Transform Coding of Audio Impulse Responses

J. van der Vorm

M.Sc. Thesis

Laboratory of Acoustical Imaging and Sound Control
Department of Imaging Science and Technology
Faculty of Applied Sciences
Delft University of Technology

Professor: Prof. dr. ir. A. Gisolf
Supervisor: dr. ir. D. de Vries

Delft, August 2003

© Copyright 2003 The Laboratory of Acoustical Imaging and Sound Control

All rights reserved. No parts of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission of the author or The Laboratory of Acoustical Imaging and Sound Control.

Graduation Committee:	dr. ir. D. de Vries Laboratory of Acoustical Imaging and Sound Control Department of Imaging Science and Technology Delft University of Technology
	prof. dr. ir. A. Gisolf Laboratory of Seismics and Acoustics Department of Imaging Science and Technology Delft University of Technology
	prof. dr. ir. L.J. van Vliet Pattern Recognition Group Department of Imaging Science and Technology Delft University of Technology
	dr. ir. R. Heusdens Information and Communication Theory Group Faculty of Information Technology and Systems Delft University of Technology
	dr. ir. D.J. Verschuur Laboratory of Seismics and Acoustics Department of Imaging Science and Technology Delft University of Technology
	drs. E. Hulsebos Laboratory of Acoustical Imaging and Sound Control Department of Imaging Science and Technology Delft University of Technology

Abstract

The main objective of the EU sponsored Carrouso project is to specify, develop and implement a new technology that can be used to transfer a sound field, generated in a real or virtual space, to a different space, preferably with full control over perceptually relevant spatial and temporal properties. The goal of this thesis can be seen as part of this: how can the amount of data that defines an 'acoustic environment' be reduced, such that it is usable for transport. Reduction is possible by using perceptual irrelevancies of the so called impulse responses, defining an 'acoustic environment'. The underlying technology is for a large part developed to compress music and speech signals, as can be found in audio codecs as MP3 and AAC.

The foundation for the recording- and playback side of the Carrouso-project is Wave Field Synthesis (WFS). The WFS technique is started by TU Delft and uses loudspeaker arrays to generate wave fronts. One of the possible methods uses dry recording (no reflections, echo or other influence of the enclosure on the signal) and does the playback with the help of a set of impulse responses. One impulse response consists of three parts, the direct sound peak, the early reflections and the reverberation tail. To reconstruct a soundfield, a set of impulse responses is convolved with the dry recording. Performing convolution over a large number of channels is computationally intensive or can cause a certain delay in playback (depending on the use of the convolution theorem), therefore the partitioned convolution is explained and used as a trade-off.

As the human ear can not analyze all components of a soundfield large parts of of audio data can be thrown away, without changing the subjective experience of a listener. Examples of properties of the human ear usable for this information reduction are the 'absolute threshold of hearing', the existence of critical bands and the masking of sound in the frequency and time domain. To exploit these properties a part of the compression calculation must be done in the frequency domain. The 'modulated lapped transform' is an appropriate transform to convert a signal to the frequency domain, because it has,

other than the discrete Fourier transform, no block-edge effects; it is critically sampled and together with the proper filterbanks its time domain aliasing cancellation leads to perfect reconstruction.

To develop an impulse response compression method, various techniques can be used. Sonke [1] has worked on a method for such compression. His method splits the impulse response in various frequency bands with the Patterson windows. The signal is then integrated over small time slices to reach the final output coefficients. To reconstruct the original impulse response a white noise signal is used. In this thesis a compression scheme is developed, which uses a reverse approach, more analog to audio compression methods. The signal is first transformed to the frequency domain in blocks of a certain length with the above mentioned 'modulated lapped transform'. Then the spectrum is saved with one parameter per critical frequency band. The block lengths are unequally divided between the early reflections and the reverberation tail of the impulse response.

This last compression model has been tested with a perceptual listening test, involving twenty volunteers. Results of this test show that the developed model works well for impulse responses with a relatively large amount of reverberation, but works less for impulse responses with almost no reverberation. It seems that careful choosing of the free parameters in the model, such as the size of the windows and the number of early reflections encoded separately, can increase the quality of the reconstruction. It can be concluded that the data stream that defines an 'acoustic environment' can be decreased with at least a factor 150, without changing the subjective experience of the listener.

Samenvatting

Het door de EU gesponsorde Carrousso project streeft ernaar om een nieuwe technologie te specificeren, ontwikkelen en implementeren die gebruikt kan worden om een geluidsveld, gegenereerd in een werkelijke of virtuele ruimte te kunnen verplaatsen naar een andere ruimte, liefst met volledige controle over de perceptueel relevante spatiële en temporele eigenschappen. Dit afstudeeronderzoek kan worden gezien als een onderdeel van het Carrouso project en houdt zich bezig met de vraag hoe de hoeveelheid data die een 'akoestische omgeving' definieert zodanig kan worden verminderd, dat deze hanteerbaar wordt voor tranpsort. Dit wordt gedaan door gebruik te maken van perceptuele irrelevantie in de zogenaamde impuls responsies die een 'akoestische omgeving' definiëren. De onderliggende technologie is grotendeels ontwikkeld om muziek en spraak te comprimeren, zoals gedaan wordt bij de MP3- en AAC-compressietechnieken.

Aan de opname- en weergavetechniek gebruikt in het Carousso-project ligt golfveldsynthese ten grondslag. De goldveldsynthese techniek is ontwikkeld aan de TU Delft en werkt door middel van het genereren van geluidsgolven met behulp van luidsprekerarrays. Eén van de methoden maakt gebruikt van een 'droog geluid' opname (geen reflecties, echo of andere invloed van de ruimte op het geluid) en geeft dit weer met behulp van een set impuls responsies, samen de 'akoestische omgeving' genoemd. Eén impuls responsie kan worden opgevat als bestaande uit drie delen: het directe geluid, de vroege reflecties en de galmstaart. Om nu een golfveld weer te geven wordt een set impuls responsies geconvolueerd met de droge opname. Wanneer een convolutie over veel kanalen tegelijk uitgevoerd wordt, is dat behoorlijk rekenintensief en veroorzaakt tevens vertraging in weergave. Gepartitioneerde convolutie is daarom geschikt als tussenoplossing.

Het menselijk gehoor heeft diverse eigenschappen die het mogelijk maken om een gedeelte van data die een geluidsveld defininieert weg te gooien, zonder dat dit de subjectieve

waarneming van de luisteraar beïnvloed. Voorbeelden van deze eigenschappen zijn onder andere de ’absoluut drempelwaarde van het gehoor’, het bestaan van kritische frequentiebanden en het maskeren van het geluid in het frequentie- en in het tijd-domein. Om dit uit te buiten moet een gedeelte van de compressie-berekeningen gedaan worden in het frequentie-domein. De ’modulated lapped transform’ is een geschikte transformatie voor conversie van een signaal naar het frequentie-domein. Deze transformatie heeft, bijvoorbeeld in tegenstelling tot de discrete Fourier transformatie, geen last van blok-band effecten en kent kritische sampling. Hierdoor kan samen met de juiste filterbanken door middel van ’time domain aliasing cancellation’ perfecte reconstructie plaatsvinden.

Om een impuls responsie compressie methode te ontwikkelen, kunnen verschillende technieken gebruikt worden. Sonke [1] heeft hier waarschijnlijk als eerste aan gewerkt. Zijn methode splitst eerst de impuls responsie in diverse frequentiebanden met behulp van Patterson-windows. Dit gewindowde signaal wordt dan in stukjes over de tijd geïntegreerd om de uiteindelijke uitgangscoëfficienten te verkrijgen. Voor reconstructie van deze coëfficienten tot een impuls responsie, wordt gebruik gemaakt van een Gaussisch ruissignaal. In het kader van dit afstudeeronderzoek is een compressiemethode ontwikkeld die omgekeert werkt, meer analoog aan de bestaande audio compressie methoden. Het signaal wordt eerst per blok omgezet in het frequentiedomein, met bovengenoemde ’modulated lapped transform’ en vervolgens wordt het spectrum opgeslagen met één uitgangscoëfficient per kritische frequentieband. De lengte van de blokken wordt ongelijk verdeeld over de vroege reflecties en galmstaart van de impulsresponsie.

Dit laatste compressiemodel is getest met behulp van een perceptuele luistertest onder twintig vrijwilligers. Hieruit blijkt dat het compressiemodel een goed model is voor impuls responsies met relatief veel galm, maar minder goed werkt voor impulsresponsies met weinig galm. Verder bleek dat het zorgvuldig kiezen van de vrije parameters in het model, zoals de grootte van de gebruikte windows en het aantal apart opgeslagen reflecties, de resultaten kan optimalizeren. Uiteindelijk kan de datastroom die een ’akoestische omgeving’ definieert met een factor 150 verminderd worden, zonder dat de luisteraar een duidelijk verschil merkt.

Contents

Abstract **vii**

Samenvatting **ix**

1 Introduction **7**

1.1 Context of this thesis 7
1.2 Transform coding of impulse responses 7
1.3 Research Goals 8
1.4 Thesis Outline 9
1.5 Thesis research 10

2 Wave Field Synthesis **13**

2.1 Introduction 13
2.2 WFS Theory 14
2.3 Practical implementation 16
2.4 Data and scaling 17

3 Impulse responses and convolution **19**

3.1 Impulse Response 19
3.2 Convolution in the frequency domain 20
3.3 Overlap-add and overlap-save 21
3.4 Partitioning 22

4 Audio Coding Basics **23**

4.1 Coding approaches 23

4.1.1 Lossless and Lossy 23
4.1.2 Hybrid and parametric 23
4.1.3 Waveform coders 24
4.2 Psychoacoustics 24
4.3 The human ear 24
4.3.1 Threshold of hearing 25
4.3.2 Critical Bands 26
4.3.3 Simultaneous Masking 27
4.3.4 Temporal Masking 27
4.4 Quantization 28

5 Lapped Transforms **31**
5.1 Introduction 31
5.2 Spectrum Estimation 32
5.3 Block Transforms 33
5.3.1 Matrix definitions 33
5.3.2 Discrete Fourier Transform 34
5.3.3 Karhunen-Loeve Transform 35
5.3.4 Discrete Cosine Transform 36
5.4 Lapped transforms 36
5.4.1 Lapped Orthogonal Transform 36
5.4.2 Perfect Reconstruction 38
5.4.3 Modulated Lapped Transform 39
5.5 Windowing 41

6 Subband Coding **43**
6.1 Properties 43
6.2 Band Filters 43
6.3 The parametrization process 44
6.4 The reconstruction process 45

7 Transform coding **47**
7.1 Overview 47

7.2 Window switching . . . 47
7.3 Window design . . . 49
7.4 Bark filterbank . . . 51
7.5 Spectral coding . . . 52
7.6 Compression Ratio . . . 54

8 Results 55
8.1 Comparison of the algorithms . . . 55
8.2 Transform coding filterbank . . . 57
8.3 Listening test . . . 59
8.3.1 Test method . . . 59
8.3.2 Evaluating environment . . . 60
8.3.3 Listening results . . . 62

9 Conclusion and Discussion 65
9.1 Conclusion . . . 65
9.2 Suggestions for future research . . . 66

A Critical Band filterbank 67

B Relation between the DFT & MDCT 69

C Window switch kernel 71

D Results of the perceptual tests 75

Bibliography 77

List of Figures

2.1 Demonstration of the Huygens principle for a spherical wavefront. . . . 14

2.2 Geometry for the Kirchhoff-Helmholtz integral formulation. The wave field inside S due to primary sources outside S is fully defined by the wave field on S. 14

2.3 Rayleigh I: The pressure in a receiver point depends on a distribution of monopoles on a plane surface. 15

2.4 Schematic representation of a virtual source generated in front of a loudspeaker array [25] 15

2.5 DSP, mixer and amplifiers, for the 160 speaker WFS system at the TU Delft 16

2.6 Speaker arrays for the TU Delft WFS system, set in: close up of the array. 17

3.1 Theoretical and measured impulse response 19

3.2 Example of an overlap-save process. Input signal $x(n)$ is split in three L-sized blocks. Convolution is done between x_i and y_i. To construct the output signal $y(n)$, the extra $M-1$-part is discarded. 21

3.3 Example of an overlap-save process. Input signal $x(n)$ is split in three L-sized blocks. Convolution is done between x_i and y_i. To construct the output signal $y(n)$, the extra $M-1$-parts are summed up. 22

3.4 Overview of partitioned convolution [23]. The input stream (showed at the top) is broken into blocks and each block is Fourier transformed (FFT in the figure), convolved, inverse transformed an overlap-saved to the output signal (at the bottom). 22

4.1 Simplified structure of the human ear 25

4.2 The Absolute Threshold of Hearing 25

4.3 Example of a non-uniform filterbank, resembling the human ear 26

4.4 Schematic Representation of Simultaneous Masking 27

4.5 Pre-echo: a) Original signal b) Reconstructed with pre-echo. In the reconstruction there is noise before the peak, because of quantization. . . 28

5.1 Signal processing with a lapped transform with 50% overlap 37

5.2 Half Sine Window for Overlap Add Sequence 41

5.3 Comparison for Window Overlap Add for Steady State and Transients . . 41

6.1 Patterson versus scaled Hanning Windows 44

6.2 Overview of the subband coding method 44

6.3 Reconstruction process of the subband coder 45

7.1 Overview of the transform coder 47

7.2 Reconstruction process of the transform coder 47

7.3 Frequency selectivity of the half-sine Window 50

7.4 Frequency selectivity of the Kaiser Bessel Derived window with $\nu = 6$ . . . 50

7.5 Frequency selectivity of the designed window 51

7.6 Plot of bark-transformed frequency against frequency in Hertz 51

7.7 Possible Bark filterbank with cosines shaped windows over 4096 samples 52

7.8 Process of Bark reconstruction 53

8.1 Time domain representation of the original and reconstructed Impulse response using the subband coder 55

8.2 Time domain representation of the original and reconstructed Impulse response using the transform coder 56

8.3 Spectrum of the original and reconstructed Impulse response using the subband coder 56

8.4 Spectrum of the original and reconstructed Impulse response using the transform coder 57

8.5 Match of the short windows with peaks in the impulse response 58

8.6 Filterbank with gradually longer windows (red) and impulse response(blue) 58

8.7 Screen shot of the program written for the listening tests 59

8.8 Time domain representation of the original and reconstructed impulse response using the transform coder 61

8.9 Spectrum of the original and reconstructed impulse response using the transform coder 61

8.10 Mean and confidence intervals of the grades, as outcome for the listening test 62

List of Tables

6.1 Time Integration Lengths of the Human Auditory System 44

8.1 Comparison of the number of parameters used in the subband and transform coder for an impulse response of 131072 samples 56

8.2 Grades and descriptions 60

8.3 Description of the sessions in the listening test 61

D.1 Difference grades for all subjects for all sessions. Columns S 1-9 represent the nine listening sessions. The rows represent the various subjects. The last row gives the means of the columns. 75

D.2 95% Confidence interval for the listening test. Rows S 1-9 represents the nine listening sessions. The 'From' and 'To' columns depict the border values for the confidence interval. 76

D.3 Anova data for the listening test. Rows S 1-9 represents the nine listening sessions. Further the results 'Between the groups' are displayed, first the Residue Sum SS, then the Mean Squares MS, the test ratio F and the probability P for F. 76

Chapter 1

Introduction

1.1 Context of this thesis

In 2001 the EU sponsored Carrouso project started for the specification, development, implementation and validation of new technology of three dimensional audio. Carrouso stands for Creative, Assessing and Rendering in real-time of high-quality audiovisual environments in MPEG-4 context. The idea for the underlying WFS technology originated from one of the partners in this project, TU Delft. WFS (Wave Field Synthesis) is a method for temporal and spatial reproduction of a sound field [20]. The Carrouso project aims at combining WFS with the flexible MPEG-4 standard. This MPEG-4 standard serves as a container for audio data and defines a number of techniques helping to transport audio in a compressed way.

This research project is not an official part of the Carrouso project. However, it shows a nice overlap between the two parts of the Carrouso project. The compression of audio impulse responses using perceptual analysis provides a lot of advantages for the implementation of a WFS-based reproduction system. Also, the coding of audio impulse responses shares a lot of ideas with the compression of digital music and speech, a task which was standardized by the first incarnation of MPEG.

1.2 Transform coding of impulse responses

Digital storage and playback of music became mainstream when the Compact Disc (CD) was introduced in 1986. Drawback of the digital representation of audio in this way is the high data rate. Conventional audio Cd's are sampled with 44.1 KHz in 16 bits, thus delivering approximately 700 kbps of data. Nowadays a lot of digital audio is downloaded or streamed from networks as the internet or the mobile telephone network. Data rates, such as those from a Cd, are too high for downloading and streaming. The WFS-approach of a 3D sound field uses more audio channels than the simple stereo setup of the CD, making the problem even worse. Fortunately the data rate can considerably be reduced without affecting the quality too much due to perceptual irrelevancies and statistical redundancies [4] of fully coded audio data.

One of the most well-known file standards for storing encoded audio is MP3. This is a shorthand notation for MPEG1, layer 3. The MPEG4-standard is successor to this standard and contains multiple usable profiles with varying behavior [3]. At this moment (2003) there is no profile for storing WFS data, but this is what the Carrouso project aims at. Example profiles in MPEG4 are the LPC (Linear Prediction Codec)-profile for encoding speech and the Main profile for the best possible reproduction of music. Note that MPEG4 also contains profiles for playback of video.

The key point in WFS is that a sound field can be reproduced by convolving a dry signal with impulse responses. The dry signal is the original music or speech signal, while the impulse responses characterize the room or space in which the origin must be reconstructed. Changing the impulse response can lead to the perception of being in a different room, standing on another spot in the room or hearing the source signal from different angles. For proper playback in such a system with at least eight channels of impulse responses are needed for varying acoustics the impulse responses have to be switched at least every few seconds. This leads to large and demanding data streams.

Impulse responses can be compressed with larger compression ratios than audio data (MP3 has a ratio around 12:1), although it requires different techniques. Sonke [1] worked on this as part of his PhD. thesis, but was probably unaware of the advances in audio coding (a lot of work in this area has been done in the last decade), and thus choose a rather unusual approach. This research builds on his work and tries to improve the compression of impulse responses using more advanced algorithms from the audio coding world.

1.3 Research Goals

The goal of this research is to develop a coding structure which allows the compression of impulse responses to a ratio much higher than current audio coders or the algorithm by Sonke. The compression type is lossy, which means that the coded version of the impulse response will not contain all information of the original. The amount of reduction that can be reached depends on two principles. The first principle is based on the characteristics of the human hearing system. A time-frequency analysis is done to compare the impulse response to psychoacoustic masking properties of the human ear, just as in audio coders. The second principle is that the characteristics of impulse responses are used to reach a meaningful reduction of data, by discriminating between the various parts of an impulse response such as the direct sound, early reflections and the reverberation. This reduction can done by carefully constructing a filterbank. The proposed model will contain free parameters, such as the number of frequency bands, the size of the windows and the quantization of the parameters which effect the quality of the compression. The difference between the compressed and the original impulse responses will be perceptually evaluated by use of listening tests.

The proposed model must be able to apply to a wide range of impulse responses (measured and artificially constructed) and must be practical in use. Therefore encoding speed on modern computers must be of the order of magnitude of real-time and decoding must be even faster than real-time.

1.4 Thesis Outline

In this chapter an introduction to this thesis and its field is given as well as an outline of the research goals. The second chapter explains the framework of this research, the Wave Field Synthesis. A lot of information about WFS is already published by our group at TU Delft; this chapter will provide more detail about the practical use of WFS leaving the theory to others [20]. The third chapter contains a deeper look into the problem of convolution and states some properties of typical impulse responses, since fast convolution is essential for playback in a WFS-system and thus for the listening tests in which the encoding model is tested. Also overlap-add and overlap-save are explained, providing a basis for the understanding of the use of lapped transforms, handled later in this thesis.

The fourth chapter deals with audio coding basics. Instead of giving a complete overview this chapter will only highlight some parts of current audio coders, particular those parts used in the proposed model. This chapter is split into two parts: one part discusses the perceptual properties of the human ear, the other part deals with various approaches to audio coding in a wide range of applications.

Chapter five deals with some advanced signal theory, mainly concerning various transforms, such as the MDCT and the use of multirate filterbanks. This chapter serves as a building block for the proposed coder, the existing coders and signal theory.

Two methods of coding are investigated in this thesis. The method, described in chapter six, originally researched by Sonke[1] and Hulsebos [7] was named parametrization of impulse responses. In this thesis it is called subband coding; this name describes the functionality better and avoids confusion, because parametrization of audio signals in the time domain is historically called parametrization, while this coder splits the signal in different frequency bands, analog to the subband coding of audio coders.

In the seventh chapter a new kind of coding is developed, based on block wise transformation of the impulse response, followed by spectral analysis of that block. The development of this method is the research goal of this thesis.

Chapter eight contains the results obtained by using the proposed coder. Starting with time/frequency plots of reconstructed impulse responses and a discussion of their consequences. Then results of a listening test are given to investigate the quality of the results perceptually. The thesis ends with a discussion of the results, the conclusions and recommendations for further research.

1.5 Thesis research

This thesis describes the development of an impulse response coder such that it can be understood by someone who is not an expert in the field of audio coders or signal analysis. For fluent reading it is not always emphasized where basic theory is explained and where new design methods are given. Downside of this approach is that it can be unclear what part of the research was already be done in the past and what part of the research is done for this thesis. This section is added to make this more clear.

The Wave Field Synthesis theory described in chapter 2 is given as the context for an impulse response coder and is not part of the research for this thesis. The research started by examining the possibilities to improve the subband coder (chapter 6) from Sonke. While some progress was made in this direction, also some boundaries of this approach were encountered, leading to a literature study about audio coders and later the transform coder.

It was also found that listening to impulse responses on an headphone was rather subjective and the results had to be tested in a WFS setup to reach more objective results. This started the part of research described in chapter 3. The first part, about the properties of impulse responses, is well-known for a long time; the second part about partitioned convolution however, contains newer insights. A program was written using the partitioned convolution theorem for playback on a WFS system to be able to test the developed impulse response coder in a full setup.

The basic principles of psycho-acoustic analysis for audio signals, as given in chapter 4, is a subset of a much wider theory. Discussion about this subject is still done, as can be seen from the diversity of underlying principles in audio coders. The choice to give exactly the information that is presented here, already points out in the direction of the transform coder for impulse responses.

In chapter 5 and 7 it is easy to point out what is already known theory and what is new material: chapter 5 contains only already known theory, chapter 8 uses and extends this theory to develop a transform coder for impulse responses. The last two chapters of this thesis, containing the results, listening tests and conclusions are, of course, all part of research for this thesis.

Chapter 2

Wave Field Synthesis

2.1 Introduction

To perfectly reconstruct a recorded three dimensional wave field is a goal which people have tried to reach for a long period of history. Various approaches were used in this context, all with their own shortcomings. Around 1930 the stereo technology was developed. Already then a schism started as an American and a British team researched the possible way of recording with two microphones. The British team, led by Alan Blumlein was interested in playback of a recording in a domestic environment, while the American team, researchers working at Bell labs, were interested in providing stereo to large audiences for use in (film) theaters.

Today this difference can still be seen. The development built upon the 'American' research has led to the current 3D-audio standard Dolby 5:1. This standard is supported by a lot of consumer electronics devices. There are also other standards based on the same principles: the audio is encoded in a number of channels, which are played back at certain locations (front, back, bass). If, and only if, the listener is in a narrow listening space (the 'sweet spot'), he experiences the correct 3D sound field. The Dolby standard also embraces compression of the audio channels, but other than that, still suffers from the sweet spot problem. [1]

Blumlein's development has evolved to more advanced systems (more control allowed), such as ambiophonics and ambisonics. These systems [9] use impulse responses of the recording room for more control of the reproduction of the sound field. They try however to reach this goal in a way which allows ‘semi’-traditional recordings. The Wave Field Synthesis (WFS) as proposed by Berkhout [20] in 1988 is a theoretical generalization of the wave field theory used by the ambiophonics system. Also Berkhout proposes methods of extrapolation of sound field information which leads to better reproducibility. These formulas originated as a parallel from seismic exploration research where complete models for the propagation of wave fields in the earth have been developed.

---

[1] The description here deals mainly with the playback part of 3D-audio. Recording the channels in a proper way is quite complicated and a lot of advancement in this field is made during the years. This is however beyond the scope of this thesis.

2.2 WFS Theory

The Huygens principle states that each point on a wave front can be regarded as the center of a disturbance, which is the source of an elementary spherical wavefront. This is demonstrated in figure 2.1 for a spherical wavefront. By substituting all disturbance centers by a loudspeaker it is possible to use this array as a wave field generator.

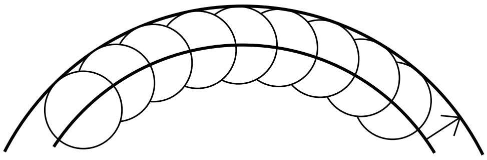


Figure 2.1 *Demonstration of the Huygens principle for a spherical wavefront.*

In this section the basic principles of the WFS-theory are given. A more thorough and fundamental treatment is given by Berkhout et al [20]. The Huygens principle is not directly applicable for use with a discrete playback system, such as a loudspeaker array. For the continuous case the Kirchhoff-Helmholtz integral states that an arbitrary sound field can be generated with a distribution of monopole and dipole sources on the surface of a closed volume

$$P(r,\omega) = \frac{1}{4\pi}\iint\limits_{S} P(r_s,\omega)\frac{\delta}{\delta n}\left(\frac{e^{-jk|r-r_s|}}{|r-r_s|}\right) - \frac{P(r_s,\omega)}{\delta n}\frac{e^{-jk|r-r_s|}}{|r-r_s|}dS \tag{2.1}$$

As can been seen in figure 2.2, $|r-r_s|$ is the distance between the listener point and the surface S. $P(r_s,\omega)$ is the sound pressure at the listener point in the frequency domain and k is the wave number ω/c.

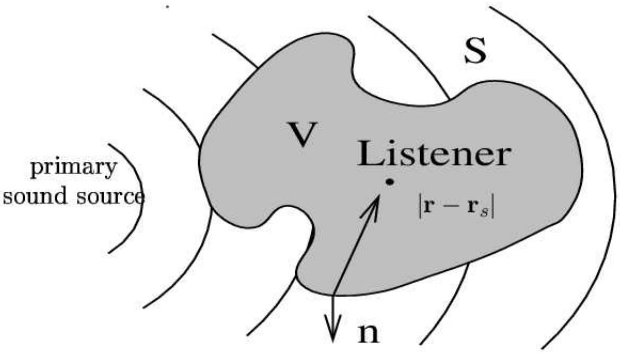


__Figure 2.2__ Geometry for the Kirchhoff-Helmholtz integral formulation. The wave field inside S due to primary sources outside S is fully defined by the wave field on S.

If the surface S is chosen to be a continuous infinite plane separating the source from the receiver area, the Kirchhoff-Helmholtz integral can be simplified and transformed into the Rayleigh I integral:

$$P(r,\omega) = \rho c \frac{jk}{2\pi} \int_S V_n(r,\omega) \frac{e^{-jk|r-r_s|}}{|r-r_s|} dS \tag{2.2}$$

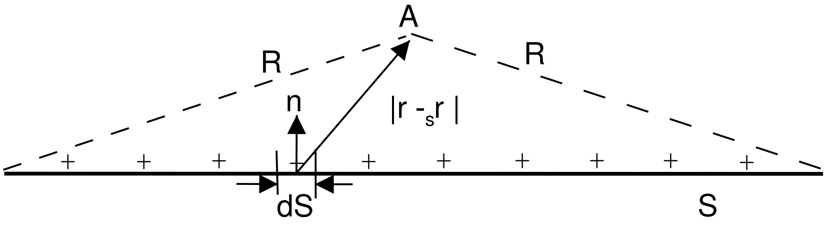


__Figure 2.3__ Rayleigh I: The pressure in a receiver point depends on a distribution of monopoles on a plane surface.

Here V is the related normal velocity of each monopole as shown in figure 2.3 and $|r - r_s|$ is the distance from listener point A to a monopole source on the plane. Thus the pressure $P(r,\omega)$ can be synthesized by means of a monopole distribution on a plane. If this is translated in practical sense, then we could physically synthesize the wave fronts at any listening point by re-radiating the sound velocity, recorded at a certain plane with loudspeakers having monopole characteristics as shown in figure 2.4.

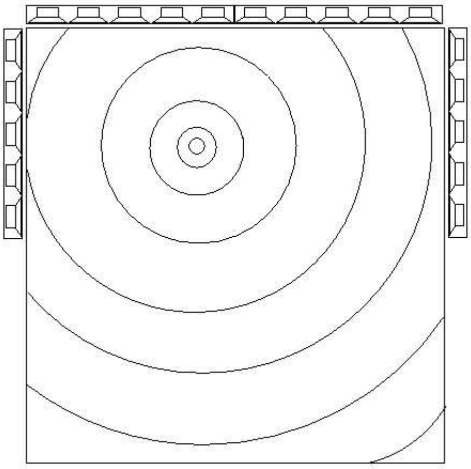


Figure 2.4 *Schematic representation of a virtual source generated in front of a loudspeaker array [25]*

Because a finite number of loudspeakers is used as secondary sources a discrete version of the Rayleigh I integral should be used. Also using a linear array instead of a planar array is much more feasible. The driving force of the speakers in such a system is given by [20].

A practical problem is that of the spatial spacing of the loudspeakers. Aliasing will occur for frequencies above

$$f_{\text{nyq}} = \frac{c}{2\Delta x} \tag{2.3}$$

It is possible to recreate sound fields [21] without spatial aliasing artifacts if

$$f_{\text{max}} < \frac{c}{2\Delta x \sin \alpha_{\text{max}}} \tag{2.4}$$

where α_{max} indicates a maximum angle between the direction of the plane wave and the loudspeaker array. In practice a loudspeaker spacing of 0.125 m gives perceptually correct results for wavefronts up to 1360 Hz [21]. Virtual sources in- and outside the listener area can be reconstructed.

2.3 Practical implementation

As shown in the previous section it is possible to recreate a sound-field in the horizontal plane using a linear array. Employing this technique in a WFS (Wave Field Synthesis) system can be done in two ways. In the first approach during a recording session the direct sound is recorded as well as the reverberation of the room on separated channels. This can be done with spot microphones and a special room configuration. Another way to make these recordings is by employing circular arrays [7]. During playback the direct sound is reproduced as a point source and the reverberated sound field is given by plane waves.

In the second approach only the direct sound is recorded during a session. The impulse responses (acoustic environment) are measured separately for the different locations of the source. During playback the direct sound is convolved with the impulse responses and the result is reproduced by the loudspeaker array using plane waves. This approach is used in this thesis.

The reproduction method of a WFS system is different compared to more conventional methods. Advantages above more conventional methods include:

- The dry recorded source signal can be reproduced in an arbitrary acoustic environment with arbitrary listener and source positions.
- Instead of the ’sweet spot’ of conventional audio playback there is a ’sweet area’ were a proper 2D sound field is constructed.
- Synthetic acoustic environments (made with acoustic modeling software) can be auralised rather precisely. (useful for testing future concert halls or simulators).

There also some problems attached with the implementation of a WFS system:

- Relatively dry source signals are needed, so ideally the recordings of the source are done in an anechoic chamber.
- Large arrays of speakers are necessary for proper playback, which gives rise to sight problems and is rather expensive.

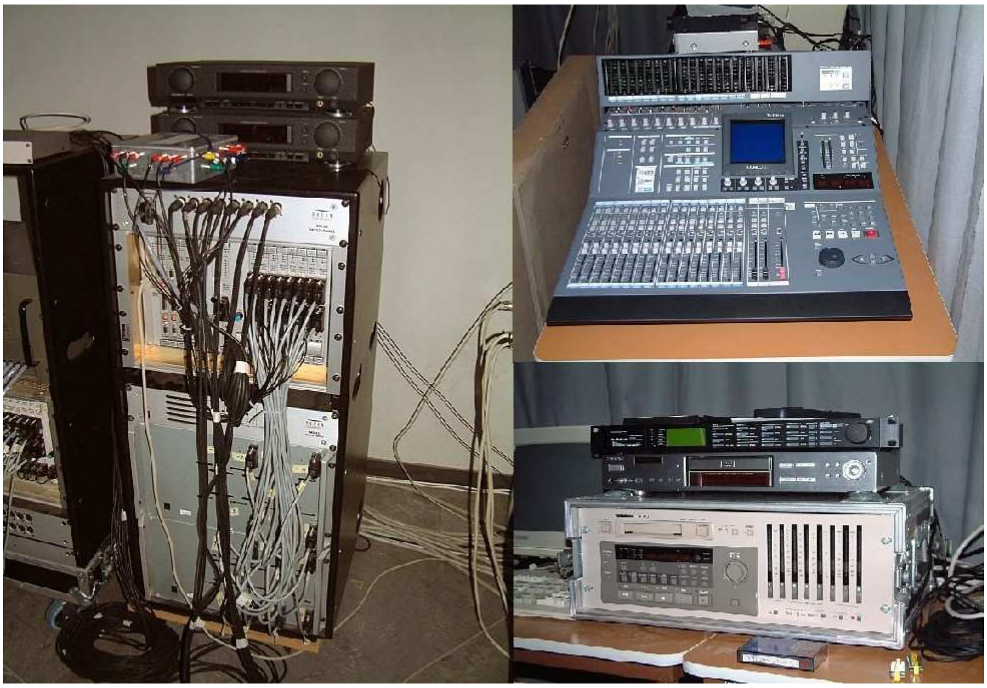


Figure 2.5 *DSP, mixer and amplifiers, for the 160 speaker WFS system at the TU Delft*

- The location of the source over time must be known.
- Convolution of all signals can be very processor intensive.

Most of these problems have practical solutions. The first problem can be circumvented with the use of close miking, which results are good enough for WFS playback. The last issue (convolution is computational intensive) can be solved by grouping sources. If, for example an orchestra is recorded, it is not so important to properly place all individual violin players. They can be grouped into fewer sources.

Instead of using speaker arrays with large numbers of speakers, research is done using distributed mode loudspeakers (DML's) for playback, although a large number of drivers remains necessary. Measuring the location of the source can of course be done by using GPS-like systems, but also measuring with microphone arrays is being investigated [11].

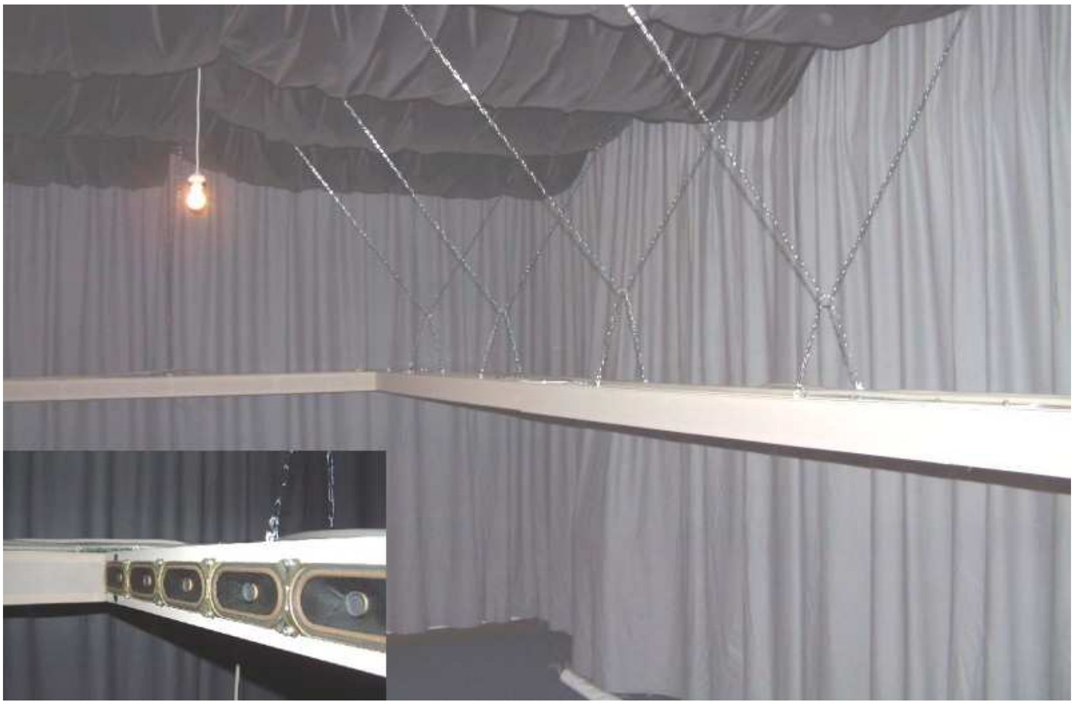


Figure 2.6 *Speaker arrays for the TU Delft WFS system, set in: close up of the array.*

2.4 Data and scaling

More channels give more data. This has consequences for storing and transmitting data via a network, such as the internet. Separation of sound sources can help in this respect. In conventional multichannel playback, the total amount of data is proportional to the number of channels. Audio channels can be compressed independently with conventional means, such as AAC (Advanced Audio Codec) compression, reducing the bit rate to approximately 192 Kbps (Kilobit per second) per channel, thus transmitting a 32 channel signal costs already 6144 Kbps. This is of course infeasible for current internet connections (approximately 33-2048 Kbps).

If a WFS-recording must be transmitted (or stored) this can be done by calculating the different channels in advance, so a situation analog to the previous example arises. If the WFS information is pre-calculated over 32 directions also 6144 Kbps is used. It is possible to compress these pre-calculated channels due to inter-channel correlation. A feasibility study for this has recently been done [25].

A different approach is separating the dry sound and the impulse responses (acoustic environment) and store/transmit them independently. This gives a total different scaling of the amount of data in a recording. In this situation the amount of data is proportional to the number of sources. However, the impulse responses alone provide still a rather large data stream: if 16 directions are used and the acoustic environment is updated every second (which is proposed by the Carrouso project), and 96KHz - 20 bit audio is used, approximately 2900 Kbps is needed. The goal of this thesis project is to compress the impulse response with a much higher factor (200x) than the current audio coders do (10x).

If the goal is minimization of the amount of data a certain trade-off exists between the number of sources, the number of playback channels and the chosen approach. More playback channels increase the data rate for the pre-calculated channel approach and more sources increase the data rate in the approach, were acoustics and sources are transmitted separately.

Another obvious advantage of the separation of sources and acoustic environment is that a recording scales naturally to an arbitrary number of playback channels (and thus to different playback systems, from small to large arrays). To make this possible sufficient information about the acoustic environment must be known, but, as stated above the amount of data of the impulse responses can be significantly reduced with compression.

Chapter 3

Impulse responses and convolution

3.1 Impulse Response

The ’acoustic environment’ is the set of impulse responses as used in a WFS system, described in the previous chapter. In this section further specification of an impulse response of a room is given.

The impulse response is literally the response of a system to an impulse. This impulse can be defined as a Dirac pulse δ, and the impulse response exposes how a system reacts to such an impulse. In the audio world an impulsive audio signal [1] can be used to measure the response of a room. If this impulse response is processed, for example for WFS playback, by convolving it with a dry signal, the impulse response is used as a FIR (Finite Impulse Response) filter. This opposed to a IIR (Infinite Impulse Response) which is a filter, requiring feedback. The IIR falls outside the scope of this project (see [2] for more information), but it is important to know its existence.

As can be seen in figure 3.1 the impulse response consists of three parts: first the direct sound, which is the impulse that is directly transferred to the listener; then the early reflections, where the sound is reflected via wall, floor and ceiling; and finally the reverberation tail. The separation between the early reflections and the reverberation tail is sometimes considered to be around 100 ms after the direct sound, but actually its position depends on the (size of the) enclosure. The reflection density increases (theoretically) with the time squared, so in the reverberation part of the impulse response

---

[1] It is not possible to play a real Dirac pulse. Therefore other signals, which have a spread in energy, are used to measure the impulse response such as noise-like signals or sweeps. The impulse response is then de convolved from the response to such a signal [17]

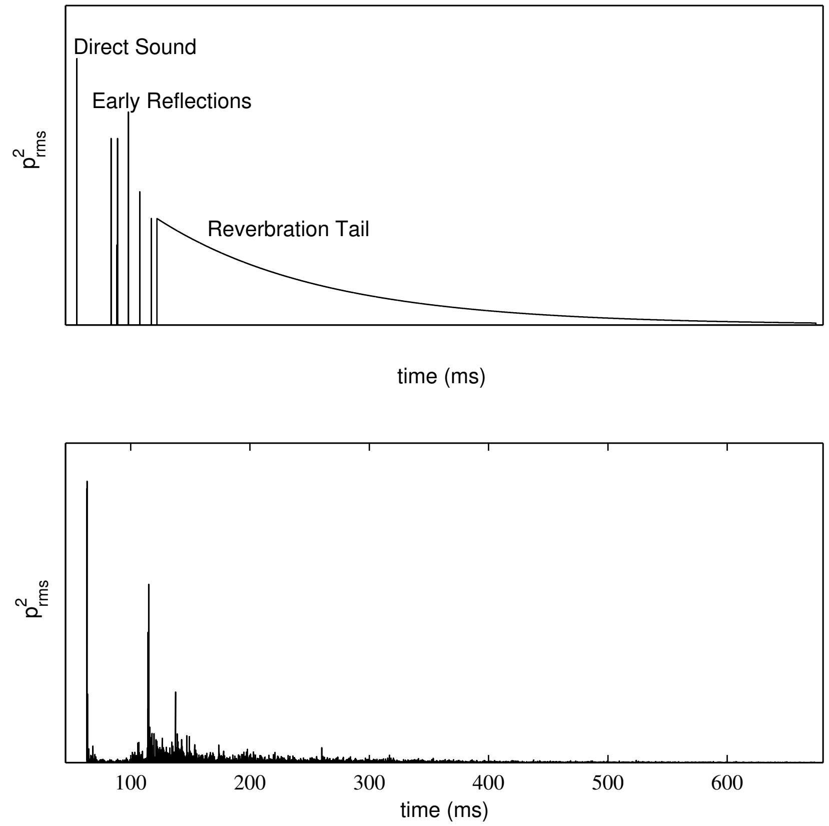


Figure 3.1 *Theoretical and measured impulse response*

this density is so high that it is usually regarded as having a statistical instead of a deterministic character. The pressure decay can be given as a function of the absorbing surface A [13] by

$$p^2_{rms}(\tau) = p^2_{rms}(0)\exp^{\frac{-Ac\tau}{4V}} \tag{3.1}$$

with p^2_{rms} the squared pressure, τ the time, V the volume of the room, $p^2_{rms}(0)$ the starting pressure. The early reflections can be calculated by the mirror image source model or ray-tracing. These techniques provide the fundamental of various hybrid models for more exact estimation of impulse responses. Nowadays multiple advanced software packages exist trying to estimate the impulse response of a certain room or space [12].

The acoustic environment is defined by a set of impulse responses. Above the impulse response is regarded as the unique response on a certain location in a certain enclosure as a result to an acoustic pulse given on a certain location. To record and use multi-trace impulse responses is regarded as being subject to the WFS system.

3.2 Convolution in the frequency domain

Convolution is necessary for utilization of all finite impulse response (FIR) filter systems, which are used for auralisation, noise control, room equalization and all kind of other digital filters. Very often one wants to reach real-time performance of this convolution when working with audio-acoustic variable systems. If the filter is rather long (for example an impulse response of an audio signal at 96 kHz and 3 seconds long is already 288 000 samples) traditional convolution in the time-domain (see equation 3.2) demands an enormous number of multiplications.

By using the convolution theorem (3.3) this problem is overcome, but a new set of problems starts when using a block-based algorithm (here a Fourier Transform) with an incoming stream of data. The calculation must be done in blocks, which can be done with the overlap-save algorithm [19] in which each block is shifted, doubled in length, convolved and summed up. In partitioned convolution the same method is used, but the impulse response is first divided in equally sized blocks. Each of these blocks is convolved by a standard overlap-save process and later summed up. At first glance this looks less efficient, because it requires more calculations, but the fundamental latency (not the calculation latency) can be much lower, because one partition of the impulse response is smaller than the total response.

3.3 Overlap-add and overlap-save

Convolution in the time-domain is given by

$$g(t) * h(t) = \int\limits_{-\infty}^{\infty} g(\tau - t) \cdot h(\tau) d\tau \tag{3.2}$$

For the frequency domain holds the convolution theorem:

$$g(t) * h(t) = G(f) \cdot H(f) \tag{3.3}$$

The overlap-save methods for dealing with large datasets and continuous data-streams can be found in [1]. The input data is divided in finite blocks of L samples (equal to the impulse response) and will be started with zeros. Only the first block is padded this way, the others will be dealt with normally. After the convolution of a block, the end of the data of this block will be polluted with wrap-around effects and thus be thrown away. This method will cause a delay of L (the size of the impulse response). An example is given in figure 3.2.

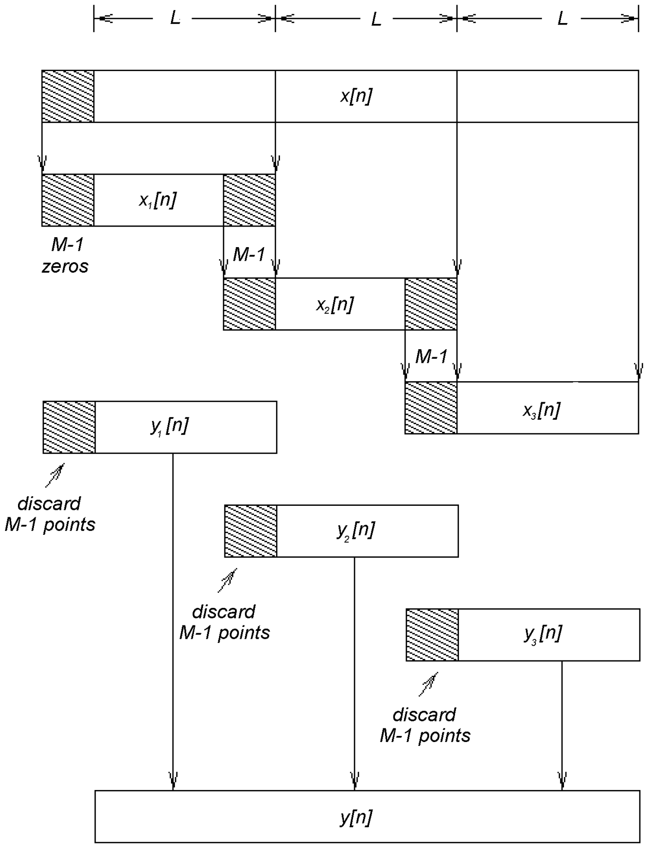


Figure 3.2 *Example of an overlap-save process. Input signal $x(n)$ is split in three L-sized blocks. Convolution is done between x_i and y_i. To construct the output signal $y(n)$, the extra $M-1$-part is discarded.*

There is also another possibility to work with data in blocks, the overlap-add method. Here each block is zero padded at both ends and then convolved. Then these pieces are added, including the overlapping regions formed by the zero-padded parts. See also figure 3.3. This process is also be useful in understanding the Lapped Transforms as given in chapter 5.

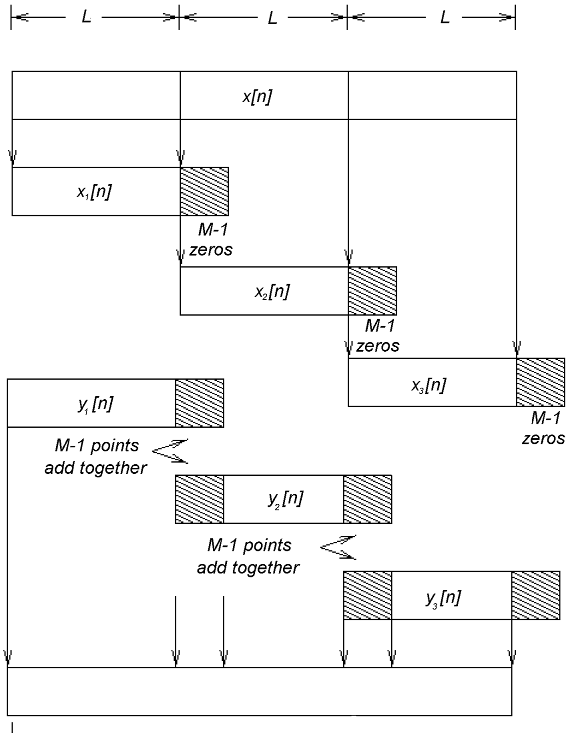


Figure 3.3 *Example of an overlap-save process. Input signal $x(n)$ is split in three L-sized blocks. Convolution is done between x_i and y_i. To construct the output signal $y(n)$, the extra $M-1$-parts are summed up.*

3.4 Partitioning

In essence this variation of calculating convolution (originally proposed by Stockham [23] in 1966) is partly done in the time domain and partly in the frequency domain. The input impulse response $h(n)$ is initially partitioned in a reasonable number of P equally sized blocks. Each block is then convolved as a separate impulse response with an overlap-save process as explained above. The resulting sub-blocks are Fourier Transformed and multiplied with a block of the input signal. At the end the sub-blocks are delayed to their original position and summed. An overview of this process is given in figure 3.4.

If the impulse response is broken up in a lot of small partitions, the speed of the Fourier

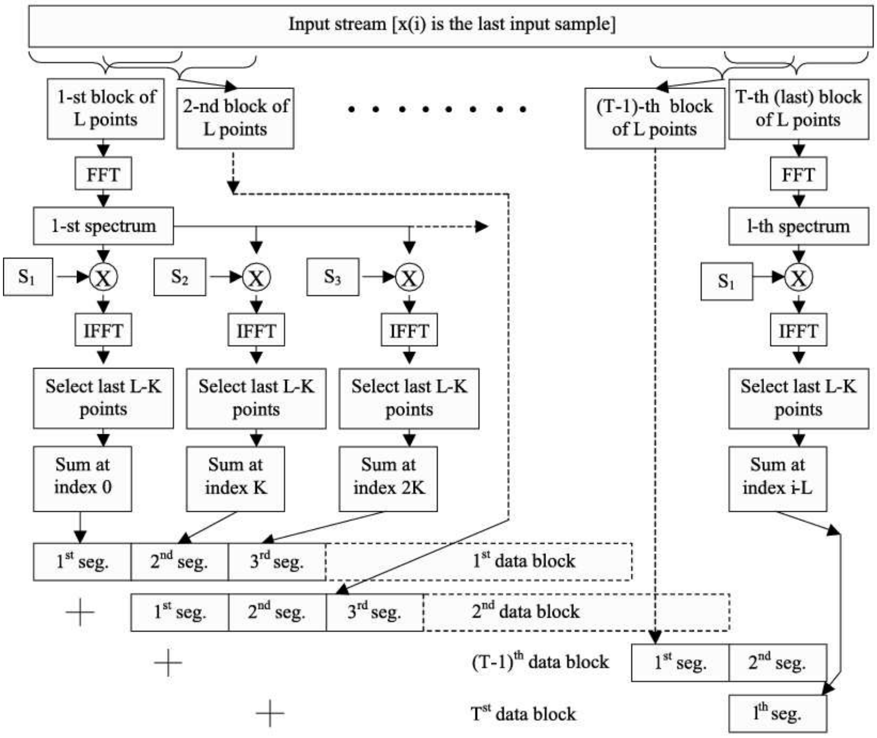

Input stream [x(i) is the last input sample]
1-st block of L points
2-nd block of L points
(T-1)-th block of L points
T-th (last) block of L points
FFT
FFT
1-st spectrum
l-th spectrum
S1
S2
S3
S1
IFFT
Select last L-K points
Sum at index 0
Sum at index K
Sum at index 2K
Sum at index i-L
1st seg.
2nd seg.
3rd seg.
1st data block
2nd data block
(T-1)th data block
Tst data block
1th seg.

Figure 3.4 *Overview of partitioned convolution [23]. The input stream (showed at the top) is broken into blocks and each block is Fourier transformed (FFT in the figure), convolved, inverse transformed an overlap-saved to the output signal (at the bottom).*

transform counts less and less. A practical implementation [23] used with a Athlon 1 GHz processor, is known to give latency of 3 ms when used with 64 samples in 128 partitions or 12 ms if 32 partitions of 256 samples are used. Compared with the measurements by Sonke ([1], 136), who finds a latency of 12.7 ms for 8192 samples on a Motorola DSP, this is a satisfying result. Of course there are a lot of other considerations like the latency of the used operating system (for example a low-latency patched Linux-kernel, can achieve a low latency), type of memory (DDR RAM is best for real high speed throughput), speed of data throughput, etc.

Because of the flexibility of a software solution, this opens the way to new and even faster convolution algorithms, taking into account the possibilities of simplifications that the properties of the human hearing system allow, as discussed in the next chapter and implementation of a WFS system.

Chapter 4

Audio Coding Basics

4.1 Coding approaches

4.1.1 Lossless and Lossy

There are two main categories of audio encoders: lossless and lossy encoders. As the name implies only lossless compression guarantees complete reconstruction of the compressed signal. In lossy compression information is thrown away in order to get better compression rates. Due to the use of psychoacoustic principles this reduction of information doesn't always lead to noticeable reduction of the sound quality.

4.1.2 Hybrid and parametric

There are various possible approaches when encoding audio lossy. In the ultra-low bit rate regime, most encoders are parametric. The parametric coders try to fit a source model to various objects in the stream. To successfully do this a certain amount of knowledge of the model must be available. This is why most of the parametric encoders are speech encoders. A more advanced class of encoders are the hybrid coders. They fill the gap between the parametric and the waveform coders discussed later. The hybrid coder works roughly the same as the parametric coder, but also sends some error information along. When a waveform, encoded with an hybrid coder, is decoded, this error information is used to deviate from the pure source model , thus giving some natural quality. The CELP-coder, provided with MPEG2 is such a hybrid coder (see [3]).

4.1.3 Waveform coders

Waveform coders can be divided into two categories: the coders trying to store the original waveform of the signal in the time domain and the coders storing the spectrum of the signal , the frequency domain coders.

Frequency domain coders form the majority of modern high quality encoders. They provide a good quality at the cost of more complexity, because they make use of the perceptual properties of the human ear. They also fall apart into two different groups [10]: the subband coders and the transform coders. The first category employs bandpass filters to split the input signal and then code the bands according to their perceptual relevance. The second category uses a fast transform to convert blocks of the input signal in frequency components and handle the psycho-acoustic properties per block. Most music compression coders (AAC, Ogg, DCC, AC-3, etc [4]) are transform coders.

The algorithm which Sonke developed [1] and which was called a parametrization of impulse responses, actually falls in the subband coder category, since no source model for impulse responses is used. The coder developed for this thesis comes close to the working of the transform coders. Actually there is no formal definition for coding of impulse response, since these coders have their own peculiarities, but in this thesis the models employed will be called subband and transform coders. One property which these coders share with some of the parametric coders is that Gaussian noise is used for reconstruction of the impulse response.

4.2 Psychoacoustics

4.3 The human ear

The peripheral part of the human auditory system converts the oscillations of air particles into neural information suited for the brain. This pre-processing of the acoustic signal performs already a frequency analysis. The structure of the ear can be divided in the outer, middle and inner ear (see figure 4.1).

The outer ear consists of the pinna (auricle), the ear canal (external auditory meatus) and the eardrum (tympanix membrane) [14]. The pinna collects the pressure waves which are amplified and conveyed to the eardrum. The ear canal is a tube, enclosing an air column, resonating at 3 kHz. The resonance increases the sound pressure level by a factor 10. By vibrating the eardrum, the energy is converted to mechanical.

The middle ear contains the hammer (malleus), anvil (incus) and stapes (stirrup). These areas of the auditory system have various functions. Important for psychoacoustics is that these areas protect against too large pressures and filter out low frequencies in noisy environments. The inner ear contains the hearing organ, a bony cone-shaped spiral called cochlea, which is filled with fluid. It converts the incoming mechanical energy to electrical impulses. This part of the ear plays the greatest role in the perception of audio.

The basilar membrane in the cochlea reacts to the pressure changes on the location where the sound wave stops. This location corresponds to a certain frequency. The basilar membrane acts thus as a spectral analyzer, converting frequency information to space information. The brain can however not separate two frequencies close to each other, due to this property. Also other perceptual problems can be made clear with the description of the ear, as described in the next sections.

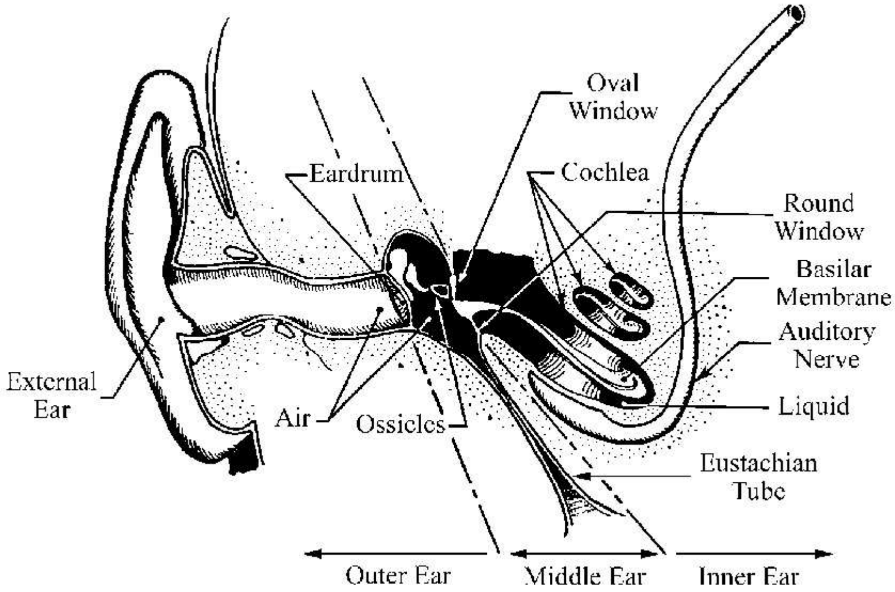

Oval Window
Eardrum
Cochlea
Round Window
Basilar Membrane
Auditory Nerve
External Ear
Air
Ossicles
Liquid
Eustachian Tube
Outer Ear
Middle Ear
Inner Ear

Figure 4.1 *Simplified structure of the human ear*

4.3.1 Threshold of hearing

The absolute threshold of hearing is characterized by the amount of energy needed for a pure tone such that it can be detected by a listener in a noiseless environment [4].

Notice that this threshold is different for different people. However in the first half of the 20th century research was done to obtain a 'standard curve' which is plotted in figure 4.2.

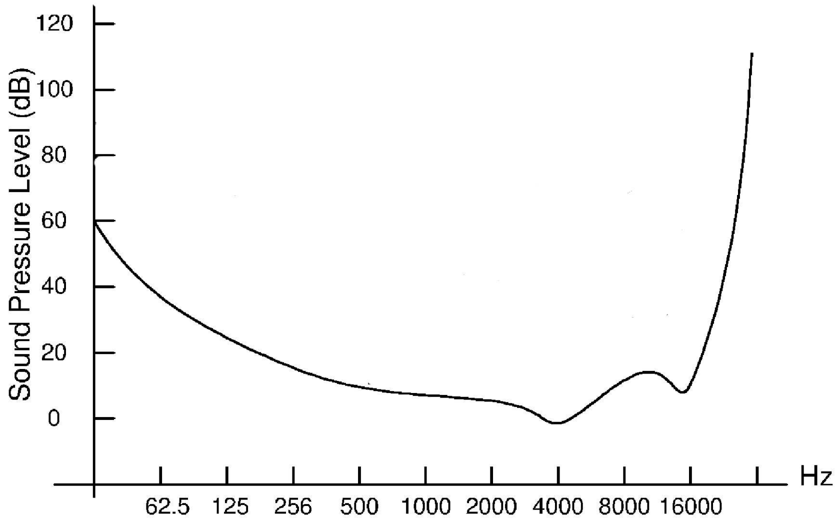


Figure 4.2 *The Absolute Threshold of Hearing*

An approximation of this threshold T_q in dB is given by

$$T_q(f) = 3.6\left(\frac{f}{1000}\right)^{-0.8} - 65e^{(-0.6\cdot\frac{f}{1000}-0.3)^2} + 10^{-3}(\frac{f}{1000})^4 \tag{4.1}$$

with f the frequency in Hz. In practice various different models for defining this threshold are used [14] and defining an accurate hearing threshold is a challenging task for modern audio coders. It is mentioned here for completeness, it is not important for coding of impulse responses, due to the fact that the loudness of the input audio signal, which is convolved with an impulse response is not known in advance.

4.3.2 Critical Bands

As explained in the previous section, the inner ear maps frequencies of a sound field to a position on the basilar membrane. This aspect explains (part of) the non-uniform frequency resolution of the auditory spectrum, since the lower frequency range has a much finer spectral resolution than the higher range. This leads to the definition of critical bands. Each point on the basilar membrane is tuned to a certain frequency called the characteristic frequency: the place at which the traveling wave caused by a stimulus reaches its maximum amplitude [10]. A bandpass filter, centered at a characteristic frequency is defined as a critical band.

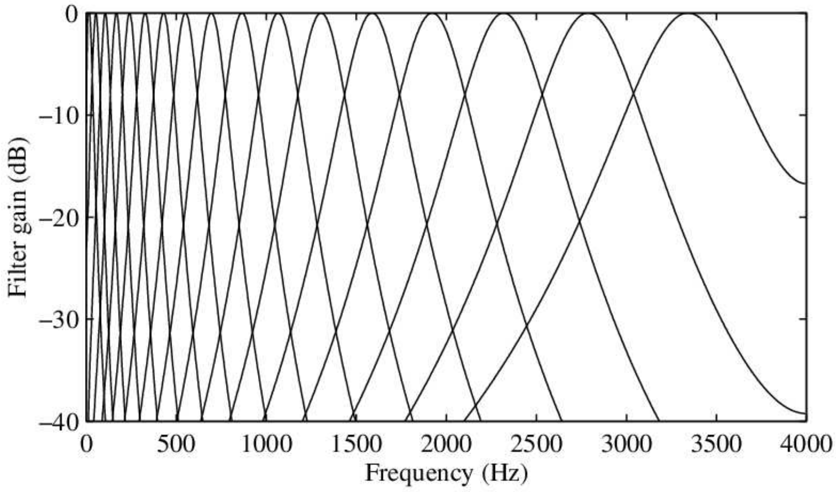


Figure 4.3 *Example of a non-uniform filterbank, resembling the human ear*

Moore [1] defines a critical band as an Effective Rectangular Band (ERB), which is the bandwidth of an ideal bandpass filter, centered at any frequency. Critical bands are distinguished, using various different methods. Masking of noise, masking of tone, stimulation with two tones, loudness of a changing frequency sinus are all properties which are dependent of the critical bands.

Experiments have shown that the width of the critical band is narrower at low frequencies. The non-linear scale on which the inner ear processes the signal is called the Bark scale. This scale is defined, such that a certain distance on the basilar membrane corresponds exactly with one Bark. A table of the Bark scale is given in appendix A. Zwicker's [14] formula to convert from frequency to Barks (z) is:

$$z(f) = 13\arctan(0.00076f) + 3.5\arctan[(\frac{f}{7500})^2] \tag{4.2}$$

An important issue is if critical band filters can be seen as discrete and slightly overlapping (or non-overlapping) or continuous. Experiments tend to indicate that they are continuous [14], but in computational use it is easier to make them discrete. Mind further the differences between the Moore and the Bark scale. They show overlap at high frequencies, but are quite different at low frequencies. Both scales are an estimation of the human auditory system at a certain input signal (for example masking of noise). They do not represent the behavior of listening to complex and compiled signals.

4.3.3 Simultaneous Masking

When a peak renders another peak inaudible, this sound is said to be masking the other. This can happen in the frequency as well as in the time domain. In the frequency domain, this property has a relationship with the critical bands. If two sounds both have components in one critical band and if one is a certain amount louder than the other, the quieter one will be imperceptible. This is called simultaneous masking , which contrasts with temporal masking, described in the next section. Physically this corresponds with the hair cells in a particular location being overstimulated and therefore unable to respond to lower magnitude vibrations.

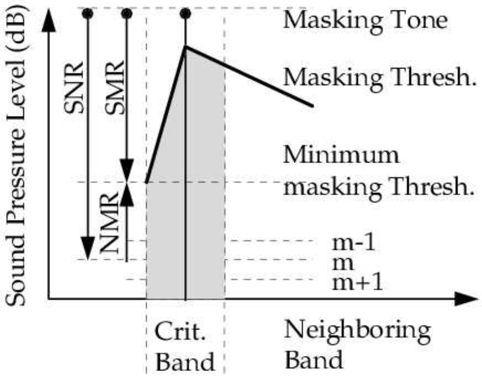


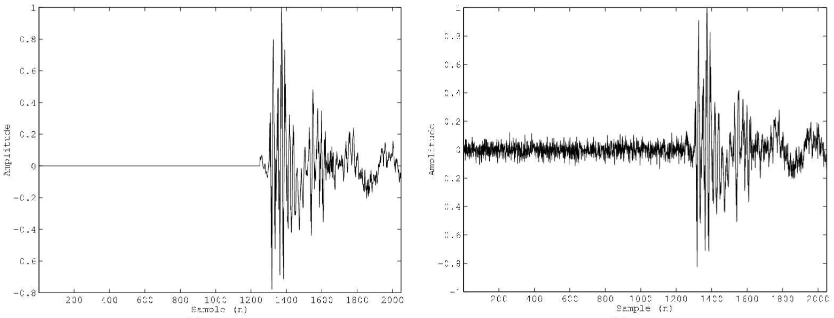
Figure 4.4 *Schematic Representation of Simultaneous Masking*

As can be seen in figure 4.4 there are several masking effects which can be exploited in an audio coder. In general a masking threshold is calculated, below which the information is thrown away. The standard practice in perceptual coding involves first classifying masking signals as either noise or tone masking, then calculating appropriate thresholds to a level where the 'just noticeable difference' (JND) lays. Then the NMR (noise to mask ratio) and the SMR (signal to mask ratio) denote the logarithmic distances from the minimum masking threshold to the masker and noise levels [27]. These thresholds are finally used to determine a certain threshold for a block of data. The function of this threshold is stored along with the quantized residue.

4.3.4 Temporal Masking

As said earlier there is also masking in time. There are two possible forms of temporal masking:

- When a loud tone masks a quieter tone which comes after it; this is called forward masking. Typical length of this effect is 50-200 ms. The amount of masking depends on the loudness of the tone and on the frequency. The frequency dependency is usually disregarded in audio coders, but the subband coder described in chapter 6 uses this property.
- A loud tone can also mask a quieter tone which comes before the louder one. This is called backward masking. The length of this masking effect is typically 5 ms. In audio coding this effect is usually disregarded.



(a) (b)

__Figure 4.5__ Pre-echo: a) Original signal b) Reconstructed with pre-echo. In the reconstruction there is noise before the peak, because of quantization.

Something related to temporal masking is the problem of pre-echo. Because most audio coders work with blocks and a threshold, silence is not reconstructed properly. Pre-echo's usually occur when a transient, (or any signal with a sharp attack) is encountered. The peak is encoded in as less bits as possible and therefore in decoding the peak, the silence is 'amplified' too much. One possible solution to this problem is the use of different window sizes, which is further researched in the next section. In figure 4.5 the pre-echo problem is visualized.

4.4 Quantization

If the spectrum of the input signal is known it still must be encoded in a proper number of bits. This is called the quantization problem. In a transform coder, first the spectrum is estimated, then the threshold is calculated using the absolute hearing threshold and subtraction/addition of all masking parameters. The residue spectrum has to be encoded, which is usually done with vector encoding. The idea is that this transformed vector is quantized with respect to the variance it has compared to the threshold function: this is the classical bit allocation problem. If an average rate of R bits per sample is wanted, a total of MR bits is available for a M samples long block. Now if the samples with index k have a geometric mean of

$$\sigma_{GM}^2 \equiv \left(\prod_{k=0}^{M-1} \sigma_k^2 \right)^{\frac{1}{M}} \tag{4.3}$$

with σ_k as variance to the threshold, then the log variance rule [19] minimizes the number of bits needed:

$$B_k = \alpha + \frac{1}{2} \log_2 \frac{\sigma_k^2}{\sigma_{GM}^2} \tag{4.4}$$

Here α is a Lagrange multiplier which depends on the bitrate wanted. B_k is the needed number of bits to store sample k. The exact way to encode a residue in bits is not important here. One should remember that the compression factor and thus certain encoding decisions earlier in the process depends on this geometrical mean.

Chapter 5

Lapped Transforms

5.1 Introduction

In this chapter some theory of transforms and spectrum estimation is given. This is done to examine which transforms can be used for an impulse response coder. As explained in the previous section one of the psycho-acoustic properties of the human ear is masking in the frequency domain. To exploit this property in a coder, some calculations have to be done on the spectrum of the impulse response.

In the next section the definition of the spectrum for a random signal is discussed, which is equal to the Fourier transform of the autocorrelation function. Since all transforms operating on a basis of sinusoids can be used as a transform, various transforms used in coders are examined. The Karhunen-Loeve Transform (KHT) is an excellent candidate due to its energy compaction and de-correlation properties, but can not be given for an unknown signal. One step further in the right direction is the Discrete Cosine Transform (DCT) which approximates the KHT. This transform is still not ideal due to block edge effects.

To solve the block edge problem, lapped transforms were researched. The MDCT or Modulated Lapped Transform seems to be a good candidate, due to its perfect reconstruction ability. Starting from a transform based on blocks, the advantage of lapped transforms is explained in section 5.4.

The chapter ends with some theory about windowing and filterbanks. Windowing of a signal is necessary when using lapped transforms. Coding an impulse response also employs a filterbank, so more information about the duality between windows and filters is researched. Some codecs use a window switching scheme to ensure proper transient and spectral coding. How this switching works and how it can be combined with the modulated lapped transform, is described in section 5.5.

5.2 Spectrum Estimation

To successfully code a signal $x(n)$ in the frequency domain, it is important to make an estimate of the power spectrum S_{xx} of this signal. The power spectrum is here defined as the distribution of the energy of the signal in the frequency domain. For a stochastic signal (such as a sampled impulse response), this spectrum is given by the Fourier transform of its autocorrelation function R_{xx} [8]:

$$S_{xx}(e^{j\omega}) \equiv \mathcal{F}\{R_{xx}(n)\} = \sum_{n=-\infty}^{\infty} R_{xx}(n)\, e^{-j\omega n} \tag{5.1}$$

where the autocorrelation R_{xx} is given by

$$R_{xx} \equiv E\left[x(m)x(m-n)\right] \tag{5.2}$$

Here E is the expectancy operator. The periodigram can be (and is classically) used as a method for spectrum estimation. This is based on the Fourier spectrum of a signal, which is only given for a deterministic signal and not for a stochastic signal. The periodigram of a block of samples is:

$$P_{xx}(e^{j\omega}) \equiv \frac{1}{M}|X(e^{j\omega})|^2 \tag{5.3}$$

The relation between the periodigram and the power spectrum is:

$$S_{xx}(e^{j\omega}) = E[P_{xx}(e^{j\omega})] \tag{5.4}$$

Here, M is the number of samples of a block over which the periodigram is taken. X is the transformed signal [1]. Hence, the power spectrum $S_{xx}(e^{j\omega})$ defines the average frequency distribution of energy for a random function $x(n)$. For spectrum estimation of a non-deterministic signal, there is no universal technique. The periodigram can be used, but is generally noisy and must be averaged over multiple blocks. Some approaches can be found in [26], but here transform methods will be discussed.

---

[1] In this chapter the capital X describes the signal in the frequency domain, while x gives the signal in the time domain.

The Discrete Fourier Transform (DFT) given by

$$X(k) = \frac{1}{M}\sum_{n=0}^{M-1} x(n)e^{-j\frac{\pi jkn}{M}} \tag{5.5}$$

can be used for the transformation of such a block of size M. But if this is done on a finite segment of a signal, then effectively the signal is windowed with a rectangular window and the DFT will not be an estimate of $X(e^{j\omega})$, but an estimate of $X(e^{j\omega})$ convolved with the frequency response of the window. A rectangular window has very nasty side lobs, therefore in practice other windows can be used, such as Hanning, Hamming and Kaiser (see also later in this chapter, for more information on windowing).

For computation of the periodigram the DFT is not the only candidate. Since the DFT is actually a projection of the signal over a set of basis functions that are complex sinusoids, any transform whose basis functions are sinusoids should serve for spectrum estimation. Furthermore the DFT is perhaps not a very effective candidate, because an M-length DFT delivers only $M/2$ frequency components. Other transforms give critical sampling , thus providing M frequency components over M samples transform (but they will generally not give phase/magnitude information). Before such transforms are investigated, some additional information about block transforms is given.

5.3 Block Transforms

5.3.1 Matrix definitions

If x is a certain block of M samples of an input signal, then the transform T of x, X, is computed by

$$X = T^T x \tag{5.6}$$

The ^{T} denotes transposing the matrix. The transform must be invertible (T^{-1} exists) to make reconstruction of the signal possible, but orthogonality of the transform is preferred:

$$T^T = T^{-1} \tag{5.7}$$

Then the reconstruction x is

$$x = TX \tag{5.8}$$

and thus there is no need to calculate T^{-1} for reconstruction. Another advantage is the conservation of energy (similar to the Parceval equations for the Fourier Transform) when using an orthogonal transform.

$$\|X\| = \|x\| \tag{5.9}$$

where $\|x\|$ represents the Euclidean norm,

$$\|x\| = \sum_{k=1}^{M} |x_k|^2 \tag{5.10}$$

5.3.2 Discrete Fourier Transform

As seen in chapter 3, the convolution principle is not immediately shared by the DFT, but by using overlap-add or overlap-save, this problem can be solved. (this can also be seen as circular convolution). As coefficients in a block matrix the DFT is defined by

$$t_{nk} = \sqrt{\frac{1}{M}} e^{j\frac{2\pi kn}{M}} \tag{5.11}$$

if t_{nk} means the element of T in the nth row and kth column (thus k can be seen as frequency component).

5.3.3 Karhunen-Loeve Transform

Another block transform is the Karhunen-Loeve transform (KLT), which is also referred to as the Hotelling transform. This transform is statistically seen an ideal transform. The KLT is an unique orthogonal transform producing a set of uncorrelated coefficients from a non-white signal.

If the covariance matrix R_{xx} of an input block x with zero mean is

$$R_{xx} \equiv E[xx^T] \tag{5.12}$$

then R_{xx} is a symmetric and a Toeplitz Matrix (thus with eigenvectors of even or odd symmetry). The covariance R_{xx} of a transformed block X is

$$R_{xx} = T_{KL}^T R_{xx} T_{KL} \tag{5.13}$$

Now the KLT transform is per definition the matrix T_{KL} that will diagonalize R_{XX} in the form

$$R_{XX} = T_{KL}^T R_{xx} T_{KL} = \text{diag}\{\lambda_0, \lambda_1, ..\lambda_{M-1}\} \tag{5.14}$$

The λ's are the eigenvalues of the system. This is also known as the principal component analysis [29] and it defines that the basis functions of the KLT are the eigenvectors of the covariance matrix of the input signal.

Equation (5.14) implies that the elements of X are uncorrelated and that their variances are given by

$$\text{Var}\{[X]_k\} \equiv \sigma_k^2 = \lambda_k \tag{5.15}$$

By diagonalizing the covariance matrix of X the KLT results in maximizing the energy compaction in X. This means that the energy is concentrated in only a few coefficients (there will be few transform coefficients with large variances and most will have small variances), which is an advantage if the goal is to give the spectrum of a block in as little output coefficients as possible.

However, the KLT is seldom used in audio coding, because it is signal dependent and the precise model of the input is not known.

5.3.4 Discrete Cosine Transform

An asymptotic equivalence exists between the Discrete Cosine Transform and the Karhunen-Loeve Transform [8]. Combined with critical sampling (M frequency components at M input samples) this makes the DCT widespread in signal coding. The definition of the DCT basis is

$$t_{nk} = c(k)\sqrt{\frac{2}{M}} \cos\left[(n+\frac{1}{2})\frac{k\pi}{M}\right] \tag{5.16}$$

where

$$c(k) \equiv \begin{cases} \frac{1}{\sqrt{2}} & \text{if } k = 0 \\ 1 & \text{otherwise} \end{cases} \tag{5.17}$$

Besides being very energy compact, the DCT does not need overlap-add or overlap-save for construction of large datasets and also does not need windowing for a smooth spectrum estimate. It has critical sampling and thus delivers M frequency components for M input samples.

This transform still suffers from the so-called block edge effects. These discontinuities in the reconstructed signal arise when the frequency component of the input blocks are independently processed. Putting them together on reconstruction gives artifacts as the different processing of the blocks can be heard.

5.4 Lapped transforms

5.4.1 Lapped Orthogonal Transform

To find a solution to the block edge effects, lapped transforms were developed. Other possible approaches are pre- and post-filtering and the short-space Fourier Transform (SSFT) [3], but these techniques give a low-pass effect around the block boundaries or in the case of the SSFT ringing around the edges. The recognition that the blocking effects are caused by the discontinuities on the basis functions of the transforms was the key to the development of the Lapped Orthogonal Transform (LOT). The basis functions of the transform must be made longer than the transform length for a lapped transform (see 5.1).

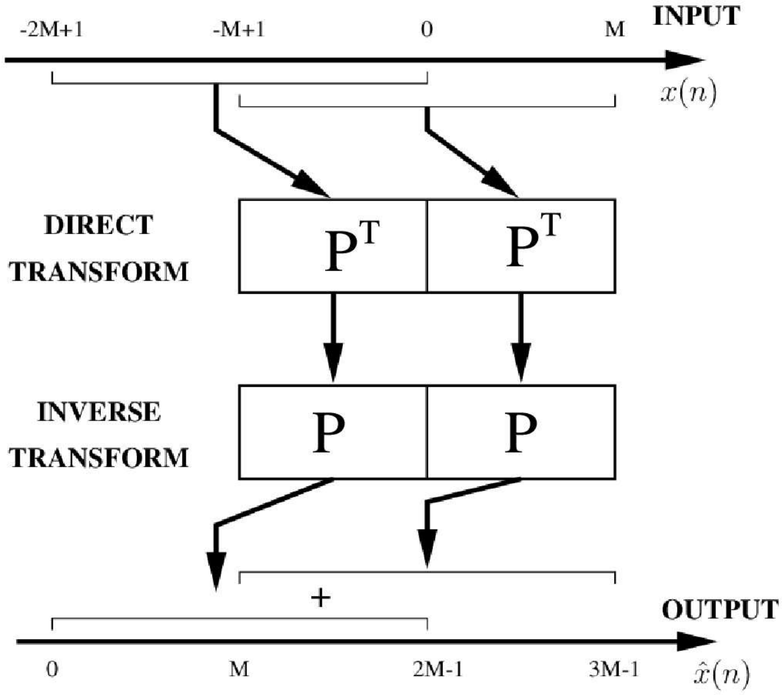

-2M+1
-M+1
0
M
INPUT
$x(n)$
DIRECT TRANSFORM
P^T
P^T
INVERSE TRANSFORM
P
P
+
OUTPUT
0
M
2M-1
3M-1
$\hat{x}(n)$

Figure 5.1 *Signal processing with a lapped transform with 50% overlap*

If a transform has the same functions for direct and inverse form, it must be orthogonal. Keeping this orthogonality at a longer basis imposes extra restrictions due to the extra degrees of freedom.

If x now has $2M$ samples, and P is the M by $2M$ transform matrix, then

$$X = P^T x \tag{5.18}$$

with:

$$x = [x(mM-2M+1) \quad x(mM-2M+2) \quad \dots \quad x(mM-1) \quad x(mM)]^T \tag{5.19}$$

with m the block index. Thus the signal x is given in overlapping blocks as shown in figure 5.1. For reconstruction of a block

$$P^T P = I \quad \text{and} \quad P^T W P = 0 \tag{5.20}$$

with

$$W \equiv \begin{pmatrix} 0 & I \\ 0 & 0 \end{pmatrix} \tag{5.21}$$

This is a very generalized description, a block transform can also be given in this way with $P^T = [0 \quad T^T \quad 0]$.

5.4.2 Perfect Reconstruction

Now extra perfect reconstruction restrictions are applied, to cancel aliasing in the time domain. Perfect reconstruction means that the transform of the spectral decomposition must be invertible. This is a non-trivial problem. If one block of the signal is transformed with the DFT and then converted back with the IDFT there is perfect reconstruction, but if we use a filterbank

$$X_k(m) = \sum_{n=-\infty}^{\infty} x(n) h_k(mM - n) \tag{5.22}$$

where $h_k(n)$ is the kth analysis filter (FIR), then the DFT does not possess perfect reconstruction. When coding a signal using a filterbank for transformation and analysis is standard practice. The reconstructed signal can be written as

$$\hat{x}(n) = \sum_{k=0}^{M-1} \sum_{m=-\infty}^{\infty} \hat{X}_k(m) f_k(n - mM) \tag{5.23}$$

where f_k is the kth synthesis filter, which is needed to transform the coefficients back to the time domain (which is usually called the synthesis bank). If equation 5.22 is substituted in 5.23

$$\hat{x}(n) = \sum_{l=-\infty}^{\infty} x(l) h_T(n, l) \tag{5.24}$$

is obtained, with the time varying impulse response given by

$$h_T(n, l) \equiv \sum_{k=0}^{M-1} \sum_{m=-\infty}^{\infty} f_k(n - mM) h_k(mM - l) \tag{5.25}$$

Then

$$h_T(n, l) = \delta(n - l - D) \tag{5.26}$$

with D optional delay between h and f. The l which was silently introduced is the index of $h(n)$ and L is total length of $h(n)$. Actually only transforms where $L = 2M$ (which means that the basis of the transform is doubled) are treated here, but the given definitions hold for the general case.

A filterbank pair $h(n)$ and $f(n)$ must satisfy condition 5.26 to obtain perfect reconstruction. It took over a decade to find such a solution as described in the next section. Here we show that there is no feasible FIR-filter bank solution for the DFT. In the DFT filterbank we need modulation to shift the center frequency to the origin, before applying filter $h(n)$. After interpolation the filter $f(n)$ is used, so

$$X_{M-k}(m) = X_k^*(m) \tag{5.27}$$

[8] shows that such a filter can have perfect reconstruction if

$$f(n) = h(n) = \begin{cases} 1, & \text{if} \quad 0 \leq n \leq M - 1 \\ 0, & \text{otherwise} \end{cases} \tag{5.28}$$

If $M > 2$ then all possible FIR solutions for $h(n)$ and $f(n)$ have polyphase components order zero [28], which means that $h(n)$ has length M and $f(n)$ is its reverse. Using the reverse of a proper window leads to bad low-pass behavior and therefore a practical solution does not exists.

Note that this doesn't mean there are no good filterbanks possible with the DFT, just that there can not be perfect reconstruction if a filterbank is used in conjunction with the DFT. If $L >> M$ the aliasing is very small and QMF-filters can also lead to good results. However, considering all other good properties of LOT and perfect reconstruction, this falls outside the scope of this thesis.

5.4.3 Modulated Lapped Transform

When $L = 2M$ filterbanks based on Perfect Reconstruction principles have Time Domain Aliasing Cancellation (TDAC). To obtain the proper filterbank and transform together, equations 5.21 and 5.26 have to be satisfied. TDAC compensates for the folded frequencies above the Nyquist frequency, by overlap-adding them in subsequent blocks.

The basis functions of a modulated lapped transform and the corresponding filterbank are given by

$$p_{nk} = f_k(n) = h(n) \cos \left[(k + \frac{1}{2})(n + \frac{L-1}{2}) \frac{\pi}{M} + \phi_k \right] \tag{5.29}$$

for $k = 0, 1, ..M - 1$ and $n = 0, 1, ..L - 1$. Assuming that $L = NM$ the phases ϕ_k are restricted by

$$\phi_k = (k + \frac{1}{2})(N + 1)\frac{\pi}{2} \tag{5.30}$$

Now the MLT (or as it is also called the Modified Discrete Cosine Transform) can be written as

$$p_{nk} = h(n)\sqrt{\frac{2}{M}} = \cos \left[(n + \frac{M+1}{2})(k + \frac{1}{2}) \frac{\pi}{M} \right] \tag{5.31}$$

where $\sqrt{\frac{2}{M}}$ was introduced for normalization. For perfect reconstruction $h(n)$ must fulfill

$$\begin{aligned} h^2(n) + h^2(n + M) &= 1 \\ h(L - 1 - n) &= h(n) \end{aligned} \tag{5.32}$$

Some examples of windows which satisfy these conditions can be found in the next section. If we give the MDCT as a function on the input signal it can be defined as

$$X(m) = \sum_{k=0}^{N-1} x(k)h(k)\cos(\frac{\pi}{2N}(2k+1+\frac{N}{2})(2m+1)) \quad m = 0,\cdots,\frac{N}{2}-1 \tag{5.33}$$

The inverse transformation is also needed and is

$$y(p) = \frac{4}{N}\,h(p)\sum_{m=0}^{\frac{N}{2}-1} X(m)\cos(\frac{\pi}{2N}(2k+1+\frac{N}{2})(2m+1)) \quad m = 0,\cdots,N-1 \tag{5.34}$$

In appendix B it is shown how the MDCT can easily be calculated from the DFT, and thus is fast and easy to calculate.

5.5 Windowing

In the previous section it was shown that lapped transforms are applied to an input signal in blocks. The length of these blocks is an important issue, because the duality between the time- and frequency domain gives you preciseness in only one domain, depending on these block length. In other words, a long block provides a detailed frequency resolution of a signal, while a short block gives a more detailed time resolution.

For a smooth overlapping of the blocks the 2:1 decimation was canceled in the inverse transformation if the window used, fulfilled the restrictions given by equation 5.32. A popular window, which satisfies these conditions is the half sine window, displayed in 5.5 and defined by

$$h(n) = \sin[\frac{\pi n}{N}] \qquad \Big\} \quad n = 0\cdots N-1 \tag{5.35}$$

where N is the size (in samples) of the window. The technique of switching windows as used in the AAC [3] according to the characteristics of the input block can be used to improve the quality of an encoder. As mentioned earlier, another method to improve the quality of a coder, especially for encoding transients, is to adopt the window size

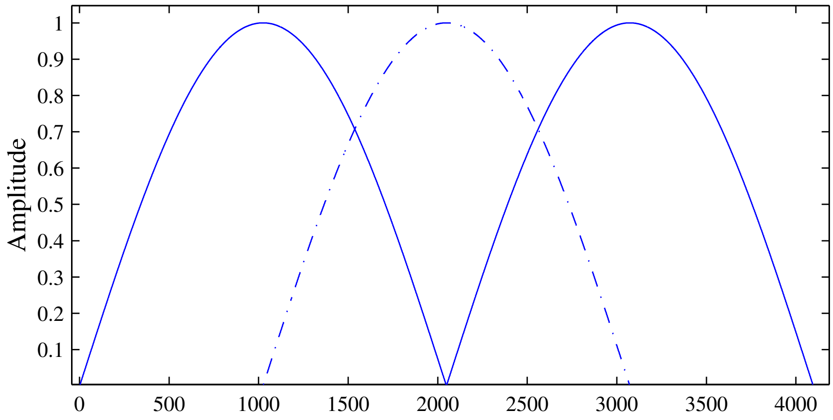


Figure 5.2 *Half Sine Window for Overlap Add Sequence*

switching. It is important to conserve the ability of perfect reconstruction. The filterbank which is formed using such a method is called a multirate filterbank. The method the AAC uses to switch to shorter windows is plotted in figure 5.5.

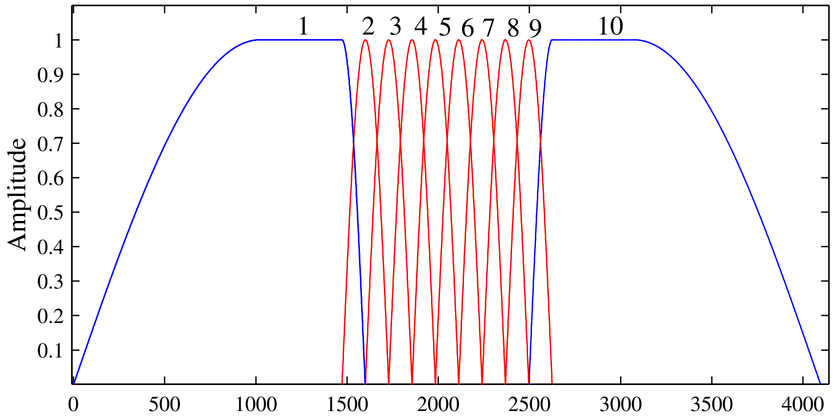


Figure 5.3 *Comparison for Window Overlap Add for Steady State and Transients*

When a transient is detected a switch to a short window occurs (windows nr. 2-9 in figure 5.5). Window number 1 and 10 are special windows, the so-called start and stop windows. The short windows come in groups of eight, because the total length of the short windows then overlaps with the large windows. This helps the alignment of the blocks if multiple channels are used, but this is not a requirement.

To maintain perfect reconstruction the start window is defined by

$$h_{start} = \begin{cases} h_{long}(n), & 0 \leq n \leq M-1 \\ 1, & M \leq n \leq M + \frac{M}{3} - 1 \\ h_{short}(n-M), & M + \frac{M}{3} \leq n \leq M + \frac{2M}{3} - 1 \\ 0, & M + \frac{2M}{3} \leq n \leq 2M-1 \end{cases} \tag{5.36}$$

The complete short window is defined by equation 5.35 for the number of samples of the next window, the long window by the number of samples of the previous window. The stop window has a similar definition (but mirrored). The example in 5.5 illustrates that for a long window of 2048 and a small window of 256 samples.

Chapter 6

Subband Coding

6.1 Properties

At the TU Delft, Sonke [1] has developed a compression method for an impulse response. Hulsebos [7] has further evaluated this method. This method is based on equations derived from Patterson, Plack & Moore [7]. According to them the amplitude spectrum of the auditory filters can be modeled by

$$A(f, f_c) = (1 + \frac{4|f - f_c|}{W(f_c)}) \exp^{\frac{4|f-f_c|}{W(f_c)}} \tag{6.1}$$

In this equation f_c is the center frequency, $W(f_c)$ denotes the equivalent rectangular bandwidth.

$$W(f_c) = 6.23 \cdot 10^{-6} f_c^2 + 93.4 \cdot 10^{-3} f_c + 28.5 \tag{6.2}$$

The temporal masking of the human ear is also taken in account. In this approach no discrimination is made between post en pre-masking. Meanwhile the frequency dependent behavior of the temporal masking is taken in account (see table 6.1).

6.2 Band Filters

In the coding process the impulse response is first split into frequency bands. This filterbank is constructed so that the sum is always unity (for keeping the gain constant))

f(Hz)	t(ms)
≤ 200	35
300	13
900	9
≥ 2700	8

Table 6.1 *Time Integration Lengths of the Human Auditory System*

and the bandwidth matches equations 6.1 and 6.2.

This filterbank is given by

$$A(f, f_i) = hann\ \{\log(\frac{b - \sqrt{b^2 - 4ac} + 2af}{b - \sqrt{b^2 - 4ac} + 2af_i}) - \log(\frac{b + \sqrt{b^2 - 4ac} + 2af}{b + \sqrt{b^2 - 4ac} + 2af_i})\} \tag{6.3}$$

where 'hann' represents a unit width Hanning window, $a = 6.23 \cdot 10^{-6}$, $b = 93.4 \cdot 10^{-3}$ and $c = 28.5$. In figure 6.1 some of these scaled Hanning windows are given and they are compared to the original Patterson windows (given by equation 6.1).

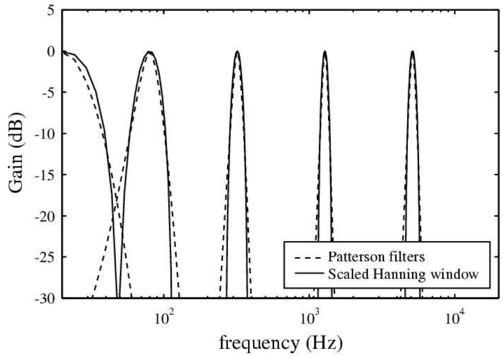


Figure 6.1 *Patterson versus scaled Hanning Windows*

6.3 The parametrization process

As figured in 6.2, the compression process consists of several phases. The first step is to use the filterbank of scaled Hanning windows and to convert the IR to the frequency domain. For the full audio spectrum, 38 bands are used. Then the signal is Hilbert transformed and the result is integrated over time using the values of table 6.1. The result can then be down-sampled to about 50 samples in the low frequency bands until 200 samples in the upper frequency bands. Only perceptual relevant data is obtained this way, according to the temporal masking principle.

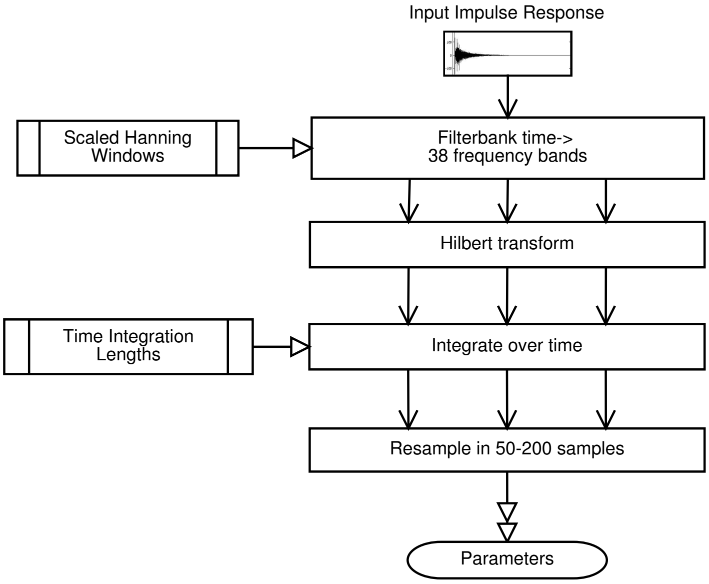

Input Impulse Response
Scaled Hanning Windows
Filterbank time-> 38 frequency bands
Hilbert transform
Time Integration Lengths
Integrate over time
Resample in 50-200 samples
Parameters

Figure 6.2 *Overview of the subband coding method*

6.4 The reconstruction process

The parameters generated by the process described in the previous section are related in the spectral and temporal properties (see also [1]). Modifying the spectral properties of a certain signal effects the temporal properties and vice versa. This mutual dependency can be quantified with a number of properties of the Fourier transform. One of this properties is the modulation property:

$$a(k) \cdot b(k) \overset{F}{\leftrightarrow} A(\omega) * B(\omega) \tag{6.4}$$

with $a(k)$ and $b(k)$ two arbitrary time domain signals and $A(\omega)$ and $B(\omega)$ their Fourier transforms respectively.

Proper reconstruction can be done by using white noise, with a dense and irregular structure in both time and frequency domain. Because of the variations in the generated noise, the parameters of the noise signals must also be accounted for. An overview of the process can be seen in figure 6.3.

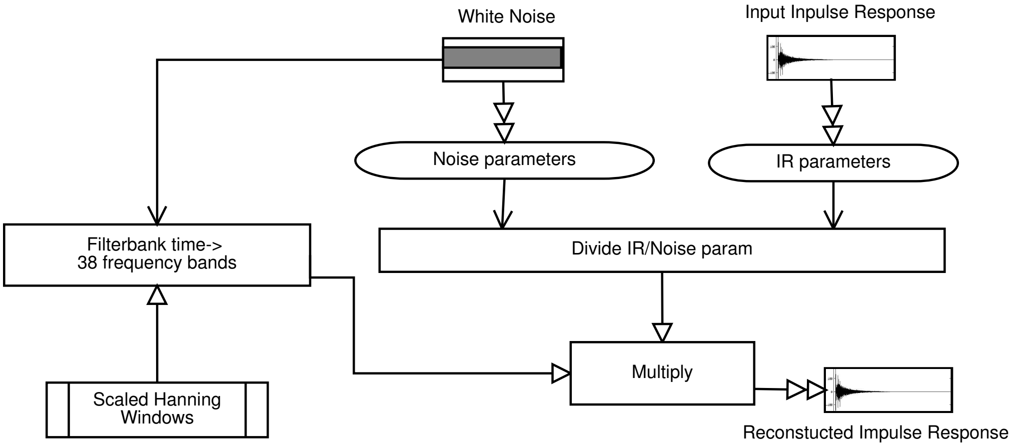

White Noise
Input Inpulse Response
Noise parameters
IR parameters
Filterbank time-> 38 frequency bands
Divide IR/Noise param
Multiply
Scaled Hanning Windows
Reconstucted Impulse Response

Figure 6.3 *Reconstruction process of the subband coder*

If $P(f_i, t)$ are the parameters for the impulse response and $N(f_i, t)$ the parameters for the noise, the reconstruction process can also be given by

$$p(t) = \sum_i \frac{P(f_i, t)}{N(f_i, t} \cdot n(f_i, t) \tag{6.5}$$

where $p(t)$ is the reconstructed impulse response, $n(f_i, t)$ is the band filtered noise signal, for which the same filterbank must be used as for the construction of the original impulse response. Naturally, using this method just returns an approximation of the original impulse response.

Chapter 7

Transform coding

7.1 Overview

In this chapter the transform coding approach of compressing an impulse response is described. Further it contains some specific methods, using the knowledge of impulse responses as described in section 3.1.

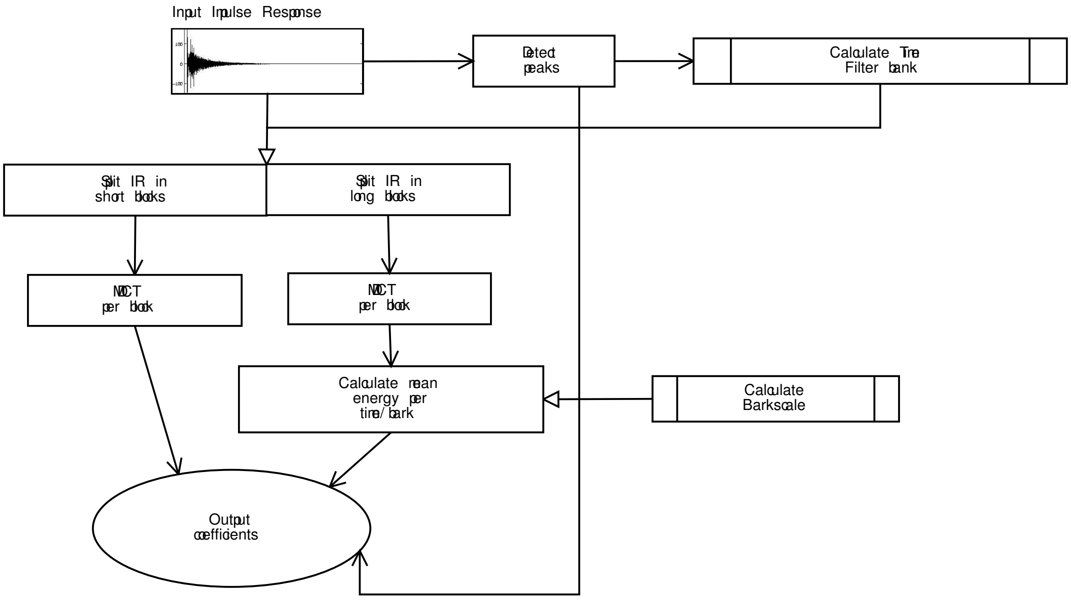

Input Impulse Response
Detect peaks
Calculate Time Filter bank
Split IR in short blocks
Split IR in long blocks
MDCT per block
MDCT per block
Calculate mean energy per time/bark
Calculate Barkscale
Output coefficients

Figure 7.1 Overview of the transform coder

In figure 7.1 and 7.2 an overview of the transform coder is given. The next section, about window switching, will give more details, starting with the explanation of 'Detect Peaks' and 'Calculate Filterbank'. Section 7.5 contains further detail about the reconstruction process.

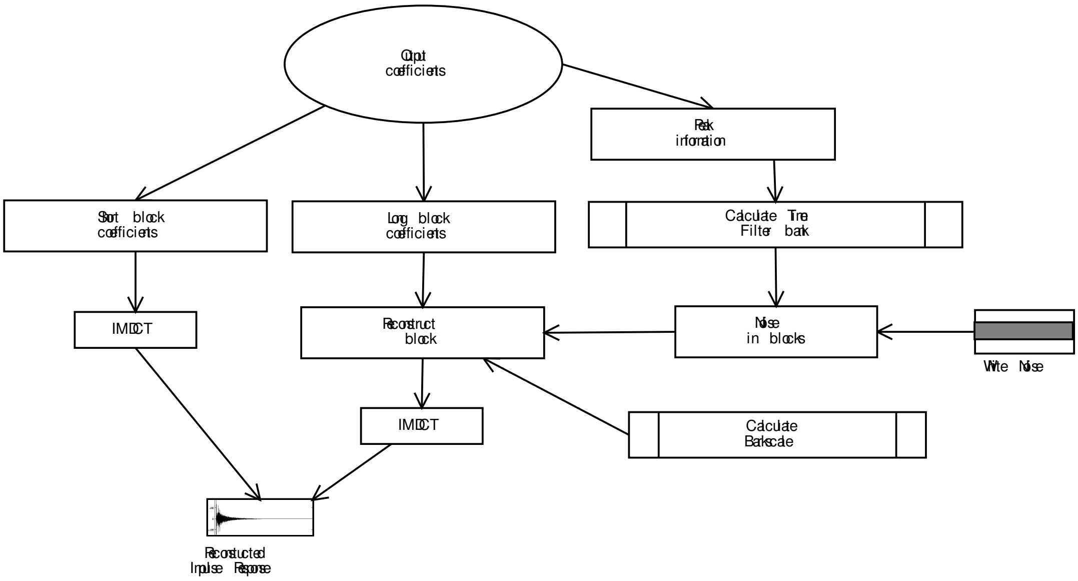

Output coefficients
Peak information
Short block coefficients
Long block coefficients
Calculate Time Filter bank
IMDCT
Reconstruct block
Noise in blocks
White Noise
IMDCT
Calculate Barkscale
Reconstructed Impulse Response

__Figure 7.2__ Reconstruction process of the transform coder

7.2 Window switching

It has been shown that the size of the windows used to approximate the spectral components of an audio signal lead to an accurate description of the signal in either the time or the frequency domain. Changing the size of a window with a certain switching technique can therefore be very useful when coding impulse responses. If the direct sound peak of a reconstructed impulse response is not a near exact copy of the original, this will be easily distinguishable. The subband coding method described in the previous chapter failed to accurately do this and therefore the original direct sound was send to the coding engine as extra data. But a precise reproduction of the early reflections without too much exposing to this problem, was found to be important either.

The window switching scheme of the AAC-coder was an interesting candidate for our coding engine. First the MPEG2-spec [3] was exactly followed and thus the switching consisted of a start window of 2048 samples (46 ms), followed by 8 windows of 256 samples (6 ms), followed by a stop window to switch to long windows again as shown in figure 5.5. To do this a criterion is needed which states at what exact position in time a switch in window-length should occur.

The traditional approach of audio coders is to compare the coding gain against the perceptual entropy for a block and transform this block with the window size which maximizes this coding gain. This coding gain can be calculated with equation 4.4. When coding an impulse response this approach is not optimal, since it is more important to provide a proper overlapping of the short windows with early reflections, than to obtain the best possible bit rate.

Some speech coders [4] use a more loose method for detecting peaks. In chapter 4 it was shown that forward temporal masking is a longer lasting effect than backward masking and therefore it is preferred to focus on the rising of the signal, instead of the falling. This effect can be used as follows: in the time domain a local estimate is made of the change in signal energy, by splitting the input in blocks of N samples and calculating the energy of the samples in each interval. The total summed energy e_j in block j is then compared to the energy in the consecutive block:

$$r = \frac{e_{j+1} - e_j}{e_j} \tag{7.1}$$

If r is higher than a certain threshold value, a transition to shorter windows will occur. After one or more short windows, a longer window is used again. If e_j is lower than a factor k of the maximum e_j the transition will be suppressed to prevent smaller, less important peaks to be coded in a small window. Another practical condition which can be used is that a switch should only occur in the first T milliseconds of the impulse response. At $t > T$ there is only reverb and no direct reflections, so switching to smaller windows and thus more data wastes bandwidth.

Typical values that can be used are

$$r = 0.2, N = 32, k = 0.2, T = 200\,\mathrm{ms} \tag{7.2}$$

This method causes an encoding in a variable bit-rate, since it is not known in advance how many short windows are going to be used. Choosing a value of the parameters in 7.2 suited for different types of inputs is difficult; the number of windows depends a lot on the level of reverb in the impulse response. Therefore another method was also tried. The number of early reflections can be estimated in advance with some extra knowledge of the acoustics of the room where the impulse responses are recorded. A fixed number is chosen (for example 16) based on this knowledge and then this number of the highest peaks in the impulse response are mapped with a short window.

Disadvantage of this last method is that the number of reflections can not be found automatically. Therefore the two last methods were tested, namely the approach with fixed number of reflections and the approach defined by equation 7.1 and 7.2.

7.3 Window design

Since the Modulated Lapped Transform is the transform used, the perfect reconstruction conditions (equation 5.36) must be fulfilled. A proper window must still be chosen. The N samples long half-sine window as given by

$$h_{hf}(n) = \sin\left[\frac{\pi}{N}(n + \frac{1}{2})\right] \tag{7.3}$$

is the unique window which also satisfies polyphase normalization [8]. This window is certainly a candidate. AAC also uses the Kaiser Bessel Derived window given by

$$h_{kbd}(n) = \sqrt{\frac{\sum_{i=0}^{n} \mathcal{W}(i)}{\sum_{i=0}^{N-1} \mathcal{W}(i)}} \tag{7.4}$$

with $\mathcal{W}(i)$ as the Kaiser Bessel kernel window function defined as follows

$$\mathcal{W}(i) = \frac{I_0\left(\pi\nu(1 - (\frac{i-N/4}{N/4})^2)\right)}{I_0(\pi\nu)} \tag{7.5}$$

where I_0 is the modified zero order Bessel function of the first kind and ν is the parameter of the window. The use of different windows has consequences for the frequency separation of the algorithm.

Figure 7.3 and 7.4 show the frequency responses of both windows. According to [14] from the point of view of transform coding and compression efficiency in particular two specific properties of the transform window are extremely important:

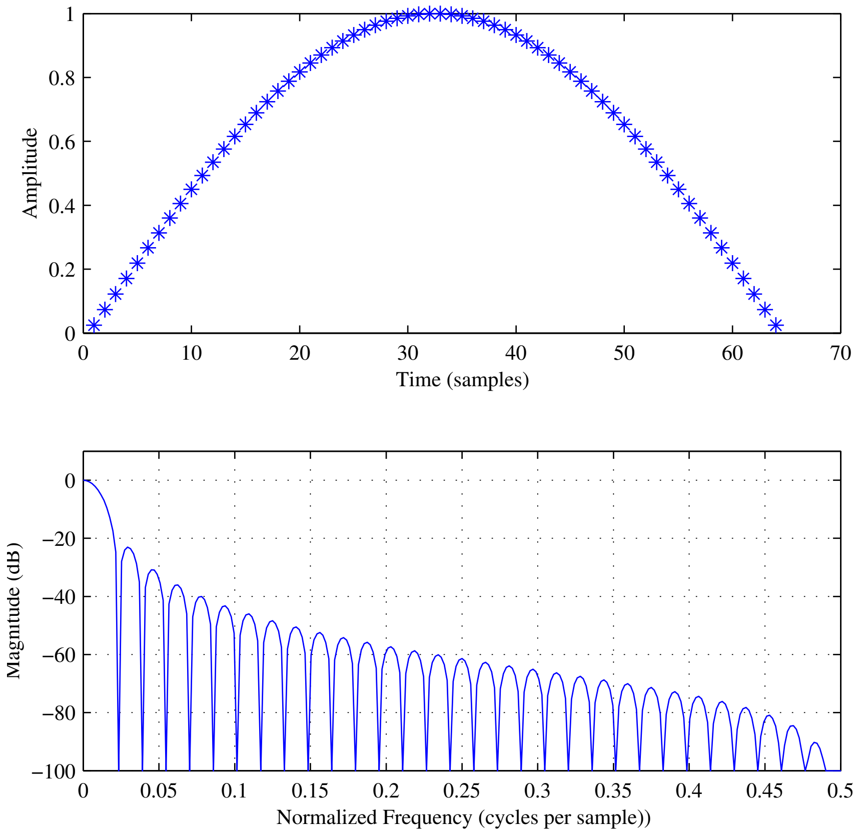


Figure 7.3 *Frequency selectivity of the half-sine Window*

- *Monotony*: the envelope of the stop-band attenuation must be monotonically decreasing or equiripple since it is desirable to confine the spreading of the frequency quantization errors. In other words it is desirable to achieve an accurate coloration of the quantization noise.

- *Selectivity*: the main lobes (pass-band) of the frequency response must be as narrow as possible and the stop-band leakage must be minimized

The half-sine window is a reasonable window, with respect to monotony, but the KBD window has better selectivity properties, although the main lob is a bit wide, the ultimate frequency rejection is better. Window design is a rather advanced topic; here just the equation of the used window is given, without an explanation of all the problems attached to the design of a window. To avoid switching between different window shapes,

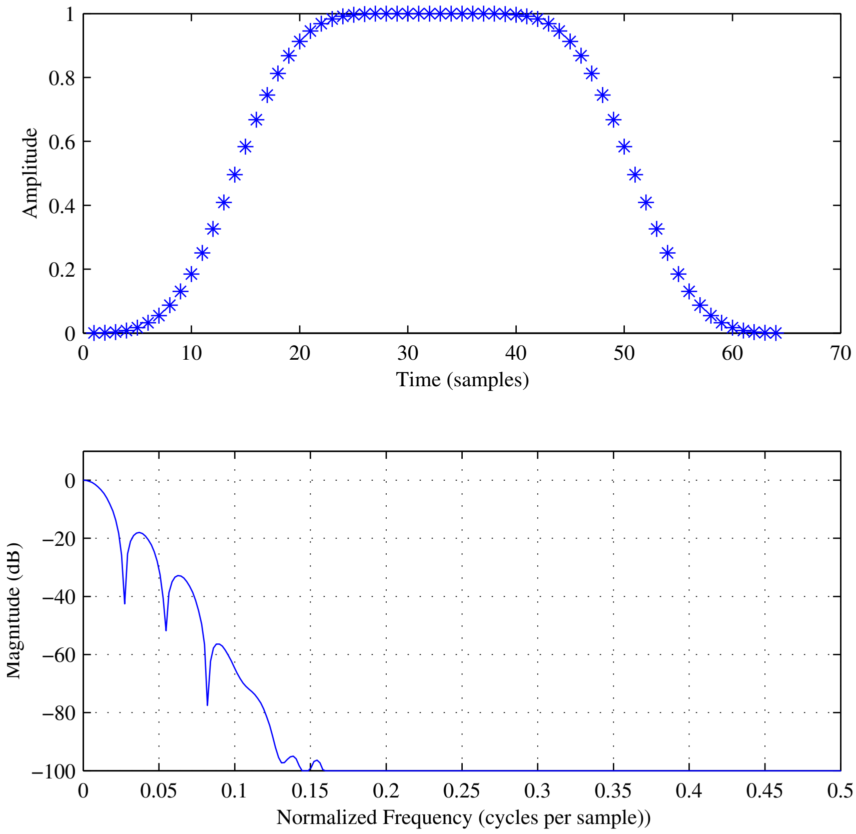


Figure 7.4 *Frequency selectivity of the Kaiser Bessel Derived window with $\nu = 6$*

the Ogg Vorbis coder [32] uses only one window, here called the 'designed' window, which is given by

$$h(n) = \sin\left[\frac{1}{2}\pi \sin\left(n + \frac{1}{2}\right)^2\right] \tag{7.6}$$

This window can be used in the MDCT because it satisfies the perfect reconstruction condition. Further the frequency response as shown in figure 7.5 combines some features of both the half-sine window and the Kaiser Bessel Derived window, thus making it the preferred candidate for this coder.

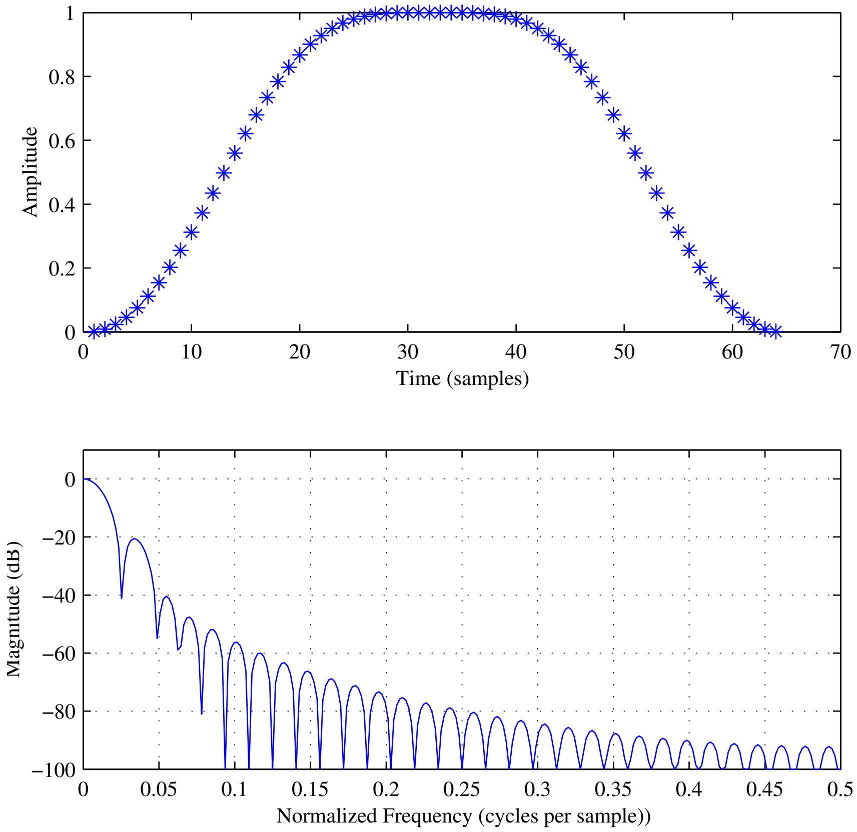


Figure 7.5 *Frequency selectivity of the designed window*

7.4 Bark filterbank

In chapter 4 the Bark scale was introduced as a scale which approximated the critical bands of the human hearing system. Most audio coders use the Bark scale to employ the masking properties of the human ear in bands. Zwicker's formula for conversion of a frequency scale to a Bark scale is:

$$z(f) = 13\arctan(0.00076f) + 3.5\arctan[(\frac{f}{7500})^2] \tag{7.7}$$

But also other conversions exist. Traunmüller proposes:

$$z(f) = \frac{26.81f}{1960 + f} - 0.53 \tag{7.8}$$

Traunmüller also proposes a different low and high frequency approximation, to obtain a curve which is in good agreement with measured statistical data.

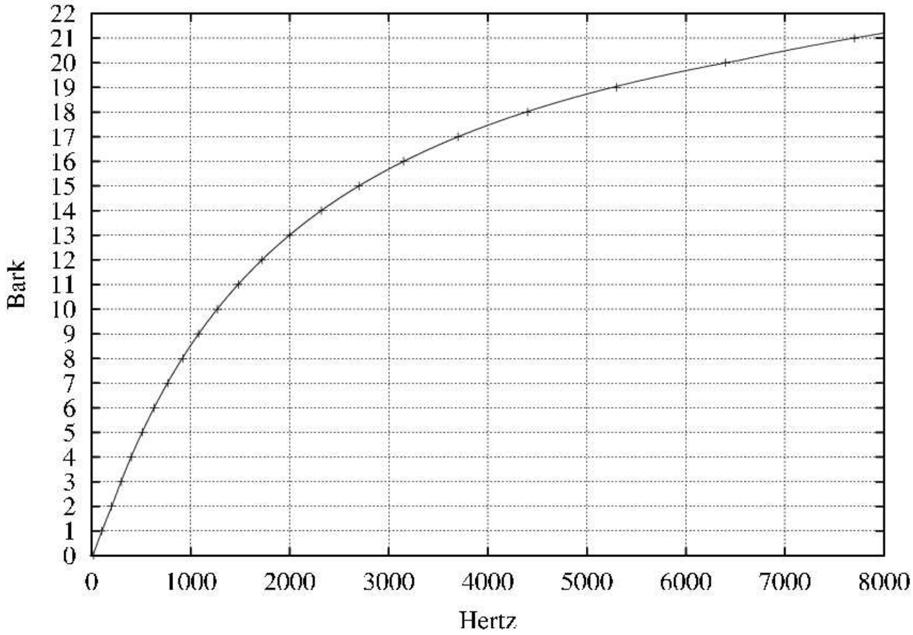


Figure 7.6 *Plot of bark-transformed frequency against frequency in Hertz*

There is no unique way to develop a filterbank from the Bark scale. If the Bark table is used to form rectangular block filters this will give sharp edges to the frequency band transitions. In the previous chapter the parameters derived from the impulse response were smoothed with a window in the time domain and the frequency domain.

Audio coders generally store all spectral components as a masking threshold function together with a residue per sample encoded in as less bits as possible. The audio artifacts are often a result of the quantization of the residues. In our impulse response coder a much better compression ratio is possible if only one output coefficient per band is saved, analog to what the subband coder in the previous chapter does in the time domain. Therefore a filterbank which approximates the critical band in the human hearing system was sought.

The curve of the Bark critical bands is not known and depends on the sort of masking effect. In audio coders it is often assumed that the windows are triangular with respect to a masked tone, however this is done to keep the computational complexity low. In figure 7.7 a filterbank is shown based on cosines windows, which was one of the proposed filterbanks for our coder. The output coefficients were derived by multiplication per window in the frequency domain and summed to obtain one parameter per band.

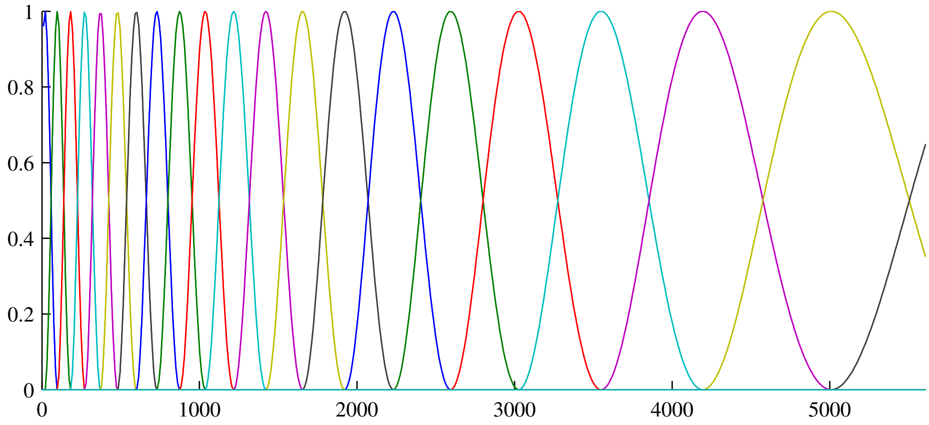


Figure 7.7 *Possible Bark filterbank with cosines shaped windows over 4096 samples*

As explained before the transform coder described here, uses blocks of different sizes in the time domain to account for the pre-echo effect. If a small frame (containing a low number of samples) is used for transformation, it is very difficult to approximate a proper Bark filterbank for that block. In other words, calculating a filter over a very small number of samples leads to a distorted filterbank. The inaccuracy in the approximation of the window can be compensated for if the original signal is reconstructed with white noise, just as in the subband coding method by dividing the parameters by the output coefficients of the noise. However the small number of noise samples in the low frequencies still give a high uncertainty factor. Further the bands are not reconstructed as intended. For this reason the MDCT-coefficients are not divided in bands, but simply store all.

7.5 Spectral coding

Spectral coding of the Bark components with one parameter per Bark is also possible without the use of a filterbank in the frequency domain. In this section a hybrid method for this reconstruction is proposed.

In the subband coding method of the previous chapter, the impulse response was constructed as a shaped noise signal. Here a similar approach is used. The coefficient in a certain band is just the summation of all energies of the samples in that particular band. The boundaries of these bands as well as the center frequencies can be found in appendix A. This can be seen as the problematic block band filter, however, since the reconstruction will make no use of filterbanks, the edges of the bands will not give problems.

First step in the reconstruction is to approximate a marginal spectrum. This is done by looking at the parameters as representing the center frequency of their Bark band. The center frequencies of the bands are given in the Bark table of appendix A. The position of these center frequencies are scaled on the n samples of a block. The process is outlined in figure 7.5. The first graph shows the original spectrum. In the second graph the summation of energies of the samples per Bark band are plotted as a function of the band.

In the third graph the parameters are plotted, scaled to the sample corresponding to the center frequency of the band. Then over these n samples a linear interpolation is carried out which results in a marginal spectrum. This linear interpolation can be seen in the same graph by the lines connecting the parameters.

At this point the spectrum of a white noise signal is calculated over the same amount of samples. The noise signal has by definition a flat spectrum; in the fourth graph of 7.5 an example is plotted logarithmically. Finally the reconstructed shaped noise spectrum is the multiplication of the noise spectrum with the interpolated Bark parameters. This MDCT spectrum can then be converted back to the time domain with the same filterbank as was used for the initial transform.

Compensation of the parameters for the noise, as in the subband coder, is not necessary because there is no erroneous quantization of the window. And unlike a transform with the DFT, with the MDCT the variance σ of a white noise signal is the same in the time- and frequency domain.

To obtain the reconstructed signal in the time domain, the windows have to be overlapped as stated in the section about windowing. The Matlab code which calculates the proper window sizes given the positions in time of the early reflection can be found in Appendix C.

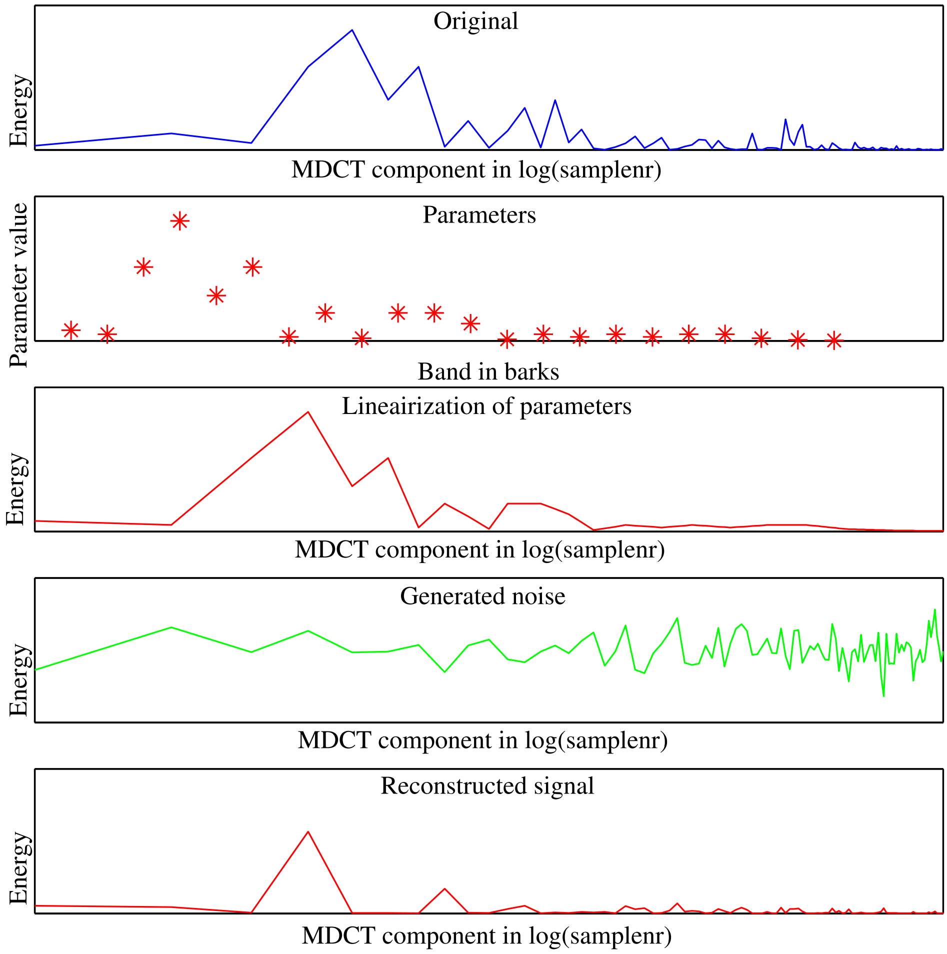


Figure 7.8 *Process of Bark reconstruction*

7.6 Compression Ratio

By looking at the transform coding proposal for compressing impulse responses it is clear that there are some degrees of freedom which determine the real compression ratio. Important parameters in this are the size of the large windows (the windows overlapping the reverberation tail and the parts between the early reflection) and the short windows, overlapping the direct sound and early reflections. Other free parameters include the number of early reflections/short windows taken into account and the quantization of the output coefficients.

The storing precision by which the output coefficients are stored determines a large part of the final compression ratio. Quantization with 4 bits, will provide $2^4 = 16$ different values for the coefficients, which is probably enough already. The linearizion and multiplication with noise helps to avoid the usual quantization errors/problems.

Besides storing the Bark band parameters, also a mechanism for storing the window sizes in time is necessary for a reconstructible encoding. Here is chosen to save the location in samples of the early reflections and direct sound (and thus of the small windows) and recalculating the filterbank with the same algorithm as during the compression (appendix C). The amount of data this delivers can be neglected in comparison with the amount of data of the output coefficients.

The algorithm does not scale with higher sample rates or higher bit rates of the input signal. In other words, compressing a 44 KHz/16 bits input signal provides the same number of output coefficients as compressing a 192 KHz/ 24 bits signal, but the compression ratio of the latter will be much higher.

Chapter 8

Results

8.1 Comparison of the algorithms

In this section the subband- and of the transform impulse response coder are compared and the differences in the obtained results discussed. Since the goal of this thesis is to develop the transform coder, the subband coder will only receive some attention here. The original impulse response shown in figures 8.1 and 8.2 is recorded in the Amsterdam 'Concertgebouw' and is indicated in this chapter as a 'much reverb' environment, since it is recorded relatively far from the stage, leading to a long diffuse tail.

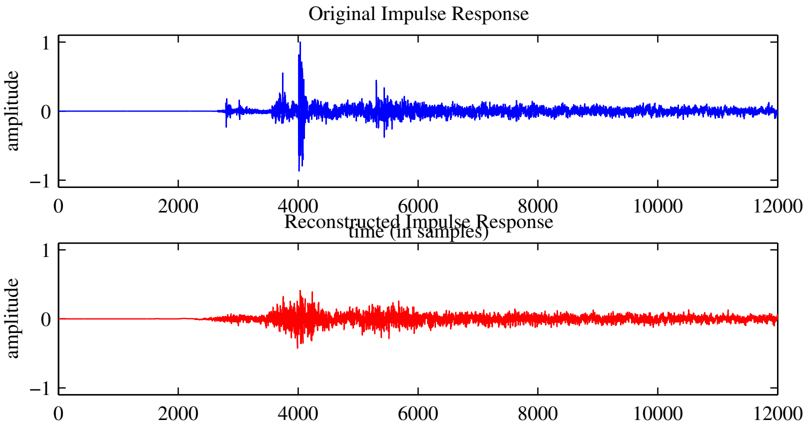


Figure 8.1 *Time domain representation of the original and reconstructed Impulse response using the subband coder*

	Subband coder	**Transform coder**
Number of frequency bands	38	26
Smallest time window	352	128
Longest time window	1543	2048
Percentage short windows	63%	12.5 %
Total number of parameters	9310	3488

Table 8.1 *Comparison of the number of parameters used in the subband and transform coder for an impulse response of 131072 samples*

The compression ratio is not only different for both encoders, it also depends on some initial parameters as described in section 7.6. In table 8.1 some typical parameters are given for compression of the 'much reverb' impulse response from figures 8.1 and 8.2. The subband coder employs various time window lengths depending on the frequency band, as given in table 6.1. In the transform coder the impulse response is first transformed to the frequency domain and the small windows are placed on a fixed number of peaks, as explained in the previous chapter. For the transform coder the windows overlap each other, thus delivering twice as much parameters per block.

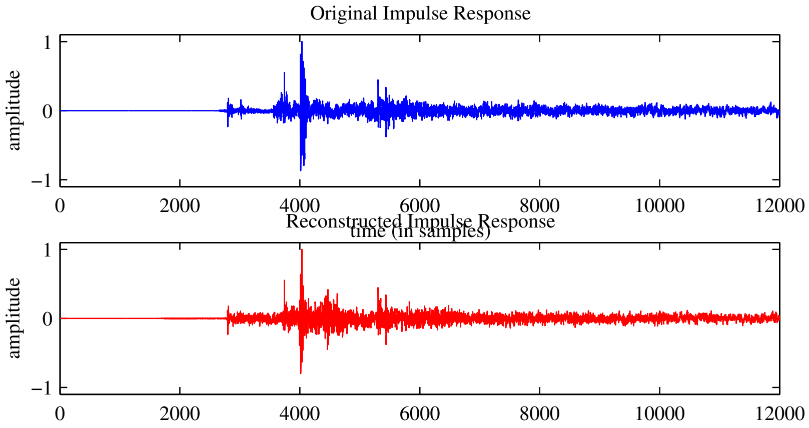


Figure 8.2 *Time domain representation of the original and reconstructed Impulse response using the transform coder*

It is clear from table 8.1 that the transform coder uses less output coefficients to store the compressed impulse response and thus reaches a higher compression ratio. More so the parameters in this method can be quantized, giving even higher compression factors (see 4.4). The parameters resulting from the subband coder may also be quantized, but this is more difficult because the quantization should be carried out in the time domain instead of the frequency domain. Also the computational complexity of the transform coder is lower. It is expected that the general calculation time of the subband coder is in the order of five times slower (most code was not optimized and written in Matlab).

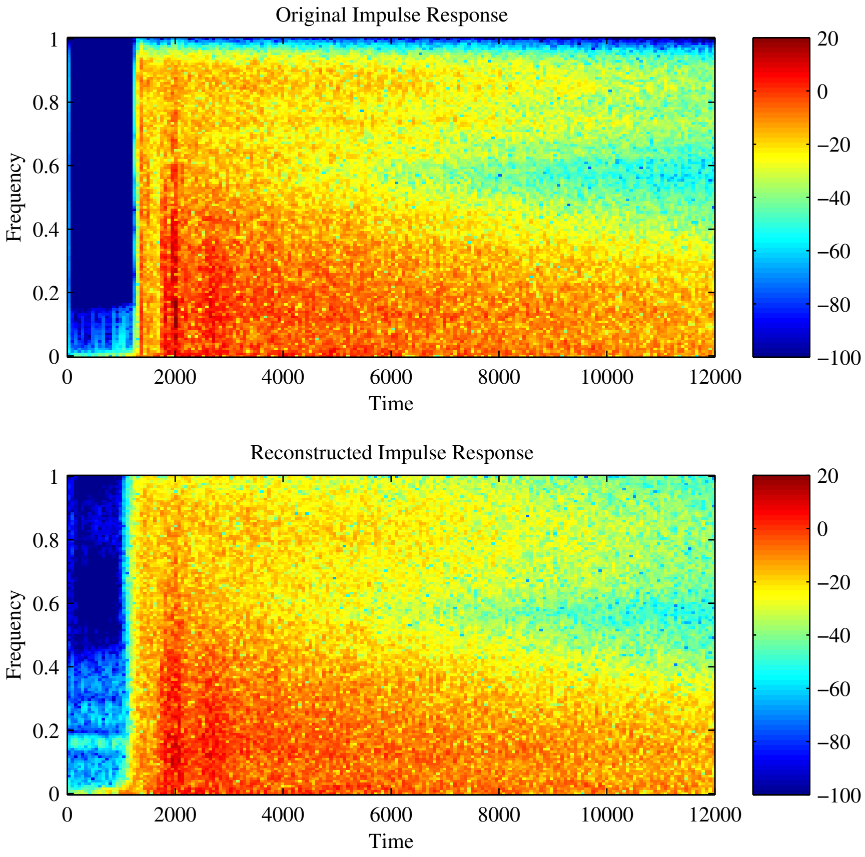


Figure 8.3 *Spectrum of the original and reconstructed Impulse response using the subband coder*

The quality of the reconstruction differs. In figure 8.1 and 8.2 the reconstructed version of an encoded impulse response using the subband respectively the transform coder are shown. The absence of peaks in the subband encoded impulse response is striking: the direct sound does not exist in the reconstruction. Hulsebos [7] assumes that the parametrization with the subband coder should include the non-encoded direct sound information and only the reverberation tail must be encoded. The early reflection part of the impulse response is not mentioned. The impulse response shown in the figures is measured at a certain angle (the listener does not look to the source position) and therefore the largest peak arrives at the listener around sample 1000 (which corresponds with 22 ms). Other peaks also disappear in the subband coder. The transform coder does not have this problem. In fact the number of peaks that must be kept intact is an initial parameter; in this example 8 peaks were kept.

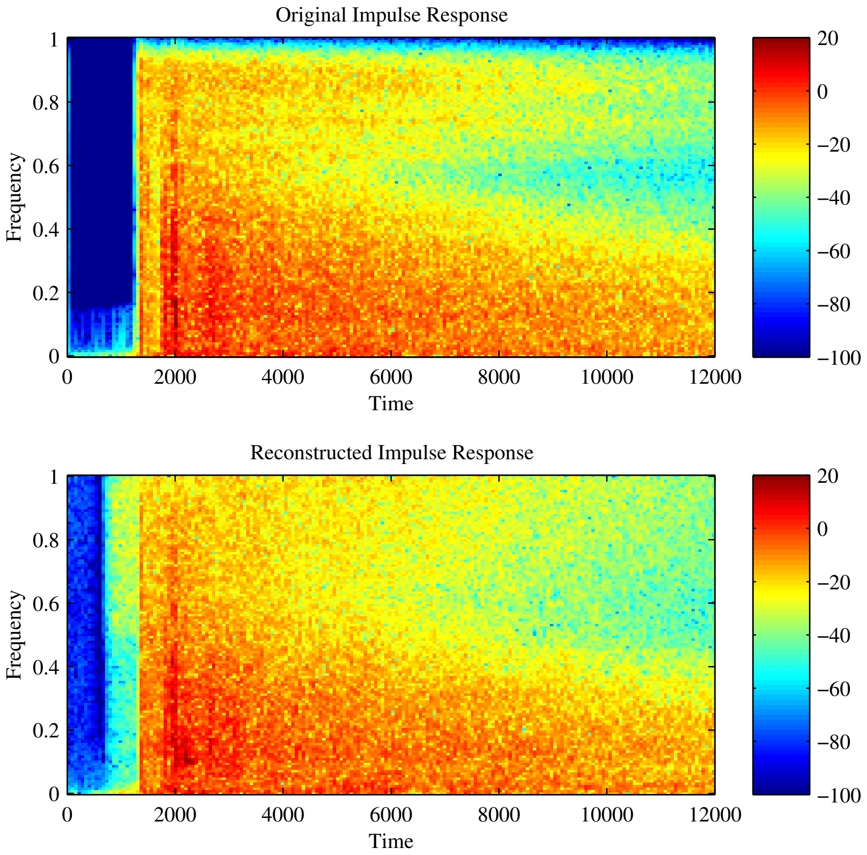


Figure 8.4 *Spectrum of the original and reconstructed Impulse response using the transform coder*

One can also compare the periodigram of the reconstructed impulse responses. The periodigrams of the 'Concertgebouw' impulse response measurements are plotted in figures 8.3 and 8.4 and show the change in the (normalized) spectrum over time. To calculate this spectrum over time the signal is windowed with Hanning windows and Fourier transformed over 256 samples.

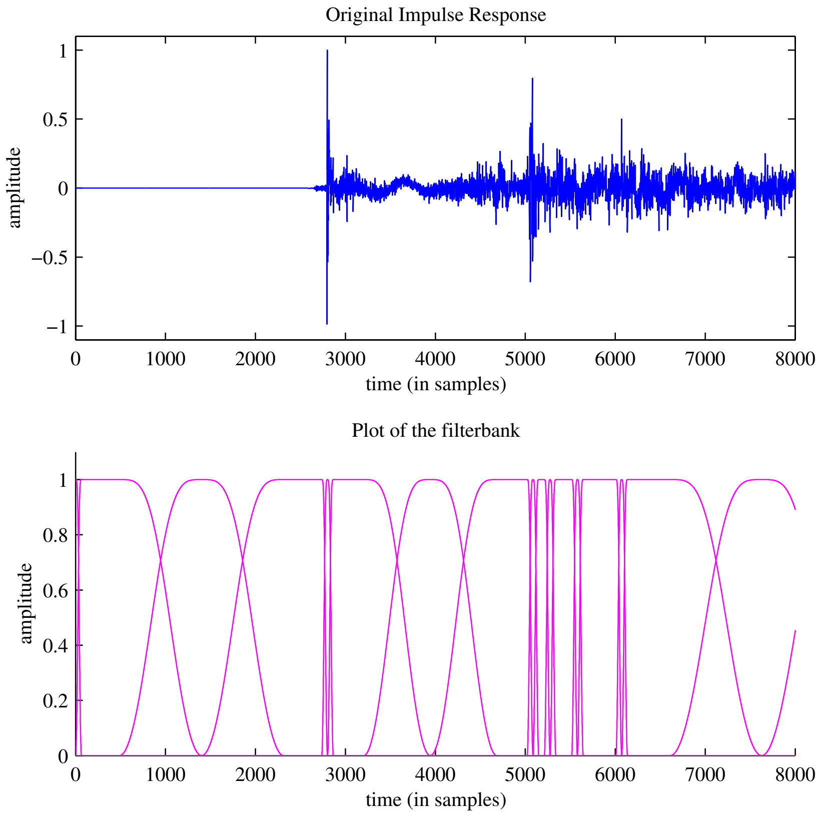
Here the subband coder seems to reach a more precise result than the transform coder. Especially in the high frequency area the reconstruction seems better. This is not really surprising as this coder uses 38 bands, while the transform coder uses 26 bands. The sharp attack problem of the subband coder can also be seen in the frequency spectrum. The transform coder shows the sharp attack properly but has an flat spectrum before the attack. The amplitude of this part is low, but the spectrum and size betray an artifact of one (half) large window with leaked white noise, due to the quantization error. It is not possible to estimate the impact of such artifacts without listening to the results. These plots help in explaining some of the expressions behind and design choices in different coders, but are not of much use without listening to the results. A presumption in these coders is that the spectrum does not have to be known precisely, but only in critical bands. The size of a certain band and the spread in frequency attached to this can not be evaluated with a plot. However the shaped noise idea can clearly be seen in both coding methods. The differences of the periodigram of a reconstruction and the original signal is relatively large, but the differences between the shaped noise reconstruction small.

8.2 Transform coding filterbank

In this section only the transform coder is considered. One of the features of this coder is the window switching scheme. As stated in the previous chapter various approaches were tried for matching the filterbank with the impulse response peaks. In figure 8.5 the fixed number approach is shown. It can be seen that the small (Ogg-)windows are mapped reasonably well on the peaks. The larger windows are properly placed between the peaks.

The original idea was that the amount of reflections in an impulse response is an increasing function in time and thus that a filterbank should follow this behavior. Such a filterbank starts with the smallest windows at $t = 0$ and than over time the windows grow slightly until the largest window is reached at the reverberant part of the impulse response. A plot of such a filterbank can be found in figure 8.6. This concept did not lead to satisfactory results, because the precise mapping of the peak and the window is very important: if a peak occurred just between two windows it was spread in the reconstruction. An example peak is marked with a green arrow in figure 8.6.

The other approach mentioned in section 7.2 was to switch to a smaller window after a rising in the signal defined by equation 7.1. This method also has the advantage of properly mapping the small windows on the peaks. In practice, only after tweaking the



Original Impulse Response

Plot of the filterbank

Figure 8.5 *Match of the short windows with peaks in the impulse response*

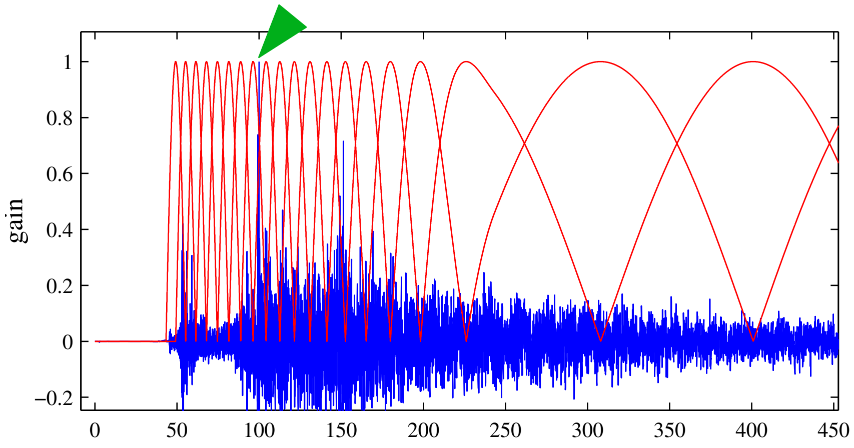


Figure 8.6 *Filterbank with gradually longer windows (red) and impulse response(blue)*

necessary parameters (given in equation 7.2) for a certain impulse response, the results were good. However if another impulse response was used (with more reverb, only discrete peaks etc.) the parameters had to be adjusted again. A value for N (the size of a block over which the rise is calculated) and r (the threshold value for the rise factor) of 32 resp. 0.2 only worked properly for some impulse responses. Other input signals could need a N of 8 and r of 0.5 for proper mapping of the filterbank with the peaks.

The disadvantage of the method, measuring a rise in signal, was that no general initial parameter set could be given. However the other method, which uses a fixed number of peaks, showed also not to be robust. The latter is the recommended option. The amount of small windows needed for a proper reconstruction can only be determined by more listening tests. The author of this thesis assumes that proper boundaries of this parameter are 8 and 16 reflections and therefore tried this parameters in the listening test.

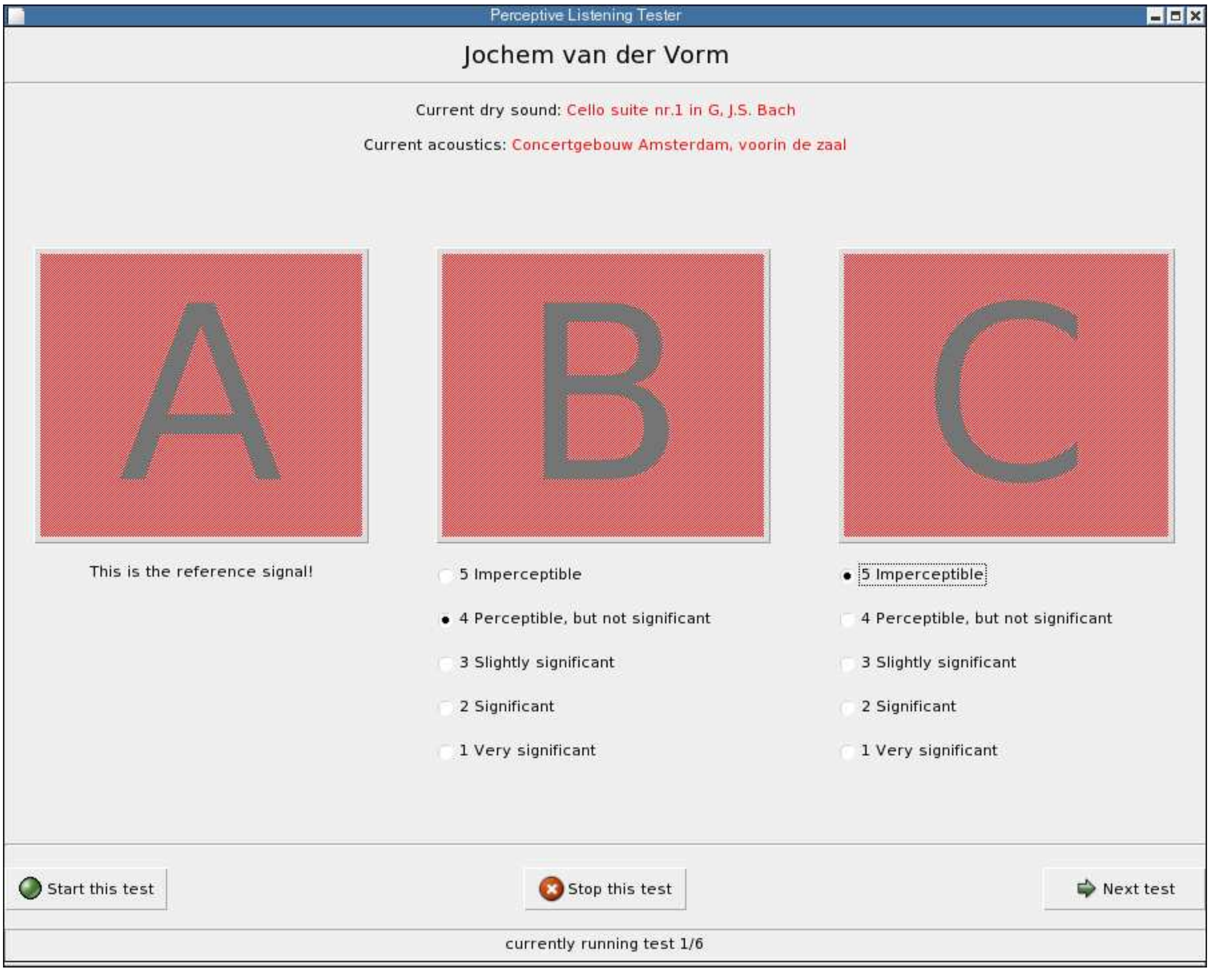
8.3 Listening test

8.3.1 Test method

During the analysis of the impulse response reconstruction it is necessary to listen to the signal to evaluate the correctness of the compression. However listening to such a signal is very subjective and as stated in chapter 4 not all human ears are similar, so you might be optimizing the result for your own hearing system. Therefore conducting a listening test for multiple subject is done to gain more insight in the quality of the coder.

A choice one can make is to let the subject listen to the impulse responses directly. This is rather difficult and it requires a lot of training to interpret the results correctly and to hear the subtle differences. For this reason it was decided to let the listeners evaluate the convolved impulse response. The context in which this thesis is done is the application of a Wave Field Synthesis system, therefore the listening tests were conducted in such an environment.

The recommendation ITU-R BS.1116-1, ”Methods for the Subjective Assessment of Small Impairments in Audio Systems Including Multichannel Sound Systems” [6] describes a way to measure the impairments in multi-channel audio systems. This recommendation is used to evaluate the quality of multi-channel audio coders, such as the MPEG-2 AAC. This is different from testing convolved coded impulse responses; then it is not



Jochem van der Vorm

Current dry sound: Cello suite nr.1 in G, J.S. Bach

Current acoustics: Concertgebouw Amsterdam, voorin de zaal

This is the reference signal!

- ○ 5 Imperceptible
- ● 4 Perceptible, but not significant
- ○ 3 Slightly significant
- ○ 2 Significant
- ○ 1 Very significant

- ● 5 Imperceptible
- ○ 4 Perceptible, but not significant
- ○ 3 Slightly significant
- ○ 2 Significant
- ○ 1 Very significant

currently running test 1/6

Figure 8.7 *Screen shot of the program written for the listening tests*

tried to evaluate an audio signal, but an acoustic environment. The listening tests done to investigate the quality of the transform coder of this thesis are therefore inspired by and not similar to tests accomplished by this paper.

To conduct subjective assessments in the case of small impairments is rather difficult. The ITU-R standard proposes a double-blind triple-stimulus with hidden reference. Such as test uses three stimuli which can be switched at any time by the user (A, B and C). The first stimulus (A) contains the reference signal. In our case this is a dry input signal convolved with the original impulse response. B or C is the hidden reference and thus equal to A. The other signal (C if B is the hidden reference and vice versa) is the

Impairment	Grade
Imperceptible	5
Perceptible, but not significant	4
Slightly significant	3
Significant	2
Very significant	1

Table 8.2 *Grades and descriptions*

same dry signal, but then convolved with the **reconstructed** impulse response.

Since long- and medium term aural memory are unreliable, the test procedure should rely exclusively on short-term memory. This is best done if a near instantaneous switching method is used in conjunction with the triple stimulus system described above. The expert listeners will be able to point out the hidden reference, while the listeners who have no clue, can be statistically left out of the final result.

Listeners give a grade for the difference between the supposed hidden reference and the supposed reconstruction as given in table 8.3.1. This table differs form the ITU-R recommendation, we used 'Significant' instead of 'Annoying'. The recommendation is written for audio coders where artifacts are easier qualified as annoying. Evaluating an acoustic environment with WFS system is more complicated as most listeners have no experience with the possible artifacts and therefore there is no standard for 'Annoying'.

Although it is better to let the listener switch between the stimuli with none visible means, so being able to move around, in this test a computer program was used for the listener to switch between the different stimuli and to grade them directly. A screenshot of the used program can be found in figure 8.7.

8.3.2 Evaluating environment

To optimize the transform coder a lot of combinations of the free parameters have to be evaluated. The result dependents on the dry input signal used. A proper listening test however should not last for more than 20-30 minutes and experience suggests that 10 to 15 trials per session should be scheduled [6]. For this reason only two versions of the parameters are used and only the transform coder is put to the test. The values of the stimuli can be found in table 8.3.2 and are somewhat arbitrarily chosen.

Session	Reflections	Small h(n)	Large h(n)	Environment	Dry signal
1	8	64	4096	Much reverb	Cello
2	8	64	4096	Much reverb	Drums
3	8	64	4096	Much reverb	Speech
4	16	128	2048	Much reverb	Cello
5	16	128	2048	Much reverb	Drums
6	16	128	2048	Much reverb	Speech
7	16	128	2048	Less reverb	Cello
8	16	128	2048	Less reverb	Drums
9	16	128	2048	Less reverb	Speech

Table 8.3 *Description of the sessions in the listening test*

Three different dry input signals were used, which tests various possible problems in the transform coder. From audio coding listening test it is known that a castanet sample can point out pre-echo problems, a harpsichord sample shows artifacts in the high frequency range and speech can give ringing artifacts [1].

The source signals in the WFS setup must be dry and therefore the choice was limited. The cello is recorded especially for this purpose in the anechoic chamber, the speech and drum samples come from a lexicon CD. The cello is used as a music signal, which could be found in the ’real world’, as could be heard in the ’Concertgebouw’ and thus fits with the acoustic environment used. The drum sample can point out problems with sharp attacks and thus with the early reflections: the perception of the convolved speech sample depends strongly on the accuracy in spectrum reconstruction. The ’less reverb’ and ’much reverb’ environments in the table are shorthand notations for the impulse responses from the ’Concertgebouw’ measurements near the source (’less reverb’) and far from the source (’much reverb’). In figures 8.8 and 8.9 an original and a reconstructed impulse response near the source (the ’less reverb’-variant) are plotted.

8.3.3 Listening results

The listening test was done by 21 listeners (one listener did the test twice). One listener made a lot of mistakes in filling in the answers and therefore his results were left out. The listener taking the test twice reached almost similar results, so a total of 20 results

---

[1] Actually speech is usually encoded with a special speech encoder, better suited for this purpose than an audio encoder, specialized in music.

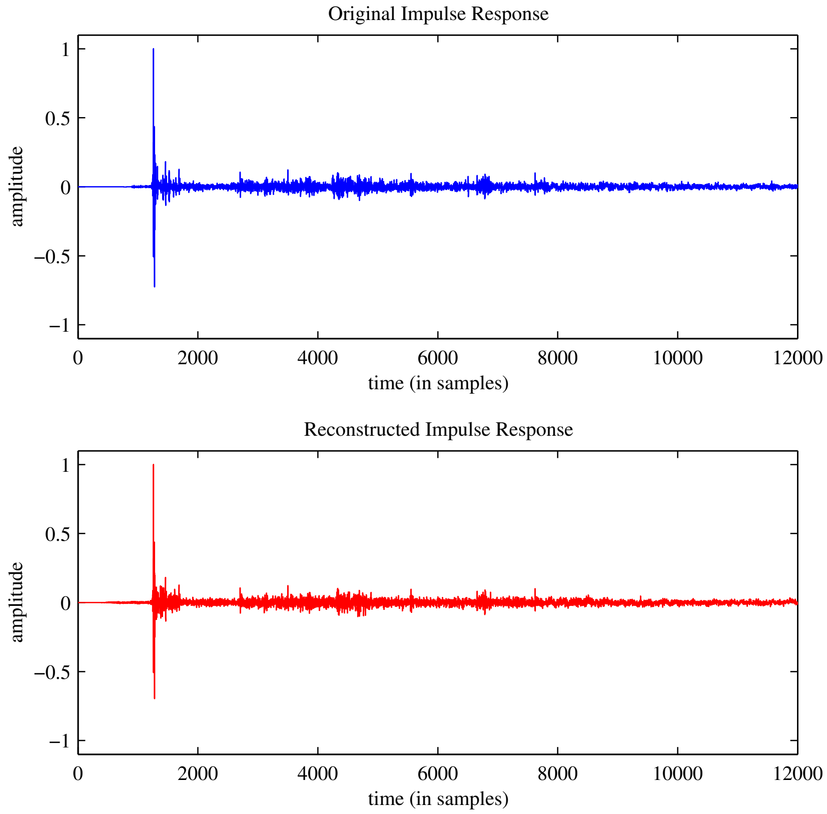


Figure 8.8 *Time domain representation of the original and reconstructed impulse response using the transform coder*

are used here.

In table 8.3.1 the grading of the test is given. In this section the difference grade is used. The difference grade is defined by the grade the listener gave for the reconstruction minus the grade awarded to the hidden reference. So, a positive difference grade means the listener choose the wrong answer (for example he gave the hidden reference 4 points and the reconstruction 5 points on the 8.3.1 scale). Logically, the lower the difference grade, the larger the impairment was, if the impairment is defined as the difference between the hidden reference and the reconstruction.

According to the ITU-R [6] recommendation the expertise of the listeners can be tested with a t-test in which an individual result is tested against the distribution of difference grades given by all users. All listeners were able to detect a substantial amount of im-

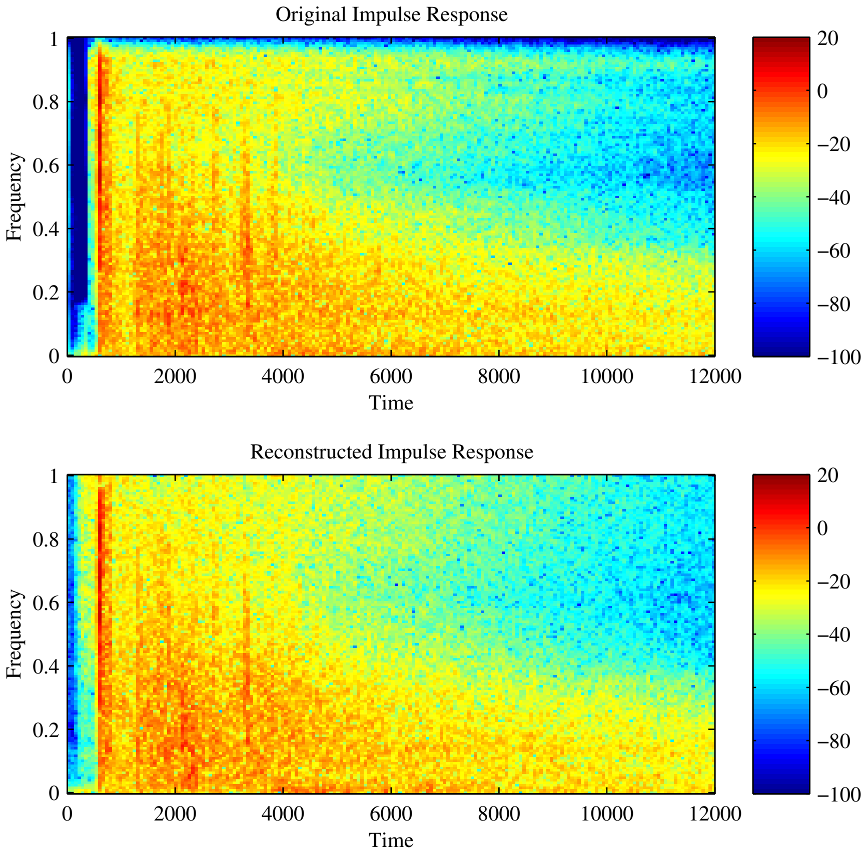


Figure 8.9 *Spectrum of the original and reconstructed impulse response using the transform coder*

pairments, therefore choosing the sessions over which the expertise test is carried out is difficult (because of the low number of candidates for such a test). Therefore the statistical pitfall of shaping the data to the experimenter's preconception is avoided here by using the data of all listeners. Downside of this approach is that it can not be stated explicit that an impairment close to zero difference grade equals a perfect reconstruction, because the best expert listeners could perhaps point out the hidden reference correctly, but their answers are obfuscated by the other results.

The raw data of the perceptual listening test is given in table D.1. The mean difference grades are plotted in figure 8.10, together with the 95%-confidence intervals found in table D.2. To test whether the listener could distinct the hidden reference from the impaired signal, an analysis of variance (ANOVA) test is done. The results can be found in table D.3. If a margin of 5% is used, in session 2, 3, 8 and 9 the hidden reference and impaired signal are distinguished successfully.

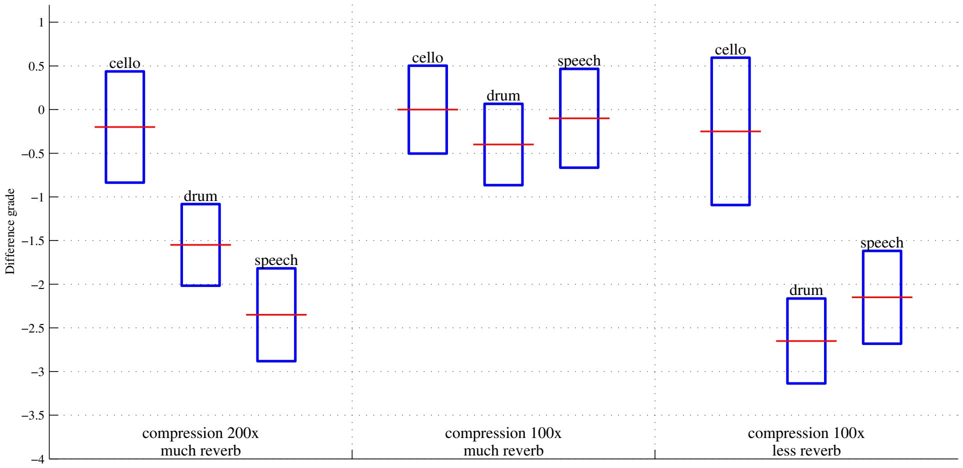


Figure 8.10 *Mean and confidence intervals of the grades, as outcome for the listening test*

The results provide some insight in the strong and weak points of the transform coding engine. The compression of the acoustic environment for playback of the recording of a cello such as in session 1, 4 and 7, is under all tested parameters indistinguishable from the original [2].

In the sessions with a dry drum or speech signal the difference between original and reconstructed acoustic environment could be detected in the highest compression setting (session 2 and 3). By just listening at the impulse response it was already suspected that the chosen compression parameters led to artifacts. These impulse responses were added to verify that using a WFS-system for the listening test is a sensible perceptual test. Since the only difference in sessions 1,2,3 and sessions 4,5,6 was the difference between carefully and recklessly chosen parameters this seems to be ascertained.

The last three sessions (7, 8, 9) were done with the same parameters as the middle three, but in session 8 and 9 the original and reconstructed acoustic environment were detected by the listeners. Apparently our transform coder, as provided until now, is better in compressing impulse responses with much reverb or it is easier to hear artifacts in impulse responses with more direct sound. The last two ways of looking at the result are comparable, both lead to the conclusion that the compression algorithm has to be improved for such situations.

---

[2] Note that the expert listeners may be able to detect small differences, as explained elsewhere in this section.

Chapter 9

Conclusion and Discussion

9.1 Conclusion

In this thesis a multidisciplinary spectrum of topics is covered, which include Wave Field Synthesis, impulse responses and acoustic environments, audio coders, the human hearing system, tools and algorithms for audio analysis and manipulation and perceptual listening tests. All these topics come together in the design of a transform coder for acoustic environments. The final design for this coder is able to compress such environments with a ratio of almost 150 for 44KHz, 16 bit input, which is in line with the original goal of this research.

The modulated lapped transform is a proper transform for an impulse response coder and makes it possible to exploit the perceptual properties of the human ear for data reduction. The basics of audio coders, such as the AAC algorithm can be utilized in an impulse encoder. A major difference lies in the switching of windows. The switching algorithm employed by the MPEG 2 Audio coder is not exact enough for use in an impulse response coder. Between the various switching models tested, the direct overlapping of short windows with the highest peaks in the original signal gave the best results.

Keeping one output coefficient per critical band in a large block can lead to a reasonable reconstruction of the frequency components. For the shorter blocks, laying together with the peaks in the impulse response, it is advantageous to keep all spectral components to maintain the peaks exact.

When coding an impulse response with much reverb and carefully selected parameters, the result was indistinguishable from the original impulse response when applied to a WFS system. But when coding an acoustic environment where the direct sound and early reflections dominate above the reverb, the chosen parameters or model were not good enough: the original and the reconstructed acoustic environment were still distinguishable by expert listeners.

9.2 Suggestions for future research

While the proposed transform coder reached good results under some, but not all circumstances, optimizing the encoder is recommend in the following way:

- Further research on the number of early reflections encoded separately, the size of the small and long windows to deploy. This can best be done by setting up a large scale listening test.
- Develop a better algorithm for reconstruction of the spectrum. Currently this is done by linear interpolation, but it was found [8] that the quality of speech encoders improved significantly using ARMA models for reconstruction of the spectrum.
- Use of codebooks in combination with vector quantization [32] to reach more compression. This can be used as an extension on the coding principles discussed in this thesis and will provide lossless compression based on the way the data is stored, not on psycho-acoustic principles.

Apart from enhancing the current coder, it may be worth the effort to investigate compression of impulse responses using the MPEG 4-CELP encoder. This parametric speech encoder uses very different algorithms than the current coder. It is very specialized in encoding speech, but perhaps some minor enhancements will make it employable for impulse responses.

A total different route one can take is to employ the principle of temporal and spectral masking and the use of bands to come up with a faster convolution engine than the ones currently in use. This is rather difficult, but the reward is high, convolving signals is a real bottleneck on a WFS system, but also on almost all digital recording systems. Furthermore a number of patents in this area make it impossible to use the most advanced algorithms freely. Psycho-acoustic principles are used to compress impulse responses to save disk space-bandwidth, but in a convolver these principles should be used to reduce the number of multiplications.

Appendix A

Critical Band filterbank

Band no.	Center freq. (Hz)	bandwidth (Hz)
1	50	-100
2	150	100-200
3	250	200-300
4	350	300-400
5	450	400-510
6	570	510-630
7	700	630-770
8	840	770-920
9	1000	920-1080
10	1175	1080-1270
11	1370	1270-1480
12	1600	1480-1720
13	1850	1720-2000
14	2150	2000-2320
15	2500	2320-2700
16	2900	2700-3150
17	3400	3150-3700
18	4000	3700-4400
19	4800	4400-5300
20	5800	5300-6400
21	7000	6400-7700
22	8500	7700-9950
23	10.500	9950-12.000
24	13.500	12.000-15.500
25	19.500	15.500-

Appendix B

Relation between the DFT & MDCT

The MDCT of a block of input signal $x(n)$ is defined as [10]

$$C(k) = \sqrt{\frac{2}{M}} \sum_{i=0}^{N-1} x(n)h(n)\cos(\frac{\pi}{M}(N+n_0)(k+0.5)) \tag{B.1}$$

where $h(n)$ is the window function, N is the length of the input block, $M = \frac{N}{2}$ is the number of transform coëfficients in each block and n_0 is a constant equal to $(M+1)/2$. Write the above formula as

$$C(k) = \sqrt{\frac{2}{M}} \sum_{i=0}^{N-1} R\left[x(n)h(n)e^{(-j\pi(n+n_0)(k+0.5))/M}\right] = \sqrt{\frac{2}{M}} F\left[e^{j\phi(k)}F(s(n))\right] \tag{B.2}$$

where R denotes the real part and F the Fourier transform.

$$\phi(k) = \frac{-\pi(N+2)(k+0.5)}{2N} \tag{B.3}$$

$$s(n) = e^{\frac{-j\pi n}{N}} x(n)h(n) \tag{B.4}$$

And finally:

$$C(k) = \sqrt{\frac{2}{M}}\, |S(k)|\, \cos[\frac{2\pi n_0(k+0.5)}{N} - \angle S(k)] \tag{B.5}$$

Appendix C

Window switch kernel

```
function filterbank = construct_filterbank(peak_pos,n_small,n_large);

% -------------------------------------------------------------------
% Author            : Jochem van der Vorm
% Last change       : Tue Apr  8 22:10:52 CEST 2003
% -------------------------------------------------------------------

% Change peak locations in window sizes
win_sizes = diff([0 sort(peak_pos)]);

% We want the switch windows according to the thesis
new_win = [];
for i=1:length(win_sizes)
    f = win_sizes(i)-n_small/2;
    n_wins = ceil(f/(n_large/2));
    f = ceil(f/n_wins);
    f = 2*round(f/2);
    new_win = [new_win; ones(n_wins,1)*f*2];
    if i < length(win_sizes)
        win_sizes(i+1) = win_sizes(i+1) + win_sizes(i) - ...
        f * n_wins - n_small/2;
    end
end

% Begin and end with an extra window to ensure proper begin and end
new_win = [128; new_win; 128];

% Now we can split the signal in filterbank
```

```
filterbank{1} = sinwin(n_small);
frame_nr = 2;
for i=2:length(new_win)-1
    % Clear variables
    window = [];
    add_small_window = 0;

    % We use this in the constuction
    tmp_large = sinwin(new_win(i));
    tmp_small = sinwin(n_small);

    % First half of the window
    if new_win(i)~=new_win(i-1)
        % first half is start part
        switch_t = (new_win(i) - n_small)*0.25;
        window = [zeros(switch_t,1); window];
        window = [window; tmp_small(1:end*.5)];
        window = [window; ones(switch_t,1)];
    else
        % first half is normal
        window = tmp_large(1:end*.5);
    end

    % Second half of the window
    if new_win(i)~=new_win(i+1)
        % second half is stop part
        switch_t = (new_win(i) - n_small)*0.25;
        window = [window; ones(switch_t,1)];
        window = [window; tmp_small(end*.5+1:end)];
        window = [window; zeros(switch_t,1)];
        add_small_window = 1;
    else
        % Second half is normal
        window = [window; tmp_large((end*.5+1):end)];
    end

    % Now we apply the constructed window
    filterbank{frame_nr} = window;
    frame_nr = frame_nr + 1;

    % Plus a n_small window on a transition
    if add_small_window==1
        filterbank{frame_nr} = tmp_small;
        frame_nr = frame_nr + 1;
    end
end
```

Appendix D

Results of the perceptual tests

S 1	**S 2**	**S 3**	**S 4**	**S 5**	**S 6**	**S 7**	**S 8**	**S 9**	**Mean**	**Variance**
1	0	-1	-1	-1	0	-2	-2	-2	-0.89	1.05
-1	-2	-2	0	1	-1	-2	-3	-1	-1.22	1.20
-1	-2	-1	1	1	2	0	-1	-3	-0.44	1.59
-1	-3	-3	0	-1	0	-3	-3	-3	-1.89	1.36
1	-1	-1	-1	-1	-1	1	-3	-1	-0.78	1.20
-1	-2	-3	1	-1	-1	-3	-3	-3	-1.78	1.39
1	-2	-2	-1	0	-1	-1	-2	1	-0.78	1.20
-1	-2	-1	-1	0	1	2	-3	-2	-0.78	1.56
2	-1	-2	1	1	1	-2	-2	-3	-0.56	1.81
-2	-1	-3	1	-1	-1	1	-3	-3	-1.33	1.58
1	-1	-3	2	-1	2	2	-3	-2	-0.33	2.12
0	-1	-1	0	0	0	1	-1	-2	-0.44	0.88
1	-1	-1	-1	1	-1	1	-1	-1	-0.33	1.00
-3	-2	-4	-1	1	2	1	-4	-4	-1.56	2.40
2	-2	-4	1	-1	-1	-1	-3	-3	-1.33	1.94
1	-3	-2	-1	-1	-1	1	-4	-1	-1.22	1.64
-1	-2	-4	1	-2	1	-2	-4	-2	-1.67	1.80
-1	1	-2	1	-1	-1	3	-1	-2	-0.33	1.66
-1	-1	-3	-2	-2	0	-2	-4	-3	-2.00	1.22
-1	-3	-4	0	0	-2	0	-3	-3	-1.78	1.56
-0.2	-1.55	-2.35	0	-0.4	-0.1	-0.25	-2.65	-2.15	-1.07	1.09

Table D.1 *Difference grades for all subjects for all sessions. Columns S 1-9 represent the nine listening sessions. The rows represent the various subjects. The last row gives the means of the columns.*

	Mean	From	To
S 1	-0.20	-0.837	0.437
S 2	-1.55	-2.017	-1.083
S 3	-2.35	-2.882	-1.818
S 4	0	-0.504	0.504
S 5	-0.4	-0.866	0.066
S 6	-0.1	-0.666	0.466
S 7	-0.25	-1.094	0.594
S 8	-2.65	-3.137	-2.163
S 9	-2.15	-2.682	-1.618

Table D.2 *95% Confidence interval for the listening test. Rows S 1-9 represents the nine listening sessions. The 'From' and 'To' columns depict the border values for the confidence interval.*

	SS	MS	F	P-value
S 1	0.40	0.40	0.73	0.40
S 2	24.03	24.03	57.97	0.00
S 3	55.22	55.22	82.14	0.00
S 4	0.00	0.00	0.00	1.00
S 5	1.60	1.60	4.22	0.05
S 6	0.10	0.10	0.21	0.65
S 7	0.63	0.63	0.62	0.44
S 8	70.23	70.23	129.86	0.00
S 9	46.23	46.23	87.17	0.00

Table D.3 *Anova data for the listening test. Rows S 1-9 represents the nine listening sessions. Further the results 'Between the groups' are displayed, first the Residue Sum* SS*, then the Mean Squares* MS*, the test ratio* F *and the probability* P *for* F*.*

Bibliography

[1] Sonke, J.J, *Variable Acoustics by Wave field synthesis*, Amsterdam, The Netherlands, 2000.

[2] Oppenheim, et al *Signals & Systems*, Prentice Hall, 1997.

[3] Bosi, M. et al, *ISO/IEC Mpeg-2 Advanced Audio Coding*, Journal of the Audio Engineering Society, No 10, Oct 1997, pp 789-813.

[4] Painter, T. & Spanias, A, *A Review of Algorithms for Perceptual Coding of Digital Signals*, IEEE, 1994.

[5] Geiger, R., *INTMDCT - A Link Between Perceptual And Lossless Audio Coding*, IEEE, 2002.

[6] ITU-R BS.1116-1, *Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems*, 1997.

[7] Hulsebos, E. and De Vries, D., *Parametrization And Reproduction Of A Concert Hall Acoustics Measured With A Circular Array Microphone Array*, AES Convention Paper (112th), 2002.

[8] Malvar, H., *Signal Processing with Lapped Transforms*, Artech House, 1992.

[9] Glasgal, R., *Ambiophonics*, 2nd edition, 1998.

[10] Najafzadeh-Azghandi, H., *Improving Perceptual Coding of Narrowband Audio Signals at Low Rates*, IEEE Conf. Acoustics, pp.913-916, 1999.

[11] Atmadja, S., *Implementation of a Source Tracking system using Cross Correlation of speech signals*, TU Delft, 2002.

[12] Vörlander, M., *Computer models in room acoustics - state of the art*, Journal Building Acoustics, no.4, p. 229-245, 1997.

[13] Kuttruff, H. *Room acoustics,* Applied Science Publishers Ltd, London, 1st ed, 1973.

[14] Ferreirra, A.J. de S., *Spectral Coding and Post-processing of High Quality audio,* Dissertation, University of Porto, 1998.

[15] Plain, S.E.M., *Bit Rate Scalability in Audio Coding,* McGill University, Montreal, Canada, 2000.

[16] Sporer, *The use of multirate filterbanks for coding of high quality digital audio,* EUSIPCO, Amsterdam, june 1992.

[17] Berkhout, A.J., *Applied Seismic Wave Theory,* Elsevier, Amsterdam-Oxford-New York-Tokyo, 1987.

[18] Wang, Y. et al, *Some Peculiar Properties of the MDCT,* Proceedings of ICSP, 2000.

[19] Press, W.H. & others, *Numerical recipes in C: The art of scientific computing,* Cambridge University Press, 1989.

[20] Berkhout, A.J., D. de Vries and P.Vogel, *Acoustic Control by Wave Field Synthesis,* J. Acoust. Soc. Am. 93(5), May 1993.

[21] Boone, M.M., *Acoustic Rendering with Wave Field Synthesis,* position paper Acoustic Campfire, Snowbird, Utah, May 26-29 2001.

[22] Omoto, A. et al, *Similarity Evaluation of Room Acoustic Impulse Responses: Visual and Auditory Impressions,* AES, Vol. 50, No.6, 2002.

[23] Torger, A. & Farina, A., *Real-time partitioned convolution for ambiophonics surround sound,* IEEE workshop on applications of signal processing 2001.

[24] Kernighan, B & Ritchie, D.M., *The C programming language, 2nd edition,* Prentice-Hall Inc. 1988.

[25] Väljamaë, A., *A feasibility study regarding implementation of holographic audio rendering techniques over broadcast networks,* Chalmers. Götenborg Sweden, 2003.

[26] Blackman, R.B. and J.W. Tukey, *The Measurement of Power Spectra,* New York, 1958.

[27] Painter, T and Spanias, A, *Perceptual coding of digital audio,* Proc. IEEE vol. 88, no. 4, 2000.

[28] Vaidyanathan, P.P., *Multirate digital filters, filterbanks, polyphase networks and application: a tutorial,* Proc IEEE vol.78, 1990.

[29] N.S. Jaynt, *Digital Coding of Waveforms*, Prentice hall, Englewood Cliffs, 1994.

[30] Kunz, D. and Aach, T., *Lapped directional transform, a new transform for spectral image analysis*, Aachen, 1997.

[31] Zelinkski, R. and Noll, P., *Adaptive Transform Coding of Speech signals*, IEEE ASSP vol. 25, 1977.

[32] J. Moffitt, *Ogg vorbis documentation.* http://www.xiph.org/ogg/vorbis/docs.html, 2002.

Index

AAC, 17
absolute threshold of hearing, 25
AC-3, 24
acoustic environment, 19, 20
ambiophonics, 13
ambisonics, 13
analysis of variance, 62
ANOVA, 62
anvil, 25
auralisation, 20
auricle, 24
autocorrelation filter, 32

Bark, 26
Bark scale, 51
bit-rate, 17
block edge effect, 36

Carrouso, 7
CELP, 23
cochlea, 25
compression ratio, 8
convolution theorem, 20
critical band, 26
critical sampling, 33

DCT, 36
deterministic, 20
DFT, 33
difference grade, 62
Dirac pulse, 19
direct sound, 20
Discrete Cosine Transform, 36
Discrete Fourier Transform, 33
double-blind triple-stimulus, 59

eardrum, 24
early reflections, 20
effective rectangular band, 26
ERB, 26

FIR, 19
Fourier Transform, 32
frequency domain, 24
Frequency scale, 51

Gaussian noise, 24
geometric mean, 28

half sine window, 41
half-sine, 50
hammer, 25
Hamming, 33
Hanning, 33
hidden reference, 59
hotelling transform, 35
Huygens, 14
hybrid coder, 23

IIR, 19
impairment, 62
impulse response, 19
incus, 25
inner ear, 24
ITU-R BS.1116.1, 59

JND, 27

Just Noticeable Difference, 27

Kaiser, 33
Karhunen-Loeve transform, 35
Kirchoff-Helmholtz, 14

Lagrange multiplier, 29
lapped transforms, 36
log variance rule, 28
lossless, 23
lossy, 23

malleus, 25
membrame, 24
middle ear, 24
modulated lapped transform, 40
monotony, 50
Moore, 26
MP3, 8
MPEG-4, 7
multirate, 41

NMR, 27

Ogg, 24
outer ear, 24
overlap-add, 21
overlap-save, 21

Parceval, 34
partitioned convolution, 20
perfect reconstruction, 38
periodigram, 32
pinna, 24
power spectrum, 32
pre-echo, 28
psychoacoustics, 23

QMF filters, 39
quantization, 28

Rayleigh integral, 15
real-time, 20
rectangular window, 33
residue spectrum, 28
reverberation, 19

seismic, 14
selectivity, 50
short-space Fourier Transform, 36
Simultaneous Masking, 27
SSFT, 36
stapes, 25
statistic, 20
stirrup, 25
sweet area, 16
synthesis bank, 38

time domain, 24
Time-Domain Aliasing Cancellation, 39
Toeplitz matrix, 35
transient, 28

variable bit-rate, 49

Wave Field Synthesis, 16
WFS, 16
window switching, 41