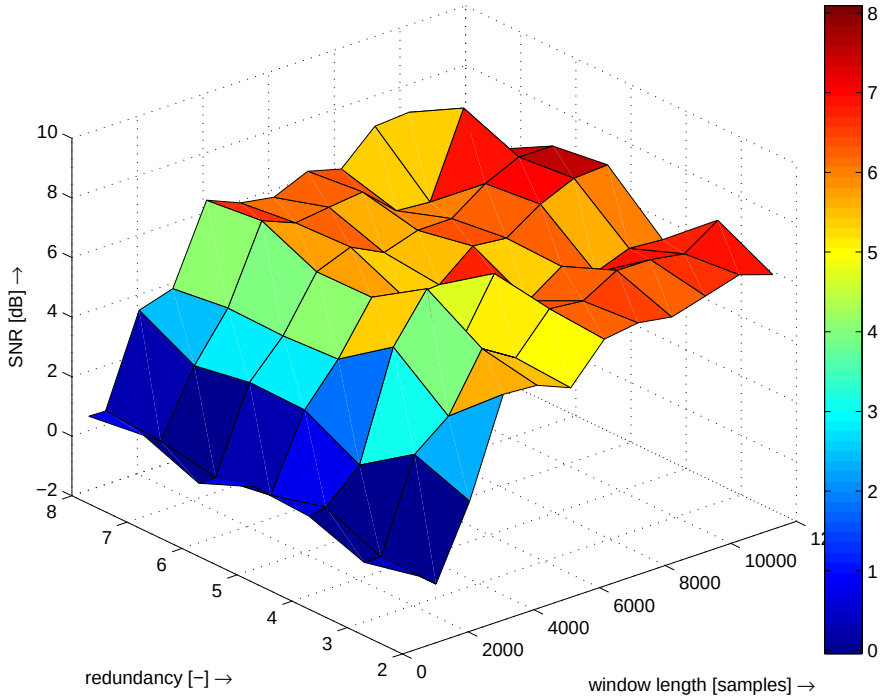


Fig. 7.23: SNR of inpainting with various redundancy and window length in synthesis model.

Tab. 7.3: Values of the gap length in the test

Gap length [ms]	10	20	30	40	50	60	70	80	90	100
Gap length [samp.]	160	320	480	640	800	960	1120	1280	1440	1600

quently, the computation is performed for each of atom separately and the weight is computed immediately. The only considered case is that the frame bounds A and B are equal to 1 (it is a Parseval tight frame)[7]. For all of the following experiments weighting is enabled.

7.6.3 Analysis vs. Synthesis model

Various wav files containing different kinds/genres of signals were used as a testing group of examples. These wav files are originally used in the SmallBox [29]. The gap length was set up to the values in Tab. 7.3.

For each size of the gap an inpainting experiment was performed with various window sizes of of $\{513, 1026, 2052,\ 3078, 4104, 5130,\ 6156, 7182, 8208,\ 9234, 10260,\ 11286\}$ samples. For each combination of gap size and window length there were 10 independent inpainting experiments with different gap position for each analysis

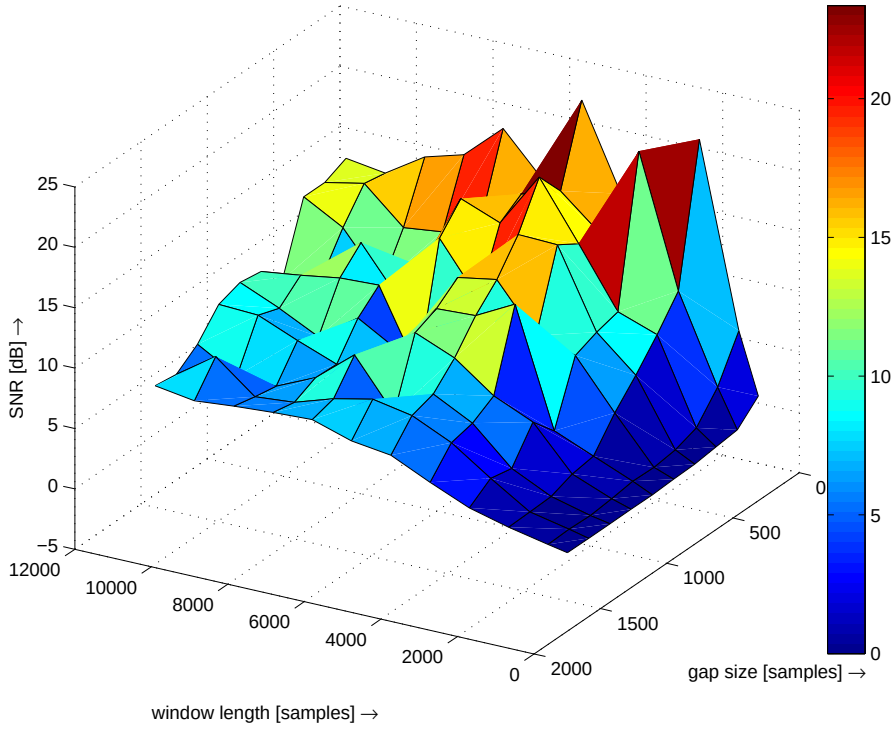


Fig. 7.24: Mean SNR of inpainting with various gap length and window length in the analysis model.

and synthesis model. From the resulting reconstruction the SNR value is computed on the non-reliable samples followed by the computation of the average SNR value and variance of the 10 experiments.

An example of a piece of music with high level of harmonicity is a music file *music11_16kHz.wav* containing a guitar playing chords. As illustrated in Fig. 7.24, the best restoration in the terms of objective SNR evaluation was obtained in the case of short gaps and a size of window length from 2000 to 9000 samples (from 125 to 560 ms). Increasing the length of the gap, the SNR is decreasing and range of the satisfying window length is narrower using the analysis model. Likewise in the synthesis model, the satisfying SNR for gap length of 1120 samples is reached using window length in the range from 5130 to 7182 samples, see Fig. 7.25.

The explanation of this behaviour is that with very small (window length)/(gap size) ratio the window is not able to find enough of a reliable pattern for inpainting such a gap. On the other hand, having large ratio of (window length)/(gap size), samples inpainted inside the gap does not have space for any fluctuation and are rather static.

Looking at the same 3D plots from another perspective in Fig. 7.26 7.27, the black line connects the points with the highest SNR for each gap size. In general the best reconstruction is reachable choosing parameters with color closest to red

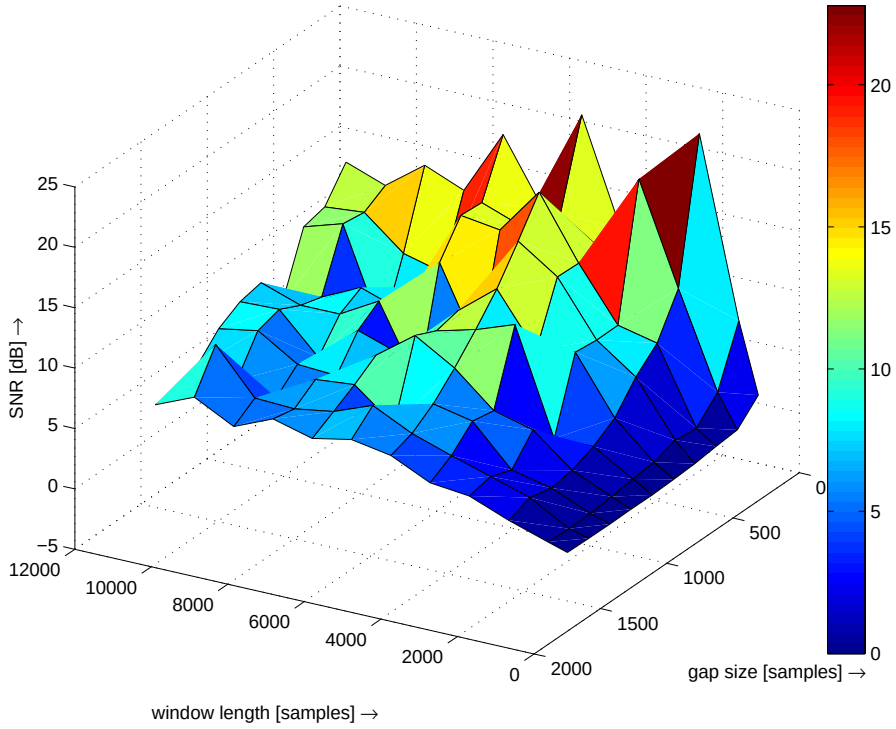


Fig. 7.25: Mean SNR of inpainting with various gap length and window length in the synthesis model.

color.

The same results are analysed from the view of the standard deviation. The standard deviation is computed as

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{7.4}$$

where $N = 10$ is a number of experiments with fixed gap size and window length, x is a current SNR value and μ is an average SNR of all N experiments. For graphical representation of standard deviation of *music11_16kHz.wav* file see Fig. 7.28 and 7.29. Detailed observation of the points with the highest SNR of all wav files denotes that these maxima rarely correspond to the maxima points in the standard deviation plot.

Table 7.4 shows the details about the maximum SNR values for the analysis and synthesis model for *music11_16kHz.wav*. In this particular file, the worst possible values of SNR 15.85 dB and 19.14 dB respectively are very satisfying. However, in most cases the point with the worst standard deviation is "very close" to the point of the maximum mean SNR. "Very close" means that the hop size to that critical point is usually in the distance of one gap size hop or one window length hop. See this behaviour in detail in Tab. 7.5. The worst case SNR is more than

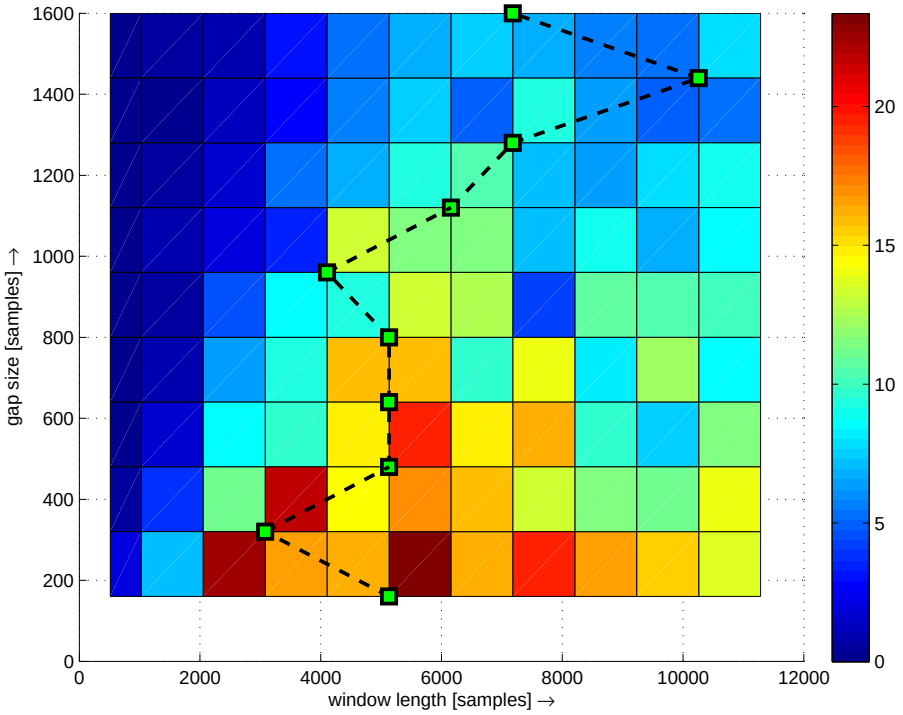


Fig. 7.26: Average SNR of inpainting with various gap length and window length in the analysis model. Dashed line with green points indicates the maximum values for each gap length.

Tab. 7.4: Best SNR results of `music11_16kHz.wav` with its standard deviation

Model	Gap size [samples]	Window length [samples]	Average SNR [dB]	Standard deviation [dB]	Worst case SNR [dB]	Best case SNR [dB]
Ana	160	5130	23.35	7.50	15.85	30.85
Syn	160	2052	22.79	3.65	19.14	26.44

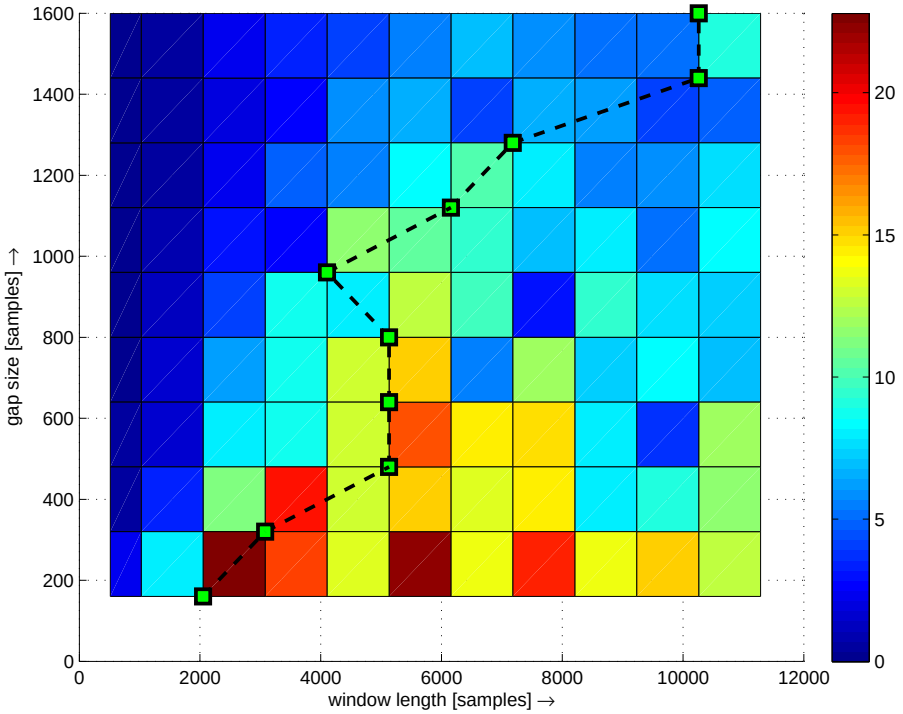


Fig. 7.27: Mean SNR of inpainting with various gap length and window length in the synthesis model. Dashed line with green points indicates the maximum values for each gap length.

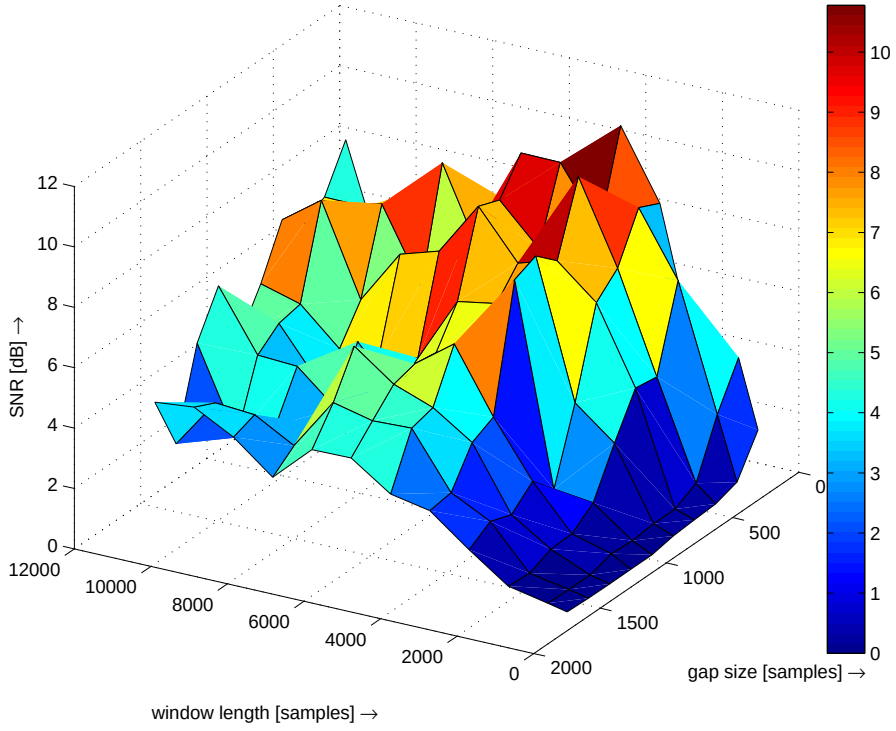


Fig. 7.28: Standard deviation of the SNR of inpainting with various gap length and window length in the analysis model.

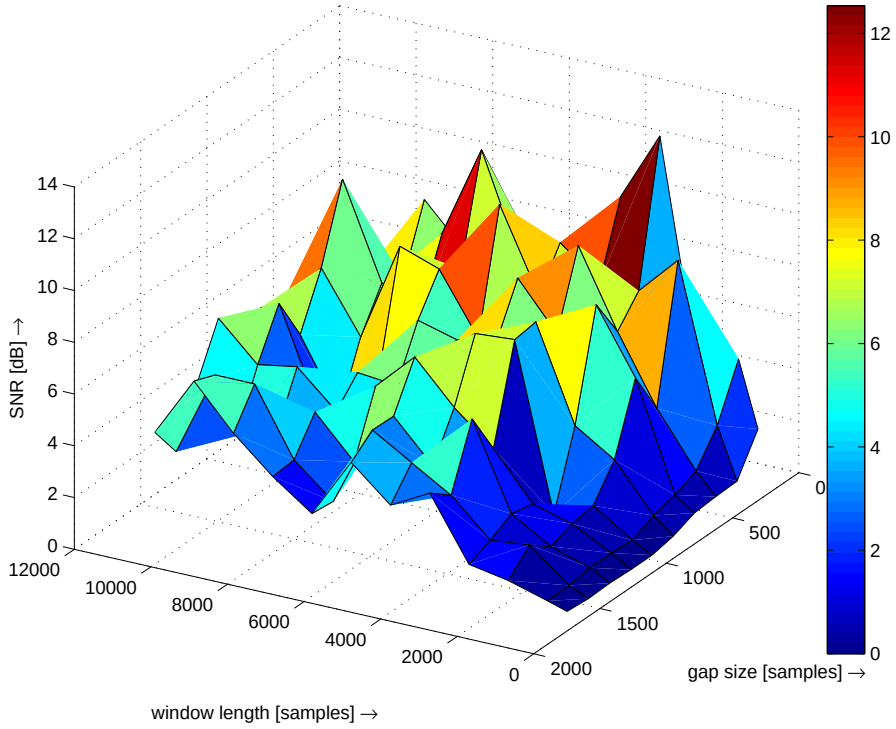


Fig. 7.29: Standard deviation of the SNR of inpainting with various gap length and window length in the synthesis model.

10 dB lower (compared to Tab. 7.4) while the window length hop size is just one step (1026 samples).

A short conclusion of this observation is that there are somehow optimal parameters for a particular gap size and gap position, nevertheless, they can be very easily missed by a slight modification of a window length. This kind of behaviour was observed in almost all other experimental sound pieces with harmonic structure.

Tab. 7.5: Parameters of points with the worst standard deviation of the SNR results of `music11_16kHz.wav`

Model	Gap size [samples]	Window length [samples]	Average SNR [dB]	Standard deviation [dB]	Worst case SNR [dB]	Best case SNR [dB]
Ana	160	4104	16.34	10.78	5.56	27.12
Syn	160	3078	18.39	12.54	5.85	30.93

Tab. 7.6: Best SNR results of `music07_16kHz.wav` with its standard deviation

Model	Gap size [samples]	Window length [samples]	Average SNR [dB]	Standard deviation [dB]	Worst case SNR [dB]	Best case SNR [dB]
Ana	160	8208	3.32	4.97	−1.65	8.29
Syn	160	8208	2.70	4.75	−2.05	7.45

7.6.4 Non-harmonic signals

In contrast with harmonic signals, inpainting of signals with non-harmonic structure with a lot of onsets and noise was not as successful as the harmonic signals. An example of such non-harmonic signal is the recording of the drums (`music07_16kHz`). In the 3D-plot of an inpainting (Fig. 7.30 and 7.31) using various window length and gap size it is obvious that the reconstruction failed in all combinations of parameters. The maximum value of SNR as an average of 10 experiments for each particular combination of window length and a gap size is in Tab. 7.6. Therefore, the Audio Inpainting of such non-harmonic signal should be performed by some other technique.

In the following Tab. 7.7 and 7.8 is the comparison of all pieces of sound with their average overall SNR, average overall standard deviation and a character of the signal. The highest SNR was reached in files containing a harmonic signal, especially when only a single instrument is playing. The worst results were obtained with rather non-harmonic records containing speech or completely non-harmonic signals.

Another notable remark is that the order of the analysis and synthesis results is the same, however, results of the analysis model are reaching better SNR values then the synthesis. Further, the standard deviation of the synthesis model is the same or higher in all of the experiments compared to the analysis model standard deviation.

7.6.5 Processing time

Considering the speed of the computation, there is quite a big difference between the analysis and the synthesis model especially using large window sizes. In Figures 7.32, 7.33 and 7.34 you can see the plot of the average processing time of 10, 50 and 100 ms long gap for various window lengths.

It is clearly visible that the processing time of the analysis model is significantly lower than the processing time of the synthesis model. In the range of

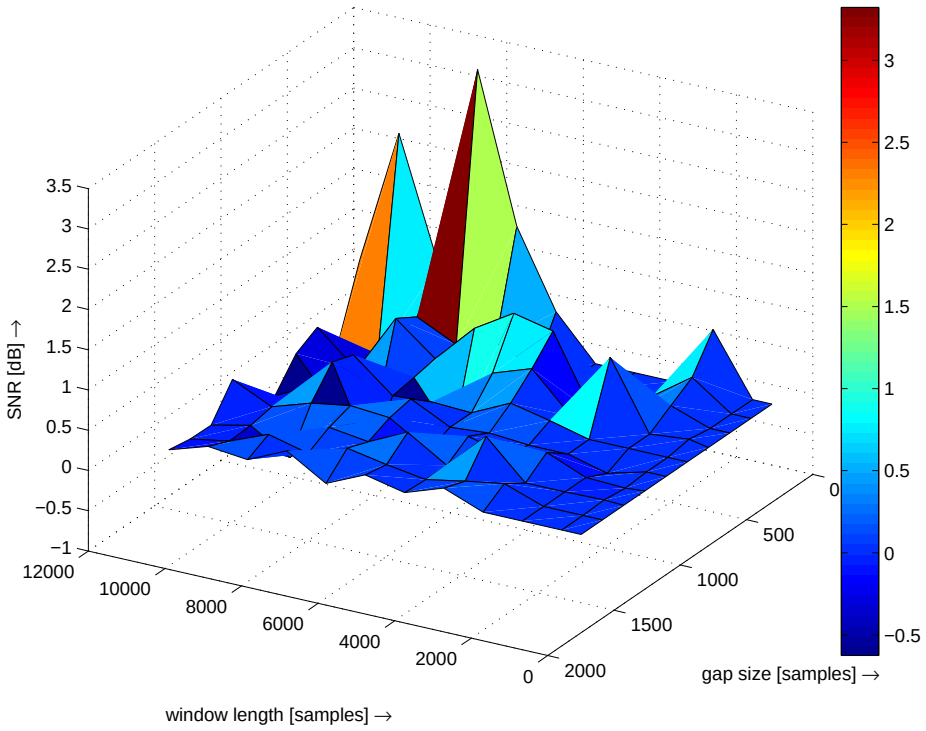


Fig. 7.30: Mean SNR of inpainting with various gap length and window length in the analysis model (music07_16kHz.wav).

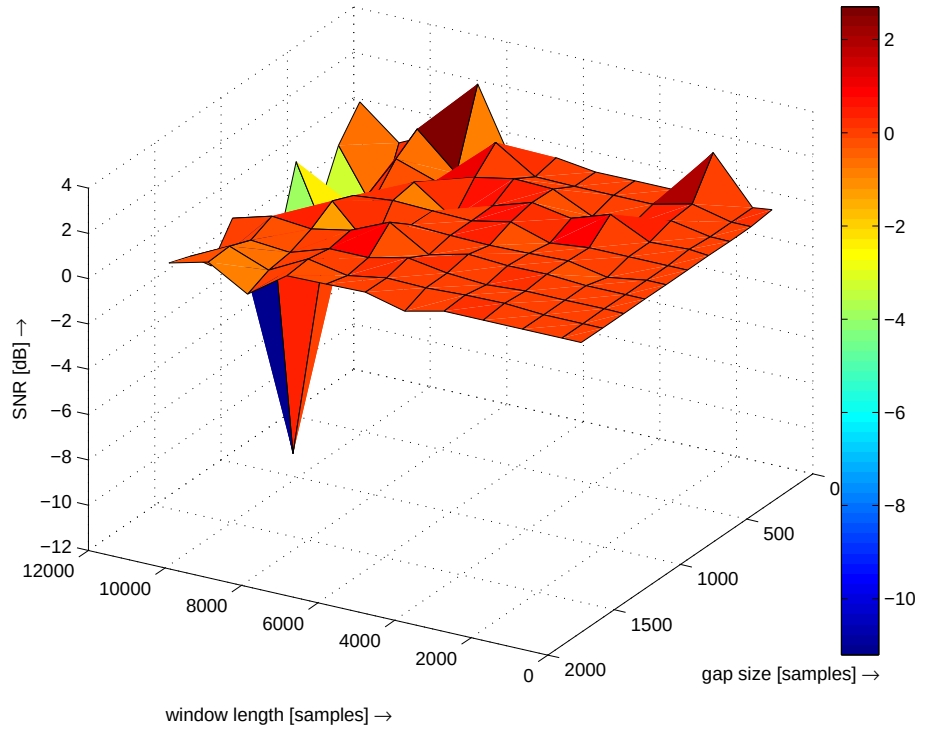


Fig. 7.31: Mean SNR of inpainting with various gap length and window length in the synthesis model (music07_16kHz.wav).

Tab. 7.7: Results of inpainting experiments over all of the sound files (analysis model, sorted by the average SNR)

Filename	Average SNR [dB]	Average STD [dB]	Character
music11_16kHz	8.1	4.4	harmonic, guitar
music03_16kHz	7.9	4.6	harmonic, guitar
music02_16kHz	5.0	3.8	harmonic, double bass
music04_16kHz	4.5	4.7	harmonic, woman singing
music10_16kHz	4.2	1.7	harmonic, orchestra
music08_16kHz	3.4	2.0	harmonic, pop music
music09_16kHz	1.7	1.7	speech, rap
music12_16kHz	1.1	1.5	speech, DJ show
music07_16kHz	0.1	0.6	non-harmonic, drums

Tab. 7.8: Results of inpainting experiments over all of the sound files (synthesis model, sorted by the average SNR)

Filename	Average SNR [dB]	Average STD [dB]	Character
music11_16kHz	7.3	4.3	harmonic, guitar
music03_16kHz	6.7	4.7	harmonic, guitar
music02_16kHz	4.4	4.0	harmonic, double bass
music04_16kHz	4.0	5.2	harmonic, woman singing
music10_16kHz	3.8	1.6	harmonic, orchestra
music08_16kHz	2.8	1.9	harmonic, pop music
music09_16kHz	1.4	1.7	speech, rap
music12_16kHz	0.8	1.6	speech, DJ show
music07_16kHz	-0.3	1.5	non-harmonic, drums

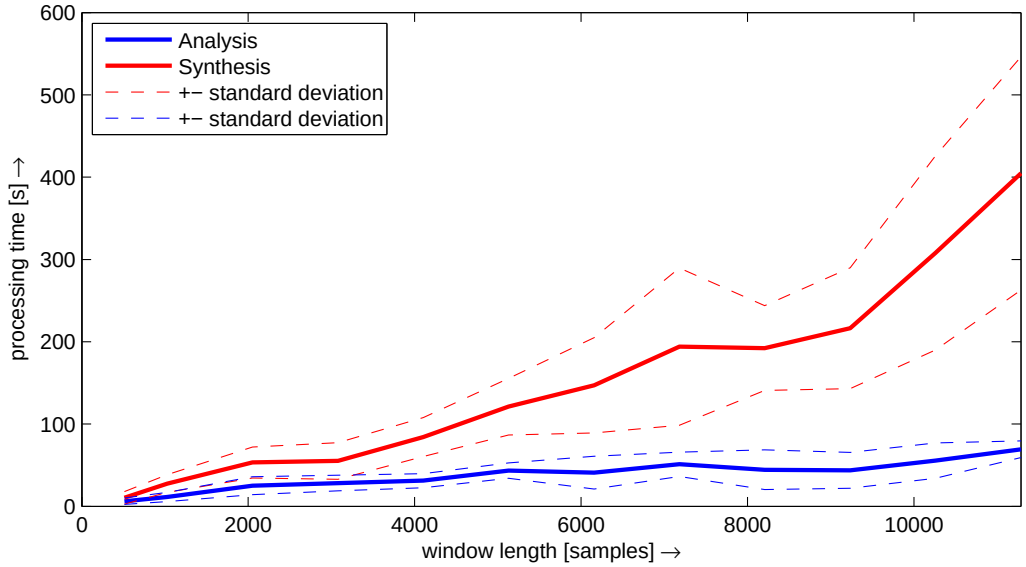


Fig. 7.32: Average inpainting processing time of file music11_16khz with gap size of 160 samples and various window length.

$\langle 0, 3000 \rangle$ samples of the window length the processing time of an inpainting process is almost the same for both models while the computational time of the synthesis model is still higher. Around the window size of 4000 samples the difference between the processing time of the models is increasing. The top values are usually reached with the largest window length.

Another remark from this experiment is that the average computational time of the inpainting process using a proximity algorithm is not dependent on the gap size. On the other hand, the standard deviation of the computational time is slightly increasing with the growing gap size.

Fig. 7.35 illustrates the ratio of processing time of the synthesis and analysis model. The ratio is computed from average processing time values for each window length and gap size. For gap lengths of 80 ms and 100 ms and window size of 512 samples the ratio reached up to the value of 6, all other ratios in the plot for all other increasing window lengths show almost the same evolution. With increasing window length the ratio of the processing time is generally ascending with the range of the ratio for each window length from 2 to 3. The maximum processing time of the analysis model is almost always smaller then the synthesis model. One of the possible reasons could be the fact that the result of the solver in the analysis model is the signal directly. On the other hand, an additional transformation of the coefficients to the time values is needed at the end of the synthesis model computations.

Fig. 7.36 represents the ratio of the number of iterations computed in the same way as described in the last paragraph. Compared with Fig. 7.35, the plots look the

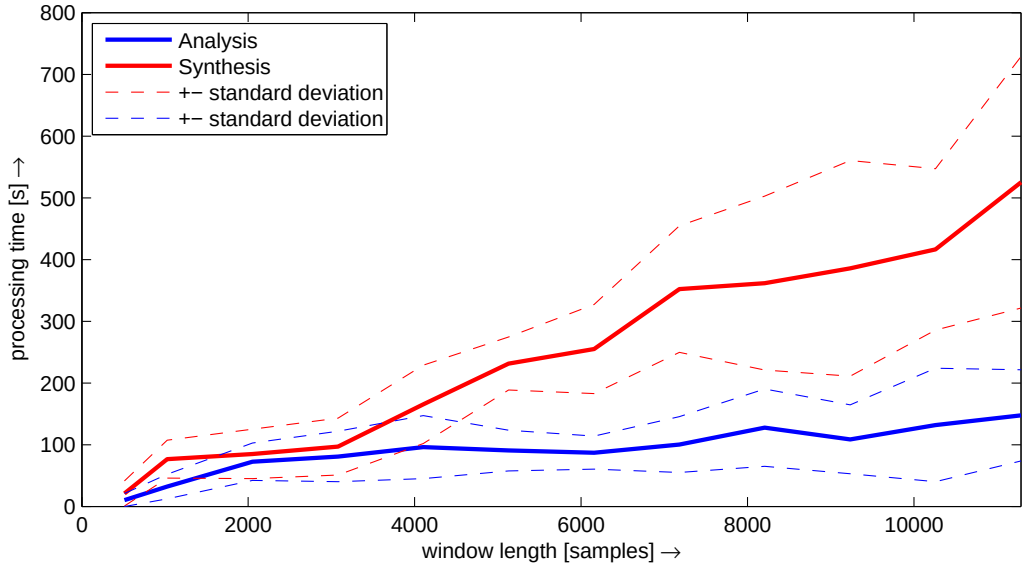


Fig. 7.33: Average inpainting processing time of file music11_16khz with gap size of 800 samples and various window length.

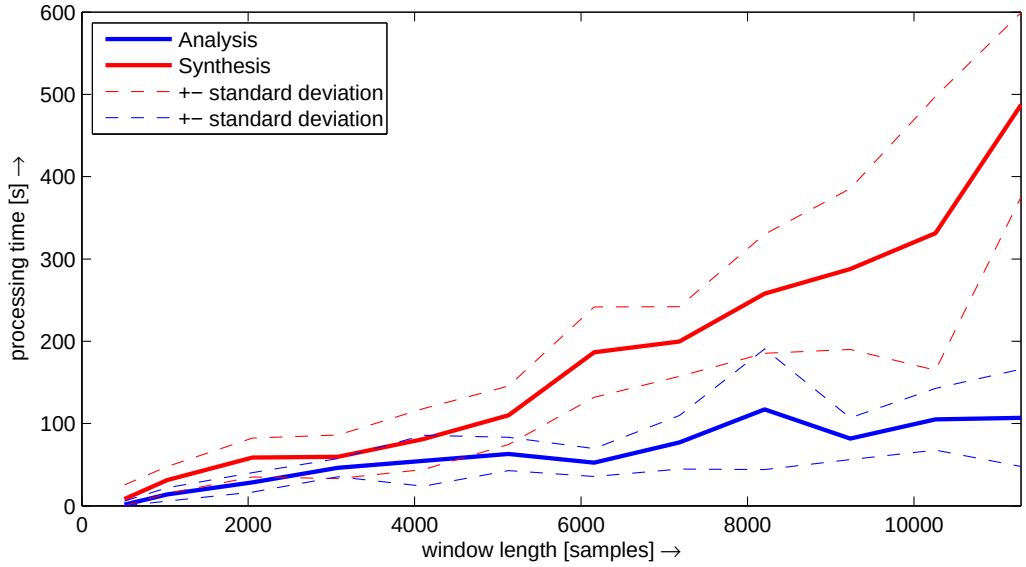


Fig. 7.34: Average inpainting processing time of file music11_16khz with gap size of 1600 samples and various window length.

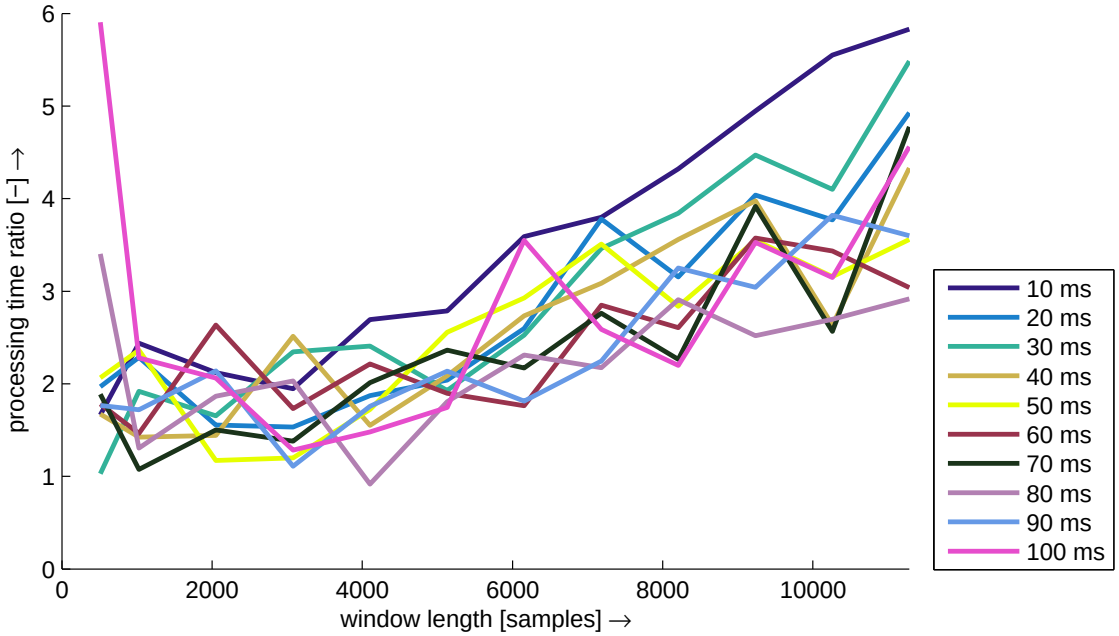


Fig. 7.35: Ratio of synthesis/analysis average inpainting processing time of file music11_16kHz.

same at first sight. Following the evolution of the curves, there are slight differences between the ratios of the processing time and number of iterations. Generally, the evolution is the same with the difference that the ratio of the number of iterations is roughly two times higher than the ratio of the processing time.

However, the interpretation of different processing times of analysis and synthesis model resulted in an almost different evolution trend of number of iterations of the proximal algorithm. The maximum value of iterations (the stopping criteria) was set to 2000 in all of the experiments. Detailed analysis of the *music11_16kHz* file (the one with the best SNR results) shows, that the stopping criterion of maximum iterations was fulfilled four times in the analysis model and six times in the synthesis model, respectively. Note that the number of experiments for each model is 1200. As a results in more than 99% of experiments the stopping criterion was the relative norm.

It was found that the size of the gap does not affect the number of iterations. On the other hand, the number of iterations is dependent on the window length, especially in the synthesis model.

In Fig. 7.37, 7.38 and 7.39 you can see the evolution of the number of iterations. In all of the cases, the number of iterations (averaged over 10 random experiments for each gap size and window length) in the synthesis model is significantly higher than in the analysis model.

Considering the synthesis model the trend of the number of iterations of the

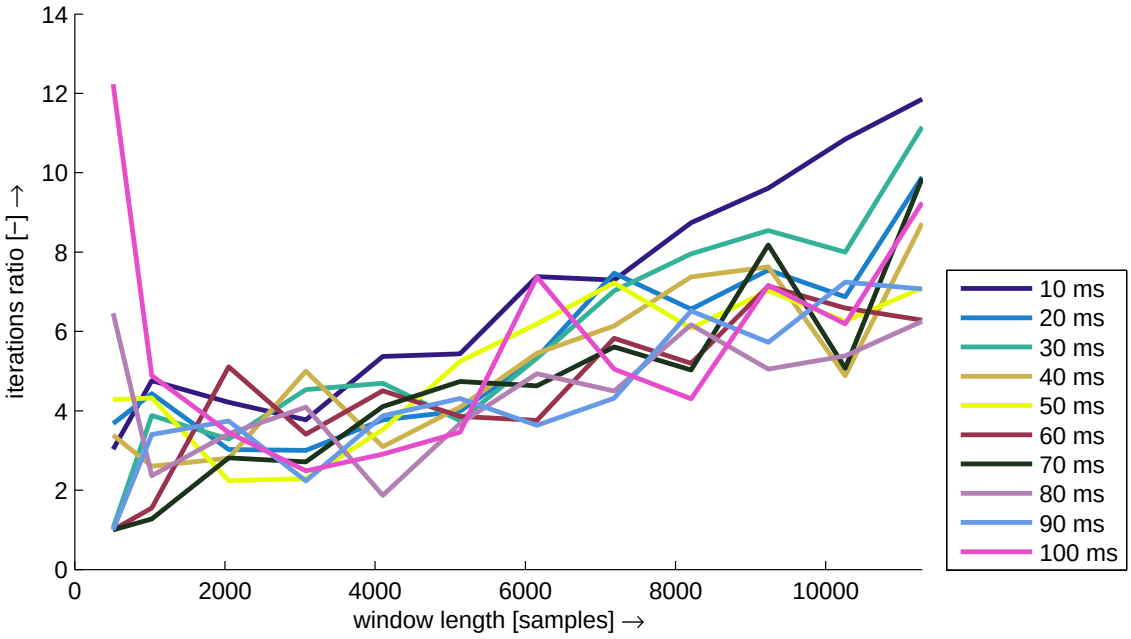


Fig. 7.36: Ratio of synthesis/analysis average proximal algorithm iterations of file music11_16kHz.

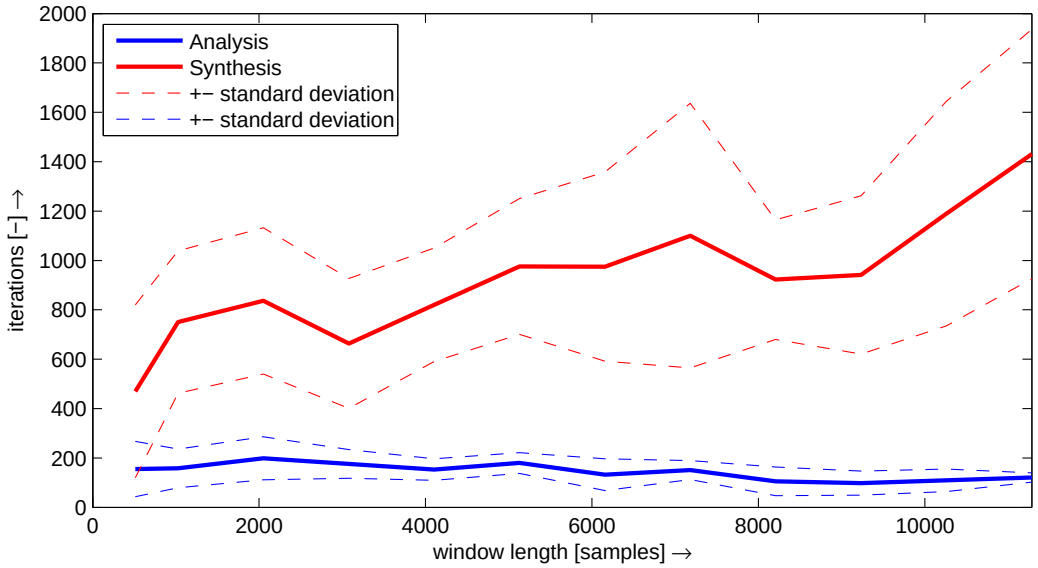


Fig. 7.37: Average inpainting iterations number of file music11_16khz with gap size of 160 samples and various window length.

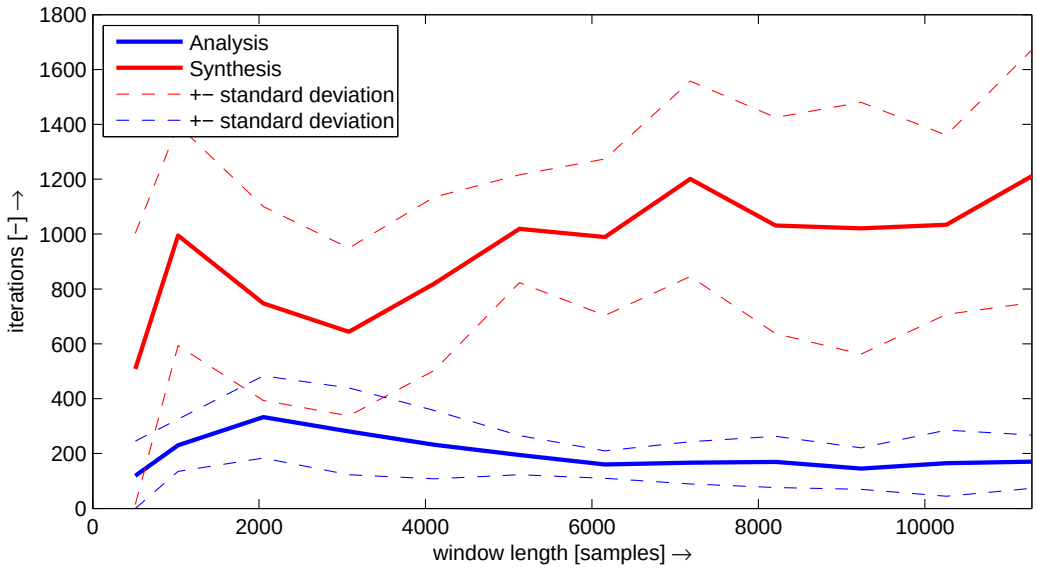


Fig. 7.38: Average inpainting iterations number of file music11_16khz with gap size of 800 samples and various window length.

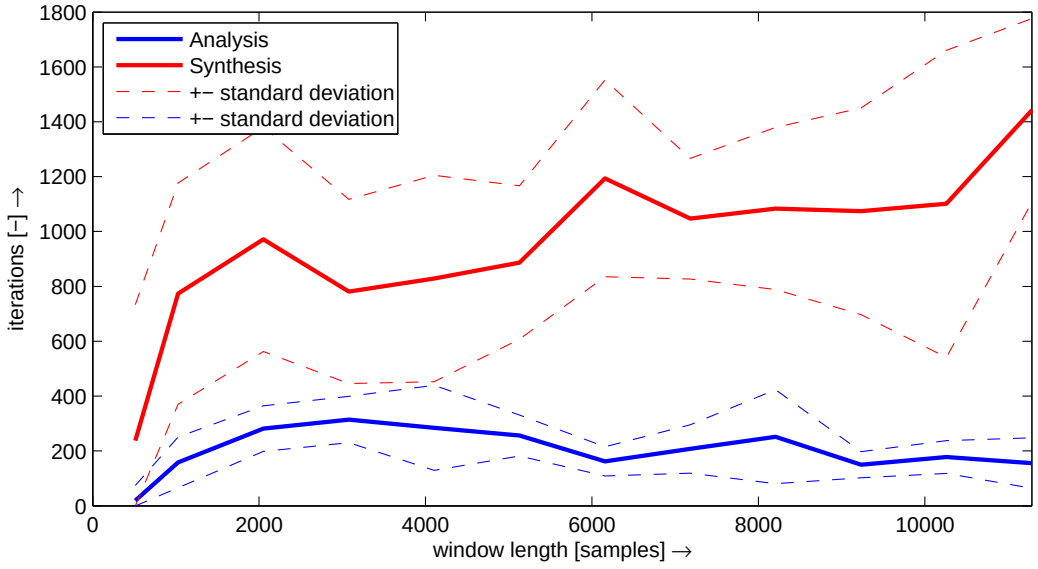


Fig. 7.39: Average inpainting iterations number of file music11_16khz with gap size of 1600 samples and various window length.

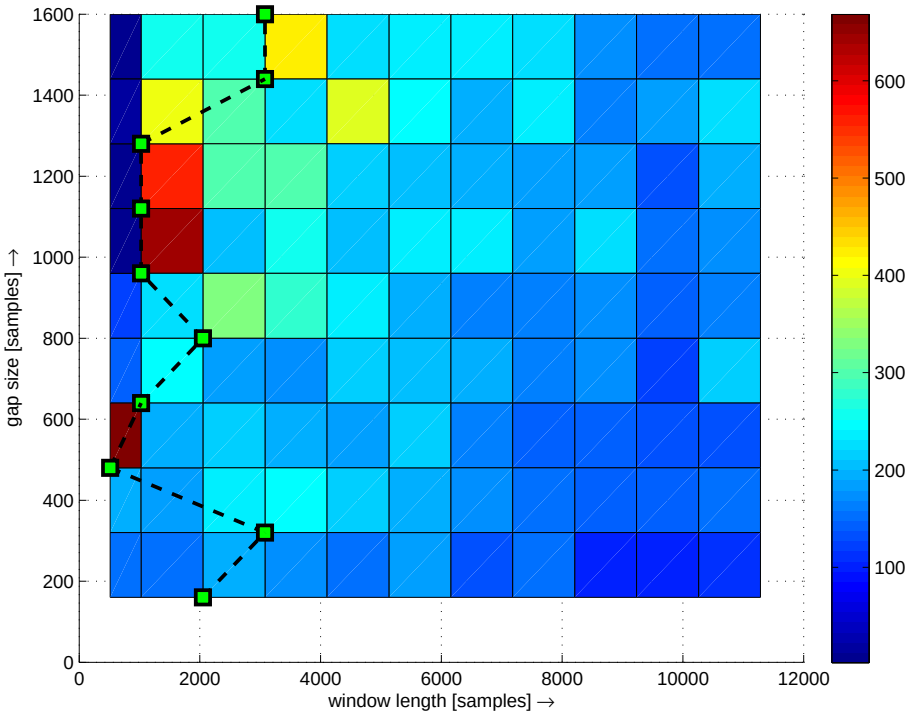


Fig. 7.40: Number of iterations of the proximal algorithm with various gap and window length in the analysis model. Dashed line with green points indicates the maximum values for each gap length.

proximal algorithm is growing with an increasing window length. The maximum number of averaged iterations is 1602 (gap size = 1120 samples, window length = 11286 samples).

In the analysis model, the number of iterations is slightly decreasing with the growing window length. The maximum number of averaged iterations is 668 (gap size = 480 samples, window length = 513 samples). Nevertheless, the processing time of both models is increasing as was shown in previous experiments.

On the other hand, the maximum number of iterations is completely different in both models. In the analysis model, the most iterations were performed using shorter windows over all gap sizes, mostly 1026 or 2052 samples (see Fig. 7.40). In the synthesis model instead, the number of iterations was highest using the longest windows, mostly 11286 samples (see Fig. 7.41). Green points in the figures are representing the maximum values for each gap size.

Recalling the relation of iterations number and SNR, these values seem to be completely independent. In Fig. 7.42 the graph of relative number of iterations and relative SNR illustrates that fact. The same unpredictable behaviour was observed through all other experiments.

Finally, in Fig. 7.43 you can see that the reconstructed signal using the analysis

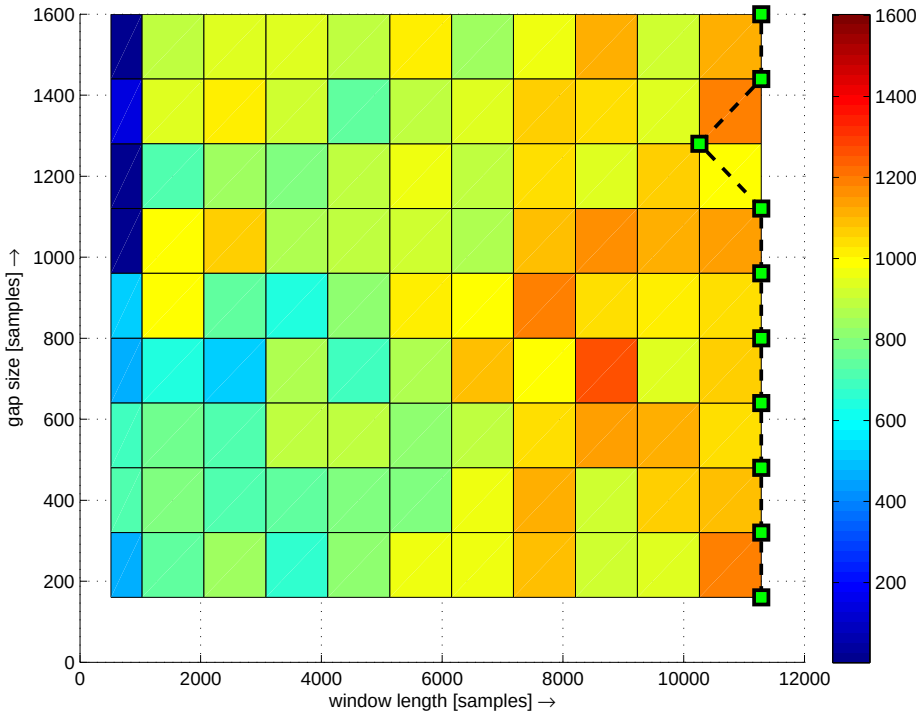


Fig. 7.41: Number of iterations of the proximal algorithm with various gap and window length in the synthesis model. Dashed line with green points indicates the maximum values for each gap length.

and synthesis model is almost the same as the original signal.

7.6.6 Structured sparsity

First experiments using the structured sparsity for audio inpainting were performed using the StrucAudioToolbox v. 02[10]. Comparing state-of-the-art methods with inpainting by ℓ_1-regression resulted towards modern inpainting methods with SNR more than 13 dB higher [7] [4]. For more detailed experiments the Windowed Group Lasso operator was reimplemented in the Brno-Wien Inpainting Toolbox connected to the LTFAT toolbox[11] especially because of efficient frames representation.

The core problem being solved is described in Eq. 5.14. The unconstrained version of the model (see Eq. 5.6) for solving the inpainting problem utilizes the penalty term λ which has to be experimentally set to produce the smallest possible error. On the other hand, the only way how to eliminate the deviation at all is to use the constrained formulation of the problem (see Eq. 5.3) and set $\delta = 0$.

Using the example gap position and size like in all other experiments the first parameter under investigation was the empirical Wiener exponent (introduced in

[10] `http://homepage.univie.ac.at/monika.doerfler/StrucAudioToolboxV02.rar`[87]
[11] `http://ltfat.sourceforge.net/`

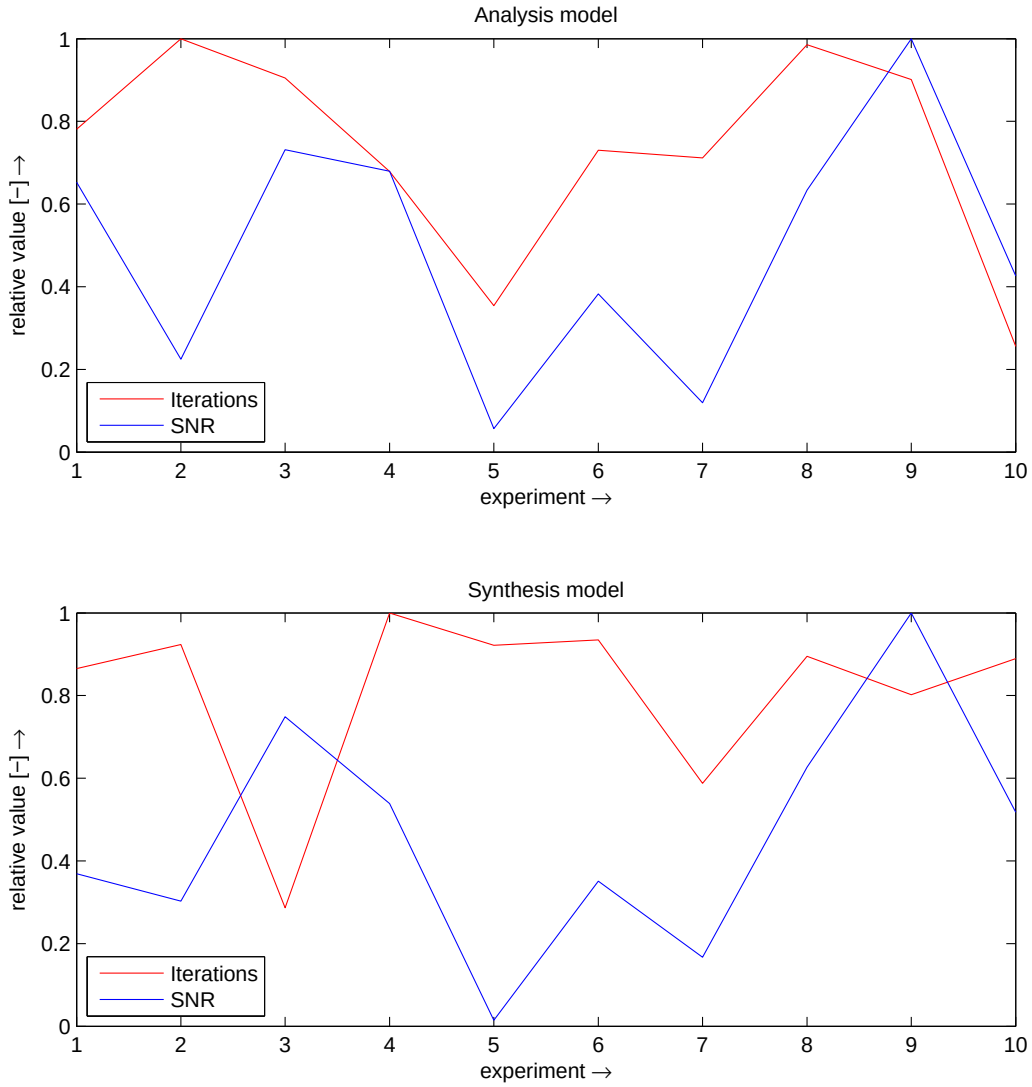


Fig. 7.42: Comparison of the relative SNR and number of iterations for file music11_16kHz, gap length of 960 samples, window lenght of 4104 samples.

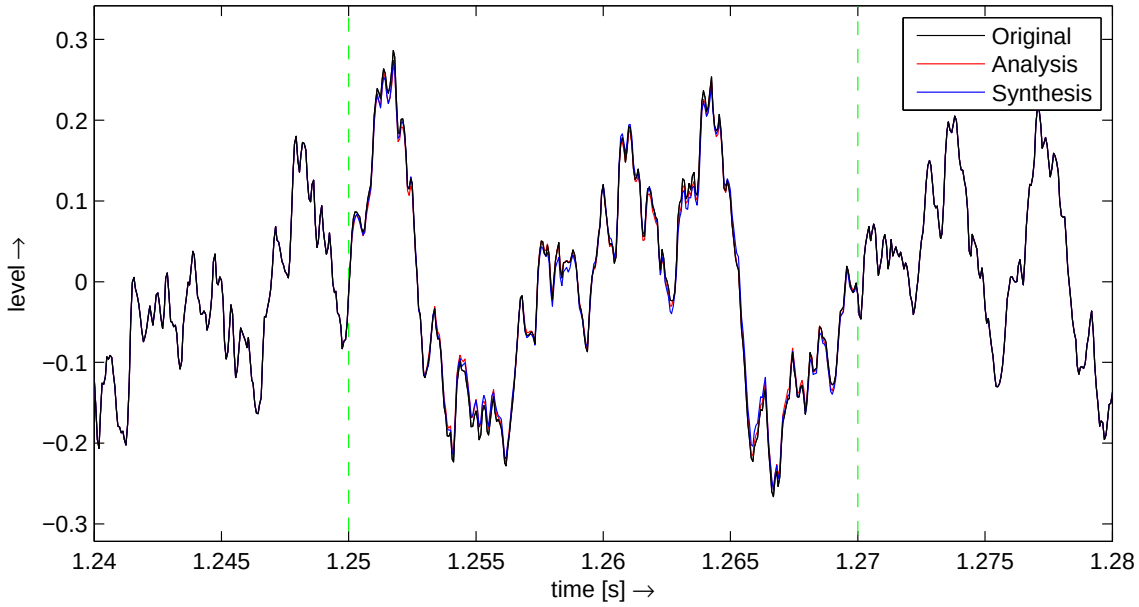


Fig. 7.43: Comparison of the signal reconstruction using analysis and synthesis model.

Eq. 5.16). Regarding Fig. 7.44 it is clear that the exponent set to 2 produces better results of inpainting. The best value of SNR = 21.17 dB was reached with $\lambda = 0.012$. Additionally, Fig. 7.45 shows that the processing time is not influenced by the selection of different exponent.

Using $\lambda = 0.012$ and expo = 2 the following batch experiment was performed. The size of 1D neighbourhood was selected from the range of $\{5, 10, \ldots, 50\}$ coefficients which corresponds to {4781, 8196, 11611, 15026, 18441, 21856, 25271, 28686, 32101, 35516} samples and the gap size was set to values of $\{10, 20, \ldots, 100\}$ ms which corresponds to $\{160, 320, \ldots, 1600\}$ samples. Each combination of such parameters was tested 10 times for an inpainting performance in the sense of SNR with different random gap position in file *music11_16kHz.wav*. Averaged results are illustrated in Fig. 7.46. It is obvious that inpainting performance decreases with increasing gap length independently of the neighbourhood size. Center of the neighbourhood was always set to the middle sample. On the other hand, the processing time increases with increasing neighbourhood size (see Fig. 7.47). One of possible outputs of this experiment is that it is not necessary to use neighbourhood of size larger than 10 coefficients.

For better illustration of power of the algorithm the example gap inpainting is in Fig. 7.48 in time and time-frequency representation. On the time plot there is almost no difference between the original and reconstructed signal. The spectral representation shows lower coefficients at higher frequencies, however, there is no

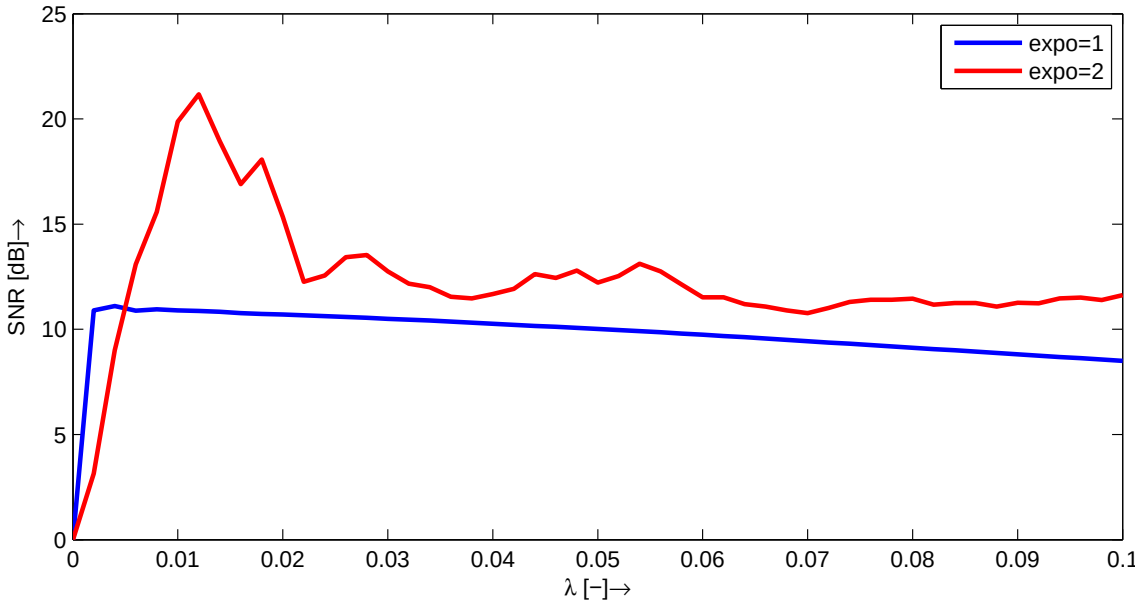


Fig. 7.44: SNR of inpainting by structured sparsity with various λ and exponent parameters.

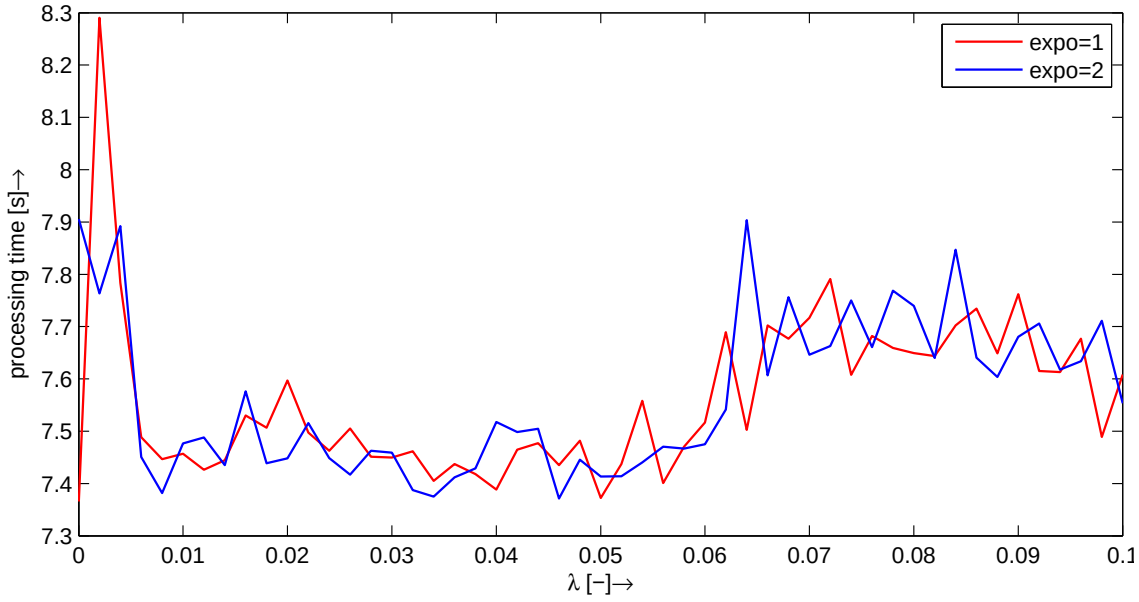


Fig. 7.45: Processing time of inpainting by structured sparsity with various λ and exponent parameters.

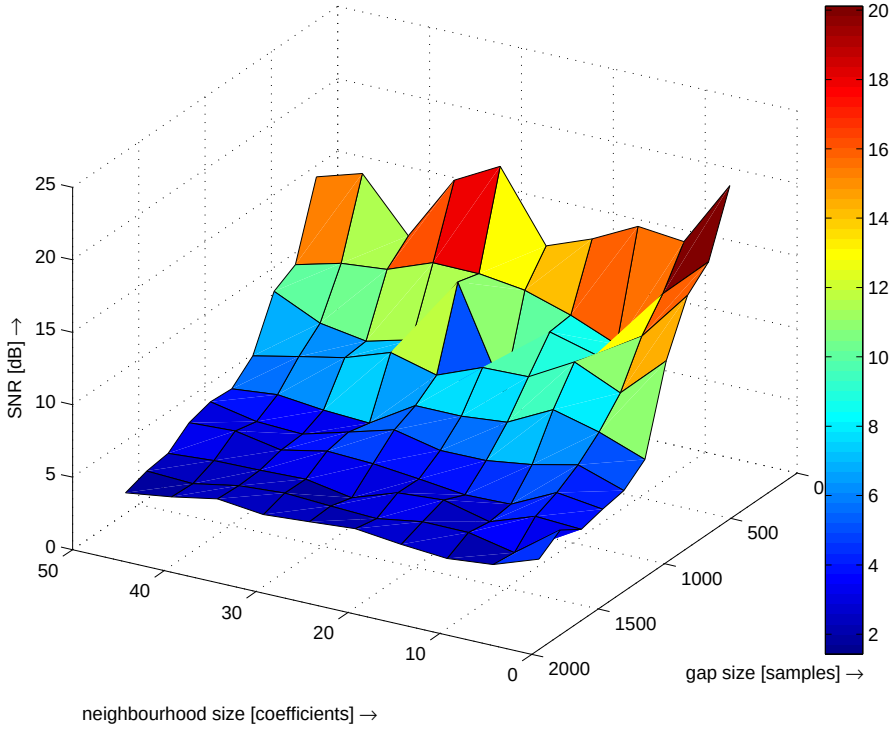


Fig. 7.46: SNR of inpainting by structured sparsity with various coefficients neighbourhood size and gap length.

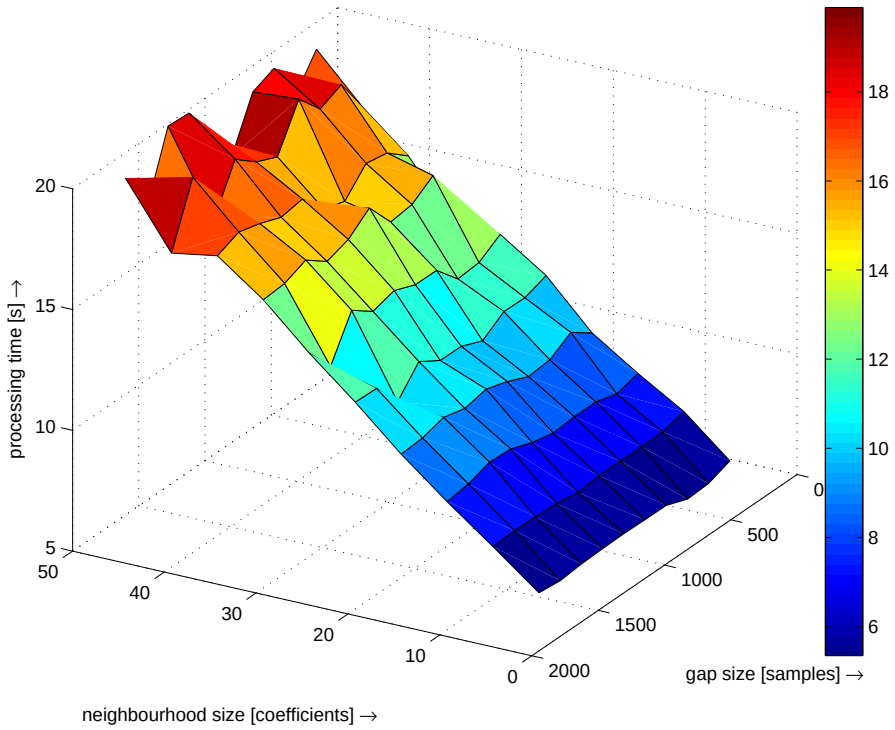


Fig. 7.47: Processing time of inpainting by structured sparsity with various coefficients neighbourhood size and gap length.

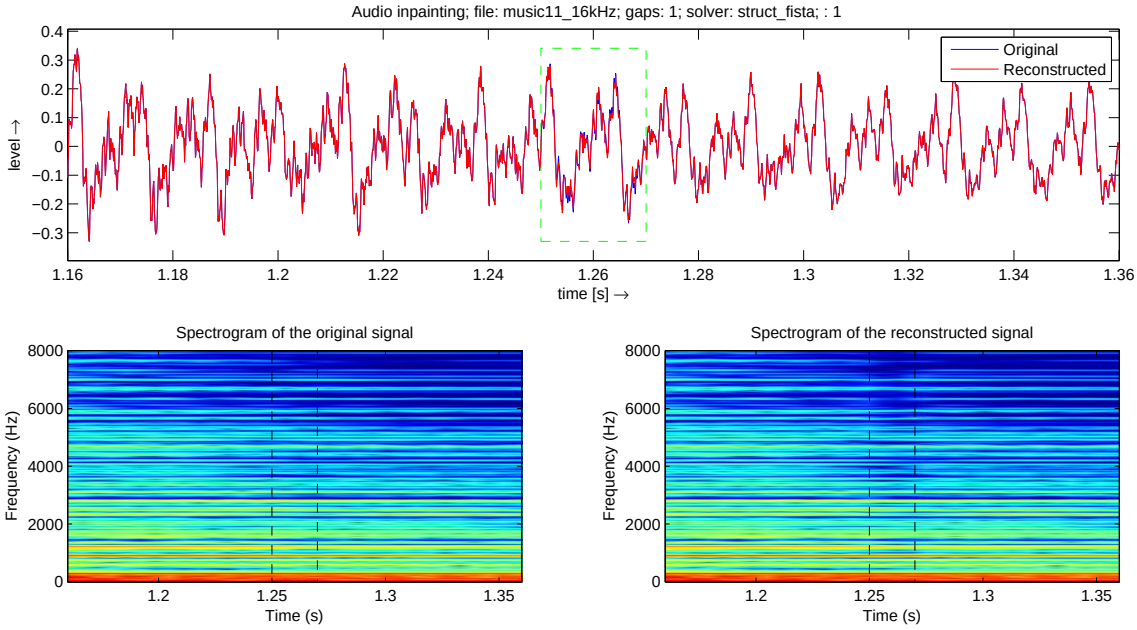


Fig. 7.48: Time and spectral representation of inpainting using the structured sparsity.

influence on the overal signal quality from the perceptual point of view and the reconstruction could be called perfect.

Audio Denoising

Another successful application of the structured sparsity is audio denoising [87] [84] [85]. As described in [2] very old recordings on wax cylinders contain periodical short-time distortions caused by fissures and mould of the material. First attempt for restoration of this kind of distortion was the process of audio inpainting. However, this approach is quite complicated because of appropriate error detection. Moreover the preliminary results were unsatisfactory.

Applying structured sparsity denoising on the digitized recordings of wax cylinders brought both successful elimination of short time distortions and reduction of the broadband noise. Music files for the experimental purposes were digitized from the original wax cylinders by The Phonogrammarchiv (Wien, Austria)[12]. The first comparator recording was captured by the Czech composer Alois Hába in 1931 and it contains the traditional music group from the East Moravian region. The second experimental recording with its restoration results is presented online on the

[12]The Phonogrammarchiv - The Austrian Audiovisual Research Archive, `http://www.phonogrammarchiv.at/`

webpage[13]. It contains a woman voice singing a traditional song that was captured in 1910 by the famous Czech composer Leoš Janáček and his collaborators. The cylinders were digitized into `.wav` files with sampling rate $f_s = 96\,\text{kHz}$ and bit depth of 24 bits.

Experiments were performed by the StrucAudioToolbox v. 02[14] and some cylinders under investigation were denoised such that only the tonal components remained and the noise was absolutely suppressed [5]. Parameters utilized for these experiments are not supposed to be used as a general rule because of the high non-stationarity of the background noise. Especially λ and neighbourhood size have to be set up very carefully depending on the signal character. The preprocessing had to be performed on the original file to suppress the strongest interferences such as crackles or low frequency rumbling using the Izotope RX II Advanced[15].

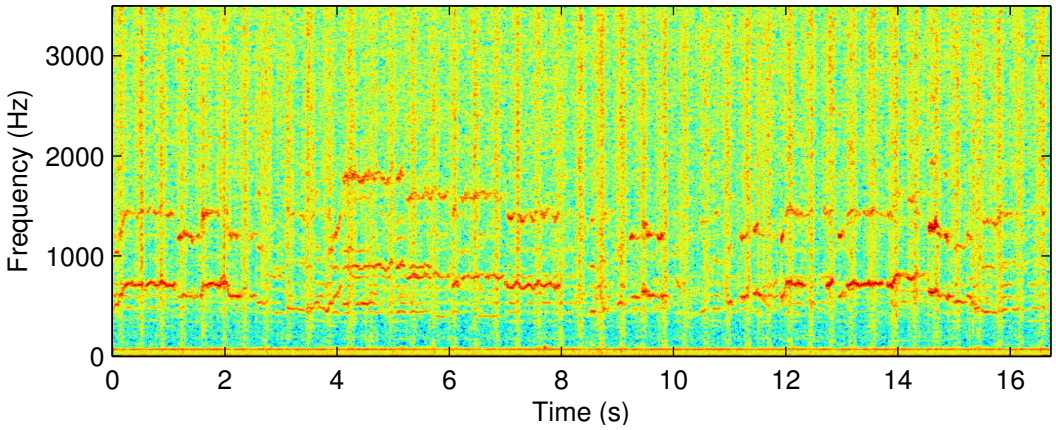


Fig. 7.49: Spectrogram of the original instrumental recording

Spectrogram of the original signal is depicted in Fig. 7.49. A lot of broadband noise, especially periodical waves of increasing noise energy are clearly visible. The frequencies above the limit of 3.5 kHz contain only noise without any harmonics and were suppressed by the lowpass filter.

Signal restoration shown in Fig. 7.50 was performed with the structured sparsity framework. All the parameters were setup according to Tab. 7.9 and the value $\lambda = 0.02$ was chosen experimentally. The consequently created Gabor frame is a Parseval tight frame. Higher value of λ produces stronger denoising, lower value preserves a lot of short-time harmonics in the whole spectrum. Fig. 7.51 shows the results of the denoising using RX software Denoiser with Musical Noise Supression (MNS) Simple Algorithm [63]. Attenuation of the broadband noise produced the best results with

[13] `http://www.utko.feec.vutbr.cz/~machv/2015_denoising_icumt/`

[14] `http://homepage.univie.ac.at/monika.doerfler/StrucAudioToolboxV02.rar`

[15] `https://www.izotope.com/en/products/audio-repair/rx`, v. 2.10.652

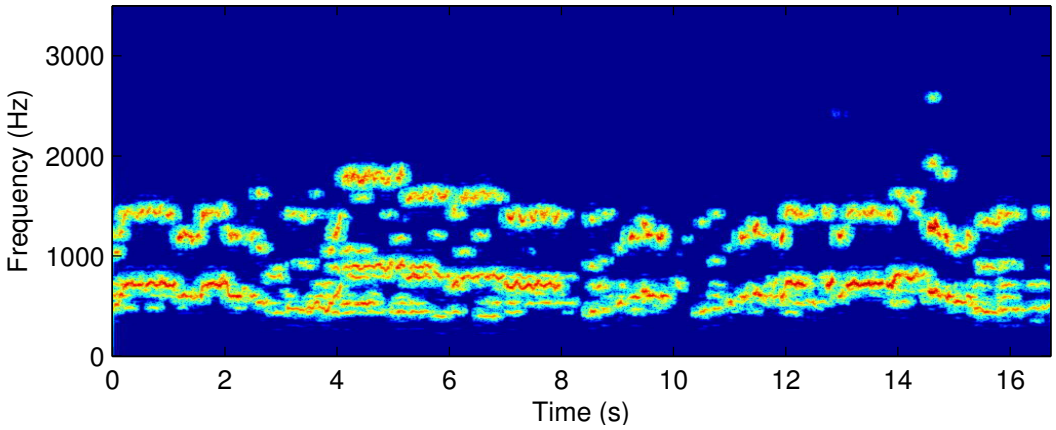


Fig. 7.50: Reconstruction by the structured sparsity

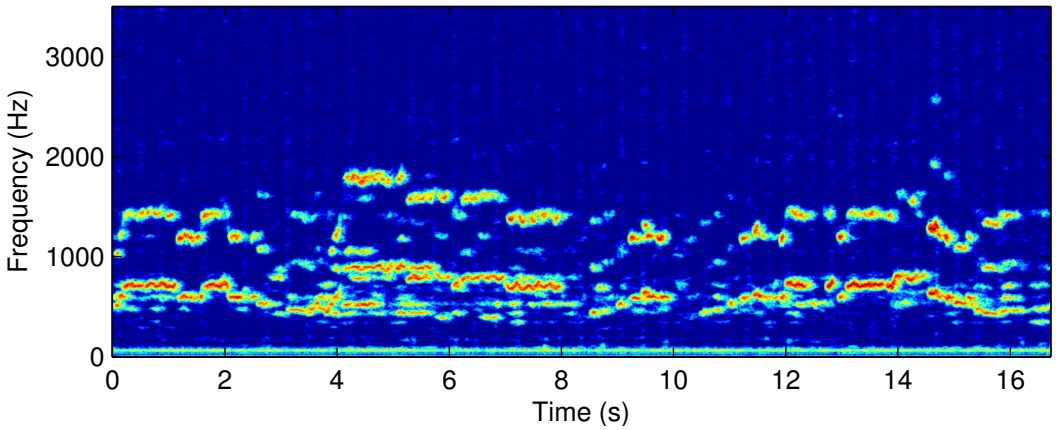


Fig. 7.51: Reconstruction by the professional software

parameter Reduction=29.2 dB and Threshold=−6.0 dB. The algorithm successfully suppressed the noise while useful harmonics remained. Some of the harmonics are slightly suppressed, however, this is hardly recognisable during listening. Setting a higher value of the *Reduction* parameter resulted in visible attenuation of harmonics. The drawback of this process was a low frequency noise (hum) which can be simply removed by the Low-Pass filter, of course. The very similar results were obtained choosing the Advanced, Extreme or combined algorithm MNS Algorithm. It is worth noting that the most time consumptive algorithm was the Advanced MNS.

It was clearly audible that the short tones from the melody were suppressed too much (e.g. between the time of 8 and 8.5 s). Generally, an instrumental music typical for The Moravia region contains more short time tones than the voice singing, therefore, the level of *Reduction* value had to be weakened (down to aprox. 20 dB) to preserve the short tones.

All the resulting experimental `.wav` files together with the detailed reconstruc-

Tab. 7.9: Structured sparsity parameters

Dictionary	Gabor
Window length	2304 samples
Window overlap	75 % of window length
Frequency channels	2304
Shrinkage type	WGL
λ	variable
Neighbourhood	5×50 coefs. ≈ 0.32 s
Center sample	[3,25]
Exponent α	$2 \approx$ Wiener

tion parameters and spectrograms are available on the web page already mentioned above.

7.7 Final evaluation and chapter summary

Using the best parameters for each of the methods, the audio inpainting / interpolation of the example gap was performed: file *music11_16kHz.wav*, gap starts on sample number 20 000 and is 20 ms long. All of the resulting music files with detailed parameters list, time plot and spectrograms are available on enclosed CD. In Tab. 7.10 results of the objective evaluation are presented.

Comparing the evaluation by SNR shows that the best reconstruction of the example gap was reached by ℓ_1-relaxation using the analysis model. The objective evaluation also corresponds to a non-official listening test where no difference between the original and reconstructed music sample is recognisable. The same result from the subjective point of view goes to another two methods: ℓ_1-relaxation using the synthesis model and structured sparsity. Both of them also reached SNR higher than 20 dB.

Very similar listening result of the OMP algorithm shows, that sparse representations definitely push the fruitfulness of the audio inpainting methods forwards.

Methods using AR signal or parameters modeling produce results slightly higher than 0 dB. More successful in terms of SNR were WFBI and LSRI with SNR close to 4 dB. Listening to the results that do not produce any disturbing noise, however, there is an evident decay of signal energy in the reconstructed signal portion. A slightly longer gap is recognisable during listening to inpainting experiment of the Sinusoidal modeling method.

The worst results were reached by the samples repetition method. This corre-

Tab. 7.10: Complex evaluation of inpainting/interpolation algorithms

Method name	SNR [dB]	PSM [-]	PSM_t [-]
Samples repetition	−3.9338	0.9839	0.5118
AR samples modeling (LSRI)	3.9738	0.9958	0.8312
AR samples modeling (WFBI)	3.9526	0.9971	0.9108
AR sinusoidal modeling	0.7873	0.9953	0.7330
Greedy algorithm (OMP)	13.7471	0.9980	0.9202
ℓ_1-relaxation (analysis model)	25.5870	0.9999	0.9961
ℓ_1-relaxation (synthesis model)	23.6777	0.9998	0.9964
ℓ_1-relaxation (structured sparsity)	21.1735	0.9997	0.9873

sponds to the disturbing experience from the listening to the reconstructed audio file.

Another objective evaluation method, the PEMO-Q (presented in Sec. 7.1.2) was performed on the resulting files. The demo version of this algorithm provides evaluation of only a single music sample of the maximum length of 4 seconds and every experiment is restricted by filling in a captcha code. Therefore, evaluation of larger set of data would be not feasible. While there is only one gap in the performed experiments, evaluation of the full length file (4 seconds) is predicative. The signal under PEMO-Q test is limited on samples no. 15000 to 25000 (approx. 0.6 seconds).

Results evaluated by PSM score with values almost at the maximum of 1 represent nearly similarity with the original signal. This is quite misleading because this similarity measure is not useful for evaluation of such a short disturbance (320 samples) in severalfold longer file. The second score (PSMt) is more suitable for evaluating of this kind of imperfection. Scores reached by this evaluation method correspond to the SNR results in almost all cases. The only exceptions is the result of the WFBI method which is higher than the LSRI compared to the difference between their SNR results.

From these results and all the previous experiments, it is obvious that results obtained from methods utilizing the sparse representations provide better refilling of the gap of the signal samples in audio files.

8 CONCLUSION

This thesis dealt with the process of audio restoration, especially the interpolation of missing data segments. In the beginning the most common types of damage to historical or current audio recordings were described. Short time interruptions like clicks, crackles or signal gaps are nowadays successfully treated by interpolation methods based on samples repetition or autoregressive modeling of either signal samples or signal parameters. These state-of-the-art methods are presented in Chapter 2.

Recently, sparse representations of signals brought novel approaches of signal analysis and synthesis and naturally penetrated into the field of audio processing. The process of signal interpolation using overcomplete dictionaries was termed the Inpainting and the main goal of this thesis was to explore these new techniques, find possible ways of improvement and compare them to the state-of-the-art methods. Necessary theoretical aspects like vectors, norms, basis and frames were described in Chapter 3 and connected to the topic of sparse representations in Chapter 4. Since the solution of overcomplete systems is not unique, possible solutions of these tasks are computed by approximation algorithms described in Chapter 5.

In the beginning of this research, the presumptions were that the process of Audio Inpainting will be feasible especially for harmonic signals and the restoration process will be most efficient in shorter gap sizes. As will be described further, these prepositions were proven by experiments.

Regarding the simplest method for interpolation, the Weighted Repetitive Substitution, both objective evaluation and subjective listening to the results is not satisfying and the listener can clearly recognise the gap position in the sound. A naturally arising question is whether this kind of interpolation method is really useful because if the signal gap was filled with zero samples the result sounds more naturally compared to the interpolation result. The only possible reason of utilizing this method for particular audio signal interpolation is the speed of computation, because the algorithm is very fast and feasible for real-time applications.

Interpolation using the AR modeling of signal samples brought improvement of the restoration in terms of SNR. The best average result of SNR = 5.78 dB was reached for a gap of length 360 samples using an AR model of order 3600. However, the standard deviation of 14 dB makes the results very unstable. The computational time of larger model orders (larger than 10 times the signal gap) makes this method unusable in real experiments because reconstruction of a single gap takes more than 1000 seconds.

Exploring the most important parameters of sinusoidal modeling resulted in an optimal setting of frequency threshold thrF = 3, amplitude threshold thrA = 0.5,

length of the vector for amplitude mean value computation $M = 60$. There are no general recommendations for selection of the core parameter, the model order, according to the gap length since the values did not show any regular evolution using various gap length and model order. Maximum average SNR did not exceed the value of 3 dB.

The Orthogonal Matching Pursuit (greedy algorithm) reached the best values of SNR = 15.64 dB for redundancy of dictionary of 9. However, the processing time of such redundancy factor is quite long, therefore, the optimal redundancy factor is red = 3. Batch experiments resulted in the highest average SNR = 7.59 dB obtained for gap size of 10 ms and neighbourhood size factor of 8 which results in the neighbourhood size of 1280 samples (80 ms). The optimal trade-off between the speed of computation and the resulting inpainting performance using OMP algorithm should be the neighbourhood size factor of at least 4.

Main contribution of this thesis is the experimental verification of audio inpainting utilizing ℓ_1-relaxation algorithms. Considering single coefficient sparsity (without relation to its neighbourhood), both synthesis and analysis model were implemented. Note that until now there was no scientific contribution on analysis model implementation for Audio Inpainting.

Fixing the atom length of the dictionary, results of the analysis model reached slightly higher SNR values than the synthesis, both with best average SNR higher than 20 dB. Further, the standard deviation of the synthesis model is the same or higher in all of the experiments compared to the analysis model standard deviation. There are some optimal parameters for a particular gap size and gap position, nevertheless, they can be very easily missed by a slight modification of a window length. This kind of unstable behaviour was observed in almost all other experimental sound pieces with harmonic structure.

Regarding the speed of computation, the analysis model is about 2 to 4 times faster than the synthesis whereas the number of iterations of the synthesis model was from 4 to 8 times higher than of the analysis model. Another remark from this experiment is that the average computational time of the inpainting process using a proximity algorithm is not dependent on the gap size. On the other hand, the standard deviation of the computational time is slightly increasing with the growing gap size. The number of iterations is dependent on the window length, especially in the synthesis model.

The highest SNR has been reached in files containing a harmonic signal, especially when only a single instrument is playing. The worst results were obtained with rather non-harmonic records containing speech or completely non-harmonic signals.

The structured sparsity for audio inpainting evaluated by SNR produced restoration results comparable to the ℓ_1-relaxation without structure. Looking at the re-

sulting time plot of the signal, there is almost no visible difference between the original and reconstructed signal. The restoration could be called perfect for small gap sizes up to 500 samples. Moreover, such comparable results were reached in a shorter time period. Finally, denoising using the structured sparsity outperformed professional software and was successfully utilized for denoising of recently found wax cylinders recordings.

This thesis proves that audio restoration could profit from sparse representations in terms of restoration quality. However, there is a long way from the theory to the real audio engineering field mainly because of the efficient implementation and optimization. Further research in this field could be focused on content based audio inpainting [14].

Like in all fields of research, new unanswered questions are arising from every answered query. There is great opportunity for the success of new methods. The ideas and results presented in this thesis are a step to contributing in this never ending journey.

BIBLIOGRAPHY OF THE AUTHOR

[1] Kurc, D., Mach, V., Orlovsky, K., and Khaddour, H. Sound Source Localization with DAS Beamforming Method Using Small Number of Microphones. In *36th International Conference on Telecommunications and Signal Processing (TSP)* (Rome, 2013), Brno University of Technology, pp. 526–532.

[2] Mach, V. Digital restoration of recordings from the phonograph cylinders and their copies. In *As recorded by the phonograph: Slovak and Moravian songs recorded by Hynek Bím, Leoš Janáček and Františka Kyselková in 1909–1912* (Brno, 2012), The Institute of Ethnology of the Academy of Sciences of the Czech Republic, v.v.i., pp. 165–176.

[3] Mach, V. Digitální restaurace záznamů z janáčkovských fonografických válců a jejich kopií. In *Vzaty do fonografu: slovenské a moravské písně v nahrávkách Hynka Bíma, Leoše Janáčka a Františky Kyselkové z let 1909–1912* (Brno, 2012), Etnologický ústav AV ČR, v.v.i., pp. 153–163.

[4] Mach, V. Structured Sparsity for Audio Inpainting. In *Proceedings of the 20th Conference STUDENT EEICT 2014* (Brno, 2014), pp. 41–45.

[5] Mach, V. Denoising Phonogram Cylinders Recordings Using Structured Sparsity. In *7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)* (2015), pp. 314–319.

[6] Mach, V., and Ozdobinski, R. Optimizing dictionary learning parameters for solving audio inpainting problem. *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems 2*, 1 (2013), 40–45.

[7] Mach, V., and Wiesmeyr, C. Methods of Completing Missing Samples in Audio Signal. In *Conference Proceedings: International Masaryk Conference for Ph.D. Students and Young Researchers 2013* (Hradec Kralove, 2013), MAGNANIMITAS, p. 3812–3820.

[8] Procházková, J., and Mach, V. Editace zvukových záznamů z fonografických (voskových) válečků uložených na brněnském pracovišti Etnologického ústavu AV ČR. In *Vzaty do fonografu: slovenské a moravské písně v nahrávkách Hynka Bíma, Leoše Janáčka a Františky Kyselkové z let 1909–1912* (Brno, 2012), Etnologický ústav AV ČR, v.v.i., pp. 187–193.

[9] Procházková, J., and Mach, V. The Editing of Sound Recording from Phonograph (Wax) Cylinders at the Brno Branch of the Institute of Ethnology of the ASCR. In *As recorded by the phonograph: Slovak and Moravian songs*

recorded by Hynek Bím, Leoš Janáček and Františka Kyselková in 1909–1912 (Brno, 2012), The Institute of Ethnology of the Academy of Sciences of the Czech Republic, v.v.i., pp. 199–206.

REFERENCES

[10] Interoperability Standards for VoIP ATM Components ED-137, 2012.

[11] Adler, A., Emiya, V., Jafari, M., Elad, M., Gribonval, R., and Plumbley, M. A constrained matching pursuit approach to audio declipping. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on* (2011), pp. 329–332.

[12] Adler, A., Emiya, V., Jafari, M., Elad, M., Gribonval, R., and Plumbley, M. Audio Inpainting. *IEEE Transactions on Audio, Speech, and Language Processing 20*, 3 (March 2012), 922–932.

[13] Aharon, M., Elad, M., and Bruckstein, A. M. K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations. *IEEE Transactions on Signal Processing 54* (2006), 4311–4322.

[14] Bahat, Y., Schechner, Y. Y., and Elad, M. Self-content-based audio inpainting. *Signal Processing 111*, 0 (2015), 61–72.

[15] Balazs, P., Doerfler, M., Kowalski, M., and Torresani, B. Adapted and adaptive linear time-frequency representations: A synthesis point of view. *Signal Processing Magazine, IEEE 30*, 6 (Nov 2013), 20–31.

[16] Balazs, P., Dörfler, M., Jaillet, F., Holighaus, N., and Velasco, G. Theory, implementation and applications of nonstationary Gabor frames. *Journal of computational and applied mathematics 236*, 6 (2011), 1481–1496.

[17] Barchiesi, D. *Sparse Approximation and Dictionary Learning with Applications to Audio Signals*. PhD thesis, Queen Mary University of London, 2013.

[18] Bayram, I., and Kamasak, M. A simple prior for audio signals. *IEEE Transactions on Acoustics Speech and Signal Processing 21*, 6 (2013), 1190–1200.

[19] Beck, A., and Teboulle, M. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences 2*, 1 (2009), 183–202.

[20] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning 3*, 1 (2011), 1–122.

[21] Bradley, K., Casey, M., Cavaglieri, S. S., Clark, C., Davies, M., and Frilander, J. *Guidelines on the Production and Preservation of Digital Audio Objects*, second ed. International Association of Sound and Audio Visual Archives, 2009.

[22] Bruckstein, A. M., Donoho, D. L., and Elad, M. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review 51*, 1 (2009), 34–81.

[23] Chen, S., Donoho, D., and Saunders, M. *Atomic decomposition by basis pursuit*. SIAM J. Sci Comput. 20 (1998), no.1, reprinted in SIAM Review, 2001.

[24] Christensen, O. *An Introduction to Frames nad Riesz Bases*. Birkhäuser, Boston-Basel-Berlin, 2003.

[25] Christensen, O. *Frames and Bases, An Introductory Course*. Birkhäuser, Boston, 2008.

[26] Combettes, P., and Pesquet, J. A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing 1*, 4 (2007), 564–574.

[27] Combettes, P., and Pesquet, J. Proximal splitting methods in signal processing. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering* (2011), 185–212.

[28] Combettes, P., and Wajs, V. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation 4*, 4 (2005), 1168–1200.

[29] Damnjanovic, I., Davies, M., and Plumbley, M. D. Sparse Representations and Dictionary Learning Evaluation Framework – SMALLbox. In *SMALL Workshop on Sparse Dictionary Learning* (London, United Kingdom, Jan. 2011).

[30] Daubechies, I. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.

[31] Deller, J., Hansen, J., and Proakis, J. *Discrete-Time Processing of Speech Signals*. An IEEE Press classic reissue. Wiley, 2000.

[32] Donoho, D. L., and Stark, P. B. Uncertainty principles and signal recovery. *SIAM J. Appl. Math. 48*, 3 (1989), 906–931.

[33] Elad, M. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[34] Elad, M., Milanfar, P., and Rubinstein, R. Analysis versus synthesis in signal priors. In *Inverse Problems 23 (200)* (2005), pp. 947–968.

[35] Elad, M., Starck, J., Querre, P., and Donoho, D. Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). *Applied and Computational Harmonic Analysis 19*, 3 (2005), 340–358.

[36] Ellis, D. P. W. Sinewave and sinusoid+noise analysis/synthesis in Matlab, 2003. Online web resource.

[37] Engan, K., Aase, S., and Hakon Husoy, J. Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on* (1999), vol. 5, pp. 2443 –2446 vol.5.

[38] Engan, K., Rao, B. D., and Kreutz-Delgado, K. Frame design using FOCUSS with method of optimal directions (MOD). In *Proc. NORSIG 1999, Tronheim, Norway* (June 1999), pp. 65–69.

[39] Etter, W. Restoration of a discrete-time signal segment by interpolation based on the left-sided and right-sided autoregressive parameters. *IEEE Transactions on Signal Processing 44*, 5 (1996), 1124–1135.

[40] Feichtinger, H. G., and Strohmer, T. *Advances in Gabor Analysis*. Birhäuser, 2001.

[41] Fink, M., Holters, M., and Zölzer, U. Comparison of Various Predictors for Audio Extrapolation. In *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13)* (Maynooth, 2013), pp. 1–7.

[42] Gersho, A., and Gray, A. *Vector Quantization and Signal Compression*. The Kluwer International Series in Engineering and Computer Science. Springer-Verlag GmbH, 1992.

[43] Godsill, S., Rayner, P., and Cappé, O. Digital audio restoration. *Applications of digital signal processing to audio and acoustics* (2002), 133–194.

[44] Goodman, D., Lockhart, G. B., Wasem, O., and Wong, W.-C. Waveform substitution techniques for recovering missing speech segments in packet voice communications. *IEEE Transactions on Acoustics, Speech and Signal Processing 34*, 6 (Dec 1986), 1440–1448.

[45] Gorodnitsky, I. F., and Rao, B. D. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing 45*, 3 (1997), 600–616.

[46] Henzlová, P. Přehled metod komprimovaného snímání. Bakalářská práce, ČVUT, 2010.

[47] Hrbacek, R., Rajmic, P., Vesely, V., and Spirik, J. Sparse signal representations: an introduction. *Elektrorevue – the online journal* (2011), 1–10. In Czech.

[48] Hrbáček, R., Rajmic, P., Veselý, V., and Špiřík, J. Řídké reprezentace signálů: úvod do problematiky. *Elektrorevue – Internetový časopis* (2011), 1–10.

[49] Huber, R., and Kollmeier, B. PEMO-Q—A new method for objective audio quality assessment usinga model of auditory perception. *IEEE Trans. Audio Speech Language Proc. 14*, 6 (November 2006), 1902–1911.

[50] Jan, J. *Číslicová filtrace, analýza a restaurace signálů, (in Czech)*. VUTIUM, 2002.

[51] Janssen, A. J. E. M. From continuous to discrete Weyl-Heisenberg frames through sampling. *J. Fourier Anal. Appl. 3*, 5 (1997), 583–596.

[52] Janssen, A. J. E. M., Veldhuis, R. N. J., and Vries, L. B. Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes. *IEEE Trans. Acoustics, Speech and Signal Processing 34*, 2 (4 1986), 317–330.

[53] Karahanoglu, N. B., and Erdogan, H. A* orthogonal matching pursuit: Best-first search for compressed sensing signal recovery. *Elsevier Digital Signal Processing* (2011).

[54] Kereliuk, C. *Sparse and structured atomic modelling of audio*. PhD thesis, McGill University, Montreal, 2012.

[55] Kereliuk, C., Depalle, P., and Pasquier, P. Audio interpolation and morphing via structured-sparse linear regression. In *Proceedings of the Sound and Music Computing Conference 2013* (Stockholm, 2013), pp. 546–552.

[56] Kitić, S., Bertin, N., and Gribonval, R. Audio declipping by cosparse hard thresholding. In *2nd Traveling Workshop on Interactions between Sparse models and Technology* (2014).

[57] Škopík, M. *Digitální konverze a archivace audiovizuálních a fotografických archivů*. Národní ústav lidové kultury, 2010.

[58] Kowalski, M., Siedenburg, K., and Dörfler, M. Social Sparsity! Neighborhood Systems Enrich Structured Shrinkage Operators. *Signal Processing, IEEE Transactions on 61*, 10 (2013), 2498–2511.

[59] Kowalski, M., and Torrésani, B. Sparsity and persistence: mixed norms provide simple signal models with dependent coefficients. *Signal, Image and Video Processing 3*, 3 (2009), 251–264.

[60] Kowalski, M., and Torrésani, B. Structured Sparsity: from Mixed Norms to Structured Shrinkage. In *SPARS'09 – Signal Processing with Adaptive Sparse Structured Representations* (2009), R. Gribonval, Ed., Inria Rennes – Bretagne Atlantique, pp. 1–6.

[61] Lagrange, M., Marchand, S., and Rault, J.-b. Long interpolation of audio signals using linear prediction in sinusoidal modeling. *J. Audio Eng. Soc 53*, 10 (2005), 891–905.

[62] Lesage, S., Gribonval, R., Bimbot, F., and Benaroya, L. Learning unions of orthonormal bases with thresholded singular value decomposition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05).* (march 2005), vol. 5, pp. v/293 – v/296 Vol. 5.

[63] Lukin, A., and Todd, J. Suppression of musical noise artifacts in audio noise reduction by adaptive 2-D filtering. In *Audio Engineering Society Convention 123* (Oct 2007).

[64] Lukin, A., and Todd, J. Parametric interpolation of gaps in audio signals. In *Audio Engineering Society Convention 125* (Oct 2008), pp. 3–6.

[65] Mailhe, B., Barchiesi, D., and Plumbley, M. Ink-svd: Learning incoherent dictionaries for sparse representations. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on* (March 2012), pp. 3573–3576.

[66] Mallat, S., and Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing 41*, 12 (1993), 3397–3415.

[67] McAulay, R., and Quatieri, T. Speech analysis/synthesis based on a sinusoidal representation. *Acoustics, Speech and Signal Processing, IEEE Transactions on 34*, 4 (Aug 1986), 744–754.

[68] Murray, J. F., and Kreutz-Delgado, K. An improved focuss-based learning algorithm for solving sparse linear inverse problems. In *Conference Record of the 35rd Asilomar Conference on Signals, Systems and Computers* (November 2001).

[69] Nam, S., Davies, M., Elad, M., and Gribonval, R. The cosparse analysis model and algorithms. *Applied and Computational Harmonic Analysis 34*, 1 (2013), 30 – 56.

[70] Necciari, T., Balazs, P., Holighaus, N., and Sondergaard, P. The erblet transform: An auditory-based time-frequency representation with perfect reconstruction. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* (May 2013), pp. 498–502.

[71] Olshausen, B., and Field, D. Natural image statistics and efficient coding. In *Computation in Neural Systems* (January 1996), vol. 7, pp. 333–339.

[72] Ozdobinski, R. Applications of dictionary learning methods for audio inpainting. Master Thesis, June 2014. in Czech.

[73] Pati, Y., Rezaiifar, R., and Krishnaprasad, P. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers* (1993), pp. 40 –44.

[74] Špiřík, J., Rajmic, P., and Veselý, V. Representation of signals: from bases to frames. *Elektrorevue – the online journal* (2010), 1–11.

[75] Špiřík, J., Rajmic, P., and Veselý, V. Reprezentace signálů: od bází k framům. *Elektrorevue – Internetový časopis* (2010), 11.

[76] Průša, Z., Søndergaard, P., Balazs, P., and Holighaus, N. LTFAT: A Matlab/Octave toolbox for sound processing. In *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research (CMMR 2013)* (Marseille, France, October 2013), Laboratoire de Mécanique et d'Acoustique, Publications of L.M.A., pp. 299–314.

[77] Rajmic, P., Bartlova, H., Prusa, Z., and Holighaus, N. Acceleration of Audio Inpainting by Support Restriction. In *Proceedings of the 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)* (2015), Brno University of Technology, pp. 325–329.

[78] Rajmic, P., and Daňková, M. *Úvod do řídkých reprezentací signálů a komprimovaného snímání*. Vysoké učení technické v Brně, 2014.

[79] Rajmic, P., and Klimek, J. Removing crackle from an LP record via wavelet analysis. In *Proceedings of the 7th international conference on digital audio effects DAFx04* (2004), pp. 100–103.

[80] Rao, K. R., and Yip, P. *Discrete cosine transform: algorithms, advantages, and applications*. Academic Press, 1990.

[81] Rášo, O. *Objective measurement and noise supression in musical signals*. PhD thesis, Brno University of Technology, 2013.

[82] Rodbro, C., Murthi, M., Andersen, S., and Jensen, S. Hidden markov model-based packet loss concealment for voice over ip. *IEEE Transactions on Audio, Speech, and Language Processing 14*, 5 (sept. 2006), 1609 –1623.

[83] Shuman, D., Wiesmeyr, C., Holighaus, N., and Vandergheynst, P. Spectrum-Adapted Tight Graph Wavelet and Vertex-Frequency Frames. *Preprint, arXiv:1401.6033* (2013).

[84] Siedenburg, K. Persistent Empirical Wiener Estimation With Adaptive Threshold Selection for Audio Denoising. In *Proceedings of the 9th Sound and Music Computing Conference* (Copenhagen, Denmark, 2012), pp. 426–433.

[85] Siedenburg, K., and Doerfler, M. Persistent time-frequency shrinkage for audio denoising. *J. Audio Eng. Soc 61*, 1/2 (2013), 29–38.

[86] Siedenburg, K., Kowalski, M., and Dorfler, M. Audio declipping with social sparsity. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on* (2014), IEEE, pp. 1577–1581.

[87] Siedenburg, Kai; Dörfler, M. Structured sparsity for audio signals. In *Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx-11)* (September 2011), pp. 23–26.

[88] Tibshirani, R. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological) 58*, 1 (1996), 267–288.

[89] Velasco, G. A., Holighaus, N., Dörfler, M., and Grill, T. Constructing an invertible constant-Q transform with non-stationary Gabor frames. In *Proc. of the 14th Int. Conference on Digital Audio Effects DAFx11* (Paris, 2011), pp. 93–99.

[90] Veldhuis, R. N. J. A method for the restoration of burst errors in speech signals. In *Signal Processing 3: Theories and Applications*. North-Holland, Amsterdam, 1986, pp. 403–406.

[91] Veselý, V., and Rajmic, P. *Funkcionální analýza s aplikacemi ve zpracování signálů*. Vysoké učení technické v Brně, Brno, 2014.

[92] Yang, J., Wright, J., Huang, T. S., and Ma, Y. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing 19*, 11 (Nov 2010), 2861–2873.

Ing. Václav Mach

Curriculum Vitae

Experience

02/2016–now **Software developer**, *Ramet a.s.*, Kunovice (CZ).
Research and development in the field of traffic monitoring, especially speed enforcement using radar sensors technology and image processing techniques. Experimental development in Matlab, implementation of real time traffic monitoring systems in C++, Qt, OpenCV for Windows and Linux embedded systems.

07/2014 – 12/2015 **C/C++ developer, Team Leader**, *ARTISYS, s.r.o.*, Brno (CZ).
Team leader of the VCS development team (up to 5 members), responsibility for the resulting product, collaboration on the company standards for source code documentation, revision control system, project management system. Coordination of the development with the project manager, presentation of the system to the customers. References: Ing. Bahula

02–06/2014 **C/C++ developer**, *ARTISYS, s.r.o.*, Brno (CZ).
Development of Voice Communication System (VCS) for Air Traffic Monitoring (ATM) and Control (ATC) according to EUROCAE, ICAO standards and customer requirements. System analysis, SW engineering, implementation and testing in the field of VoIP, interconnection of multiple communication systems, real–time audio signal processing. Collaboration on other projects dealing with audio signal processing. Administration of Linux Gentoo OS.

2010–2014 **Freelance Audio Engineer**.
Audio digitization, restoration, preservation, archiving, audio editing, postproduction, remastering, consulting, studio and sound systems design, music notation typesetting.
Important references: The Institute of Ethnology of the Academy of Sciences of the Czech Republic, v.v.i. (dr. Pospíšilová), National Institute of Folk Culture (Mgr. Škopík), The Moravian Library in Brno (Mr. Šír).

Student jobs and internships

08/2012 **Software developer (Matlab, C)**, MESIT PŘÍSTROJE S.R.O., Uherské Hradiště.

07–08/2010 **Web Developer (PHP, MySQL)**, DICOM SPOL. S R.O., Uherské Hradiště.

06–09/2008 **Video Engineer**, CZECH TELEVISION, Brno Studio.

Education

2011–now **Ph.D. student of Telecommunications**, *Brno University of Technology*, CZ.
Signal processing, Sparse signal representations, Time-Frequency analysis, Audio software development, Electroacoustics. Part-time employed (2013–2015).
Other activities: Git server administrator, Lecturing exercises.

2009–2011 **Master of Communications and Informatics**, *Brno University of Technology*.
Audio and image signal processing, multimedia, network systems, communications, cryptography, programming (C, C++, Java, Matlab, OpenCV)
Master thesis: Implementation of Voice Activity Detectors using C open-source Libraries

Staré Město – Czech Republic
✉ *mach.vasek@gmail.com*

2006–2009 **Bachelor of Teleinformatics**, *Brno University of Technology*.
Basics of electrical engineering, communications, signal processing, audio systems, basics of programming (HTML, PHP, database systems, Java, C, C++, Matlab)
Bachelor thesis: Acoustical measurements in real environment

Research grants

2013–2016 **Novel methods for missing audio signal inpainting**.
Bilateral project of Brno University of Technology, University of Vienna and Austrian Academy of Sciences, project no. 7AMB13AT021 and 7AMB15AT033. Researcher and MATLAB toolbox leading developer.

2014–2016 **Cognitive multimedia analysis of audio and image signals**.
Brno University of Technology project no. FEKT-S-14-2335.

2014–2015 **Applications of digital audio restoration methods in the process of digitization of the audio records on magnetic tapes**.
Nation Institute of Folk Culture (Strážnice, CZ).

2013 – 2015 **Center of Sensor, Information and Communication Systems (SIX)**.
Project no. ED2.1.00/03.0072.

2013 **Applications of sparse solutions in multidimensional data processing**.
Project leader, junior interfaculty research grant of BUT no. FEKT-J-13-1903.

2010–2012 **Cultural Identity and Cultural Regionalism in the Process of Forming an Ethnic Image of Europe**.
The Institute of Ethnology of the Academy of Sciences of the Czech Republic, v.v.i., project no. AV0Z90580513. Digital restoration of digitized audio recordings of wax cylinders, sound engineering and mastering.

Research interships

05/2012 NuHAG group, Faculty of Mathematics, University of Vienna, Austria

10/2013 NuHAG group, Faculty of Mathematics, University of Vienna, Austria

Publications

Book Chapters: 4

Papers published in journals: 1

Conference proceedings papers: 4

Papers indexed in ISI WoS: 1

Papers indexed in Scopus: 1

Software: 1

Honours and Awards

April 2014 2nd place in **EEICT Student Competition**, category Signal processing, Cybernetics and Automation.

Reviewer

2013 Certified Methodology for Digitization and Online Access of Gramophone Records and Other Sound Documents for Memory Institutions. Moravian Library (Brno).

Teaching

2011–2015 Occasional lectures on Audio Restoration for Audio Engineers and Ethnologists
2013–2014 Introduction to Computer Typography and Graphics, computer exercises
2012–2014 Electroacoustics, laboratory exercises
2011 Basics of Programming, computer exercises

Special Courses and Training

2015 First Aid
2014 How to lead people instead of direct, Python programming basics
2013 Audiovisual production, Adobe Premiere - advanced, Basics of Etiquette, Networking
2012 Acoustical measurement system Björg&Kæl PULSE, LaTeX for intermediate
2011 Mathematical Approach on Computational Harmonic Analysis (Germany)
2007–2011 Cisco Academy - CCNA courses passed

Computer, programming skills

Basic JAVA, PHP, MySQL, PYTHON
Intermediate HTML, Adobe multimedia software
Advanced C/C++, OpenCV, Qt, Windows, Linux, Matlab, LaTeX, MS/LibreOffice, Audio Processing SW, JIRA, Git, Redmine

Foreign languages

English **Intermediate (C1)** *Conversationally fluent*
German **Basic (B1)** *Basic spoken and written*

Skills

Communicational Several oral presentations inland/abroad in English and Czech language
Organizational Software development team leader (up to 5 members), interfaculty research grant project leader, member of musical group leadership (up to 50 members)
Personal Teamwork skilled, organized, easy-going, willing to learn, patient, listening.

Others

Driving license B-driving license, clean record
Interests Folk musician, Swimming, Running, Dancing, Reading