Rate-Distortion Optimal
Time-Frequency Decompositions
for MDCT-based Audio Coding

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 14 november 2006 om 12:30 uur
door Omar Aziz NIAMUT
elektrotechnisch ingenieur
geboren te Rotterdam.

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. R.L. Lagendijk

Toegevoegd promotor:
Dr.ir. R. Heusdens

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. R.L. Lagendijk,	Technische Universiteit Delft, promotor
Dr.ir. R. Heusdens,	Technische Universiteit Delft, toegevoegd promotor
Prof.dr.ir. A.W. Heemink,	Technische Universiteit Delft
Prof.dr.ir. A.-J. Van der Veen,	Technische Universiteit Delft
Prof. A. Ortega,	University of Southern California
Dr.-Ing. J. Herre,	Fraunhofer Institute for Integrated Circuits
Dr.ir. S. van de Par,	Philips Research Laboratories

B&W Bowers & Wilkins

The production of this thesis has been supported by B&W Group Nederland B.V.

Cover design by Eelco de Vaal
Printed by Gildeprint Drukkerijen, Enschede

ISBN-10: 90-9021159-4
ISBN-13: 978-90-9021159-6

Chapters 1,2,3 and 10:	Copyright © 2006 by O.A. Niamut
Chapters 4,5,6,7 and 9:	Copyright © 2003, © 2004, © 2005 and © 2006 by IEEE, respectively
Chapter 8:	Copyright © 2005 by AES

All rights reserved. No part of this thesis may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, any information storage or retrieval system, or otherwise, without written permission from the copyright owner.

Preface

The research presented in this thesis was conducted within the projects SiCAS and ARDOR.

SiCAS (*Sinusoidal Coding of Audio and Speech*) was an STW funded project that started in 1999 and ended in 2001. The objective was to develop a generic audio coding system that could compete with application-optimized systems and that could adapt to the input signal, user and other time-varying constraints. In this project the following parties participated: Philips Research Laboratories, the Royal Institute of Technology (KTH) in Stockholm and Delft University of Technology. SiCAS was supported by Philips Research Laboratories and the Technology Foundation STW, applied science division of NWO and the technology programme of the ministry of Economics Affairs.

ARDOR (*Adaptive Rate-Distortion Optimized sound codeR*) was an European Union funded project within the fifth framework that started in 2002 and ended in 2005. The objective was to meet the need for a universal codec as created by the emergence of time-varying heterogeneous networks and by the convergence of traditional consumer electronics with mobile communications. In this project the following parties participated: Philips Research Laboratories, France Telecom R&D, Aalborg University (CPK), University of Hannover (TNT), Royal Institute of Technology (KTH) and Delft University of Technology. ARDOR was supported by the E.U. grant no. IST-2001-34095.

Within the SiCAS and ARDOR project, Delft University of Technology investigated and developed rate-distortion optimal time-frequency decomposition algorithms for transform coding. The results of this work are presented in this thesis.

O.A. Niamut, Delft, October 2006.

Summary

Perceptual audio coding has emerged as the *de facto* solution to cope with efficient storage and transmission of digital audio. Standardized solutions are offered to consumers worldwide, that perform satisfactory if properly employed. However, the recent convergence of consumer electronics and mobile communication, and the emergence of ubiquitous heterogeneous network environments with time-varying bandwidth and delay constraints, put severe demands on the capabilities of the existing solutions and on the user that has to select from a broad range of solutions. This can easily lead to situations of application mismatch where an audio coding system is employed outside the intended application range. New schemes are required that can adapt to the conditions and constraints as imposed by the user and the network.

In this thesis we study several techniques and combinations thereof, that we consider as suitable candidates for incorporation into new audio coding schemes. Rather than to undertake the development of a complete audio coding scheme, we concentrate on the signal processing aspects and interaction of these techniques, instead. In the first part of the thesis, an overview is given of two techniques that can already be encountered in various digital signal coding schemes. These techniques serve as ingredients for the algorithms that are presented in the second part.

First, we look at operational rate-distortion (RD) optimization. With operational RD optimization, we seek to obtain the best achievable performance for coding an audio signal, given the choice of compression framework or coding environment. In this thesis we review the material on operational RD optimization, formulate the rate-constrained bit allocation problem and study solutions for this problem. Here, we are mostly interested in the interaction of such an RD optimization framework with the time-frequency decomposition of the signal. This leads to a study of best basis search algorithms and their combination with RD optimization.

In most audio coding schemes, the time-frequency decomposition is obtained using the modified discrete cosine transform, or MDCT. Thus, we investigate various properties of the MDCT, such as the conditions for perfect reconstruction, window design and fast algorithms. Moreover, we look at three distinct adaptive techniques that are available for the MDCT in order to obtain nonuniform time-frequency decompositions.

The main objective of the work presented in this thesis is to study the combination of an operational RD optimization framework with adaptive MDCT-based time-frequency decomposition techniques. In the second part, new algorithms and experimental results are presented for the three decomposition techniques, in the format of scientific papers.

We start with an investigation of adaptive frequency decomposition. Subband merging is employed to construct a nonuniform MDCT and dynamic programming is applied for fast best basis searching. We show that the proposed algorithm can lead to gains in SNR and subjective listening test. However, we observe that lossless coding of the side information associated to the obtained decompositions leads to a high side information rate and we conclude that this particular frequency domain approach does not provide a performance increase that can justify the increase in complexity.

Next, we continue with adaptive time segmentation, where dynamic programming is employed for best basis search and block switching for MDCT-based time segmentation. Three variations of the basic algorithm are constructed that cover a large range of complexity trade-offs. The effects of varying window overlap are thoroughly studied and we show that an optimal solution can be obtained in polynomial time. Furthermore, we directly compare a new audio coding system that incorporates our time segmentation algorithm with MPEG-4 standardized coding systems and obtain equally good or better listening test results for a large range of bit rates. A low-complexity variant of this audio coding scheme shows a negligible performance loss.

We then turn back to frequency decomposition and study temporal noise shaping, which employs linear prediction in the frequency domain. We combine temporal noise shaping with RD optimization to control the order of the prediction filter and the selection of quantizers. This leads to an efficient algorithm that outperforms an existing method to control temporal noise shaping. Although the algorithm interferes with the initial purpose of temporal noise shaping, the performance gain in terms of rate-distortion behavior is significant.

List of Papers

The following papers and patents have been published and filed, respectively, by the author of this thesis during his Ph.D. studies:

[1] O.A. Niamut and R. Heusdens, “Subband Merging in Cosine-Modulated Filter Banks”, in *IEEE Signal Processing Letters*, vol. 10, no. 4, pages 111–114, April 2003.

[2] O.A. Niamut and R. Heusdens, “Flexible Frequency Decompositions for Cosine Modulated Filter Banks”, in *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing (ICASSP’03)*, pages 449–452, April 2003.

[3] O.A. Niamut and R. Heusdens. Audio Coding With Non-Uniform Filter Bank. 2004. European patent application, EP1421579. Koninklijke Philips Electronics N.V.

[4] O.A. Niamut and R. Heusdens, “Rate-Distortion Optimal Audio Coding with Signal Adaptive MDCT”, in *Proceedings of the IEEE Fourth Benelux Signal Processing Symposium (SPS’04)*, pages 79–82, April 2004.

[5] O.A. Niamut and R. Heusdens, “RD Optimal Time Segmentations for the Time-Varying MDCT”, in *Proceedings of the 12th European Signal Processing Conference (Eusipco’04)*, pages 1649–1652, September 2004.

[6] O.A. Niamut and R. Heusdens, “Time Segmentation Algorithms for the Time-Varying MDCT”, in *Proceedings of the IEEE First Benelux/DSP Valley Signal Processing Symposium (SPS-DARTS’05)*, pages 101–104, April 2005.

[7] O.A. Niamut and R. Heusdens, “Optimal Time Segmentation For Overlap-Add Systems With Variable Amount Of Window Overlap”, in *IEEE Signal Processing Letters*, vol. 12, no. 10, pages 665–668, October 2005.

[8] O.A. Niamut, R. Heusdens and H.J. Lincklaen Arriëns, “Upfront Time Segmentation Methods for Transform Coding of Audio”, in *Proceedings of the 119th AES Convention*, October 2005.

[9] R. Heusdens, J. Jensen, W.B. Kleijn, V. Kot, O.A. Niamut, S. van de Par, N.H. van Schijndel and R. Vafin, “Bit-Rate Scalable Intraframe Sinusoidal Audio

Coding Based on Rate-Distortion Optimization", in *Journal of the Audio Engineering Society*, vol. 54, no. 3, pages 167–188, March 2006.

[10] O.A. Niamut and R. Heusdens, "RD Optimal Temporal Noise Shaping For Transform Audio Coding", in *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing (ICASSP'06)*, vol. V, pages 189–192, May 2006.

[11] J. Østergaard, O.A. Niamut, J. Jensen, R. Heusdens, "Perceptual Audio Coding using N-Channel Lattice Vector Quantization", in *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing (ICASSP'06)*, vol. V, pages 197–200, May 2006.

Table of Contents

Preface i

Summary iii

List of papers v

1 Introduction 1
- 1.1 Problem statement . . . 3
- 1.2 Scope . . . 5
- 1.3 Organization . . . 8
- Bibliography . . . 11

Part I - Background

2 Operational RD optimization and best basis search 19
- 2.1 Introduction . . . 19
- 2.2 Operational RD optimization . . . 22
  - 2.2.1 problem formulation . . . 22
  - 2.2.2 solution to the rate-constrained problem . . . 24
- 2.3 Best basis search methods . . . 28
  - 2.3.1 linear expansions and best bases . . . 28
  - 2.3.2 time-frequency tilings . . . 29
  - 2.3.3 time segmentation and frequency decomposition . . . 31
  - 2.3.4 binary tree best basis search . . . 33
  - 2.3.5 dynamic programming best basis search . . . 35
  - 2.3.6 best bases and operational RD optimization . . . 36
- 2.4 Conclusion . . . 38
- Bibliography . . . 39

3 The MDCT **45**
3.1 Introduction 45
3.2 Properties of the MDCT 46
3.2.1 lapped orthogonal transforms 46
3.2.2 TDAC and perfect reconstruction 49
3.2.3 window design 51
3.2.4 fast MDCT implementation 54
3.2.5 MDCT examples 55
3.2.6 related transforms 56
3.3 Adaptive time-frequency decomposition 58
3.3.1 window and block switching 59
3.3.2 temporal noise shaping 62
3.3.3 subband merging 66
3.4 Conclusion 69
Bibliography 70

Part II - Papers

4 Flexible frequency decompositions for cosine-modulated filter banks **79**
4.1 Introduction 80
4.2 Previous work 80
4.3 Flexible frequency decomposition 82
4.3.1 reduction of algorithmic complexity 85
4.3.2 coding of side information 85
4.4 Experimental results 86
4.5 Concluding remarks 87
References 88

5 Rate-distortion optimal audio coding with signal adaptive MDCT **91**
5.1 Introduction 92
5.2 The MDCT 93
5.3 Subband merging 94
5.4 Flexible frequency decomposition 95
5.5 Experimental results 98
5.6 Concluding remarks 99
5.7 Acknowledgement 99
References 100

6 Time segmentation algorithms for the time-varying MDCT 101
6.1 Introduction 102
6.2 The time-varying MDCT 102
6.3 Time segmentation algorithms 103
6.3.1 single tree time segmentation 105
6.3.2 flexible time segmentation 106
6.3.3 cost functions 107
6.4 Experimental results 108
References 111

7 Optimal time segmentation for OLA systems with variable overlap 113
7.1 Introduction 114
7.2 Problem statement 115
7.2.1 existing approach 115
7.2.2 suboptimality of the existing approach 116
7.3 Flexible time segmentation for a varying window overlap 117
7.3.1 derivation of the proposed algorithm 117
7.3.2 complexity analysis 119
7.4 Experimental results and discussion 120
References 122

8 Upfront time segmentation methods for transform coding of audio 123
8.1 Introduction 124
8.2 RD optimization 126
8.2.1 dynamic programming best basis search 128
8.2.2 suboptimality of the existing approach 130
8.3 The transform coder framework 131
8.3.1 experimental results 132
8.4 Upfront time segmentation 133
8.4.1 cost measures for upfront segmentation 134
8.4.2 computational complexity 135
8.4.3 experimental results 136
8.5 Conclusions 139
8.6 Acknowledgments 140
References 140

9 RD optimal temporal noise shaping 143
9.1 Introduction 144
9.2 Temporal noise shaping 144

9.3 RD optimal TNS . . . 146
9.4 Experimental results . . . 148
9.4.1 results for single fragment . . . 149
9.4.2 results for multiple fragments . . . 149
References . . . 153

10 Results and Recommendations 155
10.1 Summary of results . . . 155
10.2 Recommendations . . . 156
Bibliography . . . 158

A Perfect reconstruction of the MDCT 159

B Frequency domain linear prediction 163
B.1 The Hilbert transformer . . . 163
B.2 Relation Between Squared Hilbert Envelope And Spectral Autocorrelation . . . 164
B.3 Frequency Domain Linear Prediction . . . 165
Bibliography . . . 166

C Subband merging 167

Samenvatting 173

Acknowledgements 175

Curriculum Vitae 177

List of Figures

1.1 The generic perceptual audio encoder block scheme. 2
1.2 The generic perceptual audio encoder block scheme extended with a rate-distortion control block. 5

2.1 Lagrange optimization and duality gap. 26
2.2 A time-frequency tile depicts the time-frequency localization of a particular basis function. 30
2.3 Examples of time-frequency tilings. 31
2.4 Examples of time segmentation and frequency decomposition. 32
2.5 The single tree decomposition algorithm employs tree pruning to eliminate suboptimal decompositions. 34
2.6 Initial decomposition of a signal into N segments of M samples. . . . 35
2.7 The flexible decomposition algorithm employs dynamic programming to build up the optimal decomposition. 36

3.1 Analysis mapping $\mathbf{T_a}$ and synthesis mapping $\mathbf{T_s}$. 46
3.2 Forward MDCT operation. 48
3.3 Inverse MDCT operation. 49
3.4 Impulse and magnitude responses of several MDCT windows. 52
3.5 Example of the overlap-add MDCT operation and the corresponding MDCT spectra. 56
3.6 Example of the introduced time aliasing. 57
3.7 Window switching operation. 60
3.8 Window switching leads to a nonuniform time segmentation. 61
3.9 Open-loop frequency domain linear prediction scheme. 63
3.10 The combination of a filterbank and frequency domain linear prediction leads to a trade-off between time localization and frequency bandwidth. 64

3.11 Temporal noise shaping leads to reduced pre-echo artifacts. 65
3.12 Subband merging trades frequency bandwidth for time resolution. . . 68
3.13 Equivalence relation between a channel with decimation by M and 2 channels with decimation by $2M$. 69
3.14 Increase of overall time-frequency uncertainty ν as a result of subband merging. 70

4.1 Time-frequency tilings as obtained by decomposition algorithms. . . . 81
4.2 A decomposition S_k is a collection of adjacent (nonuniform) subband channels. 82
4.3 Dynamic Programming is employed to search iteratively for the optimal decomposition. 83
4.4 Resolution switching for 2 filters with subband merging. 84
4.5 A comparison between fixed uniform and variable nonuniform decomposition. 86

5.1 Subband merging trades frequency bandwidth for time resolution . . . 95
5.2 A decomposition $S_n^{(M)}$ is a collection of adjacent (nonuniform) subband channels. 96
5.3 Example of an optimized nonuniform MDCT and its representation by a binary sequence. 97
5.4 Blockscheme of the proposed perceptual audio coding system. 98

6.1 Window-switching schemes using the various MDCT windows. . . . 103
6.2 Impulse and magnitude responses for the various MDCT windows. . . 104
6.3 An example where the window tails have to be adapted depending on the optimal choice for the length of the added segment. 105
6.4 The single tree time segmentation algorithm employs tree pruning to eliminate suboptimal segmentations. 106
6.5 The flexible time segmentation algorithm employs dynamic programming to construct the optimal segmentation. 107
6.6 Results for minimum entropy flexible time segmentation of a castanet signal. 109
6.7 SNR curves for the various time segmentation algorithms, for both fixed and variable window overlap. 109

7.1 Examples of time segmentation with fixed and variable overlap windows. 115
7.2 An example where the window tails have to be adapted to retain PR when switching to different segment lengths. 116

7.3 A schematic overview of the various overlap possibilities that are considered during optimization. 118
7.4 Comparison of the various time segmentation algorithms for the entropy cost measure. 120
7.5 Comparison of the various time segmentation algorithms for the rate-distortion cost measure. 121

8.1 The MPEG-2 AAC block switching process. 125
8.2 The flexible time segmentation algorithm employs dynamic programming to construct the optimal time segmentation. 129
8.3 Symmetric MDCT windows with maximum overlap are employed during optimization. 130
8.4 Scheme of the proposed transform coder. 131
8.5 MUSHRA listening test results with the proposed transform coding system. 134
8.6 Lagrangian costs obtained for each time segmentation method, for 4 audio fragments. 136
8.7 Time segmentation of harpsichord fragment. 138
8.8 OAB score distributions for the individual fragments and for all fragments. 139

9.1 Block scheme of a standard TNS implementation based on prediction gain. 146
9.2 Block scheme of the proposed TNS implementation in an RD optimization framework. 147
9.3 Results for 3s of German male speech signal. 150
9.4 Rate vs. perceptual distortion plots for the set of four audio fragments. 151
9.5 MUSHRA test results for all versions. 151

List of Tables

4.1 A comparison between fixed uniform decomposition and variable non-uniform decomposition. . . . 87

5.1 Results of the listening test. . . . 99

8.1 The 8 test signals used in the MUSHRA test. . . . 133

Chapter 1

Introduction

With the advent of the Compact Disc [50], the benefits of digital audio became available for the mass consumer. Audio in a digital format provides various advantages over its analog counterpart, like the vinyl record [47] or magnetic tape [17], such as improved audio quality, noise and error robustness, reproducibility and suitability for post-processing operations. However, these advantages come at a price, since digitalization of audio signals inevitably incurres high data rates. While a CD can hold a fair amount of data, recently arisen transport channels such as the Internet and portable and fixed storage media strongly exposed the need for data reduction mechanisms. Digital audio coding or compression techniques have risen to meet this need for lower data rates. Indeed, digital coding of multimedia signals such as speech, music, images and video brought together the previously separate fields of information theory, digital signal processing and psychology into a mix that resulted in arguably the most successful practical applications known to these fields. Examples such as MP3, the Sony MiniDisc and Dolby Digital are known and used all over the world.

When it comes to coding digital signals in general, Shannon's seminal work on source coding and rate-distortion (RD) theory [48] sets the initial framework for further study. The theorems in his work provide us with general bounds on the minimum bit rate with which to represent a signal at a certain fidelity. A perfect representation of a signal at a lower rate can be achieved by lossless or entropy coding. Lossless coding applied to audio signals in isolation does not deliver the compression rates that are necessary for improved storage and transport, but it is often employed as part of an audio coding framework. Of much more interest is the case of lossy coding, where the reproduction fidelity or coding distortion can be traded for bit rate. Hence, we say that an audio coding scheme tries to represent an audio signal at minimum or no distortion, given a certain amount of bits.

While Shannon's theorems hold for coding of digital signals in general, audio coding constitutes a rather special case. In order to achieve perceptual transparency or minimum perceptual distortion, the incurred distortions should be compensated for, or measured according to the human auditory system. This is where the psy-

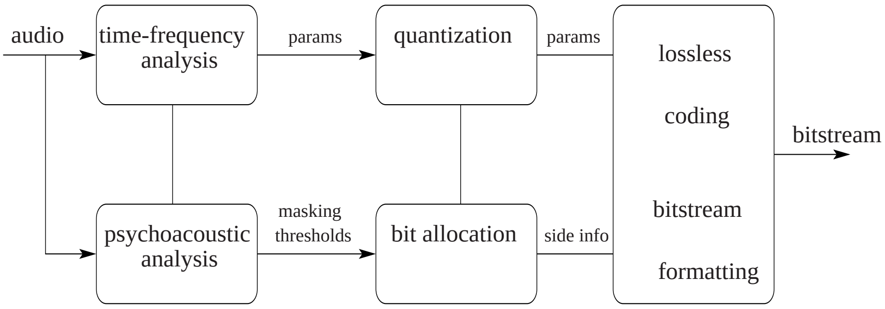


Figure 1.1: *The generic perceptual audio encoder block scheme.*

chologist steps in, since the field of psycho-acoustics, as documented thoroughly by Zwicker [57] and Moore [33], studies auditory phenomena such as simultaneous and temporal masking, critical bands and masking thresholds. The incorporation of this knowledge into a digital audio coding scheme is generally considered a cornerstone of the success of recent perceptual audio coding schemes.

The remaining parts necessary to implement an audio coding scheme come from the field of digital signal processing, where techniques such as linear prediction, filter banks, signal transformations and parametric descriptions are part of the audio coding tool set [28]. Armed with such techniques we can formulate the generic audio coding scheme depicted in Fig. 1.1. While the design of each block can have a profound impact on the resulting coding performance, arguably the most important part in an audio coder is the time-frequency analysis. It can be implemented in various manners and a proper choice is critical for the performance of the audio coding scheme under design. Most often, we encounter a multi-rate filter bank or discrete signal transform here. In a recent overview on perceptual audio coding by Painter and Spanias [40], several desirable filter bank characteristics [1] are enumerated as follows.

- The filter bank should provide a signal adaptive time-frequency tiling. In particular, it should be possible to employ a high-frequency resolution mode for stationary signals and a low-frequency resolution *critical band* mode for transient signals [43]. Moreover, it should be possible to efficiently switch between these and possibly other resolutions.

- The subband filters should posses strong stopband attenuation and provide good channel separation. Additionally, the filters should have high time-domain overlap to ensure minimum blocking artifacts.

- In the absence of quantization, perfect reconstruction of the input signal should be possible. The filter bank should be critically sampled to avoid an increase of data samples. Fast algorithms for implementation of the filter bank should be available.

---

[1] An equivalent list can be created in case of a signal transform.

A wealth of filter banks and signal transforms [31, 53, 56, 30] is made available to the designer of an audio coding scheme, whereas only a few exhibit the desired properties from the list above. While linear phase and pseudo-quadrature mirror filter banks [53] are applicable in certain situations and where the short time Fourier, discrete cosine, wavelet packet and local cosine transforms [30] each have their merits, we observe and recognize that the cosine-modulated filter bank [29], and more specifically, the modified discrete cosine transform [44, 31] (MDCT) has emerged as the most dominant signal transform in current audio coding schemes.

This dominance of the MDCT can for instance be observed from the audio coding schemes as standardized by the Motion Picture Expert Group (MPEG) of the International Standardization Organization (ISO), such as MPEG-1 layer 3 [34] (*”MP3”*) and MPEG-2/4 AAC [12, 35, 37] formats, and from Dolby AC-3 [1] (*”Dolby Digital”*) as standardized by the Advanced Television Systems Committee (ATSC). These standards document today’s most popular and often applied audio coding schemes and while the encoder implementation details typically vary heavily among the schemes, they all rely on the MDCT for time-frequency analysis.

1.1 Problem statement

The existing audio coding schemes provide a seemingly adequate performance and are widely employed. However, the ongoing change in storage media and transmission channels demands ever increasing performance and new behavior of audio coding systems. With the work presented in this thesis, the author hopes to make a contribution towards solving two specific issues that can be encountered on the road towards new and improved audio coding schemes.

Firstly, we realize that Shannon’s source coding theorems provide us with performance bounds but not the means to achieve those bounds. Furthermore, the incorporation of psychoacoustics into audio coders complicates the original source coding problem since a valid and robust perceptual distortion measure is generally hard to obtain. Hence, we observe that, while theoretically the solution to the audio coding problem can be found in Shannon’s work, other more heuristic approaches are often taken in practice. While this can still result, and clearly has resulted, in some highly efficient coding schemes, it becomes increasingly difficult to evaluate and compare the various schemes amongst each other. Thus far, new source coding results from information theory and novel heuristic approaches taken in perceptual audio coding have been proposed side by side. A crucial development in combining these two worlds is observed in the article on best wavelet packet bases by Vetterli and Ramchandran [45]. There, the application of *operational* RD optimization for transform coding of multimedia signals is presented. The use of operational RD optimization provides achievable *and* optimal results for a *given* coding environment. As such, we arrive at a situation were we indeed solve the original problem of minimizing a perceptually relevant distortion measure given a certain target rate.

Secondly, many existing coding schemes have been developed with a particular application area and range in mind. An example is the audio part of the MPEG-4 standard [37], in which a bouquet of audio coding schemes is defined for a variety of signals and bit rates. This requires an explicit user selection of the appropriate codec for a particular situation. While excellent results can be obtained if a scheme is applied in the proper situation, a performance penalty can be observed in case of a mismatch between the application and the coding scheme. Such a mismatch can occur under multiple circumstances, for example, with changing signal characteristics or varying network conditions. In light of the recent convergence of consumer electronics and mobile communication, and the emergence of ubiquitous heterogeneous network environments with time-varying bandwidth and delay constraints, the ability of a codec to adapt to time-varying signal and network characteristics becomes a critical factor.

We can come up with the following set of requirements for new coding schemes that are to be employed in future ambient intelligent landscapes.

- Transparent to the user such that no manual codec selection is required.
- Adaptive to time-varying input signals.
- Adaptive to time-varying network constraints such as bit rate and delay.
- Adaptive to time-varying networks conditions such as packet losses.
- Flexible with respect to computational complexity and power resources.

A coding scheme that adheres to these requirements is denoted by the term *universal* audio coding. It seems unlikely that a single signal processing tool can cope with this monumental task and indeed, we see that multiple coding strategies are employed in recent universal audio coding schemes [46, 11]. Again, operational RD optimization can be a valuable tool here, for the individual coding strategies [14] as well as the combined overall scheme [51, 20, 55, 26]. It allows for adaptive coding techniques where a coding scheme can adapt to signal and network characteristics such as rate and latency constraints.

The importance and relevance of operational RD optimization has been recognized by researchers from the video coding world [39], for increasing coding performance [49] as well as for benchmarking [21] purposes. Efficient implementations of the new H.264/MPEG-4 Advanced Video Coding standard [25, 36] often incorporate some form of RD control. It is interesting to note that even with relatively simple distortion measures, significantly increased performance can be obtained. The audio coding world has yet to embrace operational RD optimization as a trusted coding or benchmark tool and we see incorporation only in newer audio coding schemes such as the wavelet packet schemes in [19, 18] and the parametric schemes in [14, 20].

This thesis describes efforts to combine the technique of operational RD optimization with the MDCT as a time-frequency analysis tool, with a specific focus on adaptive time-frequency decomposition. The practical issues and difficulties are discussed and various possibilities for trade-offs are explored. In other words, we extend the

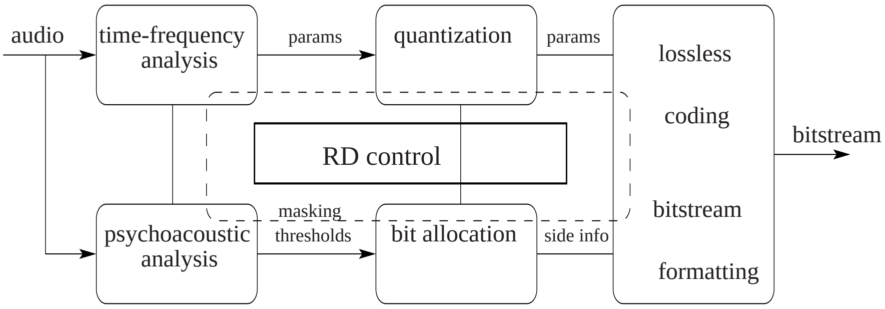


Figure 1.2: *The generic perceptual audio encoder block scheme extended with a rate-distortion control block.*

generic audio coding scheme with an RD optimization module as depicted in Fig. 1.2. As such, several new algorithms are constructed for RD optimal MDCT-based time-frequency decompositions.

1.2 Scope

The combination of an operational RD optimization framework with an MDCT-based audio coding system can take various forms. From Fig. 1.2, we can observe numerous relevant aspects of the coding system at hand that can be optimized by using the RD control block. In this thesis, we focus on the signal processing aspects of the RD control mechanism, the various techniques that exist for MDCT-based adaptive time-frequency decomposition and the interaction between the two. Therefore, in most cases our approach in developing new algorithms and evaluating their performance shall be based on simple and basic coding schemes, in which elementary forms of quantization and psychoacoustic modules will be employed, rather than the advanced versions as encountered in state-of-the-art audio coding schemes. As such, several aspects are treated superficially, or not at all. These aspects are now shortly discussed and we point to relevant articles for the interested reader to pursuit.

The first aspect under discussion is quantization. Quantization of the input data is a necessary processing step in any audio coding scheme. We ensure that data compression takes place through quantization, thus obtaining values for bit rate and distortion. In our algorithms, we explicitly incorporate selection of the optimal quantizer from predefined sets of quantizers. However, these quantizers are designed to be fairly simple, e.g. uniform and scalar, and we do not optimize their design, neither within or outside the algorithms. As a result, the quantizers we use deviate from the designs encountered in typical audio coding standards [34, 35]. Such designs, typically logarithmic or power-law scalar quantizers [22], are employed in audio coding to provide a more consistent signal-to-noise ratio over the range of quantizer values. Apart from scalar quantization, we also encounter vector quantization of the transform components in existing schemes, such as MPEG-4 Twin VQ [27, 37], Vorbis [23] and more recently in [32]. Closely related to quantization is the actual encoding of the resulting quantization indices. A variety of methods can be found, such as exponent/mantissa representation of blocks of time-frequency transform coefficients [1] or the use of scale factor bands [35, 37].

Several of these quantizer designs and encoding mechanisms can be incorporated in our RD optimization framework and in the algorithms presented in this thesis. All the scalar quantizer designs, where, apart from gain factors, the quantization levels are determined beforehand, can be employed in a straightforward manner. The more complex methods that represent blocks of transform coefficients as a single quantization index can be incorporated in our schemes as long as they operate on similar grids as the time-frequency decompositions employed in the scheme and as long as they allow rate computation for the smallest grid in the decomposition. For example, the MPEG-2/4 AAC scale factor band approach or Vorbis vector quantization can be readily employed in our time segmentation algorithms, but not directly in some of our frequency-domain methods.

While the incorporation of these more advanced quantizers and encoding mechanisms usually can lead to improved results, at the price of increased design and implementation time, it does not contribute to a significant further understanding of our problems at hand. That is, investigation of the interaction between the RD optimization framework and the time-frequency decomposition methods available for the MDCT is not clarified by incorporating advanced quantizer designs. Nevertheless, in literature several interesting examples of incorporating quantization and encoding in a RD optimization framework can be found that would combine well with our methods. In the work by Aggarwal [2, 3, 5] and in the paper by Bauer and Vinton [8], the MPEG-2/4 AAC quantization and encoding procedures are investigated and various RD optimal solutions are proposed. Their algorithms apply an RD optimization framework in which both a Lagrangian multiplier based approach as well as dynamic programming can play a role. Relating their and our work to the generic audio coding block scheme in Fig. 1.2, we observe that RD optimal solutions are indeed available for most of the constituent processing blocks. Their work provides techniques for quantization and encoding, whereas our work proposes algorithms for time-frequency decompositions, all of which are controlled by an RD optimization framework.

With respect to quantization, we would like to point to a new and promising approach as taken by Vafin in [51, 52] and Korten [41]. In their work, high-rate quantization is employed to derive analytical formulae for quantizer designs, for a given target rate. This technique allows us to directly compute the optimal quantizers for the given target rate, including the resulting rates and distortions. As such, a significant decrease of computational complexity can be obtained since it is no longer necessary to apply all possible quantizer and encoding possibilities. As it stands, the combination of this technique within an operational RD framework has been explored only for parametric audio coding schemes.

The second aspect we discuss here is the notion of perceptual distortion. In audio coding, we are naturally interested in the incurred perceptual distortion, that is, the disturbance from the original as perceived by a human listener. The study of perceptually relevant distortion measures is an important but monumental task, one that we can not hope to accomplish simultaneously with our main goals. Instead, we adopt a practical view where we either neglect perceptual aspects or we apply relatively simple perceptual models and suboptimal perceptual distortion measures. Inevitably, this limits the possibilities to compare the results as presented in this thesis, with existing commercial systems which do employ sophisticated methods to take into account auditory masking. Nevertheless, we argue that our approach is a logical choice and has its merits.

First of all, the perceptual model [54] that we employ in some of our work captures important basic aspects of auditory masking. Furthermore, it extends upon the existing ISO MPEG perceptual models by considering spectral integration. It has been applied with favorable results and has shown good correlation with subjective listening test, as is seen in e.g., [55, 14, 54]. Additionally, it allows for computation of the masking threshold independent of the segment length. This feature makes the model highly suitable for incorporation into a coding scheme in which flexible time segmentation is employed.

Secondly, the algorithms presented in this thesis support any perceptual distortion measure that can be expressed as an additive weighted mean square error (MSE) measure. That is, we assume that the perceptual distortion is computed as the difference between the original set of transform coefficients and the quantized set, weighted by a perceptually motivated weighting function. The masking threshold can serve as a basis for such a weighting function. The additivity constraint on the measure depends on the particular algorithm, i.e. whether the algorithm operates on time domain segments of frequency domain blocks. In most modern audio coding schemes, the noise-to-mask ratio (NMR) [10] is employed as a measure of perceptual audio quality. While two types of NMR are encountered, namely maximum NMR and average NMR (ANMR), we only claim support for the ANMR measure, under the aforementioned constraints. Again, we see that in [5, 8] the ANMR is also applied. The use of an ANMR-like measure allows us to update the perceptual model based on ongoing research in psychoacoustics, for example, such as encountered in the work of Dau [16, 15] and Baumgarte [9]. The details of such sophisticated and highly nonlinear psychoacoustic models can be caught in a sensitivity matrix, as proposed by Plasberg [42], which allows us to linearize the models and incorporate them in a weighted MSE measure. The relation between various forms of perceptual distortion measures is further studied in [13].

Concluding, while the use of a perceptual distortion measure will have an impact on the coding environment, e.g. for the computation of lossless codebooks, and on the performance of an audio coding system, it does not severely influence the RD optimization mechanism. In fact, an audio coding system based on RD optimization becomes relatively future proof and incorporation of new perceptual models and distortion measures, while highly relevant for the *performance* of the system, becomes a minor issue for the *operation* of the system.

As a final remark, we shortly consider scalable audio coding. RD optimization within an audio coding system enables dynamic system adaptation to the network and terminal characteristics and user requirements. In the case of bandwidth or bit rate constraints, RD optimization can be seen as a rate-control within the encoder. A different approach to cope with time-varying constraints on bit rate is taken with scalable coding. Scalable coding tries to decouple the processes of encoding, rate control and decoding. Here, multiple layers of coded audio data are created and a variety of user conditions can be handled by storing or receiving one, some or all layers. While this approach can be beneficial in various scenarios, invoking scalable tools is usually penalized by a decrease in compression performance, which is in particular true for so-called fine-granular scalable approaches and for scalability over broad ranges of bandwidths. Moreover, the combination of scalability with RD optimization can become quite complex. Scalable audio coding was studied in [24] and is defined within the MPEG-4 standard. More recently, Aggarwal investigated scalable audio coding for MPEG-2/4 AAC in [7, 6, 4].

1.3 Organization

The main body of this thesis consists of two complementary parts. First, a part with background information on the applied techniques and secondly, a part consisting of articles in which several new algorithms are proposed. The author hopes sincerely that this combination makes the thesis more readable as a whole and that the background chapters can serve as reference for further study and development of new audio coding algorithms, independent of the articles. Clearly, a drawback of this setup is the repeated occurrence of several essential parts throughout this thesis. However, the benefits of having both background and new work available may still outweigh this slightly redundant representation and the author apologizes for not having compressed this thesis to its entropy.

Part I - Background

Operational RD optimization and best basis search

In this chapter, the operational rate-distortion optimization framework is investigated. We start with formulating the audio coding problem as a rate-constrained allocation problem and then proceed to provide a step-by-step derivation of the generic operational solution. Furthermore, we look at best basis search techniques and describe two often employed fast search algorithms. Finally, we combine the two into our generic operational solution to the rate-constrained allocation problem.

The modified discrete cosine transform

The dominant signal transformation in existing audio coding schemes is the modified discrete cosine transform (MDCT). In the articles that constitute the main body of this thesis, the MDCT plays a prominent role. Therefore, a chapter devoted to the MDCT is included in this thesis to serve three purposes. First, as a whole this chapter provides background information on the MDCT and lists references to literature for further study. Second, some of the basic underlying properties of the MDCT are highlighted, such as MDCT window design and fast implementations of the transform. Finally, three techniques for obtaining adaptive time-frequency signal decompositions with the MDCT are discussed. These techniques, window switching, temporal noise shaping and subband merging, are used extensively in the articles in the second part of the thesis.

Part II - Articles

Paper I

In this paper, we develop the first of our RD optimal time-frequency decomposition algorithms. That is, a new flexible frequency decomposition algorithm is presented that jointly optimizes the MDCT structure and the bit allocation over the transform coefficients. We make use of the subband merging method, published by the author in [38], that allows for fast and efficient design of nonuniform filter banks. The new algorithm shows improvements in comparison to fixed uniform frequency decompositions, but we note that special care has to be taken to reduce the size of the decomposition overhead.

Paper II

We modify and extend the work in paper I in several ways. First, we employ a simple perceptual weighting method to obtain perceptually relevant results. Next, we simplify the bit allocation such that a single quantizer is used for a complete set of transform coefficients. Furthermore, we perform listening tests to compare the new algorithm with the situation where an MDCT is applied that leads to a uniform frequency decomposition.

Paper III

This paper presents the second of our RD optimal time-frequency decomposition algorithms. A flexible time segmentation algorithm is constructed that jointly optimizes the MDCT block lengths and the bit allocation over the individual signal segments. The combination of time-domain optimization and the MDCT windowing operation leads to certain dependency problems. We study an audio coding scheme in which an initial approach is employed where these dependencies are ignored. We then compare the behavior of the new algorithm with an existing one based on a binary tree search, for entropy and RD cost measures.

Paper IV

The flexible time segmentation algorithm outlined in paper III is suboptimal, in the sense that certain dependencies, that arise when individual signal segments are windowed and overlap-add is applied between adjacent signal segments, are ignored. In this paper, we extend the algorithm from paper III such that these dependencies are incorporated and we show that an optimal solution can be obtained in polynomial time. This algorithm gives an upper bound to the achievable performance of the existing MDCT-based time segmentation algorithms.

Paper V

Both time segmentation algorithms as presented in papers III and IV lead to a high computational complexity. In this paper it is investigated whether upfront time segmentation can reduce computational complexity without a significant decrease in performance. Upfront time segmentation can be accomplished by replacing the rate-distortion cost functional with a low-complexity cost measure that is independent of bit rate and perceptual distortion. We investigate the perceptual entropy measure and show that it can be a viable and low-complexity alternative to the rate-distortion optimal time segmentation algorithms presented earlier.

Paper VI

In this paper we study a third technique for MDCT-based adaptive time-frequency decomposition, called temporal noise shaping, which is a technique for reshaping the quantization noise in the time domain through open-loop linear predictive coding of frequency domain coefficients. We investigate its combination within a rate-distortion optimization framework, where a jointly optimal selection of the prediction filter order and the quantizer for coding the transform coefficients can be made, such that a perceptual distortion is minimized for a given target rate. A comparison is made with a scheme where temporal noise shaping is employed similar to its operation in MPEG-2/4 AAC [35, 37].

Bibliography

[1] ATSC A/52/10. United states advanced television systems committee digital audio compression (AC-3) standard, doc.A/52/10, December 1995.

[2] A. Aggarwal, S. Regunathan, and K. Rose. Trellis-based optimization of MPEG-4 advanced audio coding. In *Proceedings of the 2000 IEEE Workshop on Speech Coding*, pages 142–144, Wisconsin, USA, September 2000.

[3] A. Aggarwal, S. Regunathan, and K. Rose. Near-optimal selection of encoding parameters for audio coding. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'01)*, pages 3269–3272, Salt Lake City, USA, May 2001.

[4] A. Aggarwal, S. Regunathan, and K. Rose. Efficient scalable coding of stereophonic audio by conditional quantization and estimation-theoretic prediction. In *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, pages 465–468, Hong Kong, China, April 2003.

[5] A. Aggarwal, S. Regunathan, and K. Rose. A trellis-based optimal parameter value selection for audio coding. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2):623–633, March 2006.

[6] A. Aggarwal and K. Rose. Approaches to improve quantization performance over the scalable advanced audio coder. In *Proceedings of the 112th AES Convention*, Munich, Germany, May 2002.

[7] A. Aggarwal and K. Rose. A conditional enhancement-layer quantizer for the scalable mpeg advanced audio coder. In *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'02)*, pages 1833–1836, Orlando, USA, May 2002.

[8] C. Bauer and M. Vinton. Joint optimization of scale factors and huffman code books for MPEG-4 AAC. *IEEE Transactions on Signal Processing*, 54(1):177–189, January 2006.

[9] F. Baumgarte. Improved audio coding using a psychoacoustic model based on a cochlear filter bank. *IEEE Transactions On Acoustics, Speech, And Signal Processing*, 10(7):495–503, October 2002.

[10] R.J. Beaton, J.G. Beerends, M. Keyhl, and W.C. Treurniet. *Objective Perceptual Measurement of Audio Quality*. Audio Engineering Society, 1996.

[11] B. Bessette, R. Lefebvre, and R. Salami. Universal speech/audio coding using hybrid acelp/tcx techniques. In *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'05)*, pages III 301–304, Philadelphia, USA, May 2005.

[12] M. Bosi and et al. ISO/IEC MPEG-2 advanced audio coding. *Journal of the Audio Engineering Society*, 45:789–812, October 1997.

[13] M.G. Christensen and S.H. Jensen. On perceptual distortion minimization and nonlinear least-squares frequency estimation. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1):99–109, January 2006.

[14] M.G. Christensen and S. van de Par. Efficient parametric coding of transients. *IEEE Transactions on Audio, Speech and Language Processing*, 14(4):1340–1351, July 2006.

[15] T. Dau, B. Kollmeier, and A. Kohlrausch. Modelling auditory processing of amplitude modulation. i. detection and masking with narrowband carriers. *Journal of the Acoustical Society of America*, 102(5):2892–2905, November 1997.

[16] T. Dau, D. Püschel, and A. Kohlrausch. A quantitative model of the effective signal processing in the auditory system. i. model structure. *Journal of the Acoustical Society of America*, 99(6):3615–3622, June 1996.

[17] F.K. Engel. Magnetic tape – from the early days to the present. *Journal of the Audio Engineering Society*, 36:7606–7616, July 1988.

[18] M. Erne and G. Moschytz. Audio coding based on rate-distortion and perceptual optimization techniques. In *Proceedings of the AES 17th International Conference: High-Quality Audio Coding*, pages 220–225, Florence, Italy, September 1999.

[19] M. Erne, G. Moschytz, and C. Faller. Best wavelet-packet bases for audio coding using perceptual and rate-distortion criteria. In *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'99)*, pages 909–912, Phoenix, USA, March 1999.

[20] Heusdens et al. Bit-rate scalable intraframe sinusoidal audio coding based on rate-distortion optimization. *Journal of the Audio Engineering Society*, 54(3):167–188, March 2006.

[21] T. Wiegand et al. Rate-constrained coder control and comparison of video coding standards. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):688–703, July 2003.

[22] A.J.S. Ferreira. Optimum quantization of flattened MDCT coefficients. In *Proceedings of the 115th AES Convention*, New York, USA, October 2003.

[23] Xiph.Org Foundation. Ogg vorbis. http://www.vorbis.com/, 1994.

[24] J. Herre, E. Allamanche, K. Brandenburg, M. Dietz, B. Teichmann, B.Grill, A. Jin, T. Moriya, N. Iwakami, T. Norimatsu, M. Tsushima, and T.Ishikawa. The integrated filterbank based scalable MPEG-4 audio coder. In *105th AES Convention, Preprint 4810*, San Francisco, USA, September 1998.

[25] Telecommuncation Sector H.264 International Telecommunications Union. Advanced video coding for generic audiovisual services, 2005.

[26] IST-2001-34095. ARDOR:adaptive rate-distortion optimised sound coder. http://www.extra.research.philips.com/euprojects/ardor/, 2001.

[27] N. Iwakami and T. Moriya. Transform domain weighted interleave vector quantization (twin VQ). In *Proceedings of the 101st AES Convention*, Los Angeles, USA, November 1996.

[28] N. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice-Hall, Englewood Cliffs, NJ, 1984.

[29] R.D. Koilpillai and P.P. Vaidyanathan. Cosine-modulated FIR filter banks satisfying perfect reconstruction. *IEEE Transactions On Signal Processing*, 40(4):770–783, April 1992.

[30] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, New York, 1998.

[31] H.S. Malvar. *Signal Processing with Lapped Transforms*. Artech House, Boston, MA, 1992.

[32] N. Meine and B. Edler. Improved quantization and lossless coding for subband audio coding. In *Proceedings of the 118th AES Convention*, Barcelona, Spain, May 2005.

[33] B.C.J. Moore. *An Introduction to the Pscychology of Hearing*. Academic Press, Berlin, Germany, 1997.

[34] International Standard ISO/IEC 11172-3 (MPEG). Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s. part 3: Audio, 1993.

[35] International Standard ISO/IEC 13818-7 (MPEG). Information technology - generic coding of moving pictures and associated audio, part 7: Advanced audio coding, 1997.

[36] International Standard ISO/IEC 14496-10 (MPEG). Information technology - coding of audio visual objects, part 10: Advanced video coding, 2005.

[37] International Standard ISO/IEC 14496-3 (MPEG). Information technology - coding of audio visual objects, part 3: Audio, 1999.

[38] O.A. Niamut and R. Heusdens. Subband merging in cosine-modulated filter banks. *IEEE Signal Processing Letters*, 10(4):111–114, April 2003.

[39] A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, 15(6):23–50, November 1998.

[40] T. Painter and A. Spanias. Perceptual coding of digital audio. 88(5):451–515, April 2000.

[41] J. Jensen P.E.L. Korten and R. Heusdens. High-resolution spherical quantization of sinusoidal parameters. *to appear in IEEE Transactions on Audio, Speech and Language Processing*, 2007.

[42] Jan H. Plasberg and W. Bastiaan Kleijn. The sensitivity matrix: Using advanced auditory models in speech and audio processing. *to appear in IEEE Transactions on Audio, Speech and Language Processing*, January 2007.

[43] J. Princen and J.D. Johnston. Audio coding with signal adaptive filter banks. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'95)*, pages 3071–3074, Detroit, USA, May 1995.

[44] J.P. Princen, A.W. Johnson, and A.B. Bradley. Subband/transform coding using filter bank designs based on time domain aliasing cancellation. In *Proceedings of the 1987 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'87)*, pages 2161–2164, Dallas, USA, April 1987.

[45] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions on Image Processing*, 2(2):160–175, April 1993.

[46] S.A. Ramprashad. The multimode transform predictive coding paradigm. *IEEE Transactions on Speech and Audio Processing*, 11(2):117–129, March 2003.

[47] O. Read and W.L. Welch. *From Tin Foil to Stereo: Evolution of the Phonograph*. H.W. Sams and Co, Indianapolis, USA, 1977.

[48] C.E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423.

[49] G.J. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine*, 15(6):74–90, November 1998.

[50] Compact Disc Digital Audio System. (IEC/ANSI) CEI-IEC-908, 1987.

[51] R. Vafin and W.B. Kleijn. Entropy-constrained polar quantization and its application to audio coding. *IEEE Transactions on Speech and Audio Processing*, 13(2):220–232, March 2005.

[52] R. Vafin and W.B. Kleijn. Rate-distortion optimized quantization in multistage audio coding. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1):311–320, January 2006.

[53] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs, NJ, 1993.

[54] S. van de Par, A. Kohlrausch, G. Charestan, and R. Heusdens. A new psychoacoustical masking model for audio coding applications. In *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'02)*, pages 1805–1808, Orlando, USA, May 2002.

[55] N.H. van Schijndel and S. van de Par. Rate-distortion optimized hybrid sound coding. In *Proceedings of the 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'05)*, pages 235–238, New York, USA, October 2005.

[56] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Signal Processing Series. Prentice-Hall, 1995.

[57] E. Zwicker and H. Fastl. *Psychoacoustics Facts and Models*. Springer-Verlag, Berlin, Germany, 1990.

Part I

Background

Operational R-D can serve as the middle ground
between irrelevant theory and ad-hoc tweaking.

Ortega and Ramchandran, Signal Processing Magazine, 1998

The ease of window design and the ability to adapt
the filterbank resolution by altering a single parameter have made
the MDCT the transform of choice for most existing audio coders.

Bosi and Goldberg, Introduction to Digital Audio Coding and Standards

Chapter 2

Operational RD Optimization and Best Basis Search

This chapter contains a short overview of relevant aspects of the fields of operational rate-distortion optimization and best basis search methods and investigates some recent algorithms that combine the two. We start with an introduction and literature overview in Section 2.1. Next, in Section 2.2 the rate-constrained allocation problem is formulated and tools to solve this problem are studied. Then, in Section 2.3 on best basis search methods, we introduce the notion of time-frequency tilings and review two important best basis search algorithms. We also investigate the inclusion of best basis search methods in a rate-distortion optimization framework. Finally, we draw some conclusions in Section 2.4.

2.1 Introduction

Emerging from the field of source coding or compression, rate-distortion (RD) theory aims at the optimal approximation of a source signal under a modelling budget constraint. While an approximation can be obtained in various manners, we are considering those approximations that results from *quantization* of the source signal at hand. Quantization [37, 24] is the process of projecting the amplitudes of single or multiple samples of the source signal on a discrete set of codewords or quantization points. The quantization process inevitably leads to an approximation distortion and the modelling cost is typically expressed in bits. In the particular case of a digital audio source, we want to find the quantized representation of the audio signal that leads to the minimum perceptual distortion, at a given target bit rate.

Within an optimization framework, the search for such a representation can be formulated as a rate-constrained allocation problem. In this thesis, the RD optimization problem is considered in the *operational sense*, i.e. we seek to obtain the best achievable performance for a particular input audio signal, given the choice of compression framework or coding environment. Such a coding environment can consist of a particular signal transform, a set of quantizers and corresponding codebooks. We can expect that the *a priori* selection of a coding environment will limit the achievable performance, but we assume that such a specific coding environment can efficiently capture the relevant statistical dependencies associated with the source beforehand, while satisfying practical system requirements and constraints related to coding complexity, delay and memory usage. This operational approach searches for the best operating points for that specific system by constructing a so-called *operational rate-distortion curve*, where every point of the curve can be directly achieved by the given coding system. An operating point is represented as an (R, D)-pair, i.e. a combination of the bit rate R and quantization or coding distortion D values for a particular setting of the coding framework.

In contrast, the conventional classical RD theory, as laid out by Shannon in his seminal paper [59], is mainly focussed on the derivation of performance bounds for certain limited statistical source classes, using an *unconstrained* coding system. That is, the coding system assumes no constraints on the coding delay, available memory and the number of required computations. Although the computation of such bounds allows an insightful determination of achievable and non-achievable performance regions, these bounds are not always tight for situations of practical relevance, nor are they constructive. The derivation of the bounds relies heavily on a correct characterization of the source, whereas complex non-stationary sources such as audio are generally hard to model. Moreover, they are only valid for a limited set of distortion measures, which does not include advanced psycho-acoustically motivated distortion measures as used in modern audio coding systems.

Where the performance bounds provided by classical RD theory are typically reached when the dimensionality of the input vectors approaches infinity, high-resolution theory as pioneered by Bennett [3] can be applied for finite-length input signals. The assumption of high resolution source quantization leads to a staircase model of the probability density function of the source, such that it is constant over each quantization interval. Furthermore, quantization overload distortion is ignored. It is then possible to derive an analytic relation between the quantizer point density [1] and the resulting distortion. In most practical cases, the high resolution assumption is valid if the available rate is high and as such, the constraint on the rate generally prohibits low bit rate transform audio coding. The reader is referred to [4, 13, 44, 24] for more elaborate details and extensions of classical RD and high-resolution theory.

The initial work on operational RD optimization was done by Shoham and Gersho in their paper on efficient bit allocation [60], where they employ a specific form of Everett's Lagrange multiplier method [19]. There, the problem of rate-constrained bit allocation is formulated as an unconstrained optimization problem using a Lagrangian cost function of both bit rate and distortion. A similar approach is taken by

---

[1] In the case of scalar quantization, the quantizer point density reduces to the inverse of the quantizer stepsize.

Chou, Lookabaugh and Gray in their work on entropy-constrained vector quantization and tree-structured source coding [8, 9], where they apply the Blahut algorithm from [6]. The aforementioned papers consider independent rate-constrained bit allocation problems, in which a target rate or number of bits is set as an input parameter to the coding system and some assumptions on the independency of the rates and distortions are made. More recently, the field of operational RD optimization has been extended by Ortega, Ramchandran and Vetterli, to buffer or delay-constrained optimization [46, 56] and to algorithms for dependent quantization [54, 55]. An extensive overview of the field of operational RD optimization is provided by Ortega and Ramchandran in [45]. Although their work is mainly oriented towards image and video coding applications, most of the underlying principles and methods are generally applicable to audio coding.

Whereas quantization and RD optimization prove to be essential parts of an audio coding system, an equally significant part consists of the signal transform and its adaptivity to input signals. We concentrate on a particular class of signal transforms, given by linear expansions, that decompose a signal into elementary building blocks. Traditional linear expansions such as obtained by Fourier [47, 7], discrete cosine [58] or wavelet transforms [41, 14], are not flexible enough for representing signals with components whose localizations vary widely in time and frequency. For instance, the Fourier transform is not particularly suited for representing time-localized signals, whereas the wavelet transform is ineffective for resolving high frequency components. For signals constructed from a mixture of these components, the rigidity of linear expansions can results in highly suboptimal coding performance. Such signals require expansions into waveforms that can be adapted to the local signal structure. These waveforms are called time-frequency atoms and it is the construction of and search through dictionaries containing time-frequency atoms that lead to the best basis search problem. In this thesis, we do not treat overcomplete dictionaries that are searched by matching pursuit [42], nor do we look at dictionaries of parametric signal models, such as sinusoidal analysis [23].

We are again interested in those solutions that can be employed in practical audio coding systems and hence we require fast algorithms. We can choose from a selection of best basis search methods that are closely related to adaptive signal transforms and find their application in signal approximation problems such as estimation [48, 43], denoising [38] and compression. Coifman, Meyer and Wickerhauser formalized the concept of best bases in [12, 10, 11] where they studied tree-structured time-frequency decompositions resulting from adaptive signal transforms such as the wavelet packet and local cosine transforms [11, 40]. Recently, best basis algorithms for local cosines have been explored by Villemoes [63] and Huang *et al.* [36, 35]. The general aim of best basis search algorithms is to adaptively select, from a given dictionary or library of bases, the basis which minimizes a predefined cost measure for a given input signal. For audio coding purposes, these algorithms allow the construction of so-called time-frequency tilings, which describe the coverage of the signal by a certain basis in both time and frequency domains.

The combination of best basis search methods and operational RD optimization was first proposed by Ramchandran and Vetterli [57] who studied image compression with wavelet packets and DCT bases. Their research led to a surge of interest in algorithms that combined operational RD and best basis search methods. In [27, 29, 28, 30, 31, 32] numerous algorithms were developed for adaptive time-frequency decomposition with wavelet packet and DCT bases. Specifically in [28], the MDCT was employed for time segmentation. An important step was taken in [30] in which a dynamic programming based algorithm was proposed instead of the binary tree algorithms that were used earlier on. The work presented in Part II of this thesis relies heavily on the concepts and techniques as developed in these papers. Subsequent extensions to coding techniques such as linear prediction and sinusoidal coding and to the incorporation of side information were taken by Prandoni and Vetterli [50, 51, 49, 52]. Examples of audio coding systems that incorporate some of these algorithms can be found in, e.g., [53] for a frequency-varying MDCT, in [18, 17] for wavelet packets and in [34, 33] for sinusoidal coding.

2.2 Operational RD optimization

In this section, we formulate the problem of rate-constrained bit allocation and investigate a popular technique for solving the problem.

2.2.1 problem formulation

In most of the compression applications encountered, the bit rate is restricted to a maximum number of bits that can be used. In audio coding, this situation occurs when the coded data has to be stored on a medium with a limited storage capacity or transmitted through a channel with a limited (time-varying) bandwidth. The total number of available bits, called the target rate and denoted R_T, has to be allocated to the coded signal in such a way that an overall distortion metric is minimized. In a practical coding system it is necessary to determine at which levels of granularity the encoder is optimized by selecting an appropriate coding unit. For example, in audio coding, the basic coding unit could be a sample (time-domain or frequency-domain), a single analysis frame or a segment consisting of multiple frames.

Let an input signal be divided into N coding units. For each coding unit, a finite set of Q admissible quantizers or coding templates is available. The application of the jth quantizer for the ith coding unit is denoted $q_{i,j}$ and leads to a rate-distortion pair where the rate is denoted as $r_i(q_{i,j})$ and the distortion as $d_i(q_{i,j})$. The overall selection of coding templates for each and every coding unit is represented by a quantizer allocation vector $\mathbf{q} \in \mathcal{Q}$, where the set of allocation vectors $\mathcal{Q}$ consists of all possible ways of allocating the quantizers over the coding units, hence $|\mathcal{Q}| = Q^N$. Furthermore, R denotes the total number of bits that is allocated to the coding units and D is the resulting total distortion. Every combination of the total rate and distortion resulting from a particular rate allocation $\mathbf{q}$ leads to an (R, D)-pair and the application of all $|\mathcal{Q}|$ gives rise to the set or cloud of $\big(R(\mathbf{q}), D(\mathbf{q})\big)$-pairs for the particular coding framework and input signal.

Formulation 1. ***Rate-Constrained Bit Allocation Problem***
Given the rate constraint R_T, find the optimal bit allocation $\mathbf{q}^* \in \mathcal{Q}$ as

$$\min_{\mathbf{q}\in\mathcal{Q}} D(\mathbf{q}) \quad \textit{subject to} \quad R(\mathbf{q}) \leq R_T. \tag{2.1}$$

The problem of Formulation 1 can be generalized to the case where additional constraints exist. For example, in the case of audio streaming across a network, the rate-constrained allocation formulation does not suffice. The coding units are subject to a delay constraint, i.e. they have to be available at the decoder at a certain time in order to be played back. The characteristics of the transmission channel become important for the complexity of the allocation. It must be known whether the channel provides a constant bit rate or a variable bit rate, whether the channel delay is constant and the channel is reliable. We do not treat the delay-constrained allocation problem in this thesis, however, in Section 2.3 we refer to some studies on the influence of system delay on coding performance. Formal descriptions of delay-constrained allocation problems can be found in [56, 45].

In practise, the rate-constrained allocation problem of Formulation 1 involves bit allocation over multiple coding units. If dependencies exist between the coding units, the efficient search for a solution to the coding problem becomes a difficult process. Hence, we often assume that the constrained allocation problem is *independent*. That is, we assume that the choice and selection of a quantizer, or more general, a coding template, for a particular coding unit can be done independent of other coding units and does not affect the selection of quantizers in other coding units. If this assumption holds, we can independently quantize the coding units, which in turn leads to additive rates and distortions, given an appropriate additive distortion measure. As such, the independence property is attractive from a computational point of view, since it can lead to fast algorithms for solving the constrained allocation problem.

In many scenarios, the assumption that a selection of a coding template for a certain coding unit does not affect other units is invalid, i.e. the allocation problem is not independent. In general, two types of dependency scenarios can be identified. *Trellis-based dependencies* occur in the case where the memory of the system and the number of possible dependent cases are finite. Coding choices for a single unit depend only on a finite set of previous coding units. In that case, a Trellis diagram can be used to represent the choices and a technique such as dynamic programming can be used to minimize the overall cost function. These type of dependencies are studied in, e.g., [55, 45] and in [49]. In contrast, *tree-based dependency* denotes the situation where all possible combinations generated by successive coding template choices can be represented as a tree with the number of branches growing exponentially with the number of levels of dependency. The exponential growth in the number of combinations makes an exact solution very complex. To simplify the search for the optimal solution, good heuristics or greedy approaches can be applied, or models of the dependent operational RD curves can be employed.

2.2.2 solution to the rate-constrained problem

Obviously, the solution for the problem of Formulation 1 can be obtained by a brute-force exhaustive search through the entire set of (R, D)-pairs. Since the associated complexity of such a search is prohibitive for all practical purposes, we require more advanced techniques to obtain the solution, or an approximate solution, at a reduced complexity. A particular technique of interest is based on the discrete version of *Lagrangian* optimization, as proposed by Everett [19]. The first application in source coding, based on this framework, can be found in [60]. The technique can be described as follows.

A real-valued Lagrange multiplier $\lambda \geq 0$ is introduced and a Lagrangian cost function of both bit rate and distortion is defined as

$$J(\lambda, \mathbf{q}) = D(\mathbf{q}) + \lambda R(\mathbf{q}). \tag{2.2}$$

We can now formulate a new unconstrained problem as follows.

Formulation 2. ***Unconstrained Bit Allocation Problem***
Find the optimal bit allocation $\mathbf{q}^* \in \mathcal{Q}$ *as*

$$\min_{\mathbf{q} \in \mathcal{Q}} J(\lambda, \mathbf{q}). \tag{2.3}$$

The following theorem relates the constrained problem in (2.1) to the unconstrained problem in (2.3) through the inclusion of the target rate.

Theorem 1. ***(rate-constrained bit allocation) [60]*** *For any* $\lambda \geq 0$*, the solution* $\mathbf{q}^*$ *to the unconstrained problem in (2.3), given by*

$$\mathbf{q}^* = \arg\min_{\mathbf{q} \in \mathcal{Q}} J(\lambda, \mathbf{q}),$$

is also to the solution to the constrained problem in (2.1) with constraint $R_T = R(\mathbf{q}^*)$.

Proof: For any solution $\mathbf{q}^*$ to the unconstrained problem in (2.3) we have that

$$D(\mathbf{q}^*) + \lambda R(\mathbf{q}^*) \leq D(\mathbf{q}) + \lambda R(\mathbf{q}) \quad \forall\, \mathbf{q} \in \mathcal{Q},$$

or, equivalently,

$$D(\mathbf{q}^*) - D(\mathbf{q}) \leq \lambda \Big(R(\mathbf{q}) - R(\mathbf{q}^*) \Big) \quad \forall\, \mathbf{q} \in \mathcal{Q}. \tag{2.4}$$

Eq.(2.4) holds for all $\mathbf{q} \in \mathcal{Q}$, so it holds in particular for the subset $\mathcal{Q}^* \subset \mathcal{Q}$ given by

$$\mathcal{Q}^* = \Big\{ \mathbf{q} : R(\mathbf{q}) \leq R(\mathbf{q}^*) \Big\}. \tag{2.5}$$

Since $\lambda \geq 0$,

$$D(\mathbf{q}^*) - D(\mathbf{q}) \leq 0 \, \forall \, \mathbf{q} \in \mathcal{Q}^*. \tag{2.6}$$

and thus $\mathbf{q}^*$ is the solution to (2.1) for $R_T = R(\mathbf{q}^*)$. □

In words, (2.5) states that $\mathcal{Q}^*$ is the set of all allocations $\mathbf{q}$ that result in a rate $R(\mathbf{q})$ lower than or equal to $R(\mathbf{q}^*)$. Hence, we have that $R(\mathbf{q}) - R(\mathbf{q}^*) \leq 0$ for all allocations $\mathbf{q} \in \mathcal{Q}^*$.

Using the fact that $\lambda \geq 0$, (2.6) then shows that all allocations $\mathbf{q} \in \mathcal{Q}^*$ lead to a distortion $D(\mathbf{q})$ that is higher than or equal to $D(\mathbf{q}^*)$. Let us assume that $R_T = R(\mathbf{q}^*)$. We then have that for all allocations $\mathbf{q} \in \mathcal{Q}$ that lead to a rate $R(\mathbf{q}) \leq R_T$, the allocation $\mathbf{q}^*$ leads to the lowest distortion $D(\mathbf{q}^*)$ and hence, $\mathbf{q}^*$ is the solution to (2.1) for $R_T = R(\mathbf{q}^*)$.

Note that Theorem 1 does not guarantee the existence of a solution to (2.1). Only if $R(\mathbf{q}^*) = R_T$, i.e., when the optimal bit allocation vector $\mathbf{q}^*$ gives rise to a rate that is equal to the target rate, $\mathbf{q}^*$ is the solution to (2.1). The problem is now reduced to finding the optimal value of λ, say λ^*, such that the target rate is met. Let $W(\lambda)$ be defined as

$$W(\lambda) = \min_{\mathbf{q} \in \mathcal{Q}} \Big[D(\mathbf{q}) + \lambda \big(R(\mathbf{q}) - R_T \big) \Big]. \tag{2.7}$$

The function W is called the *dual function* corresponding to the optimization problem in (2.1) and it can be shown that W is concave in λ, see e.g. [57]. The *dual problem* corresponding to (2.1) is then defined as

$$\max_{\lambda \geq 0} \min_{\mathbf{q} \in \mathcal{Q}} \Big[D(\mathbf{q}) + \lambda \big(R(\mathbf{q}) - R_T \big) \Big], \tag{2.8}$$

that is, λ^* is obtained by maximization of W over all possible values of $\lambda \geq 0$.

The relation between the two solutions obtained from the two optimization problems in (2.1) and (2.8) is expressed through the Lagrangian duality theorems, see e.g., [25]. It turns out that the solution to the dual problem (2.8) lies on the *convex hull* of the set of (R, D)-pairs. This hull defines the boundary between achievable and non-achievable performance regions. Since we are working with a discrete set of coding units and templates, and a positive rate constraint, we only have weak duality between the problems in (2.1) and (2.8). That is, some solutions to (2.1) reside inside the convex hull and are hence infeasible solutions to (2.8). The difference between the solutions to the original problem of (2.1) and the dual problem in (2.8) is called the *duality gap*.

An example of the occurrence of the duality gap is displayed in Fig. 2.1. A randomly generated time-domain signal is divided into four segments which can be seen as coding units. Furthermore, 16 stepsizes for a normalized uniform quantizer are available as coding templates. Fig. 2.1 shows the set of RD pairs and its convex hull. The large crosses denote solutions to the dual problem for a range of target rates. In particular, the thick vertical line indicates a target rate of 18 bits. The two circles denote solutions

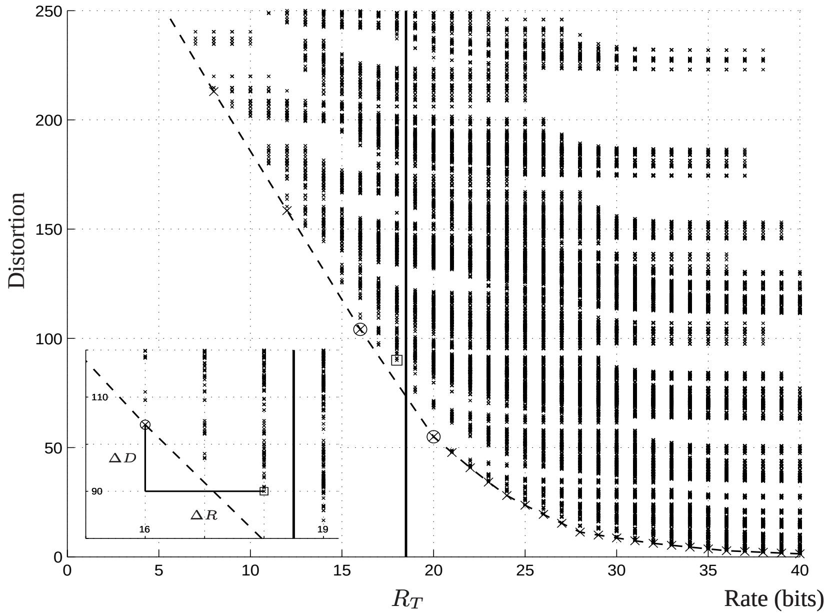


Figure 2.1: *Lagrange optimization for a random signal divided into* 4 *segments and quantized with* 16 *coding templates. The resulting set of RD pairs is shown along with its convex hull. Note that the optimal solution at the target rate of* 18 *bits can not be obtained, since the corresponding RD pair lies inside the convex hull.*

to the dual problem that lie around the target rate. The solution to the original problem is indicated by a square. The magnitude of the duality gap can be observed from the zoomed plot.

Since the Lagrange optimization technique searches only along the convex hull of the set of (R, D)-pairs rather than the entire set, the computational complexity is reduced significantly, compared to an exhaustive search procedure. We can further reduce the complexity of the optimization procedure by taking advantage of the assumptions of additivity and independence of the rates and distortions over the coding units. This allows us to simplify (2.3) as

$$\begin{aligned}
\min_{\mathbf{q}\in\mathcal{Q}} D(\mathbf{q}) + \lambda R(\mathbf{q}) \quad &= \quad (\textit{additivity}) \\
\min_{\mathbf{q}\in\mathcal{Q}} \sum_{i=1}^{N} \Big(d_i(q_{i,j}) + \lambda r_i(q_{i,j}) \Big) \quad &= \quad (\textit{independence}) \\
\sum_{i=1}^{N} \Big(\min_{j} d_i(q_{i,j}) + \lambda r_i(q_{i,j}) \Big), &
\end{aligned}$$

and the final operational RD problem is formulated as follows.

Formulation 3. ***Operational Rate-Distortion Problem***
Given the constraint R_T*, find* λ^* *as*

$$\lambda^* = \arg\max_{\lambda \geq 0} \Big(\sum_{i=1}^{N} \big(\min_j d_i(q_{i,j}) + \lambda r_i(q_{i,j}) \big) - \lambda R_T \Big). \tag{2.9}$$

Since for every λ the solution to (2.7) lies on the convex hull of the set of RD-pairs, the solution to the outer maximization in (2.9) can be obtained in an iterative way using the bisection algorithm as is done in [57]. The bisection algorithm finds the optimal λ^*, which gives rise to the zero crossing of the derivative of $W(\lambda)$. Given an uncertainty interval spanned by two initial or earlier approximate solutions, in which the solution is known to exist, the bisection method evaluates W at the midpoint of this interval and compares its sign to the existing two values [22]. The bisection algorithm can be formulated in the following manner.

Algorithm 1. ***Bisection***

```
[1] Determine $\lambda_{\min}$ and $\lambda_{\max}$ such that $R^*(\lambda_{\max}) \leq R_T \leq R^*(\lambda_{\min})$.
[2] Select a starting value $\lambda_{\min} \leq \lambda \leq \lambda_{\max}$.
[3] Compute $R^*(\lambda)$.
[4] If $R^*(\lambda) = R_T$, the optimum is found and the algorithm terminates.
    Else if $R^*(\lambda) > R_T$, set $\lambda_{\max} \leftarrow \lambda$.
    Else $\lambda_{\min} \leftarrow \lambda$.
[5] Determine the new value of $\lambda$:
        $\lambda = |D^*(\lambda_{\min}) - D^*(\lambda_{\max})| / |R^*(\lambda_{\max}) - R^*(\lambda_{\min})|$
    and goto step 3.
```

A conservative choice for the minimum and maximum values of λ in step 1 of Algorithm 1 is $\lambda_{\min} = 0$ and $\lambda_{\max} = \infty$. However, the number of iterations -and thus the algorithmic complexity- required for obtaining the optimal λ can be reduced with good initial values.

In general, the duality gap can be reduced by generating a dense set of RD pairs, which ensures that the convex hull is densely populated. If the convex hull of the RD curve consists of a few discrete operating points only, the solution found with the Lagrange method can be highly suboptimal. In such situation, a more appropriate technique for solving the rate-constrained allocation problem of (2.1) is to formulate a corresponding deterministic *dynamic programming* (DP) problem which can be seen as a multi-stage decision process. A Trellis diagram associated to the problem is created that represents all possible solutions, where the stages of the Trellis correspond to the coding units and the states represent accumulative rate and distortion. The DP algorithm traverses the trellis and prunes suboptimal branches. At the heart of the algorithm lies the *Principle of Optimality* introduced by Bellman [2], which roughly states that any subpath of an optimal path is also optimal. Various well-known algorithms rely on this principle, such as the Viterbi algorithm [20] and Dijkstra's shortest-path algorithm [15].

In contrast to Lagrange optimization, the DP approach considers all possible RD pairs, hence its solution will corresponds to that of (2.1). Hence, the complexity of DP grows quadratically with the number of coding units. Therefore, in most situations the usage of the Lagrangian method is desired and justified since the convex hull is densely populated. The DP approach can also be employed to solve (partly) dependent allocation problems, as is done in [54, 55, 49]. An interesting hybrid technique that combines the two methods is presented in [65]. We will not use the DP approach to solve the rate-allocation problem directly, but we come back to it in the next section when studying fast algorithms for best basis search.

2.3 Best basis search methods

In this section we study best basis search algorithms and their application in an operational RD optimization framework. We first introduce linear expansions and the notion of time-frequency tilings. Next, we investigate two popular fast search algorithms, that have been employed in many of the articles constituting the second part of this thesis. We conclude the section with studying the inclusion of these algorithms in the operational RD problem as formulated in (2).

2.3.1 linear expansions and best bases

Best basis search algorithms were pioneered by Coifman, Meyer and Wickerhauser in [12, 10], where they constructed bases from local cosine and wavelet packet transforms. As mentioned in Section 2.1, such transforms can be seen in the framework of linear signal expansions. When making a linear expansion of a signal $x \in \ell_2(\mathbb{Z})$ where $\ell_2(\mathbb{Z})$ is the set of square-summable sequences, we want to find a set of basis functions $\Phi = \{\varphi_k\}_{k \in \mathbb{Z}}, \varphi_k \in \ell_2(\mathbb{Z})$, called a *basis*, such that we can uniquely write x as

$$x(n) = \sum_k X(k)\varphi_k(n). \tag{2.10}$$

The set Φ is complete for the space $\ell_2(\mathbb{Z})$, i.e. all signals $x \in \ell_2(\mathbb{Z})$ can be expanded according to (2.10) and therefore, also a dual set $\{\tilde{\varphi}_k\}$ exists such that we can compute the basis expansion or signal transform coefficients X as an inner product. That is,

$$X(k) = \langle x, \tilde{\varphi}_k \rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product for $\ell_2(\mathbb{Z})$. For two real-valued sequences $x, y \in \ell_2(\mathbb{Z})$,

$$\langle x, y \rangle = \sum_n x(n)y(n).$$

Hence, if the basis functions are real-valued and the set of basis functions Φ is orthonormal, then $\tilde{\varphi}_k = \varphi_k$ and

$$\langle \varphi_k, \varphi_l \rangle = \delta(k - l).$$

The general aim of best basis search algorithms is to select, from a dictionary or library of orthonormal bases, the basis which minimizes a predefined cost measure for a given input signal. More formally stated, let $\mathcal{D} = \bigcup_{p \in \mathbb{P}} \Phi^p$ denote a dictionary of orthonormal bases for $\ell_2(\mathbb{Z})$, where each basis is given as $\Phi^p = \{\varphi_k^p\}_{k \in \mathbb{Z}}$ and $\mathbb{P}$ is a finite index set. Furthermore, let the additive cost function $\mathcal{J}(x, \Phi^p)$ for representing the input signal x in Φ^p be defined as

$$\mathcal{J}(x, \Phi^p) = \sum_{k \in \mathbb{Z}} J\Big(|\langle x, \tilde{\varphi}_k^p \rangle|^2\Big) = \sum_{k \in \mathbb{Z}} J\Big(|X^p(k)|^2\Big),$$

where J is an application-dependent cost, for example, the Lagrangian combination of rate and distortion. Any $\Phi^p \in \mathcal{D}$ that achieves the minimum cost $\mathcal{J}$ over all bases in the dictionary is called the best basis.

2.3.2 time-frequency tilings

A best basis search is only worthwhile when each basis in the given dictionary leads to a different cost for representing the input signal. That is, we assume that each basis in the dictionary $\mathcal{D}$ contains basis functions that differ from those in another basis with regard to their time and frequency responses. This difference can then be expressed in terms of the time and frequency localization of the basis functions and in terms of the resulting time-frequency decomposition. From the viewpoint of audio coding, we require adaptive signal transforms that give rise to variable time-frequency decompositions of the signal, hence we are interested in the time and frequency localization parameters of the basis functions. These parameters can be computed as follows [62, 40].

Given an orthonormal basis $\Phi = \{\varphi_k\}_{k \in \mathbb{Z}}$, the basis functions φ_k satisfy

$$\|\varphi_k\|^2 = \sum_{n=0}^{N-1} |\varphi_k(n)|^2 = 1, \quad k = 0, 1, \ldots, M-1,$$

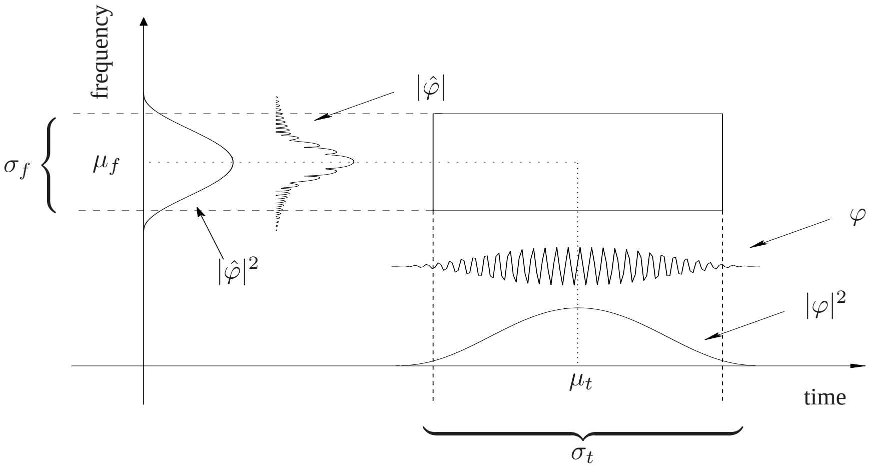


Figure 2.2: *A time-frequency tile depicts the time-frequency localization of a particular basis function. The relevant time-frequency parameters are shown.*

where N is the length of the basis functions. For any φ_k, say φ, we can interpret $|\varphi|^2$ as a probability mass function centered in time at

$$\mu_t = \sum_{n=0}^{N-1} n|\varphi(n)|^2.$$

The time localization is given as the spread around μ_t, i.e. the variance

$$\sigma_t^2 = \sum_{n=0}^{N-1} (n - \mu_t)^2 |\varphi(n)|^2.$$

By using the Parseval relation, we can equivalently define the frequency domain center μ_f and variance σ_f^2 of the Fourier transform of φ, say $\hat{\varphi}(\omega) = \sum_{n=0}^{N-1} \varphi(n) e^{-j\omega n}$. That is[2],

$$\mu_f = \frac{1}{2\pi} \int_0^{2\pi} \omega |\hat{\varphi}(\omega)|^2 \mathrm{d}\omega,$$

and

$$\sigma_f^2 = \frac{1}{2\pi} \int_0^{2\pi} (\omega - \mu_f) |\hat{\varphi}(\omega)|^2 \mathrm{d}\omega.$$

The time-frequency product of the time and frequency variances, denoted $\nu = \sigma_t^2 \sigma_f^2$, defines a so-called Heisenberg box. Computation of the set of time-frequency parameters for a variety of practical signal transforms and filter banks can be done by the methods proposed by Taswell in [61].

---

[2] It is not straightforward to uniquely describe the frequency parameters for all basis functions. We often have to use distinct formulations for the basis functions near the frequencies 0 and π.

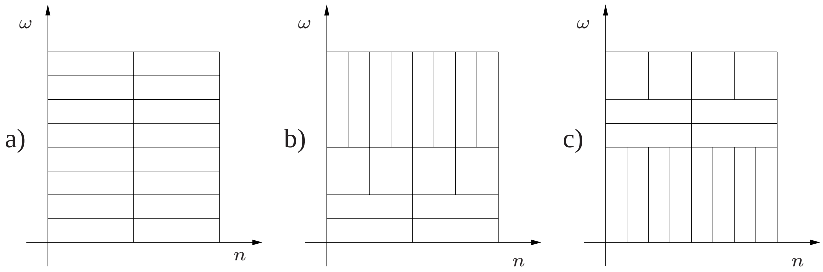


Figure 2.3: *Examples of time-frequency tilings. (a) DCT or DFT tiling. (b) Wavelet tiling. (c) Wavelet packet tiling.*

The Heisenberg box for a particular basis function can be represented by a two-dimensional time-frequency tile. This can be seen in Fig. 2.2 where for a given basis function, the corresponding tile and the relevant time-frequency parameters are shown. If we plot the time-frequency tiles for all basis function in a particular basis, we can create for each basis a time-frequency *tiling* diagram. Such a tiling diagram is an elegant visualization tool for linear expansions and signal transforms, as it depicts a decomposition of a signal both in the time and frequency domain. Tiling diagrams were first used by Gabor [21] in his original paper on time-frequency analysis. Figure 2.3 shows examples of time-frequency tilings resulting from common signal transforms.

A lower bound is imposed upon the product of time and frequency variances by Heisenberg's uncertainty principle [26, 47] as

$$\nu \geq \frac{1}{2}. \tag{2.11}$$

Eq. (2.11) implies that there are no basis functions that are arbitrarily well localized both in time and frequency. However, we can first create a dictionary of bases in which each basis contains basis functions having good time *or* frequency localization and then employ best basis algorithms to find the basis having the optimal combination of localized basis functions for the given signal. Thus, best basis search algorithms seek to construct the time-frequency tiling that is optimal for the signal x with respect to the cost measure $\mathcal{J}$.

2.3.3 time segmentation and frequency decomposition

Signal transforms are often employed on a segment-by-segment basis. That is, the time domain signals undergoing transformation are pre-segmented into consecutive signal segment. The signal transform then is separately performed on these segments, which might vary in length and might contain overlapping and windowed signal portions. Furthermore, in many situations it is convenient to regard the transform coefficients corresponding to the signal segments as a frequency domain representation of the signal portion supported within the segment. Thus, we observe that the practical

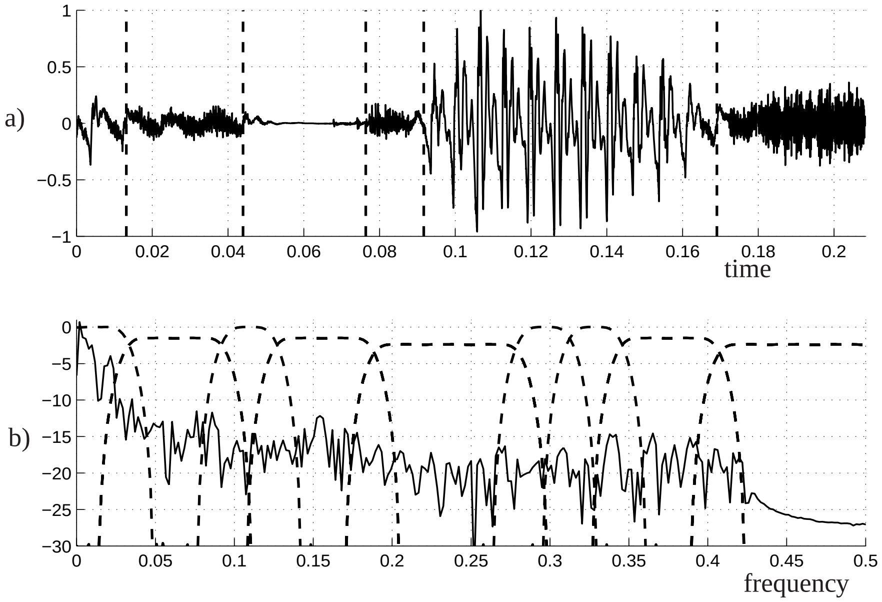


Figure 2.4: *Examples of a) time segmentation and b) frequency decomposition.*

application of signal transforms results in a segment-by-segment frequency decomposition of the underlying time domain signal. From a best basis perspective, we want to find the sequence of bases that leads to the lowest cost for representing the signal. If we expand the notion of a dictionary to the case where it can contain sequences of bases, we have the situation that both time segmentation and frequency decomposition algorithms can be described within the best basis search framework. In the case of time segmentation only, a linear expansion of a particular segment can only differ in the number basis functions compared to expansions of other segments. An example is shown in Fig. 2.4a where a time domain signal is segmented into nonuniform segments. In this example, the segment boundaries are aligned with transitions in the signal statistics. On each of the segments we can perform a frequency decomposition. Such a decomposition can lead, for example, to a basis whose basis functions have magnitude responses as depicted in Fig. 2.4b.

Both the time segmentation of a signal into segments and the subsequent frequency decomposition of the segments can be varied, which leads us to adaptive algorithms for time segmentation and frequency decomposition. While the basic concepts of these algorithms are similar, several domain-specific aspects can influence the design of the algorithms in various ways. However, the best basis search procedures employed in such algorithms are often equivalent in both domains and can therefore be combined such that jointly optimal time-frequency decomposition algorithms can be constructed.

In the following, we investigate two often used best basis search algorithms, based on tree pruning and dynamic programming, respectively. We use the term frame to denote the smallest interval in a dictionary. From the above discussion it follows that such an interval can either denote the time domain support of a basis function, or the bandwidth of the frequency response of a basis function. A segment is a combination of adjacent frames.

2.3.4 binary tree best basis search

The first best basis search algorithms were developed with the wavelet packet transform in mind [10]. This transform is a natural extension of the wavelet transform [41, 14] and provides a large dictionary of bases that are particularly useful for image coding. The wavelet packet transform can be efficiently implemented using tree-structured filter banks [41, 62]. Hence, any wavelet packet basis allows an organization of its basis functions on a binary tree. While the wavelet packet transform can be seen as a frequency decomposition, a tree-structured dictionary can also be constructed for time segmentation. Moreover, as mentioned in the previous section, it is possible to combine both tree-structured frequency decompositions and time segmentations and derive a fast best basis search method for joint time-frequency optimization.

A binary tree is organized as follows. The first node resides at either the top or the bottom and is called the root. From the root, two branches lead to lower level nodes known as children. Each child node can also function as a parent to two additional children. If a node has no children, it is considered a leaf. In the case of wavelet packets, only strongly binary trees [64] are relevant, that is, rooted trees for which the root has either zero or two branches, and all non-root branches are adjacent to either one or three branches. The vertical positions of a node relative to the root is denoted as the level or depth i of the node, whereas the horizontal position is indicated by j.

The level or depth of the node on which a wavelet packet basis function resides, corresponds to the time-frequency localization of that basis function, whereas the horizontal position of the node is an indication of its center frequency. Assuming that the root of the tree is at the bottom and has maximum depth, it is readily seen that for wavelet packets, basis functions further up the tree have better frequency localization and reduced time localization. A particular wavelet packet basis then corresponds to a particular dyadic frequency decomposition of the signal under analysis. Similarly, dyadic time segmentations can be obtained by applying a local cosine transformation, which can be organized on similar tree-structures[3], such that nodes further up the tree correspond to smaller time intervals and each local cosine basis represents a dyadic time segmentation of the signal.

A dictionary containing wavelet packet or local cosine bases can be represented as the full tree of basis function choices. An efficient algorithm for searching the dictionary

---

[3] The local cosine transform is, however, not restricted to tree-structured bases.

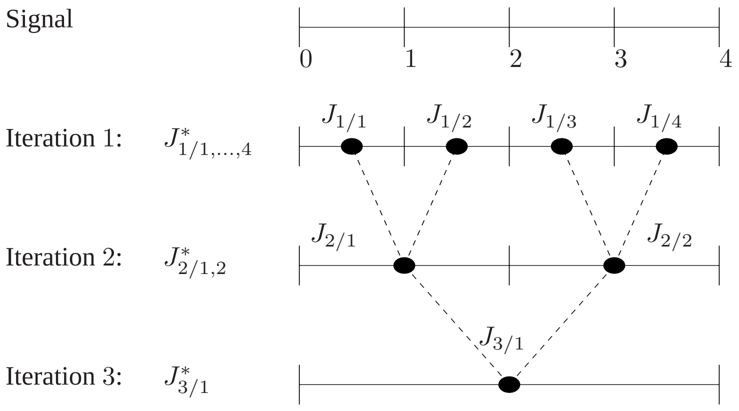


Figure 2.5: *The single tree decomposition algorithm employs tree pruning to eliminate suboptimal decompositions. At each node, the split-merge decision is made according to (2.12).*

is presented in [10]. The algorithm, denoted as the single tree (ST) algorithm in [57], involves pruning of suboptimal branches of the tree that constitutes the dictionary. The ST algorithm operates as follows. We start with a uniform division of the input signal into N frames[4] of M samples, in either the time or frequency domain. These frames are represented by the leaf nodes of the full tree. The tree associated with the dictionary is then pruned in the direction of the root by comparing the costs of each node with the accumulated costs of its two children.

Consider the example depicted in Fig. 2.5, in which a (time or frequency domain) signal is initially divided into four frames of length M, i.e. $N = 4$. Let $J_{i/j}$ be the cost for representing the jth segment of $2^{i-1}M$ samples with the jth basis function at tree level i, where the root of the tree has maximum depth $d = \log_2(N)+1$. Then, at each iteration $i = 1, \dots, d$, we evaluate the split-merge condition

$$J^*_{i/j} = \min(J_{i/j}, J^*_{i-1/2j-1} + J^*_{i-1/2j}), \tag{2.12}$$

to obtain the minimum cost, denoted $J^*_{i/j}$, for the jth segment at tree level i. This minimum cost gives rise to the best decomposition or basis, denoted $\Phi^*_{i/j}$, of the jth segment, where $j = 1, \dots, N/2^{i-1}$. After the dth iteration we obtain the minimum cost $J_{d/1}$ whereas the corresponding basis can be obtained by backtracking the sequence of split-merge decisions upwards through the tree.

For a given maximum tree depth d, the number of bases $|\mathcal{D}^d_{\text{ST}}|$ in the dictionary $\mathcal{D}^d_{\text{ST}}$ that the ST algorithm constitutes is the total number of strongly binary trees of depth at most d, which can be computed by the doubly exponential expression [1]

$$|\mathcal{D}^d_{\text{ST}}| = \lfloor c^{2^d} \rfloor,$$

---

[4] N is assumed to be a power of 2.

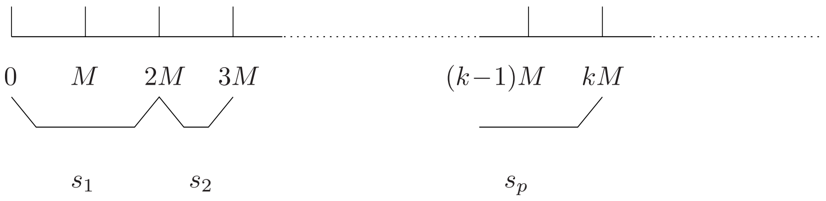


Figure 2.6: *Initial decomposition of a signal into N segments of M samples.*

where $c \approx 1.503$. The complexity of the ST decomposition algorithm for searching through the dictionary bases is $\mathcal{O}(Nd)$, that is, it depends on the initial division and the maximum tree depth that is allowed. In [29, 28], Herley *et al.* extended the ST algorithm to the double tree method, where binary tree searching algorithms in both time and frequency domains are employed and a jointly optimal time-frequency decomposition is obtained.

2.3.5 dynamic programming best basis search

The restriction of the ST algorithm to binary time segmentations and frequency decompositions was removed by Herley *et al.* in [30]. This algorithm was initially developed for time segmentation only and is often referred to as the flexible time segmentation algorithm. We derive the algorithm here for a single optimal basis or frequency decomposition, but the reader should note that an equivalent derivation can be made for a sequence of basis, corresponding to time segmentation.

The flexible decomposition algorithm employs dynamic programming [2, 5] to search for the optimal basis. It permits decomposition of *resolution* M, i.e. for an input signal x, there are N frames of M samples numbered from 0 to $N-1$ and we seek to find the optimal decomposition or basis, denoted Φ_k^*, for the subsignals $[0, kM-1]$ recursively for $k \in \{1, 2, \ldots, N\}$. The associated library $\mathcal{D}_{\text{DP}}$ is much larger than that of the ST algorithm, since $|\mathcal{D}_{\text{DP}}| = 2^{N-1}$.

Consider a signal, represented in the time or frequency domain, that is initially divided into N frames of M samples, as depicted in Fig. 2.6. Let $J_{k,l}$ denote the cost for the interval $s_{k,l} = [kM, lM-1]$, i.e. the segment that consists of frames k to l. Then, at each iteration $i = 1, \ldots, N$, the best basis Φ_i^* of the interval $[0, iM-1]$ is found by solving

$$J_i^* = \min_{0 \leq k < i} (J_k^* + J_{k,i}), \tag{2.13}$$

where J_i^* is the minimum cost for the interval $[0, iM-1]$. The minimizing argument of (2.13), say k_i^*, given by

$$k_i^* = \arg\min_{0 \leq k \leq i} (J_k^* + J_{k,i}), \tag{2.14}$$

is recorded as a split position and determines the optimal basis Φ_i^*. The algorithm terminates once J_N^* has been found and the optimal basis Φ_N^* can easily be determined

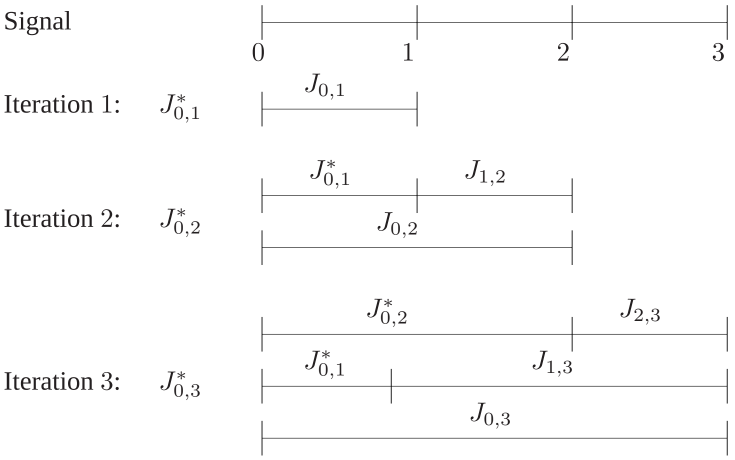


Figure 2.7: *The flexible decomposition algorithm employs dynamic programming to build up the optimal decomposition. At each iteration, the new split position is calculated according to (2.14).*

by backtracking the optimal split positions. An example of this procedure is shown in Fig. 2.7 for $N = 3$.

2.3.6 best bases and operational RD optimization

The operational RD problem as formulated in (2.9) can now be extended in a straightforward manner by including best basis search algorithms such as the single tree and dynamic programming based methods from the Sections 2.3.4 and 2.3.5 into the optimization process. This leads to RD optimal bases, i.e. for the given input signal or signal block, both the basis and the coding templates that minimize the Lagrangian cost defined in (2.2) are obtained. Let $\mathcal{D} = \bigcup_{p\in\mathbb{P}} \Phi^p$ denote a dictionary of orthonormal bases for $\ell_2(\mathbb{Z})$, where the pth basis is given as $\Phi^p = \{\varphi^p_k\}_{k\in\mathbb{Z}}$. Then we reformulate the operational RD problem from (2.9) as follows.

Formulation 4. ***Operational Rate-Distortion Problem Including Best Basis Search*** *Given the constraint R_T, find λ^* as*

$$\lambda^* = \arg\max_{\lambda\geq 0}\Big[\Big(\min_{\Phi\in\mathcal{D}}\sum_{k\in\mathbb{Z}}\big(\min_j d_k(q_{k,j}) + \lambda r_k(q_{k,j})\big)\Big) - \lambda R_T\Big]. \tag{2.15}$$

The solution to (2.15) can be found with the following step-wise procedure.

Algorithm 2. ***Operational RD Optimization***

Initialization

[1] Generate (R, D)-pairs for each possible bit allocation and for coding unit, i.e. each possible segment, transform coefficient or set of coefficients.

Optimization for a given slope λ

[2] For the given slope λ, find the minimum Lagrangian costs for each coding unit by minimizing over all coding templates.

[3] Use a best basis search algorithm to find the optimal time segmentation, frequency decomposition or joint time-frequency decomposition.

Computation of the optimal slope λ^*

[4] To find the optimal slope λ^* that corresponds to the target rate R_T, run the bisection algorithm.

Backtracking

[5] Obtain the optimal time-frequency decomposition Φ^*, the optimal allocation vector $\mathbf{q}^*$ and the corresponding coded parameters. The optimal rate R^* and distortion D^* are available or can be recomputed.

The complexity of the encoding system determines the practicality of an RD optimization technique. Three main sources of complexity can be identified.

- For obtaining the set of (R, D)-pairs from the audio source, i.e. the rate and distortion for each coding unit, several encode/decode operations have to be performed. We shall indicate this process as the *Initialization* phase. In [16, 39] examples of allocation methods that use models instead of the actual RD data can be found, to reduce the complexity. Moreover, intermediate low-complexity cost measures can be employed in certain parts of the overall RD optimization framework to reduce the complexity of the initialization phase.

- After the RD data has been found or modelled, the search for the optimal value of λ has to be performed. This part is denoted as the *Optimization* phase. The complexity depends on the delay in computing the optimal solution and the storage required by the search algorithm.

- In many situations a *Backtracking* phase is required to obtained the desired results from the output of the optimization phase. The complexity of this backtracking procedure is usually much lower that that of the previous two phases and is ignored in the rest of this thesis.

2.4 Conclusion

In this chapter we studied the theory of operational rate-distortion optimization and best basis search algorithms. We proposed several formulations of the rate-constrained allocation problem as encountered in audio coding and investigated some popular methods for solving the problem. Furthermore, we looked at two efficient best basis search algorithms which are candidates for our audio coding system. The combination of the two fields leads to interesting algorithms for audio coding purposes. However, we have not yet chosen a particular signal transform. In the introduction, we remarked that the modified discrete cosine transform is the preferred choice in many of the existing audio coders. Therefore, in the next chapter we study this signal transform in detail and investigate its possibilities for adaptive time-frequency decomposition.

Bibliography

[1] A.V. Aho and N.J.A. Sloane. Some doubly exponential sequences. *Fibonacci Quarterly*, 11:429–437, 1973.

[2] R. Bellman. *Dynamic Programming*. Princeton University Press, New Jersey, 1957.

[3] W.R. Bennett. Spectra of quantized signals. *Bell Systems Technical Journal*, 27:446–472.

[4] T. Berger. *Rate-Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[5] D.P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[6] R. Blahut. Computation of channel capacity and rate-distortion functions. *IEEE Transactions on Information Theory*, 18(4):460–473, July 1972.

[7] R. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, New York, 1965.

[8] P.A. Chou, T. Lookabaugh, and R.M. Gray. Entropy-constrained vector quantization. *IEEE Transactions On Acoustics, Speech, And Signal Processing*, 37(1):31–42, January 1989.

[9] P.A. Chou, T. Lookabaugh, and R.M. Gray. Optimal pruning with applications to tree-structured source coding and modeling. *IEEE Transactions on Information Theory*, 35(2):299–315, March 1989.

[10] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718, March 1992.

[11] R.R. Coifman, Y. Meyer, S. Quake, and M.V. Wickerhauser. Signal processing and compression with wavelet packets. In *Progress in wavelet analysis and applications*, pages 77–93. Frontières, Toulouse, France, 1993.

[12] R.R. Coifman, Y. Meyer, and M.V. Wickerhauser. Wavelet analysis and signal processing. In *Wavelets and their applications*, pages 153–178. Jones and Bartlett, Boston, USA, 1992.

[13] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley series in Telecommunication. John Wiley & Sons, New York, 1991.

[14] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, September 1990.

[15] E.W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[16] W. Ding and B. Liu. Rate control of mpeg video coding and recording by rate-quantization modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:12–20, February 1996.

[17] M. Erne and G. Moschytz. Audio coding based on rate-distortion and perceptual optimization techniques. In *Proceedings of the AES 17th International Conference: High-Quality Audio Coding*, pages 220–225, Florence, Italy, September 1999.

[18] M. Erne, G. Moschytz, and C. Faller. Best wavelet-packet bases for audio coding using perceptual and rate-distortion criteria. In *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'99)*, pages 909–912, Phoenix, USA, March 1999.

[19] H. Everett. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11:399–417, 1963.

[20] G.D. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973.

[21] D. Gabor. Theory of communication. *Journal of the Institution of Electrical Engineers*, 93.

[22] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, New York, 1981.

[23] M.M. Goodwin. *Adaptive Signal Models: Theory, Algorithms, and Audio Applications*. PhD thesis, University of California, Berkeley, USA, 1997.

[24] R.M. Gray and D.L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383, October 1998.

[25] H.J. Greenberg. *Mathematical Programming Glossary*. World Wide Web, http://www.cudenver.edu/ hgreenbe/glossary/, 1996–2005.

[26] W. Heisenberg. Ueber den anschaulichen inhalt der quantentheoretischen kinematik and mechanik. *Zeitschrift fur Physik*, 43:172–198, March 1927.

[27] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli. Arbitrary orthogonal tilings of the time-frequency plane. In *Proc. IEEE-SP Conference on Time-Frequency and Time-Scale Analysis*, pages 11–14, Victoria, Canada, October 1992.

[28] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli. Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms. *IEEE Transactions on Signal Processing*, 41(12):3341–3359, December 1993.

[29] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli. Time-varying orthonormal tilings of the time-frequency plane. In *Proceedings of the 1993 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'93)*, pages 205–208, Minneapolis, USA, April 1993.

[30] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard. Flexible time segmentations for time-varying wavelet packets. In *Proc. IEEE-SP Conference on Time-Frequency and Time-Scale Analysis*, pages 9–12, Philadelphia, USA, October 1994.

[31] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard. Flexible tree-structured signal expansions using time-varying wavelet packets. *IEEE Transactions on Signal Processing*, 45(2):333–345, February 1997.

[32] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard. Joint space-frequency segmentation using balanced wavelet packets trees for least-cost image representation. *IEEE Transactions on Image Processing*, 6(9):1213–1230, September 1997.

[33] R. Heusdens, J. Jensen, W.B. Kleijn, V. Kot, O.A. Niamut, S. van de Par, N.H. van Schijndel, and R. Vafin. Bit-rate scalable intraframe sinusoidal audio coding based on rate-distortion optimisation. *Journal of the Audio Engineering Society*, 54(3):167–188, March 2006.

[34] R. Heusdens and S. van de Par. Rate-distortion optimal sinusoidal modeling of audio and speech using psychoacoustical matching pursuits. In *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'02)*, pages 1809–1812, Orlando, USA, May 2002.

[35] Y. Huang, I. Pollak, C.A. Bouman, and M.N. Do. New algorithms for best local cosine basis search. In *Proceedings of the 2004 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'04)*, pages 773–776, Montreal, Canada, May 2004.

[36] Y. Huang, I. Pollak, C.A. Bouman, and M.N. Do. Time-frequency analysis with best local cosine bases. In *Proceedings of the 16th Annual Symposium on Electronic Imaging (IS&T/SPIE)*, San Jose, USA, January 2004.

[37] N.S. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice-Hall, Englewood Cliffs, NJ, 1984.

[38] H. Krim, D. Tucker, S. Mallat, and D. Donoho. On denoising and best signal representation. *IEEE Transactions on Information Theory*, 45(7):2225–2238, November 1999.

[39] L-J. Lin and A. Ortega. Bit-rate control using piecewise approximated rate-distortion characteristics. *IEEE Transactions on Circuits and Systems for Video Technology*, 8:446–459, August 1998.

[40] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, New York, 1998.

[41] S.G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

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

[43] P. Moulin. Signal estimation using adapted tree-structured bases and the MDL principle. In *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pages 141–143, Philadelphia, USA, June 1996.

[44] S. Na and D.L. Neuhoff. Bennett's integral for vector quantizers. *IEEE Transactions on Information Theory*, 41(4):886–900, July 1995.

[45] A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, 15(6):23–50, November 1998.

[46] A. Ortega, K. Ramchandran, and M. Vetterli. Optimal buffer-constrained source quantization and fast approximations. In *Proceedings of the 1992 IEEE International Symposium on Circuits and Systems (ISCAS'92)*, pages 192–195, San Diego, USA, May 1992.

[47] A. Papoulis. *The Fourier Integral and its Applications*. McGraw-Hill, New York, 1962.

[48] J.-C. Pesquet, H. Krim, D. Leporini, and E. Hamman. Bayesian approach to best basis selection. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'96)*, pages 2634–2637, Atlanta, USA, May 1996.

[49] P. Prandoni. *Optimal Segmentation Techniques for Piecewise Stationary Signals*. PhD thesis, cole Polytechnique Fdrale de Lausanne, March 1999.

[50] P. Prandoni, M. Goodwin, and M. Vetterli. Optimal time segmentation for signal modeling and compression. In *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, pages 2029–2032, Munich, Germany, April 1997.

[51] P. Prandoni and M. Vetterli. Optimal bit allocation with side information. In *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'99)*, pages 2411–2414, Phoenix, USA, March 1999.

[52] P. Prandoni and M. Vetterli. R/D optimal linear prediction. *IEEE Transactions on Speech and Audio Processing*, 8(6):646–655, November 2000.

[53] M. Purat and P. Noll. Audio coding with a dynamic wavelet packet decomposition based on frequency-varying modulated lapped transforms. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'96)*, pages 1021–1024, Atlanta, USA, May 1996.

[54] K. Ramchandran, A. Ortega, and M. Vetterli. Bit allocation for dependent quantization with applications to MPEG video coders. In *Proceedings of the 1993 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'93)*, pages 381–384, Minneapolis, USA, April 1993.

[55] K. Ramchandran, A. Ortega, and M. Vetterli. Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders. *IEEE Transactions on Image Processing*, 3(5):533–545, September 1994.

[56] K. Ramchandran, A. Ortega, and M. Vetterli. Optimal trellis-based buffered compression and fast approximations. *IEEE Transactions on Image Processing*, 3(1):26–40, January 1994.

[57] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions on Image Processing*, 2(2):160–175, April 1993.

[58] K.R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, Boston, 1990.

[59] C.E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423.

[60] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Transactions On Acoustics, Speech, And Signal Processing*, 36(9):1445–1453, September 1988.

[61] C. Taswell. *Wavelets in Signal and Image Analysis: From Theory To Practice*, chapter Empirical Tests for Evaluation of Multirate Filter Bank Parameters, pages 111–140. Kluwer Academic Publishers, 2001.

[62] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Signal Processing Series. Prentice-Hall, 1995.

[63] L.F. Villemoes. Adapted bases of time-frequency local cosines. Technical report, KTH Royal Institute of Technology, Stockholm, Sweden, 1999.

[64] Eric W. Weisstein. Strongly binary tree. From MathWorld–A Wolfram Web Resource http://mathworld.wolfram.com/StronglyBinaryTree.html.

[65] Y. Yoo, A. Ortega, and K. Ramchandran. A novel hybrid technique for discrete rate-distortion optimization with applications to fast codebook search for SVQ. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'96)*, pages 2040–2043, Atlanta, USA, May 1996.

Chapter 3

The Modified Discrete Cosine Transform

In this chapter we review the modified discrete cosine transform. After a short introduction in section 3.1, we study important properties of the transform in section 3.2. We then investigate three techniques for obtaining adaptive time-frequency signal decompositions with the MDCT in section 3.3 and conclude the chapter in section 3.4.

3.1 Introduction

The modified discrete cosine transform (MDCT) was first encountered in the work by Princen, Johnson and Bradley [47] as the oddly stacked filter bank based on time domain aliasing cancellation [1] (TDAC) [46]. The MDCT is a so-called lapped transform, i.e. a transform where samples from consecutive overlapping segments are windowed and transformed. In the case of the MDCT the overlap is 50%, that is, only adjacent blocks are considered for overlap. These lapped transforms, also known as lapped orthogonal transforms (LOTs), were extensively investigated by Malvar [31, 35] and can be seen as both [29] a signal transform and a multirate filter bank. Malvar proposed a specific variant of the MDCT, known as the modulated lapped transform (MLT), in [31]. As an overlapped block transform, the MDCT significantly reduces so-called blocking artifacts. These artifacts are typical for signal coding schemes that employ nonoverlapping block transformations such as the DFT [45, 3] or DCT [49]. On the other hand, the MDCT can be seen as a particular instance from the family of cosine-modulated filter banks (CMFB) [32, 23, 24, 57] and as such, it is critically sampled and possesses the perfect reconstruction property. In this chapter we mostly adhere to a signal transform viewpoint of the MDCT. Hence, time-domain descriptions rather than frequency or $\mathcal{Z}$-domain descriptions will be employed in most of the derivations and examples.

---

[1] These terms, oddly-stacked and TDAC, are explained later in the chapter.

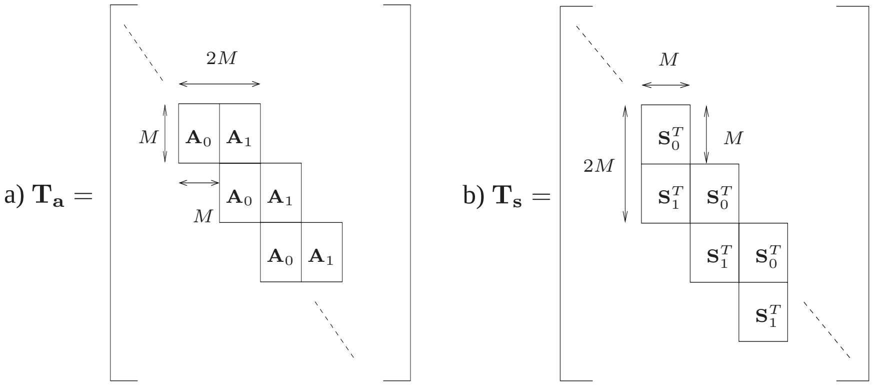


Figure 3.1: *(a) Analysis mapping $\mathbf{T_a}$ and (b) synthesis mapping $\mathbf{T_s}$.*

3.2 Properties of the MDCT

In this section we discuss the MDCT and its underlying properties. We first consider the general framework of lapped transforms and then move on to the intricacies of time-domain aliasing and perfect reconstruction. Next, some popular MDCT window designs are discussed, as well as a fast implementation algorithm. Then, some examples of the MDCT behavior are provided. The section ends with an overview of relevant transforms related to the MDCT.

3.2.1 lapped orthogonal transforms

In transform coding with lapped orthogonal transforms (LOTs), a time-domain input signal x is transformed to an output set of transform coefficients X by the analysis mapping $\mathbf{T_a}$, as $X = \mathbf{T_a}x$. The analysis mapping has an upper-triangular block-banded Toeplitz structure, as depicted in Fig. 3.1a. The synthesis mapping corresponding to the inverse transform $\hat{x} = \mathbf{T_s}X$, denoted by $\mathbf{T_s}$, is given in Fig. 3.1b.

Let the square submatrices $\mathbf{S}_\ell \in \mathbb{R}^{M\times M}$ and $\mathbf{A}_\ell \in \mathbb{R}^{M\times M}, \ell \in \{0,1\}$, contain M samples of a set of length-$2M$ synthesis and time-reversed analysis filters, f_k and h_k, respectively. That is,

$$\mathbf{A}_\ell(k,n) = h_k(2M-1-n-\ell M), \tag{3.1}$$

$$\mathbf{S}_\ell(k,n) = f_k(n+\ell M). \tag{3.2}$$

The synthesis filters f_k constitute the basis functions of the LOT. The index of the basis functions, or equivalently, a specific filter bank subband channel, is denoted $k = 0, 1, \ldots, M-1$, whereas the index $n = 0, 1, \ldots, 2M-1$ denotes the filter coefficients.

In the absence of quantization of the transform coefficients X, perfect reconstruction (PR) of x, i.e. $\hat{x} = x$, is obtained if $\mathbf{T_s}\mathbf{T_a} = \mathbf{I}$, or, equivalently, $\mathbf{T_s} = \mathbf{T_a^{-1}}$. Often, the more strict requirement $\mathbf{T_s} = \mathbf{T_a^*}$ is applied, which leads to the analysis filters being the time-reversed synthesis filters, i.e. $h_k(n) = f_k(2M-1-n)$. However, as is done in [46, 47] we do not assume this relation *a priori* in the following derivations. Using (3.1) and (3.2), the LOT PR condition $\mathbf{T_s} = \mathbf{T_a^{-1}}$ then reduces to the following set of conditions,

$$\mathbf{S_0}^T\mathbf{A_0} + \mathbf{S_1}^T\mathbf{A_1} = \mathbf{I}, \tag{3.3}$$

$$\mathbf{S_0}^T\mathbf{A_1} = \mathbf{S_1}^T\mathbf{A_0} = \mathbf{0}. \tag{3.4}$$

The MDCT is a lapped orthogonal transform where all basis functions are derived from cosine modulation of a lowpass synthesis FIR prototype filter f. It is customary to indicate f as the MDCT synthesis window. The M MDCT basis functions φ_k are then given as

$$\varphi_k(n) = f_k(n) = \gamma f(n)\cos\Big[\frac{\pi}{M}\Big(k+\frac{1}{2}\Big)\Big(n+n_0\Big)\Big], \quad k = 0, 1, \ldots, M-1,$$

where n_0 is a time shift and γ is a normalization factor, for which various choices are encountered throughout the MDCT and LOT literature. We will use $\gamma = \sqrt{2/M}$. Note that in general, even if the synthesis window f has linear phase, the MDCT basis functions do not have linear phase. Therefore, the MDCT is less suitable for image and video coding applications.

If we regard the MDCT basis functions as a set of bandpass filters, the center frequencies ω_k of these filters and the corresponding subband channels are given as

$$\omega_k = \frac{\pi}{M}\Big(k+\frac{1}{2}\Big), \quad k = 0, 1, \ldots, M-1.$$

The MDCT subband channels are offset by half a frequency bin in comparison to the DFT. Hence, an MDCT analysis splits the frequency axis between 0 and 2π into M equally spaced subband channels, each of width $2\pi/M$. This type of channel stacking in the MDCT is commonly referred to as odd stacking and the MDCT is an oddly stacked filter bank.

Let the ith time-domain input signal block x_i of length M be given as $x_i(n) = x(n + iM), n = 0, 1, \ldots, M-1$, and let the ith time-domain input signal block y_i of length $2M$ be given as $y_i(n) = x(n+iM), n = 0, 1, \ldots, 2M-1$. The M transform coefficients X_i are obtained by windowing y_i with analysis window h and by pre-multiplication with the cosine modulation matrix $\mathbf{C} \in \mathbb{R}^{M\times 2M}$, where

$$\mathbf{C}(k,n) = \sqrt{\frac{2}{M}}\cos\Big[\frac{\pi}{M}\Big(k+\frac{1}{2}\Big)\Big(n+n_0\Big)\Big],$$

$k = 0, 1, \ldots, M-1$ and $n = 0, 1, \ldots, 2M-1$. Figure 3.2 shows the forward MDCT.

Furthermore, let the ith reconstructed time-domain output signal block $\hat{x}_i$ be given as $\hat{x}_i = \hat{x}(n+iM), n = 0, 1, \ldots, M-1$ and, similarly, let the output signal block $\hat{y}_i =$

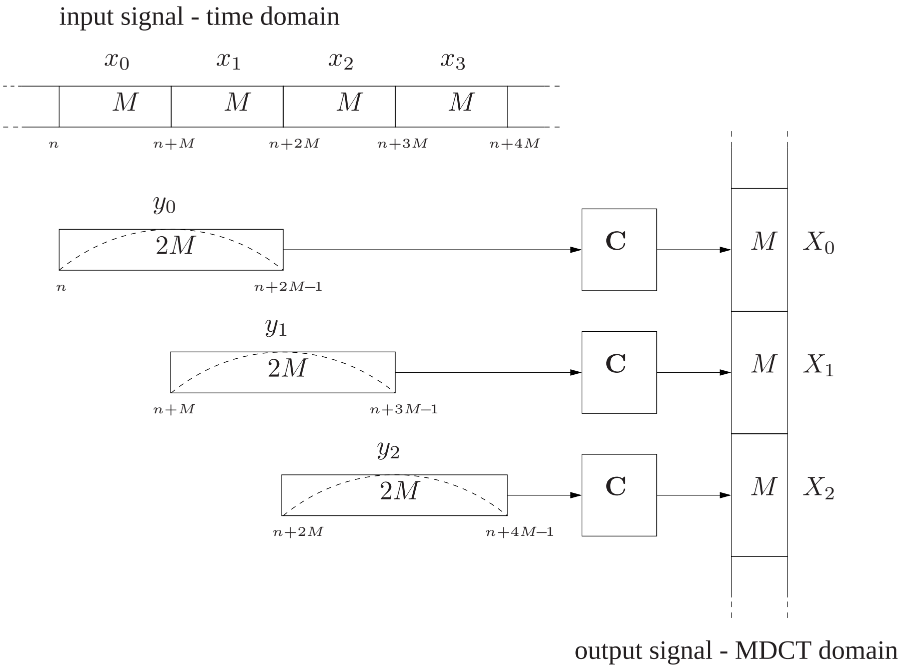


Figure 3.2: *Forward MDCT operation. The transposed input blocks are windowed with analysis window* h *and pre-multiplied with cosine modulation matrix* $\mathbf{C}$*.*

$\hat{x}(n+iM), n = 0, 1, \ldots, 2M-1$. Fig. 3.3 visualizes the inverse MDCT operation. Pre-multiplication with matrix $\mathbf{C}^T$ is applied to the transform coefficients X_i, leading to the reconstructed block $\hat{y}_i$. The ith signal block $\hat{x}_i$ is reconstructed after windowing with synthesis window f and overlap-add of the blocks $\hat{y}_{i-1}$ and $\hat{y}_i$ as

$$\hat{x}_i(n) = f(n{+}M)\hat{y}_{i-1}(n{+}M) + f(n)\hat{y}_i(n). \tag{3.5}$$

In terms of the basis expansion theory discussed in the previous chapter, section 2.3.1, an expansion of the signal block $\hat{y}_i$ is given by

$$\hat{y}_i(n) = \sum_{k=0}^{M-1} X_i(k)\varphi_k(n).$$

The M transform coefficients X_i are computed as

$$X_i(k) = \sum_{n=0}^{2M-1} y_i(n)\tilde{\varphi}_k(2M-1-n),$$

with

$$\tilde{\varphi}_k(n) = h_k(n) = \sqrt{\frac{2}{M}} h(n) \cos\Big[\frac{\pi}{M}\Big(k+\frac{1}{2}\Big)\Big(n+n_0\Big)\Big],$$

and h a lowpass analysis FIR prototype filter, indicated as the MDCT analysis window.

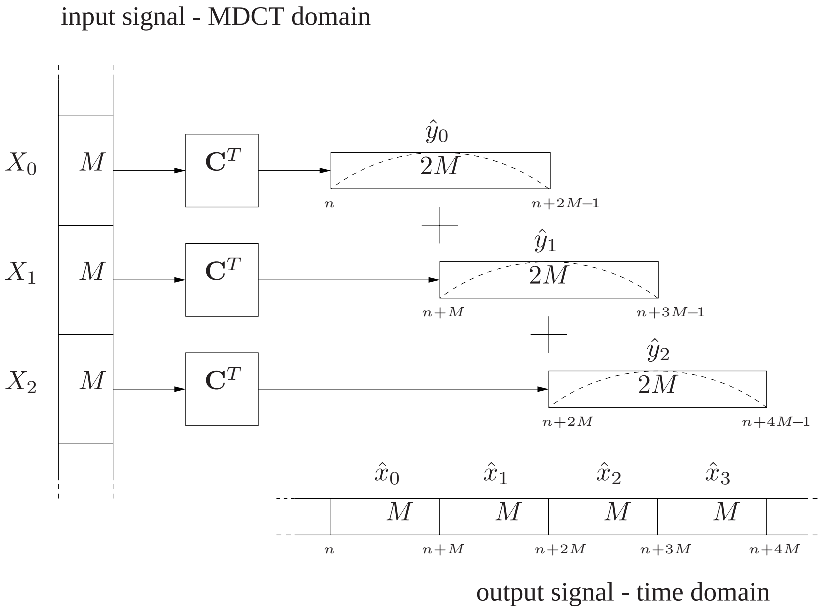


Figure 3.3: *Inverse MDCT operation. The MDCT coefficients are pre-multiplied with cosine modulation matrix* $\mathbf{C}^T$ *and subsequently windowed with synthesis window* f.

3.2.2 TDAC and perfect reconstruction

Paradoxically, the MDCT has the property that, in general, $\hat{y}_i \neq y_i$, even in the absence of quantization. Hence, considered on a block basis the MDCT is a nonorthogonal transform. Only after overlap with the preceding reconstructed signal block $\hat{y}_{i-1}$ the MDCT can satisfy conditions (3.3) and (3.4) such that $\hat{x}_i = x_i \; \forall i \in \mathbb{N}$. Perfect reconstruction of the signal x depends on the design of the analysis and synthesis windows h and f, and on the choice for the time shift n_0. We follow the method in [46, 47] in order to derive the appropriate window designs and the time shift.

Theorem 2. ***(perfect reconstruction property of the MDCT)*** *For the* i*th input signal block* y_i *and MDCT analysis window* h*, let the* $2M$ *MDCT analysis signals* X_i *be defined as*

$$X_i(k) = \sqrt{\frac{2}{M}} \sum_{n=0}^{2M-1} h(2M-1-n) y_i(n) \cos\Big[\frac{\pi}{M}\Big(k+\frac{1}{2}\Big)\Big(n+n_0\Big)\Big].$$

Furthermore, let $\hat{y}_i$ *be obtained from* X_i *as*

$$\hat{y}_i(n) = \sqrt{\frac{2}{M}} \sum_{k=0}^{M-1} X_i(k) \cos\Big[\frac{\pi}{M}\Big(k+\frac{1}{2}\Big)\Big(n+n_0\Big)\Big],$$

and let f *be the MDCT synthesis window.*

The ith signal block $\hat{x}_i$ can be perfectly reconstructed if

$$
\begin{aligned}
f(n) &= h(n), && n = 0,\dots,2M-1, && (3.6)\\
h(n) &= h(2M-1-n), && n = 0,\dots,M-1, && (3.7)\\
h^2(n) + h^2(n+M) &= 1, && n = 0,\dots,M-1, && (3.8)
\end{aligned}
$$

and $2n_0 = M+1$.

Proof: The $2M$ MDCT coefficients are not independent but satisfy

$$X(k) = -X(2M-k-1), \quad k = 0,1,\dots,M-1.$$

Therefore, only M MDCT coefficients are required for reconstruction. The ith signal block $\hat{x}_i$ is reconstructed as (3.5)

$$\hat{x}_i(n) = f(n+M)\hat{y}_{i-1}(n+M) + f(n)\hat{y}_i(n).$$

In Appendix A we show that for $2n_0 = M+1$, (3.5) is equivalent to

$$
\begin{aligned}
\hat{x}_i(n) &= x_i(n)\Big(f(n+M)h(M-1-n) + f(n)h(2M-1-n)\Big)\\
&+ x_{i-1}(2M-1-n)\Big(f(n+M)h(n) - f(n)h(n+M)\Big). \qquad (3.9)
\end{aligned}
$$

Hence, $\hat{x}_i = x_i \ \forall i \in \mathbb{N}$ if (3.6)-(3.8) are satisfied. □

We can interpret (3.9) as follows. The first term in (3.9) is the desired signal x_i, while the second term is a time alias originating from the previous signal block x_{i-1}. In order to achieve PR, this time alias has to be removed from the output. The most popular choice to achieve time domain aliasing cancellation (TDAC) is choosing $f = h$, i.e., setting the synthesis window equal to the analysis window according to (3.6). Once the aliased term is removed, it is clear that PR is obtained by designing the MDCT analysis window h such that (3.7) and (3.8) are satisfied. Note that other choices for f and h are also possible, as shortly discussed in section 3.2.3.

We now return to the LOT PR conditions (3.3) and (3.4) to see that the derived conditions on the analysis and synthesis windows lead to overall PR. Let $\mathbf{H}_\ell \in \mathbb{R}^{M\times M}$ and $\mathbf{C}_\ell \in \mathbb{R}^{M\times M}$ be defined as

$$\mathbf{H}_\ell = \mathrm{diag}\{h(n+\ell M)\},$$

and

$$\mathbf{C}_\ell(k,n) = \sqrt{\frac{2}{M}}\cos\Big[\frac{\pi}{M}\Big(k+\frac{1}{2}\Big)\Big(n+\ell M+\frac{M+1}{2}\Big)\Big],$$

respectively, for $k,n = 0,1,\dots,M-1$. By filling in the choices for h, f and n_0 in the original conditions (3.3) and (3.4) we obtain

$$
\begin{aligned}
\mathbf{H_0}\mathbf{C_0}^T\mathbf{C_0}\mathbf{H_0} + \mathbf{H_1}\mathbf{C_1}^T\mathbf{C_1}\mathbf{H_1} &= \mathbf{I},\\
\mathbf{H_0}\mathbf{C_0}^T\mathbf{C_1}\mathbf{H_1} = \mathbf{H_1}\mathbf{C_1}^T\mathbf{C_0}\mathbf{H_0} &= \mathbf{0}.
\end{aligned}
$$

Since it can be derived that

$$\begin{aligned}\mathbf{C_0}^T\mathbf{C_0} &= \mathbf{I} - \mathbf{J},\\ \mathbf{C_1}^T\mathbf{C_1} &= \mathbf{I} + \mathbf{J},\\ \mathbf{C_0}^T\mathbf{C_1} &= \mathbf{C_1}^T\mathbf{C_0} = \mathbf{0},\end{aligned}$$

we have the single PR condition $\mathbf{H_0}^2 + \mathbf{H_1}^2 = \mathbf{I}$ which is clearly satisfied if $h^2(n) + h^2(n+M) = 1$. Note that $\mathbf{J} \in \mathbb{N}^{M\times M}$ denotes the matrix with 1's along the main anti-diagonal and 0's elsewhere.

Given the conditions (3.6)-(3.8) for h, f and n_0 the definition of the MDCT is

$$X_i(k) = \sqrt{\frac{2}{M}} \sum_{n=0}^{2M-1} h(n)x_i(n) \cos\Big[\frac{\pi}{M}\Big(k + \frac{1}{2}\Big)\Big(n + \frac{M+1}{2}\Big)\Big]. \tag{3.10}$$

The inverse MDCT is then given by

$$y_i(n) = \sqrt{\frac{2}{M}} \sum_{k=0}^{M-1} X_i(k) \cos\Big[\frac{\pi}{M}\Big(k + \frac{1}{2}\Big)\Big(n + \frac{M+1}{2}\Big)\Big], \tag{3.11}$$

and the overlap-add operation as

$$\hat{x}_i(n) = h(n+M)\hat{y}_{i-1}(n+M) + h(n)\hat{y}_i(n).$$

3.2.3 window design

In the previous section we derived the constraints (3.6)-(3.8) that have to be taken into account when designing MDCT windows. This set of constraints is also known as the Princen-Bradley conditions. According to (3.6) only the analysis window h has to be designed. Constraint (3.7) can be easily satisfied by considering symmetric analysis windows only, whereas constraint (3.8) states that the window tails or halves of the MDCT analysis window must be power complementary. Additional constraints on the window design come into play when considering the MDCT in a transform coding framework. The most relevant of these constraints are:

[1] *Frequency selectivity:* The main lobe or passband of the magnitude response of the window should be as narrow as possible and stopband leakage must be minimized.

[2] *Monotony:* The envelope of the stopband attenuation should be monotonically decreasing (or at least be equiripple) to limit the spread of quantization noise over frequency regions outside the passband.

[3] *Time selectivity:* In some cases, e.g., when coding transient signals, it is desirable to constrain the quantization noise in time rather than in frequency. This requires windows having a small overlap. Another reason to reduce the overlap is to minimize the time aliasing that is introduced. This is especially beneficial when using adaptive time-frequency techniques such as temporal noise shaping, see section 3.3.2.

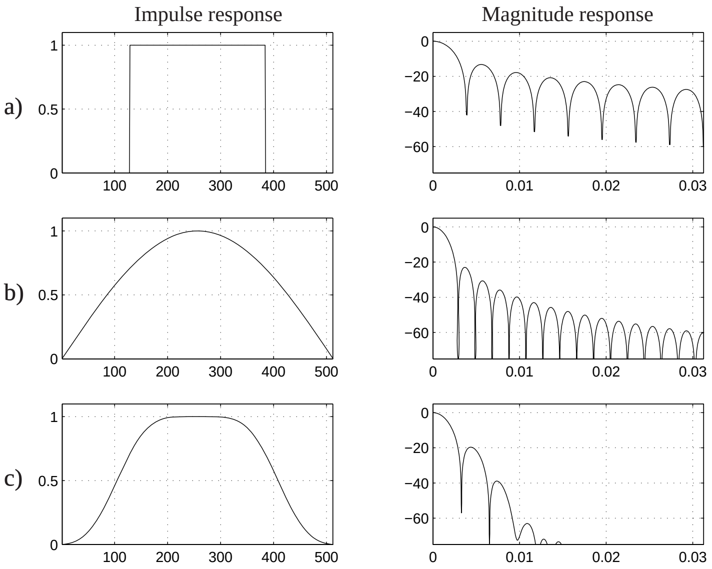


Figure 3.4: *Impulse and magnitude responses of a) trapezoidal window, b) sine window and c) KBD window.*

A variety of MDCT window designs have been explored in literature, where a trade-off is offered between the time and frequency behavior of the windows, depending on the constraints imposed by the coding framework in which the MDCT operates.

Trapezoidal windows

The trapezoidal window h_T corresponds to the minimum overlap window defined as

$$h_T(n) = \begin{cases} 0, & 0 \leq n \leq \frac{M}{2} - 1, \\ 1, & \frac{M}{2} \leq n \leq \frac{3M}{2} - 1, \\ 0, & \frac{3M}{2} \leq n \leq 2M - 1. \end{cases} \tag{3.12}$$

This window turns the MDCT into an nonoverlapping block transform, i.e. the DCT-IV. From Figure 3.4a we observe that the trapezoidal window has high time selectivity, but very poor frequency selectivity.

Sine window

Good frequency domain behavior can be obtained with a simple design based on the sine function. This window h_S is given as

$$h_S(n) = \sin\Big[\Big(n + \frac{1}{2}\Big)\Big(\frac{\pi}{2M}\Big)\Big], \qquad n = 0, \ldots, 2M-1.$$

It offers good pass-band selectivity (see Figure 3.4b), i.e., a narrow mainlobe, for selection and discrimination of tonal components spaced close to one another. Moreover, the sine window satisfies the polyphase normalization condition,

$$\sum_{k=1}^{M-1} p_k(n) = 0 \quad n = 0, 1, \ldots, M-1,$$

such that the energy of a DC (constant) signal is concentrated into the first basis function. Malvar proposed the MLT [35] as a particular instance of the MDCT, namely the MDCT where the sine window is applied. On the other hand, the time selectivity is rather low.

Kaiser-Bessel derived window

A general method to construct a suitable MDCT window h of length $2M$ from any symmetric window w of length $M + 1$ is the following:

$$h(n) = \begin{cases} \sqrt{\dfrac{\sum_{l=0}^{n} w(l)}{\sum_{l=0}^{M} w(l)}}, & 0 \leq \quad n \quad \leq M-1, \\ \\ \sqrt{\dfrac{\sum_{l=n-M+1}^{M} w(l)}{\sum_{l=0}^{M} w(l)}}, & M \leq \quad n \quad \leq 2M-1. \end{cases} \tag{3.13}$$

An example of this method is the Kaiser-Bessel derived (KBD) [65] window h_K, which leads to a good trade-off between the requirements in both the time and frequency domains. Employing (3.13), it is derived from the $M + 1$-point Kaiser-Bessel window [21] w_K, computed as

$$w_K(n) = \frac{I_0(\pi\alpha\sqrt{1-(2n/2M-1)^2})}{I_0(\pi\alpha)},$$

where $I_0(\cdot)$ is the 0th order modified Bessel function of the first kind [64], given as the series

$$I_0(n) = \sum_{l=0}^{\infty} \left(\frac{(x/2)^k}{k!} \right)^2 .$$

The parameter $\alpha \in \mathbb{R}$ determines the shape of the window, where larger values of α lead to a more narrow window. In the frequency domain, increasing values of α lead to a broader main lobe and increased stopband reduction. In Figure 3.4c the impulse and magnitude responses of a KBD window designed with $\alpha = 4$ are shown. This KBD window has high stopband attenuation and therefore compacts more energy into a single spectral component than the sine window. This is beneficial for signals with a few strong spectral components, spaced relatively far form each other. Furthermore, the window overlap is rather low, so it has good time selectivity. However, the main lobe is broader and the magnitude of the first sidelobe is higher than the sine window. Therefore, the sine window is a better choice for coding signals which contain closely-space harmonics that need to be resolved.

Other window designs

An extensive overview of MDCT window design is given by Ferreira in [14]. Additionally, an optimization method for MDCT window design is presented where, similarly to the KBD window, a tradeoff between the reduction of time aliasing and the stopband reduction can be made. In [53] Ferreira and Sinha provide a detailed comparison of some popular MDCT window designs. Moreover, in some cases, it can be advantageous to have different analysis and synthesis windows, i.e., to relax constraint (3.6). The analysis windows can then be optimized for an increase in stopband reduction, whereas the synthesis windows might be designed to be more smooth, in order to further reduce blocking artifacts. Perfect reconstruction and time-domain aliasing can still be achieved by making the MDCT a biorthogonal transform. The resulting generalized Prince-Bradley conditions are derived in [54, 6], examples of biorthogonal window designs can be found in, e.g., [40, 36].

3.2.4 fast MDCT implementation

When implemented according to (3.10), the complexity of the MDCT is $\mathcal{O}(M^2)$. One of the earliest fast implementations of the MDCT is presented by Duhamel *et al* in [11], where they show that the MDCT can be dissected as follows. Let $g = hx$ denote the windowed input block and let $z \in \mathbb{C}^{M/2}$ be given as

$$z(n) = \Big(g(2n) - g(M-1-2n)\Big) + j\Big(g(2M-1-n) + g(M+2n)\Big),\ 0 \leq n \leq M/2-1.$$

We can now compute $Z \in \mathbb{C}^{M/2}$ for $k = 0, \ldots, M-1$, as

$$\begin{aligned} Z(k) &= e^{j\pi(4Mk+4k+3M)/4M} \sum_{n=0}^{M/2-1} \Big[z(n)e^{j\pi(4n+1)/4M}\Big] e^{j4\pi kn/M}, \\ &= Z_{\text{post}}(k) \sum_{n=0}^{M/2-1} \Big[z_{\text{pre}}(n)z(n)\Big] e^{j4\pi kn/M}, \end{aligned} \tag{3.14}$$

where

$$\begin{aligned} Z_{\text{post}}(k) &= e^{j\pi(4Mk+4k+3M)/4M}, \\ z_{\text{pre}}(n) &= e^{j\pi(4n+1)/4M}. \end{aligned}$$

Disregarding normalization issues, the M MDCT coefficients X can be obtained from Z as follows.

$$\begin{aligned} X(2k) &= \operatorname{Im}\{Z(k)\}, \\ X(2k+1) &= -\operatorname{Re}\{Z(\frac{M}{2}-k-1)\}. \end{aligned}$$

Eq.(3.14) can be interpreted as follows. First, the complex time domain samples z are multiplied by the pre-twiddle factor z_{pre}, such that $z' = z_{\text{pre}}z$. Then, an $M/2$-points complex-valued IDFT operation is applied to z'. The resulting frequency domain coefficients Z' are multiplied with the post-twiddle factor Z_{post}, resulting in $Z = Z_{\text{post}} Z'$.

The pre -and post-twiddle operations have a complexity $\mathcal{O}(M)$, whereas the IDFT can be computed with an IFFT of complexity $\mathcal{O}(M \log M)$, see [7]. A similar derivation can be made for the inverse MDCT as defined in (3.11), which can be computed with an $M/2$-points complex-valued FFT. The computational complexity of the MDCT in this implementation is therefore dominated by that of an $M/2$-points FFT as $\mathcal{O}(M/2 \log[M/2])$.

It is this implementation of the MDCT that has been employed throughout all the experiments presented in Part II. Additional reductions in the number of required additions and multiplications and/or the number of memory locations can be obtained by merging the windowing operation into the pre-twiddle operation, e.g. [11, 51] or by considering efficient structures for the type-IV DCT as in [25, 38]. For a specific choice of M, as is the case for the MDCT employed in the MPEG1-layer3 standard [41], further optimizations can be made [4].

3.2.5 MDCT examples

We have seen that the MDCT leads to perfect reconstruction of the input signal after overlap-add. However, it was also observed that, in general, a single input signal block cannot be reconstructed perfectly. In a transform coding scheme where the MDCT is employed, quantization and coding decisions are typically made on block-by-block basis. This also holds for most of the algorithms presented in Part II of this thesis, where we assume independent coding of the coding units. Therefore, it is desirable to obtain further insights on the behavior of the MDCT on a block level.
Consider the example given in Fig. 3.5a, where a sinusoidal input signal divided into four overlapping block of 64 samples is shown. Each windowed input block is transformed with the MDCT, resulting in the four MDCT spectra in Fig. 3.5b. The reconstructed signal is given in Fig. 3.5c. This example emphasizes the time-varying nature of the MDCT spectrum, i.e. the MDCT spectrum is sensitive to phase shifts of the input signal.

Psycho-acoustically motivated masking curves often rely on DFT-based computation. As a result, when the MDCT is applied in combination with DFT-based psychoacoustic models, a mismatch between the DFT and MDCT spectra occurs, which can lead to coding artifacts such as time-varying bandwidth limitation artifacts. In the work by Daudet and Sandler [8, 9], a detailed analysis of this phenomenon for sinusoidal input is made and a spectrum regularization technique is proposed.

In Fig. 3.6, the four individual windowed input blocks are plotted, along with the reconstructed windowed output blocks. Note the varying amounts of time-aliasing between the first and second block, or between the third and fourth block, which lead to time-varying energy content in the reconstructed signal block. This can also be ob-

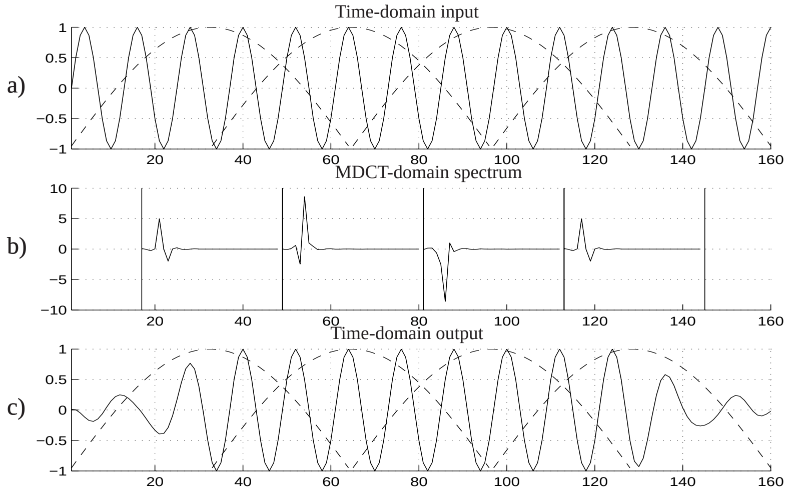


Figure 3.5: *Example of the overlap-add MDCT operation and the corresponding MDCT spectra. a) The input signal along with the windows (dashed lines). b) The resulting MDCT spectra c) The reconstructed time domain signal after overlap-add.*

served from the MDCT spectra in Fig. 3.5b. A more extensive study of MDCT spectra, reconstructed output signals can be found in the work of Wang and Vilermo [63, 60].

3.2.6 related transforms

The MDCT is closely related to a number of other signal transforms. In some coding applications, these transforms can provide increased coding performance compared to the MDCT. Moreover, in other signal processing fields such as audio and speech enhancement, such transforms can serve as a replacement for the DFT. Therefore, these transforms are briefly discussed, where we concentrate on recent transforms that can be seen as an extension of the MDCT. The relation with block transforms such as the DFT and the DCT, is explored in the articles by Yang, Yaroslavsky and Vilermo [62, 61], by making use of the shifted Fourier transforms (SDFT) [66].

Extended Lapped Transform

In some applications, a larger overlap region and thus longer windows can be desirable. The Extended Lapped Transform (ELT) [33, 35, 34] is an extension of the MDCT for which the direct ELT operation is defined as in (3.10). However, the ELT allows for larger overlapping blocks of length $2mM$, with $m \in \mathbb{N}$ the *overlapping factor*. As such it is a particular instance of the general class of perfect reconstructing cosine modulated filter banks [23, 24]. Given equivalent symmetric analysis and synthesis windows, the condition on the ELT analysis window h to achieve perfect

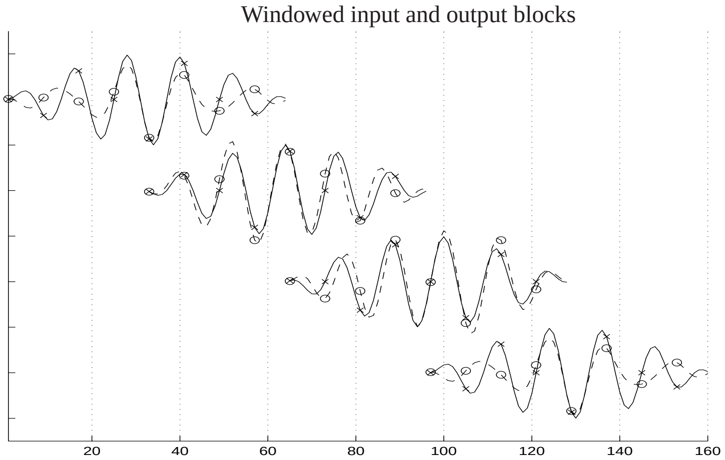


Figure 3.6: *Example of the introduced time aliasing by comparing the windowed input blocks (solid lines) with the reconstructed windowed output blocks (dashed line) before overlap-add.*

reconstruction is

$$\sum_{l=0}^{2m-2s-l} h(n+lM)h(n+lM+2sM) = \delta(s),$$

for $s = 0, \ldots, m-1$ and $n = 0, \ldots, M/2-1$. For $m = 1$, this condition reduces to (3.8) such that we can employ the MDCT windows from section 3.2.3. For $m > 2$, no simple parametrization of analysis windows exists and we have to rely on numerical optimization techniques in order to design appropriate windows.

Hierarchical Lapped Transform

The transforms as encountered thus far, for example the DFT, DCT and MDCT, result in a uniform time-frequency decomposition of the input signal. That is, given a choice for M, all the basis functions have the same length and time localization and their frequency responses have the same bandwidth. In some cases, a multiresolution approach can deliver better performance, specifically for nonstationary and transient signals. The Hierarchical Lapped Transform (HLT) [30, 35] employs the MDCT in a hierarchical or pyramidal structure. At each pyramid level, a given number of MDCT coefficients, typically corresponding to low frequency bands, are taken as input for a subsequent MDCT transform at the next level. As such, a nonuniform time-frequency decomposition (e.g., an octave band splitting) of the input signal can be obtained. The HLT is connected to wavelet packet and local cosine transforms. In practical applications, the HLT allows for progressive transmission and can increase the time resolution to prevent ringing artifacts. The concept of a nonuniform MDCT is further explored in section 3.3.

Modulated Complex Lapped Transform

It was already noted that the MDCT does not lead to perfect reconstruction of a single input block. If the critical sampling property of the MDCT is relaxed, an oversampled transform known as the modulated complex lapped transform (MCLT) [39] can be constructed, with an oversampling factor of two. The MCLT does not rely on time domain aliasing cancellation and therefore the reconstruction formula

$$\hat{y}_i(n) = \sum_{k=0}^{M-1} X(k)\varphi_k(n),$$

leads to perfect reconstruction of the block y_i, in the absence of quantization. The basis functions p of the MCLT are a combination of the MDCT from (3.10) and the modified discrete sine transform (MDST), i.e.

$$\begin{aligned}
\varphi_k(n) &= \varphi_k^c(n) - j\varphi_k^s(n), \\
\varphi_k^c(n) &= \sqrt{\frac{2}{M}}h(n)\cos\Big[\frac{\pi}{M}\Big(k+\frac{1}{2}\Big)\Big(n+\frac{M+1}{2}\Big)\Big], \\
\varphi_k^s(n) &= \sqrt{\frac{2}{M}}h(n)\sin\Big[\frac{\pi}{M}\Big(k+\frac{1}{2}\Big)\Big(n+\frac{M+1}{2}\Big)\Big].
\end{aligned}$$

The MCLT is a good candidate to replace the DFT for applications such as noise suppression and acoustic echo cancellation. Furthermore, it can be employed for equalization tasks such as in the recent spectral bandwidth replication technique [10, 13].

Due to the oversampling, the MCLT is less suitable for (audio) coding purposes, although in [50, 5] results for a nonuniform MCLT in an audio coding application are presented with some favorable results. An attractive feature of the MCLT is the fact that both the MDCT and MDST coefficients can be obtained directly from the complex coefficients, which allows joint coding/enhancement operations.

3.3 Adaptive time-frequency decomposition

The increased length of the MDCT offers substantial advantages over nonoverlapping block transforms such as the DCT. Most noticeably, a reduction of blocking artifacts is obtained. On the other hand, the larger support of the MDCT basis functions results in an increase of ringing artifacts, e.g., pre-echo and reverberation artifacts. An adaptive time-frequency decomposition can reduce these artifacts. The various window designs presented in section 3.2.3 allow for a limited time-frequency trade-off in the MDCT behavior. Since further adaptation of the time-frequency resolution than the MDCT offers is desired, more advanced techniques are required.

In this section three techniques for obtaining MDCT-based adaptive time-frequency signal decompositions are considered. These techniques will be used extensively in the rate-distortion optimization algorithms in Part II. First, we study the block switching approach, which will allow us to vary the MDCT window length and thus the number of basis functions. Secondly, we look at temporal noise shaping, which employs frequency domain linear prediction and an open-loop quantization scheme. The combination of a uniform filter bank with frequency domain linear prediction leads to nonuniform filter banks. Thirdly, subband merging is investigated. This method allows for flexible nonuniform filter bank designs where the subband filters have specific properties.

Various other methods for adaptive time-frequency decompositions exist that are not discussed in this thesis. For adaptive time segmentation, Bosi and Davidson [2] propose a phase shift in the MDCT kernel to allow shortened transition windows. The gain modification techniques presented in [58, 27] are preprocessing operations that adjusts the temporal envelope of the input signal prior to the MDCT transform. Nonuniform frequency decompositions can be obtained with the frequency-varying MDCT based on a dual-stage MDCT recombination method in [48]. Although some of these techniques have found their way into existing audio coding standards such as Dolby AC-3 [55] and MPEG-2/4 AAC [42, 43], they remain less popular than the techniques described in this section.

3.3.1 window and block switching

An effective approach for counteracting temporal unmasking artifacts is to constrain the support of the basis functions and thus the support of the introduced quantization error. Furthermore, since transient phenomena typically require a high(er) bit rate, from a compression perspective it is beneficial to constrain these high rates to short time intervals. This requires an adaptation of the window length, typically referred to as block switching. Adaptation of the window length essentially results in a nonuniform time segmentation of the input signal under consideration.

An important question that arises when changing the length of an MDCT window, is whether the perfect reconstruction property of the MDCT in the case of overlapping windows can be retained. Furthermore, it is desirable that the switch to a different window length can be made instantaneous. In order to derive these PR conditions for the time-varying MDCT we first study the situation where we vary the window shape on a block basis, a procedure called window switching. A first solution to this problem for overlapping transforms was presented by Edler [12] and later by Vetterli and Kovacevic in [26]. Here the method by Edler is discussed.

Consider the situation as depicted in Fig. 3.7a where two nonidentically shaped analysis windows of length $2M$ are employed, denoted by h_{i-1} and h_i respectively, on adjacent signal blocks x_{i-1} and x_i. Assuming equivalent analysis and synthesis windows, the reconstructed signal block in the overlapping region $n = 0, 1, \ldots, M-1$ can be derived using (3.9) as

$$\begin{aligned}\hat{x}_i(n) &= h_{i-1}(n+M)^2 x_i(n) + h_{i-1}(n+M)h_{i-1}(2M-1-n)x_{i-1}(2M-1-n)\\ &+ h_i(n)^2 x_i(n) - h_i(n)h_i(M-1-n)x_{i-1}(2M-1-n).\end{aligned}$$

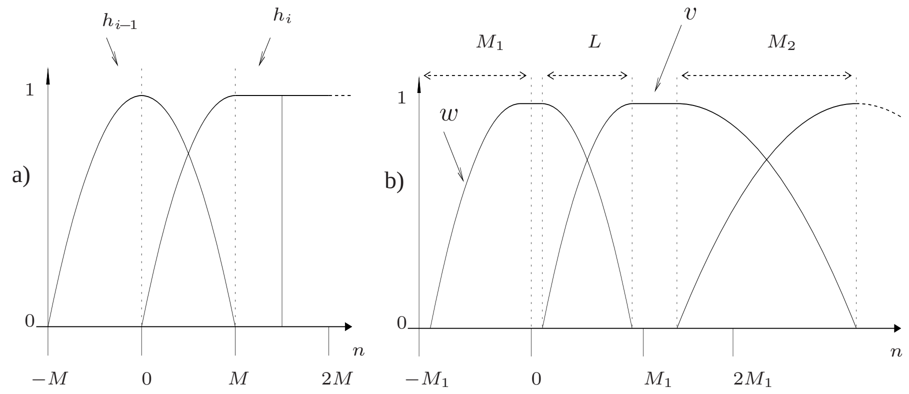


Figure 3.7: *a) Switching between two nonidentical windows of the same length. b) Switching between two nonidentical windows of different lengths* $2M_1$ *and* $2M_2$ *with mutual overlap* L*.*

Using the fact that $x_{i-1}(2M-1-n) = x_i(M-1-n), n = 0, 1, \ldots, M-1$, leads to

$$\begin{aligned}\hat{x}_i(n) &= h_{i-1}(n+M)^2 x_i(n) + h_{i-1}(n+M)h_{i-1}(2M-1-n)x_i(M-1-n) \\ &+ h_i(n)^2 x_i(n) - h_i(n)h_i(M-1-n)x_i(M-1-n).\end{aligned}$$

Hence, the conditions for perfect reconstruction in case of varying window shapes can be written as

$$h_{i-1}(n+M)h_{i-1}(2M-1-n) = h_i(n)h_i(M-1-n), \tag{3.15}$$

$$h_{i-1}(n+M)^2 + h_i(n)^2 = 1. \tag{3.16}$$

We see that the conditions (3.15) and (3.16) only involve the second half or right tail of the window h_{i-1} and the first half or left tail of the window h_i. Hence, the window h_i used in block x_i can have independently designed window tails as long as these tails satisfy (3.15) and (3.16) in relation with the windows used for adjacent blocks, h_{i-1} and h_{i+1}, respectively.

This property can be used not only to change the shape, but also the length of the applied MDCT windows from block to block. This is displayed in Fig. 3.7b where a transition from a window w of length $2M_1$ to a window v of length $2M_2, M_1 < M_2$ occurs. If the mutual overlap between the two windows is $L \leq M_1$ samples, then the right tail of w is given as

$$w(n+M_1) = \begin{cases} 1, & n = 0, \ldots, \frac{M_1-L}{2} - 1, \\ w_L(n - \frac{M_1-L}{2} + L), & n = \frac{M_1-L}{2}, \ldots, \frac{M_1+L}{2} - 1, \\ 0, & n = \frac{M_1+L}{2}, \ldots, M_1 - 1, \end{cases}$$

and the left tail of v as

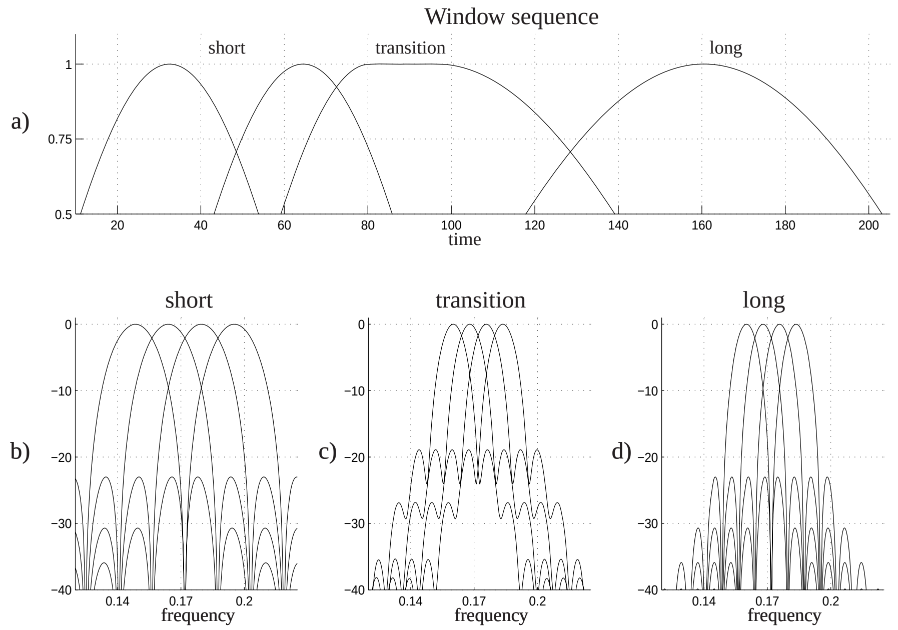


Figure 3.8: *Window switching leads to a nonuniform time segmentation. a) Window transition sequence from short to long segments. b) Magnitude response of* 4 *short (size-*32*) MDCT basis functions. c) Magnitude response of* 4 *transition (size-*64*) MDCT basis functions. d) Magnitude response of* 4 *long (size-*64*) MDCT basis functions.*

$$v(n) = \begin{cases} 0, & n = 0, \ldots, \frac{M_2-L}{2}-1, \\ v_L(n - \frac{M_2-L}{2}+L), & n = \frac{M_2-L}{2}, \ldots, \frac{M_2+L}{2}-1, \\ 1, & n = \frac{M_2+L}{2}, \ldots, M_2-1, \end{cases}$$

where w_L and v_L are windows of length $2L$ that satisfy the conditions (3.15) and (3.16) over the mutual overlap range $n = 0, \ldots, L-1$.

In general, window or block switching leads to asymmetric transition windows. Similar design constraints as listed in section 3.2.3 are relevant here. That is, a maximum overlap is required in case a high frequency resolution and maximum reduction of blocking artifacts is desired. On the other hand, a low overlap ensures that quantization noise is constrained to a limited region. Moreover, the use of the time-varying MDCT leads to a nonuniform time segmentation of the input segment. An example is provided in Fig. 3.8a, where the window sequence corresponding to a switch from a 32-channel MDCT to a 64-channel MDCT is shown. We can discern three window types, that is, short, transition and long windows. Magnitude responses of four basis functions for each of the window types are shown in Fig. 3.8b-d. Note the decrease in stopband reduction when the transition window is employed, as seen from Fig. 3.8c.

Although the window and block switching techniques provide us with enhanced time-frequency resolution adaptation, they have several limitations. For instance, the resulting frequency decompositions remain uniform, whereas in some cases a nonuniform decomposition might be desired. Furthermore, since windows having maximum overlap are often desired, asymmetric transition windows are required when switching between blocks of different length. The frequency domain properties of such windows are severely compromised [52]. That is, the transition windows have suboptimal magnitude responses compared to their symmetric counterparts, which can lead to reduced coding performance. Therefore, frequent switching between the different window lengths is not advisable. In [59] minimum mean-square error window designs are proposed to improve the properties of transition windows. Specifically when coding certain relatively non-stationary signals, e.g. speech, block switching either leads to frequent switching between the available window lengths and hence, reduced coding efficiency, or in usage of long windows only, such that temporal artifacts occur.

In most of the current audio coding standards that employ block switching, only two different window lengths are possible, denoted by *long block mode* and *short block mode*, respectively. The usage of these long and short block modes is then controlled by, for example, transient detection mechanisms based on time-domain energy measures [12, 22] or perceptual entropy [20]. These mechanisms are not necessarily optimal. In Part II of this thesis, we develop RD optimized block switching techniques that circumvent some of the aforementioned problems by allowing a larger set of window lengths and a fast searching algorithm to obtain proper time segmentations.

3.3.2 temporal noise shaping

Temporal noise shaping (TNS), proposed by Herre and Johnston in [17, 18, 16], is a form of envelope-adapted processing [19] that provides acces to the temporal fine structure of a signal within a transform window. TNS allows for reshaping the quantization noise in the time domain through open-loop linear predictive coding (LPC) of frequency domain coefficients. This results in the temporal quantization noise following the signal more closely such that most of the quantization noise will reside in signal regions with significant energy in the time domain, thereby avoiding temporal unmasking problems in coding transient and speech signals.

Consider the open-loop linear prediction scheme depicted in Fig. 3.9. For a time-domain input signal x, let $X(k)$ denote its DFT and $X(z)$ its $\mathcal{Z}$-transform. In the encoder, a pth order prediction of $X(k)$ is made with the linear prediction FIR filter h, whose $\mathcal{Z}$-transform is denoted $A(z) = \sum_{i=1}^{p} a_i z^{-i}$. The prediction results in a frequency domain prediction error signal R, given as

$$R(z) = \Big(1 - A(z)\Big)X(z).$$

The prediction error is quantized to U, i.e. $U(k) = \mathcal{Q}\{R(k)\} = R(k) + q(k)$, where q is an additive quantization error. In the decoder, the signal is reconstructed from the received signal V to $\hat{X}$. The final resulting coding error is $E = X - \hat{X}$. In the

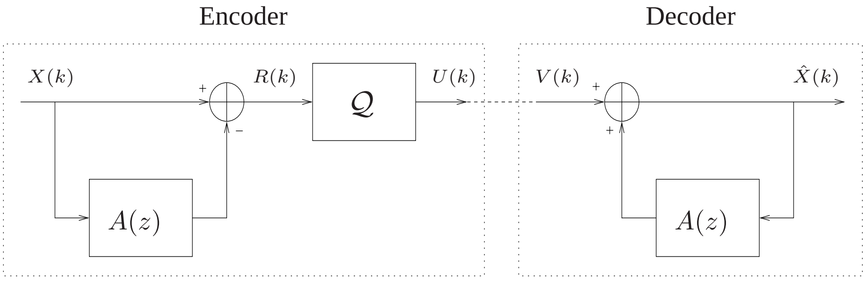


Figure 3.9: *Open-loop frequency domain linear prediction scheme.*

following, we assume $V = U$.

It turns out that E is a temporally shaped version of the quantization error q, that is, $E(z) = Q(z)/\Big(A(z) - 1\Big)$. This can be derived as follows.

$$\begin{aligned} E(z) &= X(z) - \hat{X}(z) \\ &= X(z) - U(z) - A(z)\hat{X}(z) \\ &= X(z) - \Big(1 - A(z)\Big)X(z) - Q(z) - A(z)\hat{X}(z) \\ &= A(z)X(z) - Q(z) - A(z)\hat{X}(z) \\ &= A(z)E(z) - Q(z) \\ &= \frac{Q(z)}{A(z) - 1}. \end{aligned} \tag{3.17}$$

We observe that the overall coding error is shaped by an IIR filter a', whose $\mathcal{Z}$-transform is $A'(z) = 1/\Big(A(z) - 1\Big)$. Due tot the existence of a duality between the squared Hilbert envelope of a signal and its spectral autocorrelation sequence, this inverse or synthesis IIR filter is a pth order estimate of the Hilbert envelope of the time-domain signal x. In Appendix B we provide further details on the duality between the squared Hilbert envelope and the spectral autocorrelation sequence, as well as on frequency domain linear prediction.

TNS is part of the MPEG-2/4 AAC standard [42, 43] where it is applied to MDCT coefficients. In [17] the authors propose a detection scheme based on *prediction gain* to control TNS usage, where the prediction gain is computed as the ratio between the signal energy and the error signal energy. The filter order is determined by a thresholding operation. Moreover, TNS can be applied to selected frequency regions and both forward and backward prediction schemes can be employed. The TNS operation performs an in-place filtering operation on the MDCT coefficients and instead of the MDCT coefficients, the prediction residual along with the coded prediction filter coefficients are stored in the bitstream.

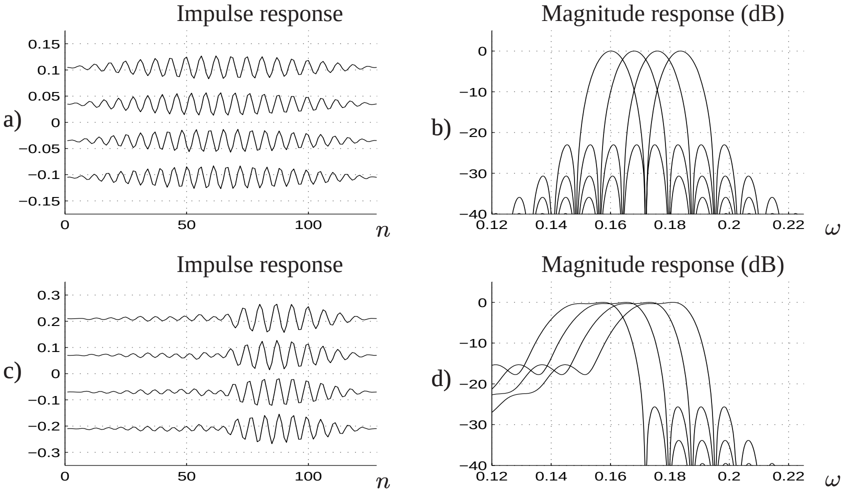


Figure 3.10: *The combination of a filterbank and frequency domain linear prediction leads to a trade-off between time localization and frequency bandwidth. (a) and (b) show the impulse and magnitude responses of* 4 *filters of a* 64*-channel uniform MDCT, (c) and (d) show the impulse and magnitude responses after prediction. A KBD window was applied.*

The use of the MDCT, rather than the DFT, introduces time domain aliasing effects in the temporally shaped noise. That is, the shaped quantization noise appears mirrored in both the left and right window half. Since the final reconstructed output is obtained after application of an MDCT synthesis window and overlap-add, the aliased noise components can be attenuated by selecting an MDCT window with low overlap. This is done in the MPEG-4 Low Delay audio coding scheme [1], which exclusively uses TNS as a method for pre-echo control. However, such a low overlap window leads to a reduction of frequency selectivity.

The combination of the MDCT and frequency domain prediction can be interpreted as a continuously signal adaptive filter bank [18]. In contrast to the block switching approach of section 3.3.1, an adaptation to the input signal characteristics within an MDCT window is provided, leading to nonuniform filter banks. An example is given in Fig. 3.10, where it is shown that frequency domain prediction in the MDCT trades time resolution for frequency bandwidth. The upper plots (Fig. 3.10a and b) show four MDCT basis functions and their magnitude responses before prediction. In the lower plots (Fig. 3.10c and d) basis functions and corresponding magnitude responses are shown after prediction with an 20th order filter has been performed. From the lower plots it can be observed that the basis functions obtained after filtering have increased time localization, centered at the same position, and an increased bandwidth. In general, for signals that display high correlation between adjacent spectral components, frequency resolution is traded for increased temporal resolution, represented

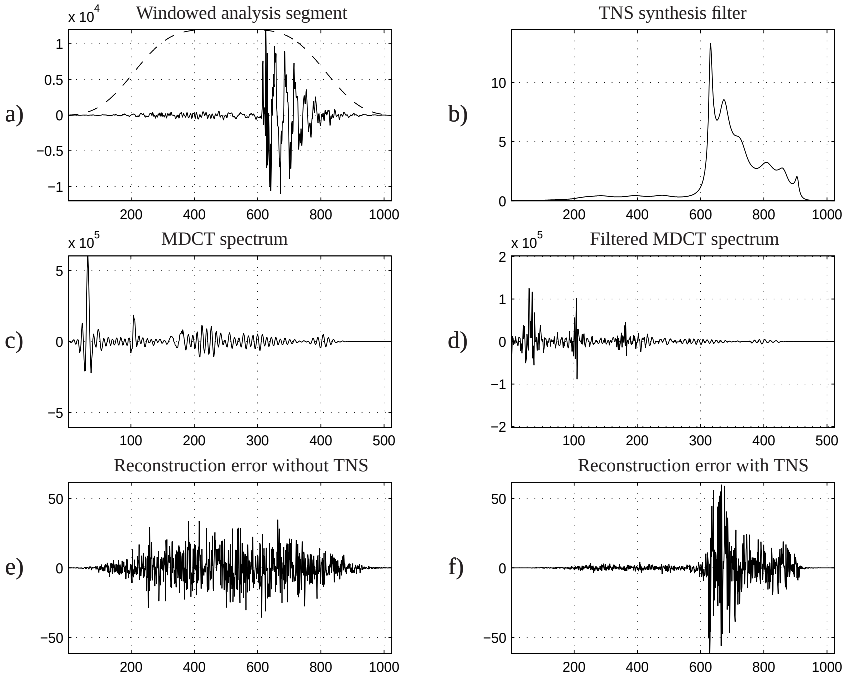


Figure 3.11: *Temporal noise shaping leads to reduced pre-echo artifacts. a) The windowed input segment and analysis window. b) The response of the TNS synthesis filter. c) The MDCT spectrum before TNS filtering. d) The MDCT spectrum after TNS filtering. e) Quantization noise without TNS. f) Quantization noise with TNS.*

by a set of basis functions having similar time localization and increased frequency bandwidths. Moreover, multiple filters can be applied to selected frequency ranges and thus highly flexible nonuniform filter banks can be obtained. However, the structure of the resulting filter banks and the frequency decomposition they give rise to are not directly transparent from the TNS filter coefficients.

A practical example of the TNS operation on real data is displayed in Fig. 3.11. A single analysis block of 1024 samples, obtained from a castanet input signal, is shown in Fig. 3.11a. The dotted lines denote the KBD window that is employed for windowing. Since the signal content is highly localized, with an attack starting in the middle of the block, pre-echos are to be expected if no additional provisions are taken to prevent temporal unmasking. If we apply TNS, we obtain the TNS reconstruction filter with the temporal envelope shown in Fig. 3.11b, which clearly resembles the envelope of the windowed analysis block. In Figs. 3.11c and d, the MDCT spectra before and after TNS filtering are displayed. The resulting quantization noise signals with and without TNS are presented in Figs. 3.11e and f, respectively. Clearly, the application of TNS leads to a temporally shaped noise signal, such that the majority of the quantization noise energy coincides with the signal energy and thus pre-echos are eliminated.

The TNS algorithm combined with a selection method based on prediction gain does not always lead to desired coding result. Apart from artifacts introduced by time-domain aliasing, some additional artifacts are described in [28]. There, the authors propose a selection method in which the perceptual entropy measure [20] is employed. In Part II of this thesis, we apply TNS within an RD optimization framework for efficient selection of the TNS usage and filter order.

3.3.3 subband merging

A simple and elegant method to construct a nonuniform MDCT is Malvar's subband merging algorithm [36, 37]. By taking linear combinations of the constituent basis functions of a uniform MDCT, the time resolution can be increased locally at the expense of a larger bandwidth, resulting in a nonuniform MDCT. Unlike most methods to design signal transforms that give rise to nonuniform frequency decompositions, the number of basis functions is not reduced. The subband merging algorithm is devised such that the operation of merging the basis functions is invertible, thereby retaining the PR property of the underlying uniform MDCT. Furthermore, the proposed method allows for an efficient implementation of a time-varying MDCT without the need for transition filters. It was shown that subband merging can be used beneficially for reducing ringing artifacts in audio and speech coding [37].

The design method proposed in [36, 37] was restricted to combinations of two or four basis functions only, and no systematic design procedure was given. An extension to the subband merging approach is proposed by Niamut and Heusdens [44]. The more general case of combining an arbitrary integer number, say $p \leq M$, of adjacent filters in an arbitrary uniform cosine-modulated filter banks (CMFB) is investigated. Necessary and sufficient conditions are derived on the underlying uniform CMFB and the way to combine the constituent filters such that resulting combined filters possess good frequency selective properties and flat passband response. A short overview of the method is given below. Since subband merging is best understood from a filter bank viewpoint [57], a $\mathcal{Z}$-domain notation is adopted to facilitate the necessary derivations.

Let a closed-form expression of the impulse response h_k of the kth analysis filter of an M-channel maximally decimated CMFB be given by [15]

$$h_k(n) = 2p_0(n)\cos\Big[\frac{\pi}{M}\big(k+\frac{1}{2}\big)\big(n-\frac{\alpha}{2}\big)\Big], \quad n = 0,\ldots,N-1, \tag{3.18}$$

where $\alpha \in \mathbb{Z}$ is called the modulation phase and p_0 is the CMFB prototype filter (e.g. an MDCT window). Perfect reconstruction of the CMFB can be obtained with suitable choices for α and p_0. Furthermore, let $H_{p,k}(z)$ be a linear combination of p adjacent filters starting from the kth filter in the CMFB, i.e.

$$H_{p,k}(z) = \sum_{i=0}^{p-1} b_{k+i}H_{k+i}(z), \tag{3.19}$$

with $b_k = e^{j\varphi_k}$ the combinatorial coefficients of magnitude 1. The index of the first filter can be $k = 0, \ldots, M-p$.

If $|H_{p,k}(z)|^2$ is equal to $\sum_{i=0}^{p-1} |H_{k+i}(z)|^2$, then $H_{p,k}(z)$ has flat passband response and a transition bandwidth similar to those of the underlying uniformly spaced subband filters. The following theorem gives necessary and sufficient conditions on the modulation phase and the combinatorial coefficients such that the resulting combined filters indeed exhibit the required frequency behavior.

Theorem 3. ***(subband merging)*** *Let p_0 denote a real-coefficient linear-phase lowpass prototype filter of length N for an M-channel PR uniform CMFB, satisfying*

$$|P_0(e^{j\omega})| = 0 \quad \textit{for } |\omega| \geq \frac{\pi}{2M} + \varepsilon,\ \varepsilon < \frac{\pi}{2M}, \tag{3.20}$$

and let $b_k = e^{j\varphi_k},\ k = 0, \ldots, M-1$. Furthermore, let the analysis filters h_k of the CMFB be defined as in (3.18).

Then

$$\left|\sum_{i=0}^{p-1} b_{k+i} H_{k+i}(z)\right|^2 = \sum_{i=0}^{p-1} \left|H_{k+i}(z)\right|^2,$$

for $1 \leq p \leq M$ and $0 \leq k \leq M-p$, if and only if $\alpha = (N-1) - M(2m+1),\ m \in \mathbb{Z}$, and $|\varphi_k - \varphi_{k+1}| = n\pi,\ n \in \mathbb{N}$.

Proof: The proof is provided in Appendix C.

Eq.(3.20) sets a condition on the prototype filter p_0 that can never be satisfied in practical applications since it requires infinite length filters. Hence, overlapping terms in the frequency domain of non-adjacent filters do exist and will result in ripples in the passband of the combined filters. By keeping the stopband attenuation of the prototype filter high, these ripples are kept to a minimum. The condition on α is satisfied by many of the existing CMFB designs, including the MDCT. Furthermore, subband merging can be implemented using integer matrices as a post-processing operation on the CMFB subband channel signals. This significantly reduces the computational complexity of the algorithm, since only one analysis operation is required to obtain multiple frequency decompositions.

An example of the subband merging method is given in Fig. 3.12, where it is shown that subband merging trades frequency bandwidth for time resolution. From the lower plots (Fig. 3.12c and d), it can be observed that the merged filters have increased bandwidths, centered around the same center frequency and increased time localization, centered at different positions. We can express the time localization of the filters using their time means μ_t as defined in chapter 2.3.2. The difference in time localization between the merged filters can be exactly calculated using the equivalence relation displayed in Fig. 3.13. That is, the kth filter bank channel, represented by the filter $H_k(z)$ and decimation with a factor M, can be replaced by two channels that

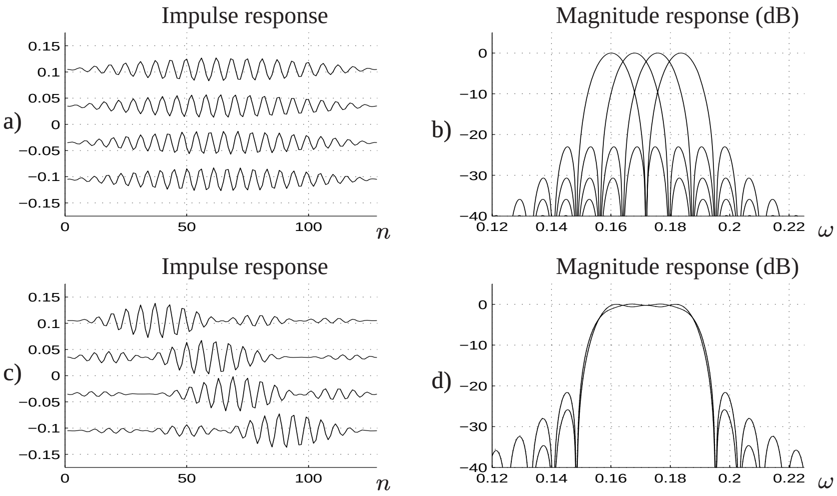


Figure 3.12: *Subband merging trades frequency bandwidth for time resolution. (a) and (b) show the impulse and magnitude responses of* 4 *filters of a* 64*-channel uniform CMFB, (c) and (d) show the impulse and magnitude responses after subband merging.*

apply the same filter $H_k(z)$ and decimation factor $M/2$, where one of the channels is delayed by $M/2$ samples. Extending this relation to p filters leads to a difference of M/p samples between the time means of merged filters.

The time-frequency resolution trade-off of the subband merging algorithm is not optimal, in the sense that the Heisenberg product of the time and frequency variances of the filters increases with the number of merged filters. We analyze this behavior in the following experiment, where we compare two sets of filter banks. The CMFBs in the first set employ filters of different lengths, resulting in 1024 to 32 distinct channels. This can be seen as a *direct* design method. The second set is created from a uniform CMFB having 1024 channels. Subband merging is employed to combine adjacent filters such that new CMFBs are obtained with reduced frequency resolution and increased time localization, but a constant number of filters. If we merge the appropriate number of filters, we can thus create filter banks that have 1024 to 32 channels, but 1024 filters. For all CMFBs, average time and frequency variances as defined in chapter 2.3.2 and denoted σ_t^2 and σ_f^2, respectively, are computed according to [56]. Similarly, values for the time-frequency localization ν are obtained.

The results are shown in Figure 3.14. Figure 3.14a shows the time variance and it is seen that employing the direct design method, the time localization increases much faster than when subband merging is applied. The increase in time variance can be expected, since we optimized subband merging for frequency domain behavior without constraints on the resulting impulse responses. On the other hand, from the frequency variances displayed in Figure 3.14b it is observed that subband merging leads to a slightly increased frequency localization compared to a direct design. The

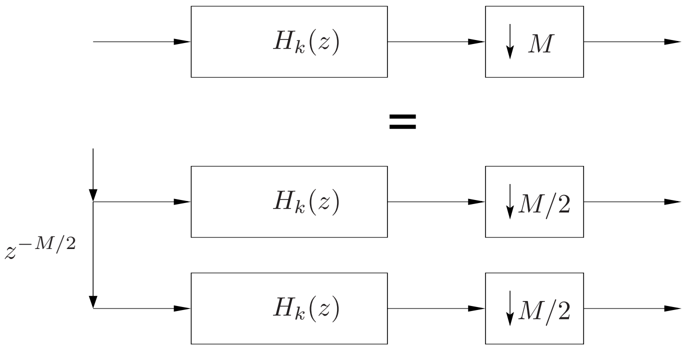


Figure 3.13: *Equivalence relation between a channel with decimation by M and 2 channels with decimation by $2M$.*

time-frequency localization is shown in Figure 3.14c, from which we see that the direct design method results in a constant time-frequency localization when varying the number of channels. In contrast, subband merging leads to an increase of the time-frequency uncertainty when a large number of filters is merged. Therefore, from this experiment it is concluded that an upperbound should be set on the number of adjacent channels that are merged, since the time-frequency localization of filters obtained by merging a large number of subbands is suboptimal. In Part II of this thesis, we consider an operational RD optimization framework that employs subband merging for nonuniform filter bank design and MDCT-based audio coding.

3.4 Conclusion

In this chapter, we studied the modified discrete cosine transform in detail. We derived the conditions for perfect reconstruction and time domain aliasing cancellation. Furthermore, we investigated several window designs and a fast implementation. Moreover, we analyzed three methods for adaptive time-frequency decompositions. In Part II of this thesis we combine these methods with the RD optimization and best basis search techniques from the previous chapter.

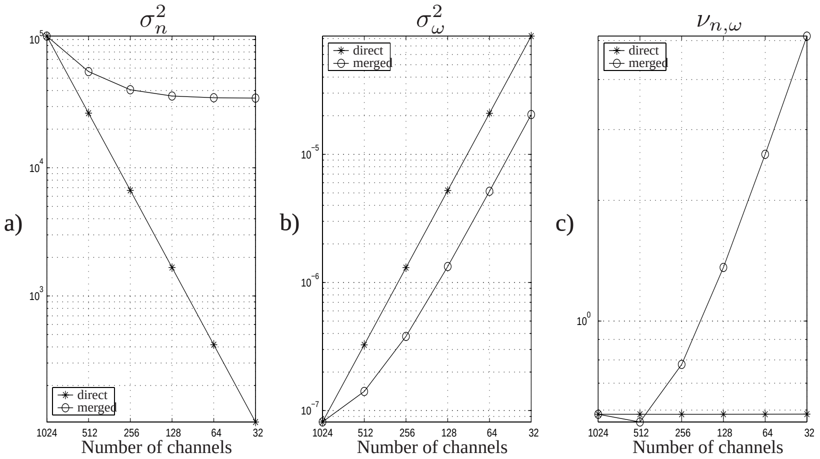


Figure 3.14: *Increase of overall time-frequency uncertainty ν as a result of subband merging. For decreasing number of channels, a)time variances, b) frequency variances and c)time-frequency localization values are displayed.*

Bibliography

[1] E. Allamanche, R. Geiger, J. Herre, and T. Sporer. MPEG-4 low delay coding based on the AAC codec. In *106th AES Convention, Preprint 4929*, Munich, Germany, May 1999.

[2] M. Bosi and G.A. Davidson. High-quality, low-rate audio transform coding for transmission and multimedia applications. In *Proceedings of the 93rd AES Convention*, San Francisco, USA, October 1992.

[3] R. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, New York, 1965.

[4] V. Britanak and K.R. Rao. An efficient implementation of the forward and inverse MDCT in MPEG audio coding. *IEEE Signal Processing Letters*, 8(2):48–51, February 2001.

[5] S. Cheng and Z. Xiong. Audio coding and image denoising based on the nonuniform modulated complex lapped transform. *IEEE Transactions on Multimedia*, 7(5):817–827, October 2005.

[6] S. Cheung and J.S. Lim. Incorporation of biorthogonality into lapped transforms for audio compression. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'95)*, pages 3079–3082, Detroit, USA, May 1995.

[7] J. W. Cooley and J. W. Tukey. An algorithm for the machine computation of the complex fourier series. *Mathematics of Computation*, 19:297–301, April 1965.

[8] L. Daudet and M. Sandler. MDCT analysis of sinusoids and applications to coding artifacts reduction. In *Proceedings of the 114th AES Convention*, Amsterdam, The Netherlands, March 2003.

[9] L. Daudet and M. Sandler. MDCT analysis of sinusoids: Exact results and applications to coding artifacts reduction. *IEEE Transactions on Speech and Audio Processing*, 12(3):302–312, May 2004.

[10] M. Dietz, L. Liljeryd, K. Kjorling, and O. Kunz. Spectral band replication, a novel approach in audio coding. In *112th AES Convention, Preprint 5553*, Munich, Germany, April 2002.

[11] P. Duhamel, Y. Mahieux, and J.P. Petit. A fast algorithm for the implementation of filter banks based on TDAC. In *Proceedings of the 1991 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'91)*, pages 2209–2212, Toronto, Canada, April 1991.

[12] B. Edler. Codierung von audiosignalen mit uberlappender transformation und adaptiven fensterfunktionen (in german). *Frequenz*, 43(9):252–256, 1989.

[13] Per Ekstrand. Bandwidth extension of audio signals by spectral band replication. In *Proc. IEEE First Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002)*, pages 53–58, Leuven, Belgium, November 2002.

[14] A.J.S Ferreira. Convolutional effects in transform coding with TDAC: an optimal window. *IEEE Transactions on Speech and Audio Processing*, 4(2):104–114, March 1996.

[15] R.A. Gopinath and C.S. Burrus. Some results in the theory of modulated filter banks and modulated wavelet tight frames. *Applied Computational Harmonic Analysis*, 2:303–326, March 1995.

[16] J. Herre. Temporal noise shaping, quantization and coding methods in perceptual audio coding: A tutorial introduction. In *Proceedings of the AES 17th International Conference: High-Quality Audio Coding*, Florence, Italy, September 1999.

[17] J. Herre and J.D. Johnston. Enhancing the performance of perceptual audio coders by using temporal noise shaping. In *101st AES Convention, Preprint 4384*, Los Angeles, USA, November 1996.

[18] J. Herre and J.D. Johnston. Continuously signal-adaptive filterbank for high-quality perceptual audio coding. In *Proceedings of the 1997 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'97)*, New Paltz, USA, October 1997.

[19] J. Herre and J.D. Johnston. Exploiting both time and frequency structure in a system that uses an analysis/synthesis filterbank. In *Proceedings of the 103rd AES Convention*, New York, USA, September 1997.

[20] J. D. Johnston. Estimation of perceptual entropy using noise masking criteria. In *Proceedings of the 1988 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'88)*, pages 2524–2527, New York, USA, April 1988.

[21] J. F. Kaiser. *Digital Filters*. Wiley, New York, NY, 1966.

[22] J. Kliewer and A. Mertins. Audio subband coding with improved representation of transient signal segments. In *Proceedings of the 9th European Signal Processing Conference (Eusipco'98)*, pages 1245–1248, Rhodes, Greece, September 1998.

[23] R.D. Koilpillai and P.P. Vaidyanathan. New results on cosine-modulated FIR filter banks satisfying perfect reconstruction. *Proceedings of the 1991 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'91)*, pages 1793–1796, April 1991.

[24] R.D. Koilpillai and P.P. Vaidyanathan. Cosine-modulated FIR filter banks satisfying perfect reconstruction. *IEEE Transactions On Signal Processing*, 40(4):770–783, April 1992.

[25] C.W. Kok. Fast algorithm for computing discrete cosine transform. *IEEE Transactions on Signal Processing*, 45(3):757–760, March 1997.

[26] J. Kovačević and M. Vetterli. Time-varying modulated lapped transforms. In *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*, pages 481–485, Pacific Grove, USA, November 1993.

[27] M. Link. An attack processing of audio signals for optimizing the temporal characteristics of a low bit-rate audio coding system. In *Proceedings of the 95th AES Convention*, New York, USA, October 1993.

[28] C.M. Liu, W.C. Lee, and T.W. Chang. The efficient temporal noise shaping method. In *Proceedings of the 116th AES Convention*, Berlin, Germany, May 2004.

[29] H.S. Malvar. The LOT: a link between block transform coding and multirate filter banks. In *Proceedings of the 1988 IEEE International Symposium on Circuits and Systems (ISCAS'88)*, pages 835–838, Helsinki, Finland, June 1988.

[30] H.S. Malvar. Efficient signal coding with hierarchical lapped transforms. In *Proceedings of the 1990 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'90)*, pages 1519 –1522, Albuquerque, USA, April 1990.

[31] H.S. Malvar. Lapped transforms for efficient transform/subband coding. *IEEE Transactions On Acoustics, Speech, And Signal Processing*, 38(6):969–978, June 1990.

[32] H.S. Malvar. Modulated QMF filter banks with perfect reconstruction. *Electronics Letters*, 26(13):906–907, June 1990.

[33] H.S. Malvar. Extended lapped transforms: Fast algorithms and applications. In *Proceedings of the 1991 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'91)*, pages 1797–1800, Toronto, Canada, April 1991.

[34] H.S. Malvar. Extended lapped transforms: Properties, applications and fast algorithms. *IEEE Transactions on Signal Processing*, 40(11):2703–2714, November 1992.

[35] H.S. Malvar. *Signal Processing with Lapped Transforms*. Artech House, Boston, MA, 1992.

[36] H.S. Malvar. Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts. *IEEE Transactions on Signal Processing*, 46(4):1043–1053, April 1998.

[37] H.S. Malvar. Enhancing the performance of subband audio coders for speech signals. In *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS'98)*, pages 98–101, Monterey, USA, June 1998.

[38] H.S. Malvar. Fast algorithms for orthogonal and biorthogonal modulated lapped transforms. In *Proceedings of the 1998 IEEE Symposium on Advances in Digital Filtering and Signal Processing*, pages 159–163, Victoria, Canada, June 1998.

[39] H.S. Malvar. A modulated complex lapped transform and its applications to audio processing. In *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'99)*, pages 1421–1424, Phoenix, USA, March 1999.

[40] G. Matviyenko. Optimized local trigonometric bases. *Applied and Computational Harmonic Analysis*, 3(4):301–323, October 1996.

[41] International Standard ISO/IEC 11172-3 (MPEG). Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s. part 3: Audio, 1993.

[42] International Standard ISO/IEC 13818-7 (MPEG). Information technology - generic coding of moving pictures and associated audio, part 7: Advanced audio coding, 1997.

[43] International Standard ISO/IEC 14496-3 (MPEG). Information technology - coding of audio visual objects, part 3: Audio, 2001.

[44] O.A. Niamut and R. Heusdens. Subband merging in cosine-modulated filter banks. *IEEE Signal Processing Letters*, 10(4):111–114, April 2003.

[45] A. Papoulis. *The Fourier Integral and its Applications*. McGraw-Hill, New York, 1962.

[46] J.P. Princen and A.B. Bradley. Analysis/synthesis filter bank design based on time domain aliasing cancellation. *IEEE Transactions On Acoustics, Speech, And Signal Processing*, 34(5):1153–1161, October 1986.

[47] J.P. Princen, A.W. Johnson, and A.B. Bradley. Subband/transform coding using filter bank designs based on time domain aliasing cancellation. In *Proceedings of the 1987 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'87)*, pages 2161–2164, Dallas, USA, April 1987.

[48] M. Purat and P. Noll. A new orthonormal wavelet packet decomposition for audio coding using frequency-varying modulated lapped transforms. In *Proceedings of the 1995 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'95)*, pages 183–186, New York, USA, October 1995.

[49] K.R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, Boston, 1990.

[50] A.S. Scheuble and Zixiang Xiong. Scalable audio coding using the nonuniform modulated complex lapped transform. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'01)*, pages 3257–3260, Salt Lake City, USA, May 2001.

[51] D. Sevic and M. Popovic. A new efficient implementation of the oddly stacked princen-bradley filter bank. *IEEE Signal Processing Letters*, 1(11):166–168, November 1994.

[52] S. Shlien. The modulated lapped transform, its time-varying forms, and its applications to audio coding standards. *IEEE Transactions on Speech and Audio Processing*, 5(5):359–366, July 1997.

[53] D. Sinha and A.J.S. Ferreira. A new class of smooth power complementary windows and their application to audio signal processing. In *Proceedings of the 119th AES Convention*, New York, USA, October 2005.

[54] G. Smart and A.B. Bradley. Filter bank design based on time domain aliasing cancellation with non-identical windows. In *Proceedings of the 1994 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'94)*, pages III/185–III/188, Adelaide, Australia, April 1994.

[55] United States Advanced Television Systems Committee Digital Audio Compression (AC-3) Standard. Document a/52/10, December 1995.

[56] C. Taswell. *Wavelets in Signal and Image Analysis: From Theory To Practice*, chapter Empirical Tests for Evaluation of Multirate Filter Bank Parameters, pages 111–140. Kluwer Academic Publishers, 2001.

[57] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs, NJ, 1993.

[58] T. Vaupel. *Ein Beitrag zur Transformationscodierung von Audiosignalen unter Verwendung der Methode der Time Domain Aliasing Cancellation (TDAC) und einer Signalkompandierung im Zeitbereich*. PhD thesis, Universitat Duisberg, Duisberg, Germany, April 1991.

[59] G. Wang and U. Heute. Time-varying MMSE modulated lapped transform and its applications to transform coding for speech and audio signals. *Signal Processing*, 82(9):1283–1304, September 2002.

[60] Y. Wang and M. Vilermo. Modified discrete cosine transform–its implications for audio coding and error concealment. *Journal of the Audio Engineering Society*, 51(1):52–61, January 2003.

[61] Y. Wang, L. Yaroslavsky, and M. Vilermo. Energy compaction property of the MDCT in comparison with other transforms. In *Proceedings of the 109th AES Convention*, Los Angeles, USA, September 2000.

[62] Y. Wang, L. Yaroslavsky, and M. Vilermo. On the relationship between MDCT, SDPT and DFT. In *5th International Conference on Signal Processing Proceedings (ICSP'00)*, pages 44–47, Beijing, China, August 2000.

[63] Y. Wang, L. Yaroslavsky, M. Vilermo, and M. Vaananen. Some peculiar properties of the MDCT. In *5th International Conference on Signal Processing Proceedings (ICSP'00)*, pages 61–64, Beijing, China, August 2000.

[64] Eric W. Weisstein. Modified bessel function of the first kind. From MathWorld–A Wolfram Web Resource http://mathworld.wolfram.com/ ModifiedBesselFunctionoftheFirstKind.html.

[65] Wikipedia. Kaiser-bessel derived window. http://en.wikipedia.org/wiki/Kaiser_window, May 2003.

[66] L.P. Yaroslavsky. *Digital Signal Processing*, chapter Shifted Discrete Fourier Transforms, pages 69–74. Academic Press, London, 1980.

Part II

Papers

Families of orthogonal bases are created every day.
This game may however become tedious if not motivated by applications.

Stéphane Mallat, A Wavelet Tour of Signal Processing

Chapter 4

Flexible Frequency Decompositions for Cosine-Modulated Filter Banks

@ 2003 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE.

This chapter has been published as “Flexible Frequency Decompositions for Cosine-Modulated Filter Banks”, by O.A. Niamut and R. Heusdens in the Proceedings of the *IEEE International Conference on Audio, Speech and Signal Processing*, April 2003.

Abstract

We investigate the use of nonuniform cosine-modulated filter banks for audio coding. A rate-distortion framework is employed, similar to the work in [1], to select the filter bank structure from a large library of possible frequency decompositions. A new flexible frequency decomposition algorithm is proposed that jointly optimizes the filter bank structure and the bit allocation over the subband channels. Experimental results for both synthetic and real audio signals are provided. The new algorithm shows significant improvements in comparison with fixed uniform frequency decompositions, but special care has to be taken to reduce the size of the decomposition overhead.

4.1 Introduction

In most of the current audio coding standards a cosine-modulated filter bank (CMFB) is employed [2], using either a polyphase or lapped transform implementation. These filter banks provide a uniform frequency decomposition, i.e. a decomposition where all the subband channels are uniformly spaced in frequency. However, for more efficient coding of audio and speech signals, a larger library of filter bank structures is required in order to adapt the time-frequency resolution of the filter bank to the signal's changing characteristics [3].

A large library of filter bank structures is for instance provided by wavelet packets [4]. Various algorithms have been proposed that choose the optimal wavelet packet basis and corresponding quantizers per time segment, where optimality is defined in a rate-distortion (R-D) sense [5]. The resulting frequency decompositions are no longer restricted to uniform band divisions. On the other hand, for CMFBs only few algorithms exist to obtain time-varying nonuniform frequency decompositions [6, 7]. However, when compared to wavelet packets, CMFBs possess interesting properties for audio coding such as good frequency selectivity and simple design of transition filters.

In this paper, we propose a new algorithm to obtain a rate-distortion optimal frequency decomposition of an audio signal using CMFBs. By combining techniques for the design of nonuniform filter banks and dynamic programming-based R-D optimization, we construct the flexible frequency decomposition algorithm. The organization of this paper is as follows. In Section 4.2 some previous methods to obtain time-varying frequency decompositions are discussed. Section 4.3 describes the new algorithm in detail. In Section 4.4 some examples are provided and a comparison with fixed uniform decompositions is made. Section 4.5 contains the conclusions and recommendations for future work.

4.2 Previous work

For audio coding, several methods for adapting the time-frequency resolution of the analysis system have been proposed. In [8], the window-switching algorithm is presented. The time-frequency resolution is adapted by switching the analysis block

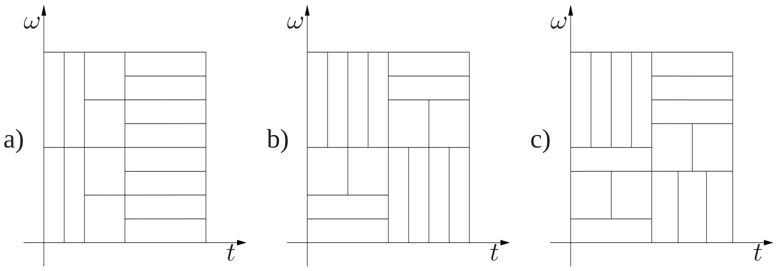


Figure 4.1: *Time-frequency tilings as obtained by decomposition algorithms. (a)Window-switching tiling (b)Single Tree tiling (c)Flexible Frequency Decomposition tiling.*

length, typically between a long-duration/high-frequency resolution mode and a short-duration/low-frequency resolution mode. The short window applied to a frame containing a transient will tend to minimize the temporal spread of quantization noise (which results in a reduction of pre-echos). Furthermore, it is desirable to constrain the high bit rates associated with transients to the shortest possible temporal regions only.

Although implemented in most of the current audio coding standards, the window-switching technique has some drawbacks. For instance, special transition windows have to be employed when switching between resolutions. This introduces extra coder delay and the spectral properties of these windows are poor compared to those of the original windows [9]. Moreover, the resulting frequency decompositions are still uniform and therefore limited in their ability to model non-stationary fragments correctly. See Figure 4.1a for an example of the time-frequency tilings that can be obtained using window-switching.

A frequency-varying decomposition method based on wavelet packets (WP) is disclosed in [10], where the *Single Tree* algorithm jointly finds the WP basis and bit allocation that are optimal in a rate-distortion sense. A Lagrange optimization technique is employed that searches along the convex hull of the R-D curve to determine the jointly optimal WP basis and corresponding quantizer choices. However, the use of wavelet packets in the Single Tree algorithm has several drawbacks. First of all, the frequency decompositions are limited to dyadic intervals (i.e. binary decompositions) only. Figure 4.1b shows an example of a tiling that can be obtained, while Figure 4.1c shows a tiling that cannot be achieved with the Single Tree algorithm. Secondly, carefully designed filters are needed at the segment boundaries [11] when the Single Tree is combined with time-segmentation algorithms. Moreover, the subband filters have poor frequency responses due to the cascaded implementation of the WP filter bank. Some work on a frequency-varying CMFB has been reported in [12]. However, this algorithm starts from a decomposition that resembles the critical band structure. Within each critical band, only binary decompositions are possible.

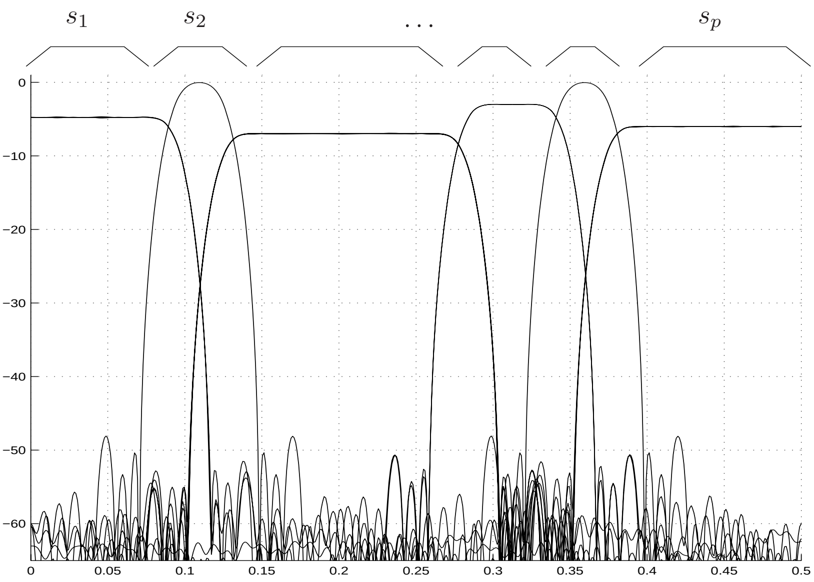


Figure 4.2: *A decomposition* S_k *is a collection of adjacent (nonuniform) subband channels* $s_1, \ldots, s_p$*.*

4.3 Flexible frequency decomposition

Given an M-channel uniform CMFB, we want to minimize the total distortion over all possible frequency decompositions and all possible ways of quantizing the corresponding subband signals such that the total required bit rate does not exceed a certain target rate R_t. If we limit ourselves to the case where every possible decomposition consists of subband channels having a bandwidth that is an integer multiple of a predefined minimum bandwidth (i.e. the bandwidth of the filters of the underlying uniform CMFB), this problem becomes the frequency equivalent of the flexible time segmentation algorithm proposed in [1]. To state the problem more formally we introduce some notation.

Let $S = \{S_1, \ldots, S_{2^{M-1}}\}$ be the set of all possible frequency decompositions, where $S_k = \{s_1, \ldots, s_p\}$ is a collection of adjacent (nonuniform) frequency intervals. Figure 4.2 shows an example of such a decomposition. Furthermore, assume that we are given a set of quantizers $\{q_n\}$ to quantize the subband samples in a decomposition and let $Q = \{Q_l, \ldots, Q_N\}$ denote the set of all possible ways of quantizing the different decompositions S_k, where $Q_l = \{q_1(s_1), \ldots, q_p(s_p)\}$. The problem that we want to solve can then be expressed as

$$\begin{aligned} &\min_{S} \min_{Q} D(S_k, Q_l) \\ &\text{subject to} \quad R(S_k, Q_l) \leq R_t. \end{aligned} \tag{4.1}$$

Clearly, Eq. 4.1 can be solved by introducing a Lagrange multiplier $\lambda \geq 0$ and solving

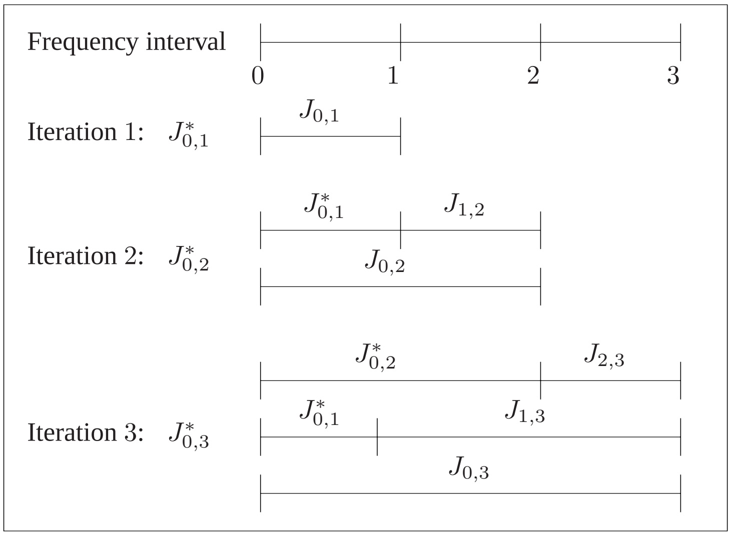


Figure 4.3: *Dynamic Programming is employed to search iteratively for the optimal decomposition.*

the unconstrained minimization problem

$$\min_S \min_Q J(\lambda) = \min_S \min_Q \sum_{i=1}^{p} J_i(\lambda, s_i, q_i(s_i)), \tag{4.2}$$

where we assume that rate and distortion are additive over the subband channels.

Solving Eq. 4.2 directly would require an exhaustive search of computational complexity $\mathcal{O}(2^M)$. However, if we can assume that the different subband channels are mutually uncorrelated, the search for the optimal quantizer strategy given a particular decomposition can be done on a channel-by-channel basis, that is,

$$\min_Q \sum_{i=1}^{p} J_i(\lambda, s_i, q_i(s_i)) = \sum_{i=1}^{p} \min_{q_i(s_i)} J_i(\lambda, s_i, q_i(s_i)). \tag{4.3}$$

This assumption is the key step in reducing the search complexity since we now can solve Eq. 4.2 using the dynamic programming technique [13], which results in a computational complexity of $\mathcal{O}(M^2)$.

The optimal frequency decomposition is now found recursively. Let $J_{k,l}$ denote the Lagrangian cost for encoding the frequency range $s_{k,l} = [\frac{\pi}{M}k, \frac{\pi}{M}l)$. Then, at each iteration i, the best frequency decomposition of the interval $[0, \frac{\pi}{M}i)$ is found by solving

$$J^*_{0,i} = \min_{0 \leq k \leq i} (J^*_{0,k} + J_{k,i}), \quad i = 1, \ldots, M, \tag{4.4}$$

where $J^*_{0,i}$ is the minimum cost for coding the interval $[0, \frac{\pi}{M}i)$. Figure 4.3 illuminates this procedure. After having found $J^*_{0,M}$ we can easily determine the optimal frequency decomposition by backtracking all the optimal split positions.

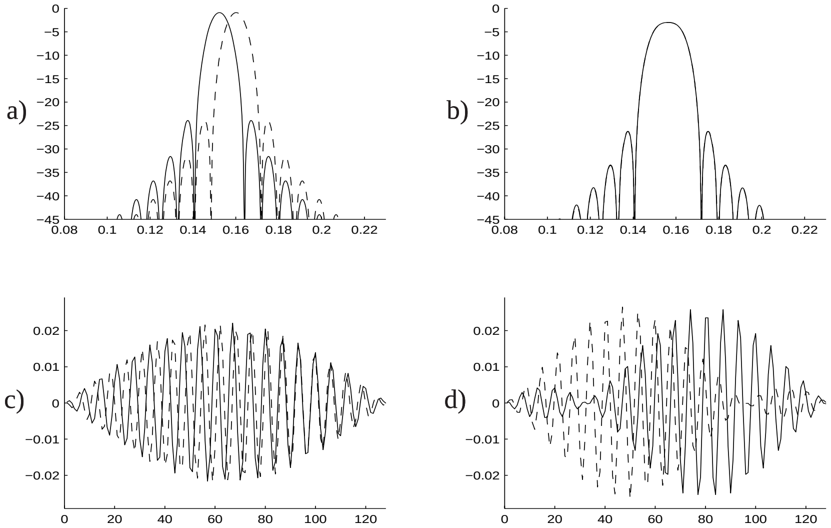


Figure 4.4: *Resolution switching for* 2 *filters with subband merging. (a)Magnitude response of unmerged filters, (b)magnitude response of merged filters, (c)time localization of unmerged filters and (d)time localization of merged filters.*

Obviously, if we do not know the right λ in advance, we have to repeat the aforementioned procedure for different values of λ in order to determine the optimal λ (i.e. the one that gives rise to $R = R_t$). Since the rate is a convex function of the distortion, efficient algorithms exist to find the optimal λ in a few iterations, e.g. the bisection method [10]. The computation of the Lagrangian costs for solving Eq. 4.4 can become very complex. In general, if we replace two adjacent subband channels by two double-bandwidth channels, the perfect reconstruction property is lost so that the other channel filters have to be modified as well and thus the subband signals. A complete signal transformation is then necessary for each and every possible decomposition, 2^{M-1} in total, which is unacceptable in most applications.

If the subband merging technique presented in [7] is employed, we can reduce the number of required signal transformations to only one, since the other decompositions can be derived by a simple post-processing of the subband signals of the underlying uniform CMFB. This is the main reason for applying this technique to the design of nonuniform frequency decompositions.

It is important to note that the merging operation does not reduce the number of channels by itself. For example, merging 2 adjacent channels results in 2 double-bandwidth channels, each having a different time localization. See Figure 4.4 for an example. As a result, in order to find the optimal bit allocation for a particular frequency interval $s_{k,l}$ we need different quantizers for the subband channels that constitute the interval under consideration.

Summarizing, the flexible frequency decomposition algorithm can be implemented as follows:

---

[1] For $k \in \{1, 2, \ldots, M\}$, compute every possible decomposition S_k of the frequency interval $[0, \frac{\pi}{M}k)$.

[2] For every decomposition S_k, compute all possible ways Q_l of quantizing the i subband samples, where $Q_l = \{q_1(s_1), \ldots, q_p(s_p)\}$, and record the resulting distortions and bit rates.

[3] For an initial value λ, find the optimal decomposition $S^*_{0,i}$ of the frequency interval $[0, \frac{\pi}{M}k)$, resulting in the minimum cost $J^*_{0,i}$ for $i = 1, \ldots, M$, where $J^*_{0,0} = 0$.

[4] Find the optimal value of λ, that corresponds to the target rate R_t, using the bisection algorithm [11].

4.3.1 reduction of algorithmic complexity

Several steps can be undertaken to reduce the complexity of the algorithm. For instance, instead of considering every possible combination of subband filters, we can limit the number of adjacent channels merged to powers of 2. As shown in [7], this restriction results in orthonormal nonuniform CMFBs, assuming that the underlying uniform CMFB is also orthonormal. Orthonormal filter banks are desirable, since in the quantization distortion can then be evaluated in the frequency domain only, so that the inverse filter bank operation is not needed at the encoder.

A second reduction in complexity is obtained by setting an upperbound on the number of adjacent channels that are merged. However, this restriction does not necessarily lead to a severe degradation of performance, because the time-frequency localization of filters obtained by merging a large number of subbands is suboptimal.

4.3.2 coding of side information

The decoder has to be informed about the selected filter bank structure. This structure can be represented as a binary sequence of length $M - 1$, where a one denotes a split between adjacent subband filters and a run of m zeros denotes that $m + 1$ adjacent subband filters are merged. As shown in [14], the information rate for such sequences is close to 1 bit/sample, even if we restrict the maximum number of channels to be merged significantly. Such a decomposition overhead is clearly unacceptable. However, initial coding experiments showed that using simple Huffman coding of the runlengths of ones and zeros already reduces the overhead by a factor 5, resulting in an overhead rate of 0.2 bit/sample.

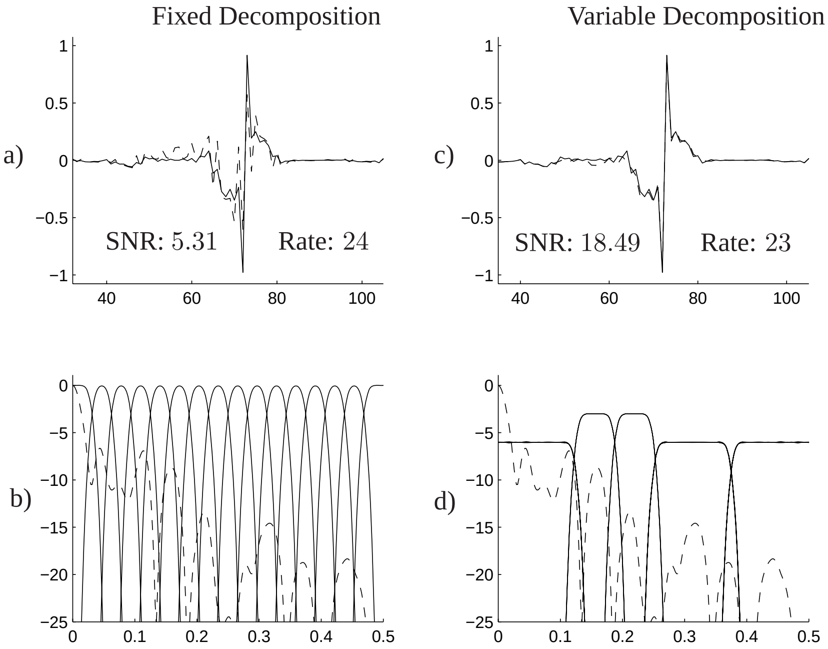


Figure 4.5: *A comparison between fixed uniform and variable nonuniform decomposition. (a)Original (solid) and reconstructed (dashed) signal for uniform decomposition, (b)uniform filter bank and signal magnitude response, (c)original (solid) and reconstructed (dashed) signal for nonuniform decomposition and (d)nonuniform filter bank and signal magnitude response.*

4.4 Experimental results

The flexible frequency decomposition algorithm was implemented in a generic CMFB-based audio codec. The M subband samples were scaled by a single scale factor (the largest absolute sample value). A normalized quantizer was employed, where the quantizer resolution for quantizing the subband signals was varied according to the allocated number of bits.

Figure 4.5 demonstrates the algorithm performance for a 1st-order AR signal with $\rho = 0.9$. The subband samples from a 16-channel filter bank are coded at a target rate R_t of 24 bits using 8 different quantizer resolutions. Clearly, the use of a variable frequency decomposition results in a better modelling of the signal and a higher SNR. In the example given, pre-echos are reduced significantly.

Several audio fragments taken from the SQAM [15] reference disc were coded using the aforementioned coding scheme and compared for both fixed uniform and variable nonuniform decompositions. The filter bank used to obtain the uniform frequency decomposition and applied in the subband merging algorithm was a 512-channel uniform CMFB. The target rate was set to 1.5 bit/sample for both cases, resulting in a decomposition overhead of 0.2 bit/sample.

Table 4.1: *A comparison between fixed uniform decomposition and variable nonuniform decomposition. Average segmental SNRs are presented. The first column shows the results for a fixed uniform decomposition coded at* 1.5 *bit/sample. The second contains the SNRs for variable decompositions coded at* 1.5 *bit/sample, while the last column presents the result for a fixed uniform decomposition coded at* 1.7 *bit/sample.*

Fragment	**Fixed (**1.5**)**	**Variable (**1.5**)**	**Fixed** (1.7)
Castanets	11.7	17.7	13.3
Suzan Vega	19.4	22.8	21.6
German Male	24.8	27.7	27.6

Table 4.1 shows the resulting average segmental SNRs for three cases. The second column shows the SNR for the uniform decomposition case, while the third column presents the SNR for the nonuniform decomposition, where we did not include the overhead rate.

Clearly, a significant improvement in SNR is obtained for all fragments. To compare these result to the case where we could spent an extra 0.2 bit/sample for the fixed uniform decomposition, the last column shows the SNRs. It is clear that for some fragments (e.g. German Male Speech) a further reduction of the overhead rate is necessary.

4.5 Concluding remarks

A new algorithm for rate-distortion optimal frequency decompositions using cosine-modulated filter banks was proposed. The flexible frequency decomposition algorithm jointly optimizes the filter bank structure and the bit allocation over the subband channels. The decomposition overhead was reduced by a simple entropy coder. Experimental results for both synthetic and real audio signals showed that the new algorithm outperforms a fixed uniform frequency decomposition.

The new algorithm is currently being compared to the existing algorithms. Further reduction of the decomposition overhead is necessary to ensure an increase of SNR for all audio signals. Moreover, the incorporation of a perceptual distortion metric that considers both frequency and temporal masking is planned to employ the algorithm in a perceptual audio coder. The flexible frequency decomposition algorithm can then easily be combined with the window-switching technique to increase the adaptive nature of the time-frequency analysis.

Bibliography

[1] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard. Flexible time segmentations for time-varying wavelet packets. In *Proceedings of the IEEE-SP Conference on Time-Frequency and Time-Scale Analysis*, pages 9–12, Philadelphia, USA, October 1994.

[2] R.D. Koilpillai and P.P. Vaidyanathan. Cosine-modulated FIR filter banks satisfying perfect reconstruction. *IEEE Transactions On Signal Processing*, 40(4):770–783, April 1992.

[3] J. Princen and J.D. Johnston. Audio coding with signal adaptive filter banks. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'95)*, pages 3071–3074, Detroit, USA, May 1995.

[4] R.R.Coifman, Y.Meyer, and M.V.Wickerhauser. Wavelet analysis and signal processing. In *Wavelets and their applications*, pages 153–178. Jones and Bartlett, Boston, MA, 1992.

[5] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard. Flexible tree-structured signal expansions using time-varying wavelet packets. *IEEE Transactions On Signal Processing*, 45(2):333–345, February 1997.

[6] M. Purat and P. Noll. A new orthonormal wavelet packet decomposition for audio coding using frequency-varying modulated lapped transforms. In *Proceedings of the 1995 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'95)*, pages 183–186, New York, New York, October 1995.

[7] O.A. Niamut and R. Heusdens. Subband merging in cosine-modulated filter banks. *IEEE Signal Processing Letters*, 10(4):111–114, April 2003.

[8] B. Edler. Codierung von audiosignalen mit uberlappender transformation und adaptiven fensterfunktionen (in german). *Frequenz*, 43(9):252–256, 1989.

[9] S. Shlien. The modulated lapped transform, its time-varying forms, and its applications to audio coding standards. *IEEE Transactions on Speech and Audio Processing*, 5(5):359–366, July 1997.

[10] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions On Image Processing*, 2(2):160–175, April 1993.

[11] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli. Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms. *IEEE Transactions On Signal Processing*, 41(12):3341–3359, December 1993.

[12] M. Purat and P. Noll. Audio coding with a dynamic wavelet packet decomposition based on frequency-varying modulated lapped transforms. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'96)*, pages 1021–1024, Atlanta, USA, May 1996.

[13] D.P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[14] R. Heusdens and S. van de Par. Rate-distortion optimal sinusoidal modeling of audio and speech using psychoacoustical matching pursuits. In *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'02)*, pages 1809–1812, Orlando, USA, May 2002.

[15] Sound quality assessment material recordings for subjective tests. Technical Centre of the European Broadcasting Union, April 1988.

Chapter 5

Rate-Distortion Optimal Audio Coding With Signal Adaptive MDCT

@ 2004 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE.

This chapter has been published as "Rate-Distortion Optimal Audio Coding with Signal Adaptive MDCT", by O.A. Niamut and R. Heusdens in the Proceedings of the *IEEE Fourth Benelux Signal Processing Symposium*, April 2004.

Abstract

In this paper, we present a perceptual audio coder based on a signal adaptive MDCT. A rate-distortion optimization framework is employed to obtain a frequency-varying nonuniform MDCT. The applied algorithms for subband merging and flexible frequency decomposition are explained and several examples are given. Listening tests show that the use of an optimized nonuniform MDCT can outperform the standard uniform MDCT.

5.1 Introduction

Perceptual audio coding has emerged as the standard solution to meet the demand for high-quality digital audio delivery at low bitrates. In recent years, a multitude of audio codecs have seen the light in both scientific and commercial applications. However, the field of perceptual audio coding still offers many scientific challenges.

A key element of all perceptual audio coder is the time-frequency analysis. In many current audio coding standards, a filterbank or signal transformation is used to perform this operation. The filterbank function is threefold [1]. First, the filterbank generates a set of parameters that is amenable to quantization in accordance with a perceptual distortion metric. Furthermore, it provides information about the distribution of the signal and masking power over the time-frequency plane in order to identify perceptual irrelevancies. Additionally, a filterbank is used to reduce statistical redundancies.

It has been recognized by various researchers [2, 3] that the ideal audio coder should make adaptive decisions regarding the optimal time-frequency decomposition. Therefore, the analysis filterbank should have time-varying resolutions both in time and frequency domains. In [1], a number of filterbank characteristics that are highly desirable for audio coding is listed:

- signal adaptive time-frequency tiling
- high-frequency resolution mode
- low-frequency resolution *critical band* mode
- efficient resolution switching
- minimum blocking artifacts
- good channel separation
- strong stopband attenuation
- perfect reconstruction
- critical sampling
- availability of fast algorithms

Several solutions have been proposed to meet these requirements. The work in [2] explains the need for adaptive resolution filterbanks in more detail and proposes a hybrid filterbank solution. An interesting approach is taken in [3], where tree-structured filterbanks are designed by optimizing a cost function that is a combination of both coding distortion and bit rate.

The rate-distortion optimization framework is a promising one, but tree-structured filterbanks are not used frequently in audio coding. In most of the current audio coding standards [4], the modified discrete cosine transform (MDCT) is employed. It satisfies many of the desired filterbank requirements and several techniques for adjusting the time-frequency resolution already exist.

In this paper, we propose a perceptual audio coding application where the MDCT is employed in a rate-distortion optimization framework to obtain frequency-varying filterbanks. We start in Section 5.2 with introducing the MDCT and describe its properties. Next, in Section 5.3, the subband merging method is explained. In Section 5.4 we show that the combination of subband merging and rate-distortion optimization leads to the flexible frequency decomposition algorithm. Experimental results are given in Section 5.5 and we draw some conclusions in Section 5.6.

5.2 The MDCT

The modified discrete cosine transform (MDCT) [5] stems from the family of perfect reconstructing cosine-modulated filterbanks (CMFB) and is an overlapped block transform, i.e. samples from consecutive blocks are windowed and transformed. In the case of the MDCT, the support of the analysis window spans 2 blocks. This greatly reduces blocking artifacts, which are heard as periodic clicks in audio coding. The direct MDCT is defined as

$$X(k) = \sum_{n=0}^{2M-1} x(n) p_{n,k}, \qquad k = 0, 1, \ldots, M–1,$$

where

$$p_{n,k} = h(n)\sqrt{\frac{2}{M}} \cos\left[\frac{(2n+M+1)(2k+1)\pi}{4M}\right] \tag{5.1}$$

are the MDCT basis functions and h is the MDCT window (the transform window is equal to the time-reversed impulse response of the prototype filter of a CMFB). To reconstruct the signal, the inverse transform results of both the current and the previous block are used in an overlap-add procedure, i.e.

$$x(n) = \sum_{n=0}^{M-1} [X(k)p_{n,k} + X^P(k)p_{n+M,k}],$$

where X^P denotes the transform of the previous block.

In the absence of quantization, the perfect reconstruction (PR) of x depends on h satisfying the linear phase and Nyquist constraints, that is,

$$h(2M-1-n) = h(n),$$

and

$$h^2(n) + h^2(n+M) = 1,$$

for $n = 0, 1, \ldots, M - 1$. The window design is a trade-off between satisfying the PR requirements and achieving a good coding performance when the transform coefficients are quantized. An often used window is the sine window, defined as

$$h(n) = \sin\Big[(n + \frac{1}{2})(\frac{\pi}{2M})\Big],$$

but other windows are also available, such as the Kaiser-Bessel derived window.

The MDCT can be used as a time-varying filterbank by employing the window-switching technique. A simple approach would be to use boundary windows (asymmetric windows with a rectangular tail) to start and stop a particular MDCT sequence. When using a fixed window overlap, the transition to a different resolution can be made instantaneously. Although these methods ensure the PR property of the MDCT, they effectively remove the advantage of using 50% overlapping windows. A better solution is to use special transition windows. In the MDCT case, it turns out that these transition windows can be easily derived from the standard MDCT windows [6]. The MDCT can be implemented by fast FFT-based algorithms, some of which incorporate the windowing operation.

Although window-switching can be employed to change the time resolution (and inherently, the frequency resolution), the frequency decomposition remains uniform. A flexible method to achieve adaptive frequency decompositions is provided by the subband merging algorithm.

5.3 Subband merging

Subband merging as described in [7] is a post-processing method to obtain frequency-varying CMFBs. By taking linear combinations of the constituent subband filters of a uniform CMFB with high frequency resolution, the time resolution can be increased locally at the expense of a larger bandwidth. This results in a nonuniform filter bank, i.e. a filterbank where the subband filters have different bandwidths. The subband merging algorithm is devised such that the operation of merging filters is invertible, thereby retaining the PR property of the underlying uniform CMFB. Furthermore, it has been investigated in what manner the filters have to be merged such that the combined filters have a flat passband response. If the individual subband filters provide high stopband reduction and the underlying uniform CMFB satisfies some design constraints, subband merging results in nonuniform filterbanks with suitable frequency responses. Unlike other methods to design nonuniform filterbanks, the number of filterbank channels is not reduced. In Fig. 5.1 it is shown how subband merging trades time resolution for frequency bandwidth. From the lower plots (Fig. 5.1c and d), it can be observed that the merged filters have increased time localization, centered at different positions, and an increased bandwidth, centered around the same center frequency.

The subband merging algorithm can be implemented efficiently as a matrix post-multiplication, where the multiplication matrix is a real-coefficient block-diagonal unitary matrix. Consider the example where we want to combine 2 MDCT basis functions starting from an arbitrary basis function index k. Let M denote the number

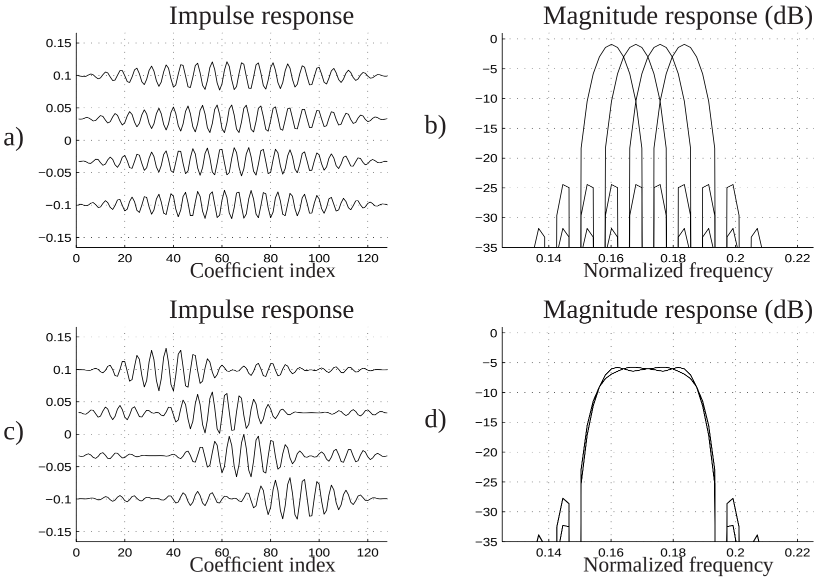


Figure 5.1: *Subband merging trades frequency bandwidth for time resolution. (a) and (b) show the impulse and magnitude responses of* 4 *filters of a* 64*-channel uniform MDCT, (c) and (d) show the impulse and magnitude responses after subband merging.*

MDCT basis functions (or filterbank channels) and $\mathbf{P}$ denote the $M \times 2M$ matrix consisting of the coefficients $p_{n,k}$ in Eq. 5.1. Then we can create a new matrix $\mathbf{P}'$ by the matrix multiplication $\mathbf{P}' = \mathbf{SP}$, where

$$\mathbf{S} = \frac{1}{2}\sqrt{2} \quad \begin{pmatrix} 1 & & & & \\ & \ddots & & & \varnothing \\ & & \begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix} & & \\ & \varnothing & & \ddots & \\ & & & & 1 \end{pmatrix} \in \mathbb{C}^{M \times M}.$$

Since the matrix multiplication is a post-processing operation on the impulse responses of the underlying uniform CMFB, subband merging decouples the design of the uniform CMFB and the adaptation to the signal characteristics. This property will greatly reduce complexity when subband merging is applied in an audio coding situation.

5.4 Flexible frequency decomposition

Subband merging provides a large library of frequency decompositions. To select the best filterbank structure from this library in an audio coding environment, we apply the flexible frequency decomposition (FFD) algorithm [8]. Similar to the work in [3, 9] a

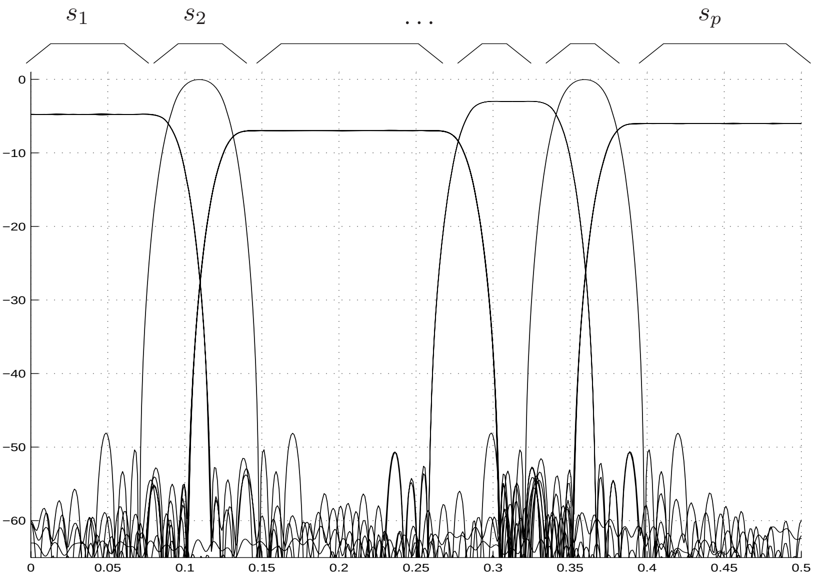


Figure 5.2: *A decomposition* $S_n^{(M)}$ *is a collection of adjacent (nonuniform) subband channels* $s_1, \ldots, s_p$.

rate-distortion optimization framework is employed where the quantization distortion is minimized subject to some target coding entropy, i.e.

$$\begin{aligned} &\min_{\mathcal{Q}} \min_{\mathcal{S}} D(S_n^{(M)}, Q_n) \\ &\text{subject to } H(S_n^{(M)}, Q_n) \leq H_t. \end{aligned} \tag{5.2}$$

In Eq. 5.2, $\mathcal{S} = \{S_1^{(M)}, \ldots, S_{2^{M-1}}^{(M)}\}$ denotes the set of all possible frequency decompositions, where $S_n^{(M)} = \{s_1, \ldots, s_p\}$ is a collection of adjacent (nonuniform) frequency intervals, as shown in Fig. 5.2. Furthermore, by $\mathcal{Q} = \{Q_1, \ldots, Q_N\}$ we denote a set of coding templates, i.e. the set of all possible ways of quantizing the transform coefficients in a particular decomposition $S_n^{(M)}$.

If we define a Lagrangian cost function as $J(\lambda) = D + \lambda H$ with Lagrange multiplier $\lambda \geq 0$, the constrained optimization problem becomes an unconstrained optimization problem, expressed as

$$\max_{\lambda \geq 0} \left(\min_{\mathcal{Q}} \min_{\mathcal{S}} \Big(\sum_{i=1}^{p} \min J_i[\lambda, s_i, Q_n(s_i)] \Big) - \lambda H_t \right). \tag{5.3}$$

Dynamic programming techniques can be applied to reduce computational complexity and avoid an exhaustive search. Further reductions of the computational complexity can be obtained by restricting the maximum number of merged filters.

The solution to Eq. 5.3 is obtained in a 3-step procedure. Although this procedure

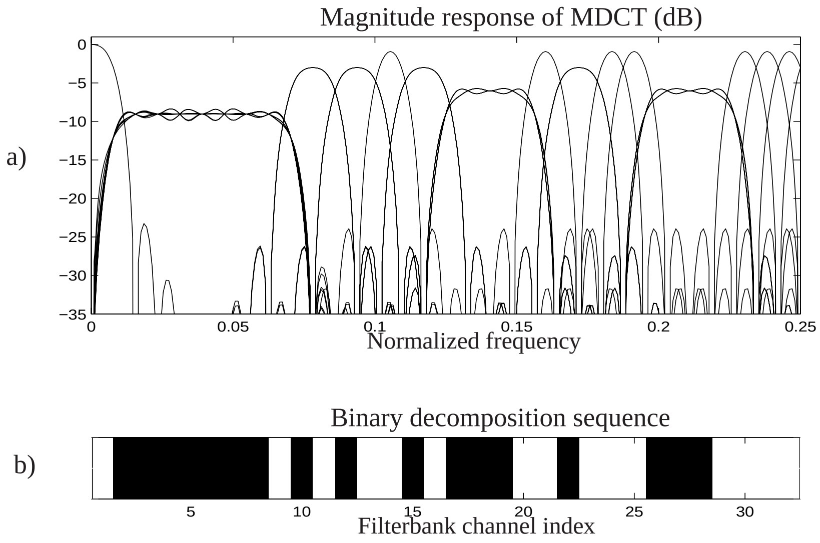


Figure 5.3: *Example of an optimized nonuniform MDCT and its representation by a binary sequence. (a) shows the magnitude response of* 32 *channels of a* 64*-channel MDCT, and (b) shows the filterbank structure expressed as a binary sequence. The white parts denote splits and the black parts denote merged filters.*

is described for a single set of M transform coefficients, it can be easily extended to the case where a target entropy is specified for coding multiple blocks.

---

[1] **Initialization** Start by computing, for $k \in \{1, 2, \ldots, M\}$, every possible decomposition $S_n^{(k)}$ of the frequency interval $[0, \frac{\pi}{M}k)$, and for every decomposition $S_n^{(M)}$, code all M transform coefficients with all possible coding templates from the set $\mathcal{Q}$ and record the resulting distortions and coding entropies.

[2] **Phase I** For an initial value of λ and the first coding template Q_1, find the optimal decomposition $S_*^{(i)}$ of the interval $[0, \frac{\pi}{M}i)$, resulting in the minimum cost $J_*^{(i)}$, for $i = 1, \ldots, M$, where $J_*^{(0)} = 0$. Repeat for all other coding templates and find the coding template Q_* and corresponding decomposition $S_*^{(M)}$ that result in the minimal Lagrangian cost $J_*^{(M)}$.

[3] **Phase II** Repeat Phase I until the optimal value of λ, that is, the one that corresponds to the target coding entropy H_t, is found.

Since the decoder has to be informed about the selected filterbank structure, side information that describes a particular frequency decomposition has to be sent together with the encoded transform coefficients. For an M-channel MDCT, a binary

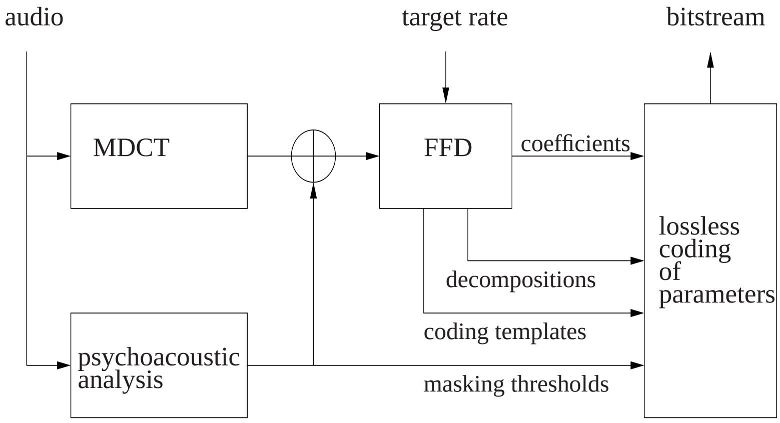


Figure 5.4: *Blockscheme of the proposed perceptual audio coding system.*

sequence of length $M-1$ describes this structure accurately, where a one denotes a split between adjacent subband filters and a run of m zeros denotes that $m+1$ adjacent subband filters are merged. Fig. 5.3 shows an example of a nonuniform MDCT-structure, obtained by the FFD algorithm and its representation by a binary sequence. To minimize the side information rate for sending these sequences, they can be coded using a combination of runlength and Huffman coding.

5.5 Experimental results

We have implemented the FFD algorithm in a perceptual audio coding system. Fig. 5.4 shows a schematic representation of this coding system. The analysis of the audio signal starts with an MDCT operation on overlapping signal blocks. The perceptual model described in [10] is taken to compute masking thresholds for each of the analysis blocks. To incorporate this psycho-acoustic model in the system, for each analysis block the transform coefficients are divided by the masking thresholds. Perceptually weighted distortions are then computed when the coefficients are quantized.

In the following block the FFD algorithm is applied on the perceptually weighted transform coefficients. First, all possible frequency decompositions are computed using subband merging. The resulting weighted and merged coefficients are coded with a scalar uniform quantizer, where the stepsize of the quantizer can be seen as a coding template. The quantized coefficients are replaced by codewords from pre-computed Huffman codebooks and the codeword lengths are taken as the coding entropies. The optimal frequency decompositions and corresponding coding templates are then computed. The bitstream holds the codewords for the quantized coefficients and information describing the masking thresholds, optimal decompositions and coding templates.

Several monophonic audio fragments (16 bits PCM, sampled at 48 kHz), were encoded at a target entropy of 55 kbps with the FFD-based coding system. The same set of fragments was coded at 60 kbps, without subband merging but optimal coding

Table 5.1: *Results of the listening test. Method A denotes uniform MDCT, method B denotes the optimized nonuniform MDCT. The general preference is for method B*

fragment	**preference A (in %)**	**preference B (in %)**
castanets	6.7	93.3
jazz	16.7	83.3
bass guitar	30	70
pop	16.7	83.3

templates were selected. The difference of 5 kbps can be used to encode the decomposition sequences in the case of the FFD algorithm. An informal listening test was performed where the listeners had to choose between the 2 encoded versions of each fragment. The results can be found in Table 5.1. It can be seen the the general preference of the listeners was for method B, i.e. the coding system that employs the subband merging and FFD algorithms.

5.6 Concluding remarks

A perceptual audio coding system has been presented that operates in a rate-distortion optimization framework. The standard MDCT is combined with algorithms for subband merging and flexible frequency decomposition to obtain a signal-adaptive and frequency-varying MDCT. Subband merging can be employed to adapt the filterbank structure to the input signal characteristics. Flexible frequency decomposition applies subband merging and rate-distortion optimization to obtain a nonuniform MDCT that is optimal for coding applications. Listening tests show that the new coding system can outperform a non-optimized uniform MDCT.

Further research will concentrate on efficient coding methods for the side information, i.e. the binary decomposition sequences and the masking thresholds. Suboptimal, greedy solutions that reduce computational complexity are currently investigated.

5.7 Acknowledgement

The authors would like to thank Huib Lincklaen Arriëns for his assistance with the software to obtain masking curves.

Bibliography

[1] T. Painter and A. Spanias. Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(5):451–515, April 2000.

[2] J. Princen and J.D. Johnston. Audio coding with signal adaptive filter banks. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'95)*, pages 3071–3074, Detroit, USA, May 1995.

[3] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions On Image Processing*, 2(2):160–175, April 1993.

[4] M. Bosi and R.E. Goldberg. *Introduction to digital audio coding standards*. Kluwer Academic Publishers, Norwell, MA, 2003.

[5] S. Shlien. The modulated lapped transform, its time-varying forms, and its applications to audio coding standards. *IEEE Transactions on Speech and Audio Processing*, 5(5):359–366, July 1997.

[6] B. Edler. Codierung von audiosignalen mit uberlappender transformation und adaptiven fensterfunktionen (in german). *Frequenz*, 43(9):252–256, 1989.

[7] O.A. Niamut and R. Heusdens. Subband merging in cosine-modulated filter banks. *IEEE Signal Processing Letters*, 10(4):111–114, April 2003.

[8] O.A. Niamut and R. Heusdens. Flexible frequency decompositions for cosine-modulated filter banks. In *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, pages 449–452, Honk Kong, Honk Kong, April 2003.

[9] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard. Flexible tree-structured signal expansions using time-varying wavelet packets. *IEEE Transactions On Signal Processing*, 45(2):333–345, February 1997.

[10] S. van de Par, A. Kohlrausch, G. Charestan, and R. Heusdens. A new psychoacoustical masking model for audio coding applications. In *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'02)*, pages 1805–1808, Orlando, USA, May 2002.

Chapter 6

Time Segmentation Algorithms For The Time-Varying MDCT

@ 2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE.

This chapter has been published as “Time Segmentation Algorithms for the Time-Varying MDCT”, by O.A. Niamut and R. Heusdens in the Proceedings of the *IEEE First Benelux/DSP Valley Signal Processing Symposium*, April 2005.

Abstract

In this paper, several time segmentation algorithms for the time-varying MDCT are discussed and compared. A time-varying MDCT is employed in an audio coding system. Time segmentation optimization procedures based on fast tree pruning and dynamic programming are investigated. MDCT windows having fixed and variable overlapping tails are considered and both entropy and rate-distortion based cost functions are applied. Experimental results in the form of SNR curves are presented. The obtained results show a clear trade-off between performance and computational complexity over a large range of bit rates, with a performance gap of 3 dB between low and high-complexity systems.

6.1 Introduction

One of the fundamental operations of an audio coding system is the time-frequency analysis, for which typically a filterbank or linear signal transformation is applied. Ideally, a signal transform that has time-varying resolutions both in time and frequency domains is required, such that it can be applied to construct arbitrary time-frequency tilings to cover the signal energy in an optimal manner.

A related problem is the construction of a time segmentation of an input signal that is optimal with respect to a specified cost measure. Although an exhaustive search procedure can provide the desired results, the computational complexity associated with the solution grows exponentially with the signal length.

A variety of time segmentation algorithms exists to solve the problem in polynomial time [1, 2, 3, 4, 5]. However, most of these algorithms were developed for wavelet packets, whereas in many audio coding applications an MDCT [6] is applied. The MDCT has desirable properties, such as good channel separation, strong stopband attenuation, minimum blocking artifacts, efficient resolution switching and the availability of fast algorithms.

In this paper, several time segmentation algorithms for the time-varying MDCT, similar to those for wavelet packets, are discussed and compared. In Section 6.2 the time-varying MDCT and transition window designs are described. In Section 6.3, time segmentation optimization algorithms based on fast tree pruning and dynamic programming are investigated and both entropy and rate-distortion based cost functions are introduced. Experimental results for encoding single and multiple audio fragments with an MDCT-based audio coding application are given in Section 6.4.

6.2 The time-varying MDCT

The modified discrete cosine transform (MDCT) [6], stemming from the family of cosine-modulated filter banks, is an overlapped block transform, i.e. a transform where samples from consecutive overlapping blocks are windowed and transformed. The support of the analysis window is two blocks. From a segment of length $2M$, a set of M transform coefficients $X(k)$ is computed by applying the direct MDCT,

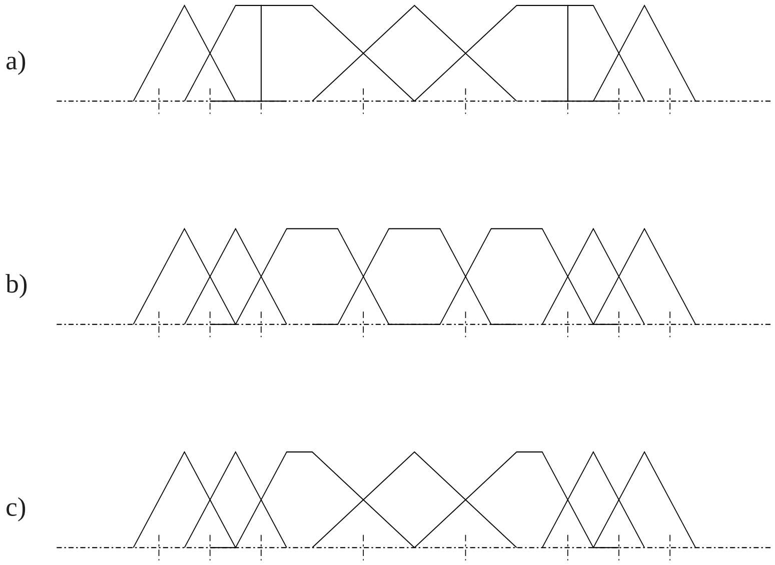


Figure 6.1: *Window-switching schemes with a) boundary windows, b) fixed overlap windows and c) variable overlap windows.*

which is defined as

$$X(k) = \sum_{n=0}^{2M-1} x(n) p_{n,k}, \qquad k = 0, 1, \ldots, M\text{–}1, \tag{6.1}$$

where

$$p_{n,k} = h(n)\sqrt{\frac{2}{M}}\cos\Big[\frac{(2n+M+1)(2k+1)\pi}{4M}\Big],$$

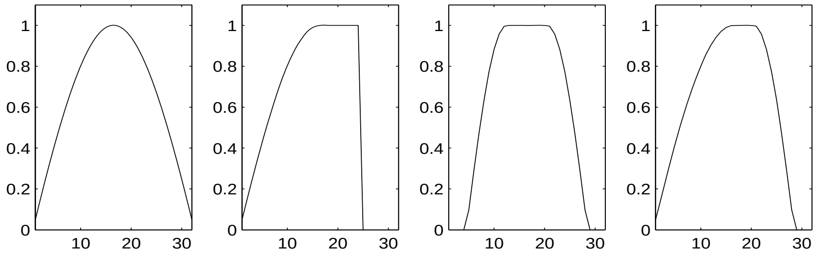
are the M basis functions and h is a prototype window.

The MDCT can be applied as a time-varying transform when window-switching is employed [7, 8]. In this case, the length of the window or, equivalently, the number of transform coefficients, is varied over time to account for nonstationarity of the signal and to prevent pre-echos. In order to retain the perfect reconstruction (PR) property of the MDCT, special transition windows are required at transition boundaries. Fig. 6.1 displays 3 time segmentation possibilities incorporating transition windows, whereas Fig. 6.2 shows time- and frequency-domain properties of the corresponding transition windows. It can be seen that, in general, variable overlap windows have better frequency domain properties, e.g. stopband reduction, than fixed overlap windows.

6.3 Time segmentation algorithms

Given a signal x that is divided into N non-overlapping frames of F samples, a time segmentation of this signal is a collection of p adjacent time intervals or seg-

a) Impulse responses



b) Magnitude responses

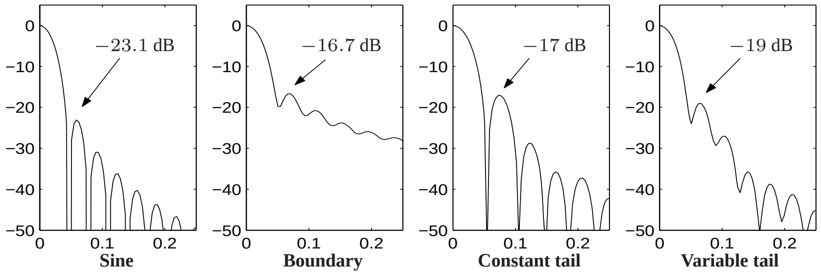


Figure 6.2: *a) Impulse and b) magnitude responses for an MDCT sine window, and some derived transition windows such as the boundary window, the fixed overlap window and the variable overlap window. The stopband reduction of the first sidelobe is indicated by an arrow.*

ments, where each segment is constructed by combining an integer number of adjacent frames. Therefore, the minimal segment length is equal to the framesize F, whereas a maximum segment length of NF is considered, i.e. a segment that comprises the complete signal x.

Let T_N denote such a time segmentation, where T_N is taken from a library of possible time segmentations, say $\mathbb{T}$. The problem at hand is to minimize a cost measure J over all possible segmentations in $\mathbb{T}$, i.e.

$$\min_{T_N \in \mathbb{T}} J. \tag{6.2}$$

If it is assumed that the cost measure is additive over p the segments, the problem can be simplified as

$$\min_{T_N \in \mathbb{T}} \sum_{n=1}^{p} J_n, \tag{6.3}$$

that is, the costs can be minimized on a segment-per-segment basis. Note that in practice, segmental costs are computed on a time interval that can be larger than the segment under consideration to account for an overlap-add procedure. Such a time interval is then windowed by a window that overlaps with adjacent segments. In general, segments of the same length can be used with different window shapes. Fig. 6.1c shows an example of such a situation.

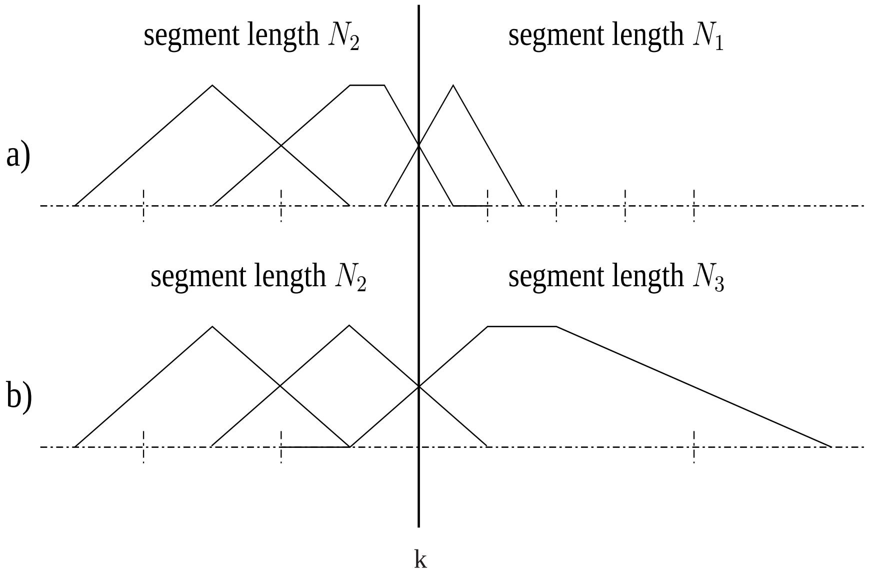


Figure 6.3: *An example where the window tails have to be adapted depending on the optimal choice for the length of the added segment.*

In order to apply the fast algorithms from e.g. [3], the segmental costs may not change during the optimization. More specifically, once the optimal segmentation T_k^* of the signal up to the kth frame and its corresponding minimum cost J_k^* have been found, they may not be influenced by the addition of new segments. In overlap-add systems that employ windows having variable overlap, this constraint is not satisfied, since the mutual overlap of the windows corresponding to two adjacent segments depends of the length of *both* segments. Hence, *a-priori* computation of the costs is therefore not possible. Fig. 6.3 shows two possibilities that occur at transitions between windows with variable tail shapes. It is clear that the windows used when switching from a segment length N_2 to $N_1 < N_2$ (Fig. 6.3a) or $N_3 > N_2$ (Fig. 6.3b) are different, so that there is a clear dependency between J_k^* and J_{k+1}^*.

An existing approach to this dependency problem [2, 5] was implemented in the experiments. The dependency between costs due to window overlap was neglected and an overlap was selected that only depends on the length of the segment under consideration. However, in the subsequent coding stage the segmented signal has to undergo additional windowing operations, such that windows with the correct overlap are applied. An exact solution to this dependency problem has been derived by the authors, but will not be discussed here.

6.3.1 single tree time segmentation

The single tree (ST) algorithm [2, 9] can be employed to search through a library of dyadic time segmentations $\mathbb{T}_{ST}$, i.e. segmentations that result from binary tree

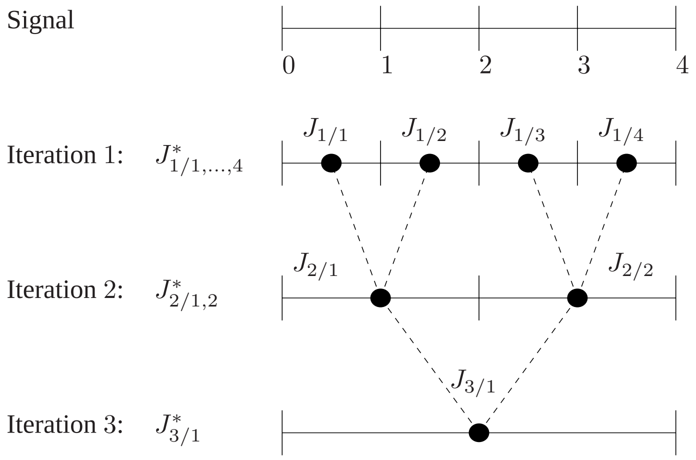


Figure 6.4: *The single tree time segmentation algorithm employs tree pruning to eliminate suboptimal segmentations.*

structures. Each segment can be seen as a node in a binary tree. Starting from a uniform segmentation into N frames, this tree is pruned in the direction of the root, as depicted in Fig. 6.4 for $N = 4$.

Let $J_{i/j}$ be the cost for the jth segment of length 2^{i-1} at tree level i, where the root of the tree has the maximum depth $\log_2(N)+1$. Then, at each iteration $i = 1, \dots, \log_2(N)+1$, the best time segmentation $T^*_{i/j}$ is found by solving

$$J^*_{i/j} = \min(J_{i/j}, J^*_{i-1/2j-1} + J^*_{i-1/2j}),$$

where $J^*_{i/j}$ denotes the minimum cost for the jth segment at tree level i. A limitation of the ST algorithm is its restriction to dyadic segmentations. This can result in segmentations that are inefficient for the given statistics of the signal.

6.3.2 flexible time segmentation

The flexible time segmentation (FT) algorithm [3, 4] searches through a much larger library of possible segmentations $\mathbb{T}_{FT}$. Eq. 6.3 defines a minimization over an additive sum of independent terms, which suggests to use the standard approach of dynamic programming [10], as shown in Fig. 6.5 for $N = 3$.

Let $J_{k,l}$ denote the cost for the time interval $t_{k,l} = [kF, lF - 1]$, i.e. the segment that consists of frames k to l. Then, at each iteration $i = 1, \dots, N$, the best time segmentation T^*_i of the interval $[0, iF - 1]$ is found by solving

$$J^*_i = \min_{0 \le k < i} (J^*_k + J_{k,i}), \tag{6.4}$$

where J^*_i is the minimum cost for the interval $[0, iF - 1]$. The minimizing argument of Eq. 6.4, say k^*_i, given by

$$k^*_i = \arg\min_{0 \le k \le i} (J^*_k + J_{k,i}),$$

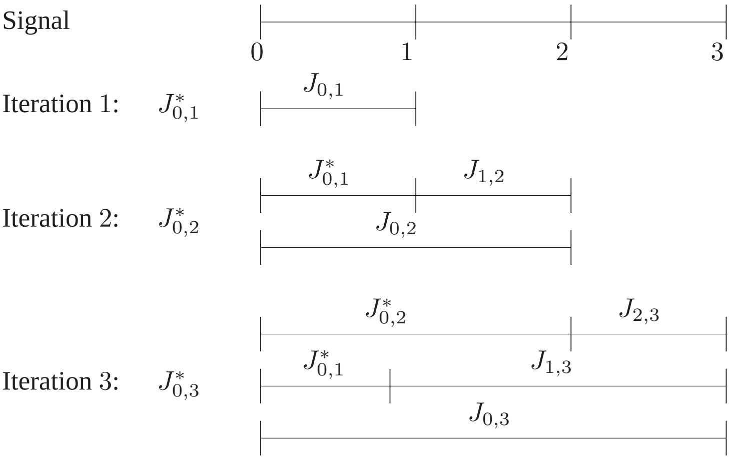


Figure 6.5: *The flexible time segmentation algorithm employs dynamic programming to construct the optimal segmentation.*

is recorded as a split position and determines the optimal segmentation T^*_i. The algorithm terminates once J^*_N has been found and the optimal time segmentation T^*_N can easily be determined by backtracking the optimal split positions.

6.3.3 cost functions

Comparisons were made for two additive cost measures. First, the cost measure from [1], the Coifman-Wickerhauser entropy, was used, which is defined as

$$J_E(\mathbf{x}) = -\sum_n x[n]^2 \log(x[n]^2),$$

where $\mathbf{x}$ denotes the M MDCT coefficients corresponding to a segment. Since the cost function is computed directly on the data samples (in our case, the MDCT transform coefficients), no coding steps are involved to obtain a time segmentation. However, the relation to coding constraints, such as bit rate and distortion, is not straightforward.

Secondly, a rate-distortion (RD) cost measure [9] was taken. It is particularly suited for coding applications since it minimizes distortion for a target bit rate. The coding rate R and quantization distortion D are combined through a Lagrange multiplier $\lambda \geq 0$, i.e.

$$J_{RD}(\mathbf{x}, \lambda) = D(\mathbf{x}) + \lambda R(\mathbf{x}).$$

An additional iterative loop over λ has to be performed to satisfy the rate constraint, see [9] for details. The distortion D must be an additive measure, e.g. squared error distortion. Perceptual aspects can be included by proper perceptual weighting of the individual coefficient distortions. The complexity of the overall algorithm increases, since all transform coefficients have to be quantized, possibly with multiple coding templates (e.g. quantizer stepsizes).

6.4 Experimental results

Both ST and FT algorithms were implemented and combined with a time-varying MDCT. A frame size of 128 time samples was used and at most 8 frames could be combined, i.e. the ST algorithm could choose between windows having lengths $256, 512, 1024$ or 2048, whereas the FT algorithm could select any window length that is a multiple of 256, up to 2048. The segments were transformed by applying Eq. 6.1 to obtain transform coefficients. For quantization of the transform coefficients, a uniform quantizer was taken. The set of coding templates consisted of 9 quantizer stepsizes. Efficient coding of long runs of zero-valued high frequency coefficients was employed. Side information for sending the selected coding template and the segmentation pattern was included.

For the experiments where the entropy-based cost J_E was minimized, an additional RD optimization procedure was performed, where, given the time segmentation that minimized J_E, optimal coding templates were selected for all segments, such that the total distortion was minimized, subject to a target rate. In the experiment where J_{RD} was taken as a cost measure, the selection of optimal coding templates was performed concurrently with the computation of the optimal time segmentation.

Experiments were performed on a total of 9 audio fragments representing various musical genres. The fragments (16 bits, mono, sampling frequency of 48 kHz), representing various musical genres (e.g. jazz, pop, single instruments and speech), were coded at bit rates ranging from 0.5 to 2 bits/sample, for both fixed overlap and variable overlap window types. Additionally, the fragments were coded using a uniform time segmentation with segments of length 1024.

Fig. 6.6 displays the results for minimization of the entropy cost measure, by applying the flexible time segmentation algorithm and windows with variable overlap, for a castanet signal. The upper plot shows the segmentation of the input signal. Clearly, small segments are selected for the signal regions with strong energy variations of a short timespan. The middle plot shows the bit allocation over the segments. As expected, the majority of the available bits goes to short segments containing transients. The lower plot displays the entropy per segment, for the various segment lengths. From this plot, the choice for short segments at the transient positions can be readily explained.

Fig. 6.7 shows composite SNR curves for the various time segmentation algorithms. The upper plot presents the results for fixed overlap windows, whereas the lower plot shows the curves that were obtained with variable overlap windows. As a first observation, the gap between minimization of the entropy cost and the RD cost can be as much as 1.5 dB. However, the increased performance of the RD optimal algorithms comes at a significant increase in complexity. Moreover, the results obtained with variable overlap windows, are on average slightly higher than those obtained with fixed overlap windows.

Furthermore, the difference between single tree and flexible time segmentation appears to be small. For windows with a fixed overlap, a small improvement can be observed when applying flexible time segmentation to minimize the RD cost. This can be explained by inspection of the spectral properties of fixed overlap windows.

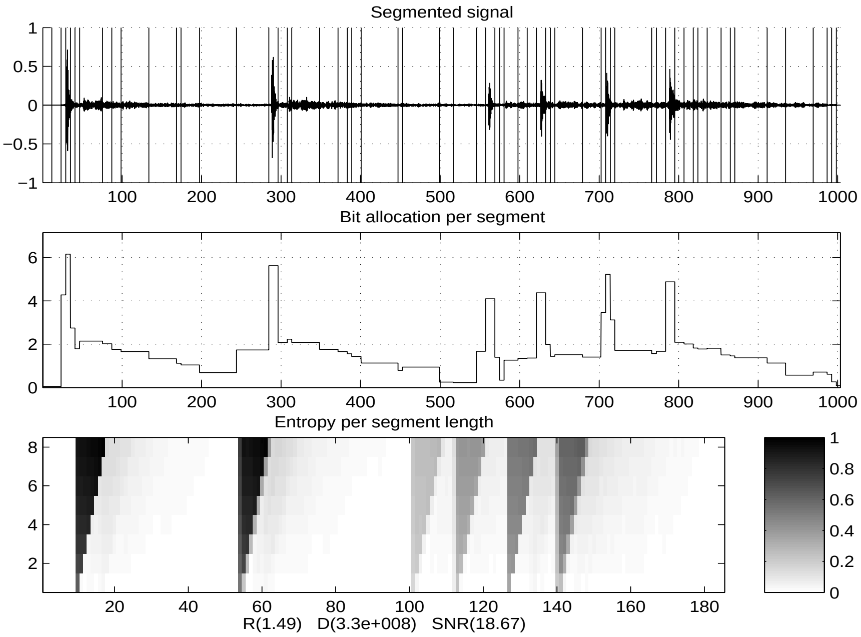


Figure 6.6: *Results for minimum entropy flexible time segmentation of a castanet signal.*

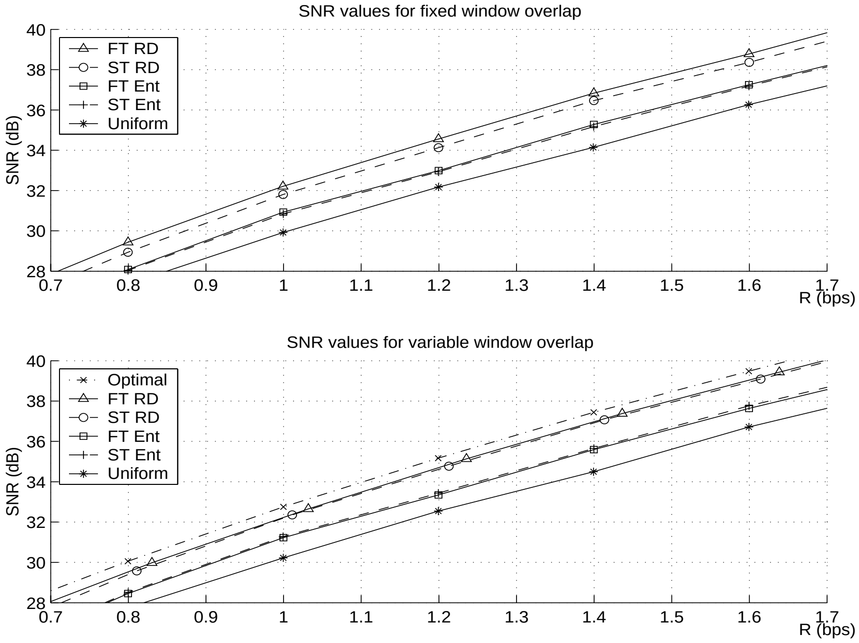


Figure 6.7: *SNR curves for the various time segmentation algorithms, for both fixed and variable window overlap.*

The magnitude of the first side lobe increases with increasing segment lengths, which results in suboptimal frequency localization for long segments and, therefore, a preference for short segments. The single tree algorithm is particularly inefficient when selecting short segments, since the remaining choices for segment lengths are then severely restricted. The flexible time segmentation does not suffer from this problem. For windows with a variable overlap, the differences between the two algorithms are quite small, for both cost measures.

In the case of variable window overlap, an ad-hoc solution to the dependency problem, as noted in Section 6.3, was implemented. An exact solution, that incorporates the window overlap during optimization and runs in polynomial time, is however possible. Experimental results that were obtained for this new algorithm are also presented in the lower plot. An additional average improvement of 0.5 dB can be observed.

The SNR curves from Fig. 6.7 show the complete trade-off between performance and complexity of the experimental MDCT -based audio coding system. On the lower end, a low complexity uniform time segmentation can be applied. On the upper end, a highly complex optimization procedure can be employed, that results in an average SNR improvement of almost 3 dB.

Bibliography

[1] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions On Information Theory*, 38(2):713–718, March 1992.

[2] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli. Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms. *IEEE Transactions On Signal Processing*, 41(12):3341–3359, December 1993.

[3] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard. Flexible time segmentations for time-varying wavelet packets. In *Proceedings of the IEEE-SP Conference on Time-Frequency and Time-Scale Analysis*, pages 9–12, Philadelphia, USA, October 1994.

[4] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard. Flexible tree-structured signal expansions using time-varying wavelet packets. *IEEE Transactions On Signal Processing*, 45(2):333–345, February 1997.

[5] O.A. Niamut and R. Heusdens. RD optimal time segmentations for the time-varying MDCT. In *Proceedings of the 12th European Signal Processing Conference (Eusipco'04)*, pages 1649–1652, Vienna, Austria, September 2004.

[6] J.P. Princen and A.B. Bradley. Analysis/synthesis filter bank design based on time domain aliasing cancellation. *IEEE Transactions On Acoustics, Speech, And Signal Processing*, 34(5):1153–1161, October 1986.

[7] B. Edler. Codierung von audiosignalen mit uberlappender transformation und adaptiven fensterfunktionen (in german). *Frequenz*, 43(9):252–256, 1989.

[8] J. Kovačević and M. Vetterli. Time-varying modulated lapped transforms. In *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*, pages 481–485, Pacific Grove, USA, November 1993.

[9] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions On Image Processing*, 2(2):160–175, April 1993.

[10] D.P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

Chapter 7

Optimal Time Segmentation For Overlap-Add Systems With Variable Amount Of Window Overlap

@ 2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE.

This chapter has been published as “Optimal Time Segmentation for Overlap-Add Systems with Variable Amount of Window Overlap”, by O.A. Niamut and R. Heusdens in the *IEEE Signal Processing Letters*, October 2005.

Abstract

In this paper we propose a new best basis search algorithm for computing the optimal time segmentation of a signal, given a predefined cost measure. The new algorithm solves a problem that arises when the individual signal segments are windowed and overlap-add is applied between adjacent signal segments. When windows having a variable tail shape are employed, the minimization of a cost measure is faced with dependencies between segmental costs due to varying window overlap. A dynamic programming based algorithm is presented that takes into account these dependencies. It computes both the optimal split positions and the optimal amount of window overlap at these split positions in polynomial time. The proposed algorithm gives an upper bound to the achievable performance of existing algorithms. Experimental results for an MDCT-based processing system are presented, both for entropy and rate-distortion cost measures. These results show a performance gain over existing schemes at the cost of an increased computational complexity.

7.1 Introduction

Best basis search algorithms have received quite some attention over the years [1, 2]. A subclass of these algorithms deals with the problem of obtaining a time segmentation of an input signal that is optimal with respect to a specific cost measure. Although a solution based on an exhaustive search solves the problem, its computational complexity grows exponentially with the signal length. Under the assumptions of additivity of the cost measure and independency of the costs over segments, dynamic programming [3] can be employed to solve the segmentation problem in polynomial time. Such conditions are met in e.g. orthogonal transform coding where a rate-distortion cost is minimized [2], if segments are coded independently. Furthermore, in sinusoidal and linear prediction systems [4, 5], that do not strictly conform to these conditions, good results have been reported using dynamic programming based minimization of a rate-distortion cost function.

The segmentation of a signal into non-overlapping segments can result in discontinuity artifacts at the segment edges. To reduce these artifacts, overlap-add techniques can be applied, where overlapping time intervals are multiplied with power-complementary windows in order to retain perfect reconstruction (PR) in the absence of further processing. There are various possibilities for the amount of overlap, e.g. a fixed number of samples (*fixed overlap*) or an overlap that varies with the segment length (*variable overlap*), see Figure 7.1. Windows allowing for variable amount of overlap often provide better spectral resolution. This can be beneficial, e.g. in an audio coding applications a higher coding efficiency is obtained. In overlap-add procedures where fixed overlap windows are employed, costs for each segment can be computed prior to the optimization procedure. However, in the case of variable window overlap, *a-priori* computation of the costs is not possible, as will be discussed in Section 7.2.2.

In this paper, a new dynamic programming algorithm is described that takes into account the amount of overlap, or equivalently, the window tail shape, during the optimization. Experimental results for an audio processing system based on the mod-

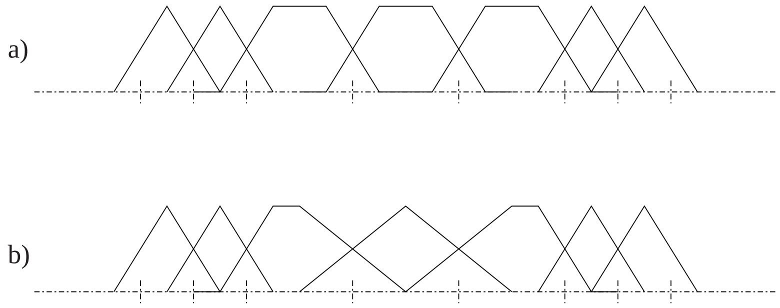


Figure 7.1: *Examples of time segmentation with (a) fixed and (b) variable overlap windows.*

ified discrete cosine transform (MDCT) [6, 7] are presented. Two experiments are performed, one for an entropy cost measure and one for a rate-distortion functional.

7.2 Problem statement

We are given a signal x that is divided into N non-overlapping frames of F samples. A time segmentation of this signal is a collection of adjacent segments that completely spans the signal, where each segment is constructed by an integer number of adjacent frames. Therefore, the minimal segment length is equal to the framesize F, whereas a maximum segment length of NF is considered, i.e. a segment that comprises the complete signal.

Let T_N denote such a time segmentation of the signal, where T_N is taken from a dictionary of possible time segmentations, say $\mathbb{T}$. The problem at hand is to minimize a cost measure J over all possible segmentations in $\mathbb{T}$, i.e.

$$\min_{T_N \in \mathbb{T}} J(T).$$

If it is assumed that the cost measure is additive over the segments and that the costs are computed independently over segments, then the problem can be described as a minimization over an additive sum of independent terms, which suggests to use the standard approach of dynamic programming. This is done by the flexible time segmentation algorithm in [8, 2], which we will briefly discuss.

7.2.1 existing approach

Let $J_{k,l}$ denote a segmental cost for the time interval $t_{k,l} = [kF, lF-1]$, i.e. the segment that consists of frames k to l. Furthermore, let J_k^* be the minimum or optimized cost for the interval $[0, kF-1]$. Then, at each iteration $i = 1, \ldots, N$, the best time segmentation T_i^* of the interval $[0, iF-1]$ is found by solving

$$J_i^* = \min_{0 \le k < i} (J_k^* + J_{k,i}), \tag{7.1}$$

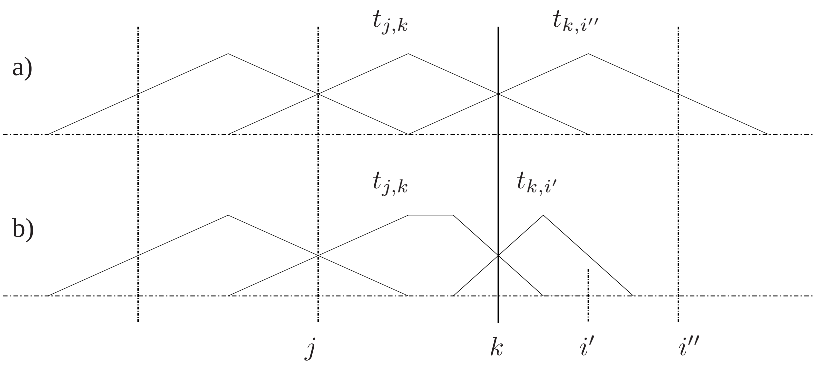


Figure 7.2: *An example where the window tails have to be adapted to retain PR when switching to different segment lengths.*

with $J_0^* = 0$. The minimizing argument of Eq. 7.1, say k_i^*, given by

$$k_i^* = \arg\min_{0 \le k \le i} (J_k^* + J_{k,i}),$$

is referred to as the split position and determines the optimal segmentation T_i^* at iteration i. The algorithm terminates once J_N^* has been found, and the optimal time segmentation T_N^* can easily be determined by backtracking the optimal split positions k_i^*.

7.2.2 suboptimality of the existing approach

When the flexible time segmentation algorithm is applied in combination with windowing and overlap-add, the segmental cost $J_{k,i}$ is computed on a time interval that can be larger than the segment $t_{k,i}$ under consideration. Such a time interval is then windowed by a window that overlaps with adjacent segments. If the window overlap between adjacent segments is given, e.g. by having a fixed window overlap, the existing algorithm from Section 7.2.1 still provides the optimal solution.

However, in general, segments of the same length can be used with different window shapes, since the mutual overlap of the windows corresponding to two adjacent segments depends of the length of *both* segments. Hence, independent computation of costs for the individual segments is no longer possible. Figure 7.2 shows an example of such a situation. It is clear that the windows used for segment $t_{j,k}$ when switching to segment $t_{k,i''}$ (Figure 7.2a) or $t_{k,i'}$ (Figure 7.2b) are different, so that there is a clear dependency between $J_{j,k}$ and $J_{k,i}$. Therefore, one cannot compute the optimal split position k, without knowing the split position i and the optimization problem becomes dependent, as was mentioned in [9].

An existing approach to solve this problem is to neglect the dependency between costs and window overlap and to select an overlap during optimization that only depends on the length of the segment under consideration [9, 10]. However, the cost thus obtained is, in general, not equal to the minimum cost that can be achieved if the overlap is taken into account during optimization. Moreover, the selection of a window overlap that only depends on the length of the segment under consideration results in a non-PR overlap-add system. As a result, in any subsequent processing stage (e.g. coding) the segmented signal has to undergo additional windowing operations, such that windows with the correct overlap are applied.

7.3 Flexible time segmentation for a varying window overlap

A new flexible time segmentation algorithm is proposed that takes into account the dependency between costs for adjacent segments due to varying window overlap. The length of the segment under investigation, and therefore the window length, determines the number of overlap possibilities. For a segment that spans i frames, $i+1$ possible window overlap situations for each of the window tails are considered, i.e. the amount of overlap ranges from 0 to $iF/2$ samples. Clearly, the number of possible window tails is equal to the possible amounts of window overlap.

7.3.1 derivation of the proposed algorithm

From Eq. 7.1 it can be seen that the optimal time segmentation T_N^* of the signal is obtained by iteratively computing the minimum costs J_i^*, where i denotes the end of the ith frame. Therefore, at iteration i, only a single minimum cost J_i has to be computed with standard dynamic programming. Since we allow $i+1$ overlap possibilities between adjacent segments, $i+1$ minimum costs have to be computed with the proposed algorithm, one for every possible window overlap at the end or right side of a time segmentation. This right window overlap at iteration i is denoted by $m = 0, \ldots, i$ and the minimum costs and corresponding time segmentations up to the ith frame for the $i+1$ possible window tails are denoted $J_{i/m}^*$ and $T_{i/m}^*$, respectively.

In Eq. 7.1, a minimization over the split position k is performed to determine the optimal time segmentation. A segmented part of the signal, described by time segmentation T_k^*, is combined with the segment $t_{k,i}$, for all possible values of k. In the proposed algorithm, in addition to the minimization over k, it is also necessary to perform a minimization over the mutual overlap, denoted by n, between the segmented signal up to position k and the added segment $t_{k,i}$ at split position k. There are n_0 possible overlap situations, where n_0 depends on the length $i-k$ of the added segment $t_{k,i}$ and the length k of the previously segmented signal. It follows that $n = 0, \ldots, n_0$ and $n_0 = \min(i-k, k)$, i.e., the minimum of the length of the added segment and the length of the segmented signal part, now described by $T_{k/n}^*$. The window that is used at segment $t_{k,i}$ has a left overlap n and right overlap m. The corresponding cost is denoted $J_{k,i/n,m}$. Figure 7.3 displays the relation between window overlap, time segmentations and the costs they give rise to.

The problem at hand can now be formulated as solving

$$J_{i/m}^* = \min_{0 \leq k \leq i-m} \min_{0 \leq n \leq n_0} (J_{k/n}^* + J_{k,i/n,m}), \tag{7.2}$$

for $m = 0, \ldots, i$ and $i = 1, \ldots, N$, where $J_{0/0}^* = 0$. To compute the optimal

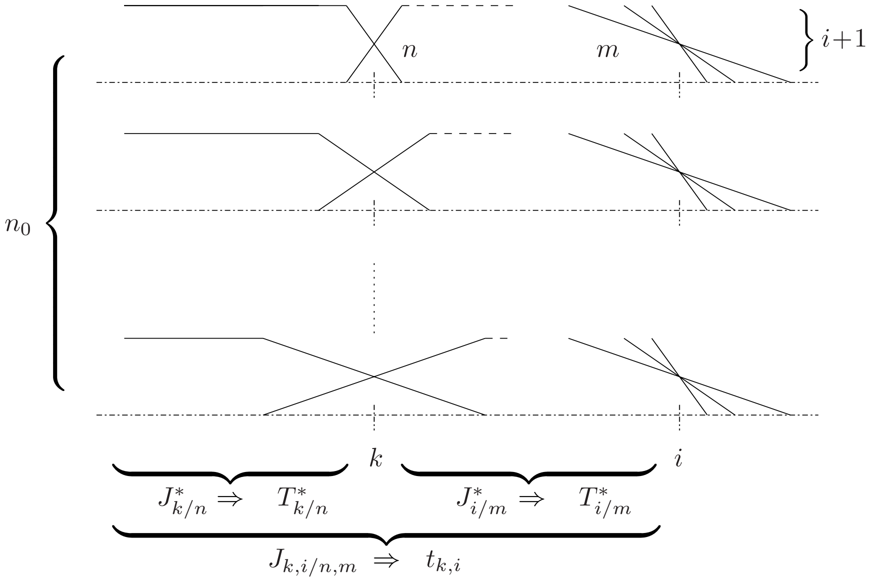

n
m
$i+1$
n_0
k
i
$J^*_{k/n} \Rightarrow T^*_{k/n}$
$J^*_{i/m} \Rightarrow T^*_{i/m}$
$J_{k,i/n,m} \Rightarrow t_{k,i}$

Figure 7.3: *A schematic overview of the various overlap possibilities that are considered during optimization. For segment $t_{k,i}$, both the left overlap n at position k and right overlap m at position i have to be selected.*

cost $J^*_{i/m}$, i.e. the cost for a time segmentation up to the ith frame that ends with a window overlap m, Eq. 7.2 is solved in 2 sequential steps. First, for each split position $0 \leq k \leq i-m$, the optimal overlap between the previously segmented signal up to the kth frame, described by $T^*_{k/n}$, and the segment $t_{k,i}$ is selected. These overlap values, say n^*_{imk}, are found by solving

$$n^*_{imk} = \arg\min_{0 \leq n \leq n_0} (J^*_{k/n} + J_{k,i/n,m}), \tag{7.3}$$

and are stored temporarily. Next, the optimal split position, say k^*_{im}, is obtained by solving

$$k^*_{im} = \arg\min_{0 \leq k \leq i-m} (J^*_{k/n^*_{imk}} + J_{k,i/n^*_{imk},m}). \tag{7.4}$$

The optimal split position k^*_{im} that is thus obtained also determines which of the $i-m+1$ overlap values n^*_{imk} that were computed in Eq. 7.3 is kept for backtracking purposes. This overlap value is denoted n^*_{im}.

From Eq. 7.2– 7.4 we can derive the matrix structure that is maintained in memory to store all the values needed during optimization and for backtracking. As an example, the matrix of optimal costs $\mathbf{J}^* \in \mathbb{R}^{N+1\times N}$ holds all the values $J^*_{i/m}$.

$$\mathbf{J}^* = \begin{pmatrix} J^*_{1/0} & J^*_{2/0} & \cdots & J^*_{N/0} \\ J^*_{1/1} & J^*_{2/1} & \cdots & J^*_{N/1} \\ & J^*_{2/2} & \cdots & J^*_{N/2} \\ & & \ddots & \vdots \\ & & & J^*_{N/N} \end{pmatrix}. \tag{7.5}$$

Similarly, the optimal split positions k^*_{im} are stored in a matrix $\mathbf{K}^*$ and the optimal overlap values n^*_{im} are stored in $\mathbf{N}^*$. Both these matrices have a structure similar to Eq. 7.5.

The algorithm terminates once the minimum costs $J^*_{N/m}$ have been obtained, i.e. the costs for segmenting the complete signal, ending with all possible amounts of overlap. A final minimization over the Nth column of $\mathbf{J}^*$ provides the best window overlap m^*_N at the end of the segmentation, i.e.

$$m^*_N = \underset{0\le m\le N}{\arg\min}(J^*_{N/m}).$$

The optimal time segmentation T^*_N can now be backtracked from the matrices $\mathbf{K}^*$ and $\mathbf{N}^*$.

7.3.2 complexity analysis

The new algorithm searches through a larger dictionary than the standard flexible time segmentation algorithm. This flexibility comes at the cost of an increased complexity, which is analyzed for two separate stages of the algorithm. First we consider the *initialization* stage, where all costs are computed. Since $i{+}1$ possible overlap situations are considered for each of the window tails of a window that corresponds to a segment of i frames, we can construct $(i+1)^2$ different windows. There are $N-i+1$ such segments in a signal of length N. Hence, the total number of computations $C_I(N)$ is given by

$$C_I(N) = \sum_{i=1}^{N} (i+1)^2 (N-i+1) = \frac{N(N+1)(N^2+7N+16)}{12}.$$

Therefore, the complexity for generating the costs for all segments is $\mathcal{O}(N^4)$, as compared to $\mathcal{O}(N^2)$ for the standard algorithm as described in [2]. If we assume that, for a segment of length i, a signal transform with complexity $i\log_2 i$ is applied and that the computation of a segmental cost has a complexity i, the complexity for the initialization stage increases to $\mathcal{O}(N^5\log_2 N)$.

The complexity of the optimization stage is derived from Eq. 7.2. The constraint on n is relaxed such that $n_0 = i - k$. This will results in a small overestimation of the complexity. The number of computations $C_O(N)$ to be performed is now given by

$$C_O(N) = \sum_{i=1}^{N} \sum_{m=0}^{i} \sum_{k=0}^{i-m} (i-k+1) = \frac{N(N+4)(N^2+4N+7)}{12},$$

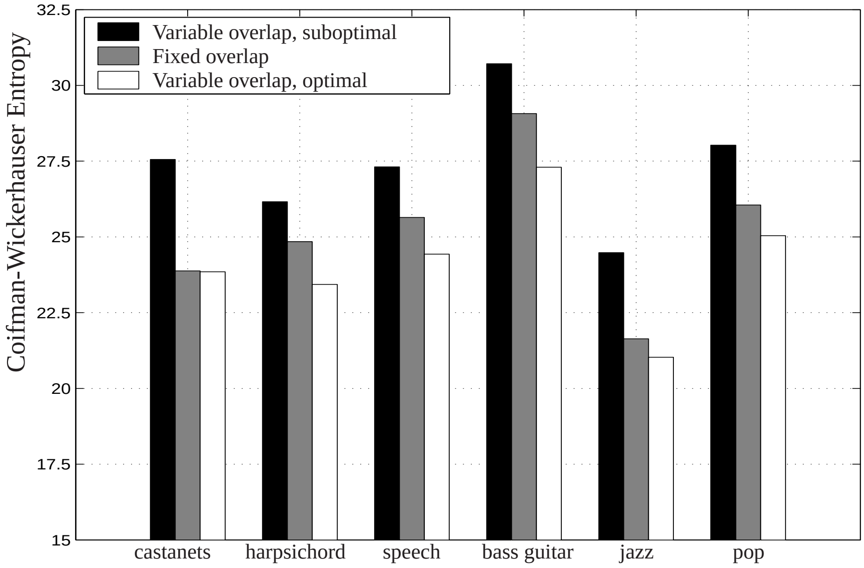


Figure 7.4: *Comparison of the various time segmentation algorithms for the entropy cost measure.*

and the complexity is therefore $\mathcal{O}(N^4)$. Note that the standard dynamic programming algorithm has a complexity $\mathcal{O}(N^2)$, see e.g. [2].

7.4 Experimental results and discussion

Time segmentation algorithms based on both the existing and the new approach were evaluated in an MDCT-based audio processing system. For additional comparisons, an experiment with fixed overlap windows was also performed. A frame size of 128 was used and at most 8 frames could be combined, i.e. the algorithms could select window lengths as an integer multiple of 256 up to 2048. Experiments were performed on a total of 6 audio fragments (16 bits, mono, sampling frequency of 48 kHz) representing various musical genres (e.g. jazz, pop, single instruments and speech). Comparisons were made for two additive cost measures.

First, we used the Coifman-Wickerhauser entropy [1]. Figure 7.4 displays results from an experiment where this entropy cost measure was minimized. It can be observed that the new algorithm always performs better than the existing methods. It gives an average improvement of 12% over the suboptimal variable overlap case and an average improvement of 4% over the fixed overlap case.

Secondly, a rate-distortion (RD) cost measure [11] was used. The MDCT coefficients were quantized by a uniform quantizer with 9 possible quantizer stepsizes. The resulting distortions were summed over all coefficients in a segment. For all quantizer stepsizes, Huffman codebooks were computed. Coding of side information was

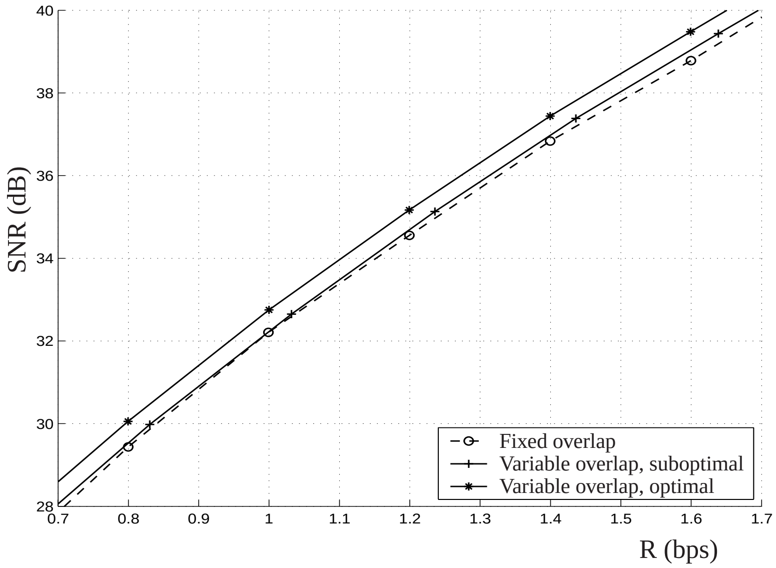


Figure 7.5: *Comparison of the various time segmentation algorithms for the rate-distortion cost measure.*

restricted to the selected stepsizes and, in the case of the new algorithm, the amount of overlap between segments. All fragments were coded at bit rates ranging from 0.6 to 2.0 bits/sample and composite SNR curves were constructed. From Figure 7.5 it is observed that an average gain in SNR of 0.5 dB can be obtained with the new method.

In both experiments, the new algorithm outperforms the existing one. However, the performance gain comes at the cost of increased computational complexity. Therefore, one of the contributions of the proposed algorithm is that it allows us to evaluate the exact loss in performance that occurs when either a fixed overlap is chosen, or when the overlap is neglected in the case of variable overlap windows, without performing an exhaustive search.

Bibliography

[1] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions On Information Theory*, 38(2):713–718, March 1992.

[2] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard. Flexible tree-structured signal expansions using time-varying wavelet packets. *IEEE Transactions On Signal Processing*, 45(2):333–345, February 1997.

[3] D.P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[4] P. Prandoni, M. Goodwin, and M. Vetterli. Optimal time segmentation for signal modeling and compression. In *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, pages 2029–2032, Munich, Germany, April 1997.

[5] P. Prandoni and M. Vetterli. R/d optimal linear prediction. *IEEE Transactions on Speech and Audio Processing*, 8(6):646–655, November 2000.

[6] J.P. Princen and A.B. Bradley. Analysis/synthesis filter bank design based on time domain aliasing cancellation. *IEEE Transactions On Acoustics, Speech, And Signal Processing*, 34(5):1153–1161, October 1986.

[7] H.S. Malvar. *Signal Processing with Lapped Transforms*. Artech House, Boston, MA, 1992.

[8] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard. Flexible time segmentations for time-varying wavelet packets. In *Proceedings of the IEEE-SP Conference on Time-Frequency and Time-Scale Analysis*, pages 9–12, Philadelphia, USA, October 1994.

[9] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli. Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms. *IEEE Transactions On Signal Processing*, 41(12):3341–3359, December 1993.

[10] O.A. Niamut and R. Heusdens. RD optimal time segmentations for the time-varying MDCT. In *Proceedings of the 12th European Signal Processing Conference (Eusipco'04)*, pages 1649–1652, Vienna, Austria, September 2004.

[11] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions On Image Processing*, 2(2):160–175, April 1993.

Chapter 8

Upfront Time Segmentation Methods For Transform Coding Of Audio

This convention paper has been reproduced from the authors advance manuscript, without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

This chapter is a slightly modified version of “Upfront Time Segmentation Methods for Transform Coding of Audio”, by O.A.Niamut, R.Heusdens and H.J.Lincklaen Arriëns in the Proceedings of the *119th AES Convention*, October 2005.

Abstract

We study a transform coder that employs a dynamic programming based rate-distortion optimization framework for time segmentation. Although this coder exhibits a high performance, its computational complexity makes it unfeasible for many practical applications. It is investigated whether upfront time segmentation can reduce computational complexity without a significant decrease in performance. Upfront time segmentation can be accomplished by replacing the rate-distortion cost functional with low-complexity cost measures that are independent of bit rate and perceptual distortion. Through both quantitative and qualitative evaluation it is shown that dynamic programming based upfront time segmentation for minimization of perceptual entropy can be a viable alternative to rate-distortion optimal time segmentation.

8.1 Introduction

Within most perceptual audio coders a time-frequency analysis is performed. Typically, a filterbank or signal transformation is used to perform this operation. We can discern three functions of this signal transform. First, the signal transform generates a set of parameters that is amenable to quantization in accordance with a perceptual distortion metric. Furthermore, it provides information about the distribution of the signal and masking power over the time-frequency plane in order to identify perceptual irrelevancies. Additionally, a signal transformation is used to reduce statistical redundancies.

The dominant signal transform in audio coding systems is the modified discrete cosine transform (MDCT) [1, 2]. It is an overlapped block transform, i.e. an operation is performed where overlapping samples from consecutive blocks are windowed and transformed. In the case of the MDCT, the support of the analysis and synthesis windows is twice the number of basis functions, resulting in a 50% overlapping transform. The overlapping windows greatly reduce blocking artifacts, which are heard as periodic clicks in audio coding. Moreover, the MDCT can be seen as a particular instance from the family of cosine-modulated filterbanks (CMFB) and as such, it is critically sampled and possesses the perfect reconstructing property. The MDCT window is equal to the time-reversed impulse response of the prototype filter of a CMFB and several window design methods exist to achieve high stopband reduction and good frequency selectivity. It is applied in most of the current audio coding standards, such as MP3 [3] and AAC [4].

The MDCT can be used as a time-varying signal transform by changing the analysis window length on a block-by-block basis (window switching). Essentially, this will result in a nonuniform time segmentation of the signal under analysis. At transitional positions between segments of different lengths, special care has to be taken with respect to the jointly overlapping window tails, to retain the perfect reconstruction property of the transform. For most signal transforms, this amounts to a complicated design procedure for transition or boundary windows. In the MDCT case, it turns out that these transition windows can be easily derived from the standard MDCT windows [5, 6].

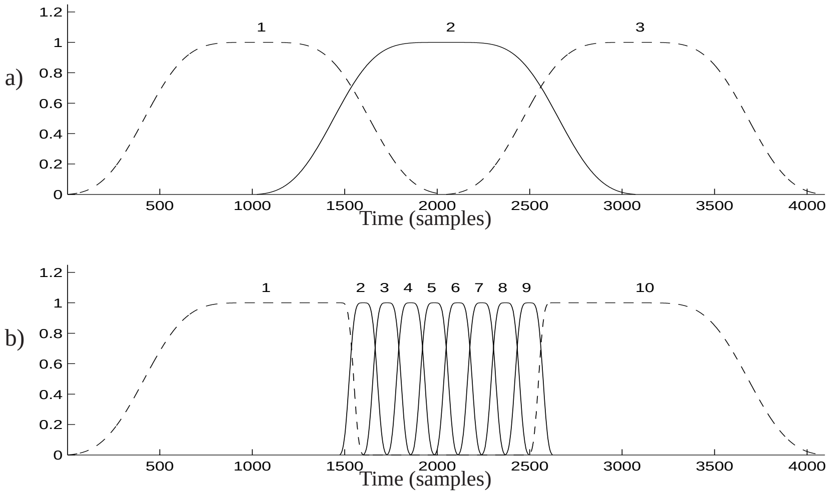


Figure 8.1: *The MPEG-2 AAC block switching process. a) Window sequence during steady state conditions. b) Window sequence during transient conditions.*

In the AAC codec, this solution is implemented to handle transient conditions as follows [7]. During steady state conditions, e.g. for harmonic signals, the MDCT in AAC operates in a so-called *long block* mode. A long transform window of 2048 samples is applied with the maximum amount of overlap. If a transient is detected, the MDCT switches to a *short block* mode where a sequence of eight 256 sample windows is applied. As dictated by the window switching paradigm, both the window preceding this short window sequence and the window directly after it need to be adapted to ensure perfect reconstruction. Figure 8.1 displays the two window sequences. In Figure 8.1a an example of the long block mode is displayed, whereas Figure 8.1b shows an example of switching between long and short block modes.

The detection of a transient region can be done in various manners. In typical AAC implementations perceptual entropy [8] is monitored on long signal blocks. If for a given long block the perceptual entropy rises above a predefined threshold, the switch to short block mode is performed within that long block. Monitoring of energy variations [5, 9] can also be applied. Both the windowing sequence and the transient detection in AAC are rather limited. The window switching scheme allows for a larger variety of window sizes to be employed. Furthermore, the transient detection method based on perceptual entropies from long blocks only can be inaccurate. Setting a threshold on the detection algorithm requires extensive tuning, especially if bit rate scalability is an issue.

An alternative approach is to employ operational rate-distortion (RD) optimization techniques [10], where a Lagrangian cost function of both the (perceptual) distortion between the original and the coded fragment and the bit rate demand is minimized, for a given coding environment. This allows for flexible adaptation to signal characteristics and bit rate scalability. Rate adaptivity of an audio coding system becomes particularly important when considering the emergence of time-varying heterogeneous networks and the convergence of traditional consumer electronics with mobile communications. Additionally, efficient best basis search algorithms can be applied to obtain time-frequency decompositions that correspond well to the signal characteristics. A drawback of these methods is the increased computational complexity. However, once a practical bound on the RD performance has been found, low complexity techniques can be benchmarked against this bound, thereby providing system designers a clear trade-off.

An example of an audio coding environment that includes an operational RD optimization framework is the transform coding system presented in [11, 12]. Within this framework, an optimal time segmentation of the signal under consideration is found by employing the dynamic programming based method in [13], such that the perceptual distortion is minimized subject to a bit rate constraint. In this article, we study the time segmentation algorithm in more detail. Moreover, we investigate the use of low complexity cost measures to replace the Lagrangian RD cost measure.

This article is organized as follows. First, in Section 8.2, operational RD optimization and the dynamic programming based best basis search are discussed. Next, in Section 8.3, we describe the transform coding system that implements an RD optimal time segmentation algorithm. A set of audio fragments is coded with this system and results from a MUSHRA test are presented. Then, in Section 8.4 we discuss a modification of this scheme by performing upfront time segmentation. We present experimental results using the modified system and we draw conclusions in Section 8.5.

8.2 RD optimization

In this section, we give a formal description of the operational RD optimization algorithm for adaptive time segmentation, that uses a Lagrangian combination of coding distortion and bit rate as a cost function together with dynamic programming. Given a signal x, we impose a grid of time resolution N on the signal. That is, the complete signal is divided into N non-overlapping frames of M samples. A segmentation of the signal is a collection of p adjacent segments, where each segment is constructed by combining an integer number of adjacent frames. Therefore, the minimal segment length is equal to the framesize M, whereas a maximum segment length of NM is considered, i.e. a segment that comprises the complete signal.

Furthermore, let S denote a dictionary of possible time segmentations of the signal, provided by the window switching method [5, 6]. Given a particular segmentation $\mathbf{s} \in S$ we can define a coding template $\mathbf{c} \in C$, that describes how the data in each of the segments is quantized and coded. Let the perceptual distortion and bit rate resulting from coding the segmentation $\mathbf{s}$ with coding template $\mathbf{c}$ be denoted as $D_\pi(\mathbf{s}, \mathbf{c})$ and $R(\mathbf{s}, \mathbf{c})$, respectively, and assume that we are given a target rate R_T. Formally stated, the constrained problem that we want to solve is given as

$$\begin{aligned} &\min_{\mathbf{s}\in S}\min_{\mathbf{c}\in C} D_\pi(\mathbf{s}, \mathbf{c}) \\ &\text{subject to} \quad R(\mathbf{s}, \mathbf{c}) \leq R_T. \end{aligned} \tag{8.1}$$

We can convert the constrained problem of Eq. 8.1 into the following unconstrained problem,

$$\begin{aligned}&\min_{\mathbf{s}\in S}\min_{\mathbf{c}\in C} J(\mathbf{s},\mathbf{c},\lambda) = \\ &\min_{\mathbf{s}\in S}\min_{\mathbf{c}\in C} \left(D_\pi(\mathbf{s},\mathbf{c}) + \lambda R(\mathbf{s},\mathbf{c})\right),\end{aligned} \tag{8.2}$$

using a Lagrangian multiplier $\lambda \geq 0$.

Assume that a decomposition $\mathbf{s}$ provides us with $p_{\mathbf{s}}$ segments. If the rates and distortions are additive over these segments and if the segments are coded independently, Eq. 8.2 can be simplified as

$$\begin{aligned}&\min_{\mathbf{s}\in S}\min_{\mathbf{c}\in C} J(\mathbf{s},\mathbf{c},\lambda) = \\ &\min_{\mathbf{s}\in S}\sum_{k=1}^{p_{\mathbf{s}}}\min_{c_k} \left(D_\pi(s_k,c_k) + \lambda R(s_k,c_k)\right).\end{aligned} \tag{8.3}$$

In this case, the selection of the optimal coding template turns out to be a simple minimization operation independently performed for each segment.

Note that the problem stated in Eq. 8.3 is solved for a particular value of λ. The resulting bit rate $R(\mathbf{s},\mathbf{c})$ does not necessarily correspond to the desired target rate R_T. An additional operation is required to obtain the value of λ such that the bit rate satisfies $R(\mathbf{s},\mathbf{c}) \leq R_T$. However, since this is a convex problem in λ, fast algorithms exist to solve the problem, like the bisection algorithm in [10].

The complete problem is now stated by Eq. 8.4 as

$$\max_{\lambda\geq 0}\left[\left(\min_{\mathbf{s}\in S}\sum_{k=1}^{p_{\mathbf{s}}}\min_{c_k\in C_k} J(\lambda,s_k,c_k)\right) - \lambda R_T\right], \tag{8.4}$$

and we apply a step-wise procedure to solve this problem as follows:

Algorithm 8.2.1.

Initialization

[1] Generate (R, D_π) pairs for each coding template and for each possible segment.

Optimization for a given slope λ

[2] For the given slope λ, find the minimum Lagrangian costs for each segment by minimizing over all coding templates.

[3] Use a best basis search algorithm to find the optimal time segmentation.

Computation of the optimal slope λ^*

[4] To find the optimal slope λ^* that corresponds to the target rate R_T, run the bisection algorithm from [10].

Backtracking

[5] Obtain the optimal segmentation $\mathbf{s}^*$, the optimal coding template $\mathbf{c}^*$ and the corresponding coded parameters. The optimal rate R^* and distortion D_π^* can then be computed.

We now focus ourselves on a particular best basis search algorithm for finding the optimal time segmentation, given λ. In general, the best basis search problem can be formulated as finding the orthonormal basis from a given dictionary to represent the signal x such that a particular cost function J is minimized. In the MDCT case, the dictionary is constructed by combining the window-switching technique with the MDCT. We will discuss an often employed search technique based on dynamic programming. An optimal basis for a pre-defined cost J can be obtained with the following method.

8.2.1 dynamic programming best basis search

Eq. 8.3 defines a minimization over an additive sum of independent terms, which suggests to use the standard approach of dynamic programming [14]. The flexible time segmentation algorithm from [15] employs dynamic programming to search through a dictionary of time segmentations.

Let $J_{k,l}$ denote the cost for the segment $s_{k,l} = [kM, lM-1]$, i.e. the segment that consists of frames k to l. Then, at each iteration $i = 1, \ldots, N$, the best segmentation $\mathbf{s}_i^*$ of the time interval $[0, iM-1]$ is found by solving

$$J_i^* = \min_{0 \leq k < i} (J_k^* + J_{k,i}), \tag{8.5}$$

where J_i^* is the minimum cost for the interval $[0, iM-1]$. The minimizing argument of Eq. 8.5, say k_i^*, given by

$$k_i^* = \arg\min_{0 \leq k \leq i} (J_k^* + J_{k,i}),$$

is recorded as a split position and determines the optimal decomposition $\mathbf{s}_i^*$. The algorithm terminates once J_N^* has been found and the optimal decomposition $\mathbf{s}_N^*$ can easily be determined by backtracking the optimal split positions. In Figure 8.2 the iterative process of building up the time segmentation is depicted for $N = 3$.

The selection of the flexible time segmentation algorithm to perform the best basis search has several implications on the computational complexity of algorithm 1. The complexity of the search procedure (step 3 of algorithm 1) is $\mathcal{O}(N^2)$. Furthermore, the dictionary that is searched with dynamic programming also determines the complexity

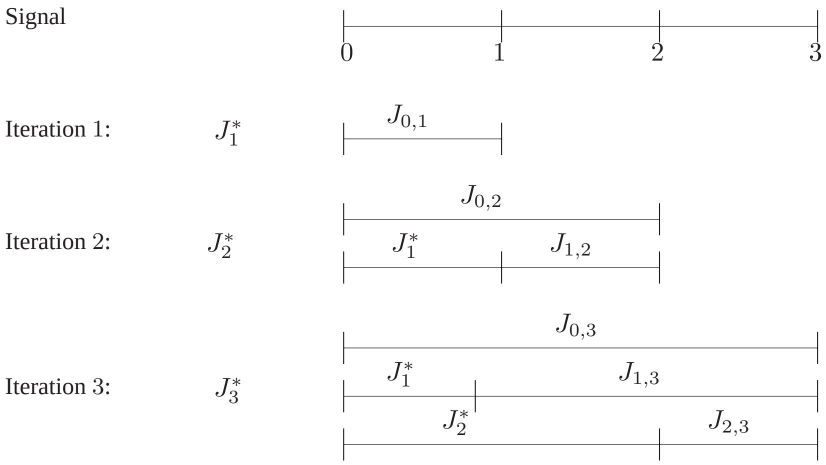


Figure 8.2: *The flexible time segmentation algorithm employs dynamic programming to construct the optimal time segmentation.*

of the *initialization* part where the (R, D_π) pairs are computed (step 1 in algorithm 1). In general, for a signal consisting of N frames, the costs for $N - i + 1$ segments of i frames are to be generated, for $i = 1, \ldots, N$. Assume that for a segment of i frames and length Mi, a signal transform with complexity $Mi \log_2(Mi)$ is required. Moreover, assume that we have $|C|$ coding templates available and the computation of the cost for a segment has complexity Mi. The complexity of step 1 is then limited to $\mathcal{O}(MN^3(\log_2 MN + |C|))$. Moreover, in step 2 of algorithm 1, the minimal Lagrangian costs are obtained for every possible segment. A useful memory structure for storing these costs is the matrix **J**, given as

$$\mathbf{J} = \begin{pmatrix} J_{0,1} & J_{1,2} & \cdots & J_{N-1,N} \\ \infty & J_{0,2} & \cdots & J_{N-2,N} \\ \infty & \infty & \ddots & \vdots \\ \infty & \infty & \cdots & J_{0,N} \end{pmatrix},$$

where the rows of **J** represent increasing segment lengths and the columns indicate the end position of a segment. This structure allows for an efficient vectorized implementation of Eq. 8.5, where J_i^* is computed as

$$J_i^* = \min \left\{ \begin{bmatrix} J_{i-1}^* \\ J_{i-2}^* \\ \vdots \\ J_1^* \\ 0 \end{bmatrix} + \begin{bmatrix} J_{i-1,i} \\ J_{i-2,i} \\ \vdots \\ J_{1,i} \\ J_{0,i} \end{bmatrix} \right\}. \tag{8.6}$$

Note that the first vector in Eq. 8.6 contains costs corresponding to optimal segmentations of the signal up to frame $i-1$. The second vector in Eq. 8.6 is the ith column in **J**.

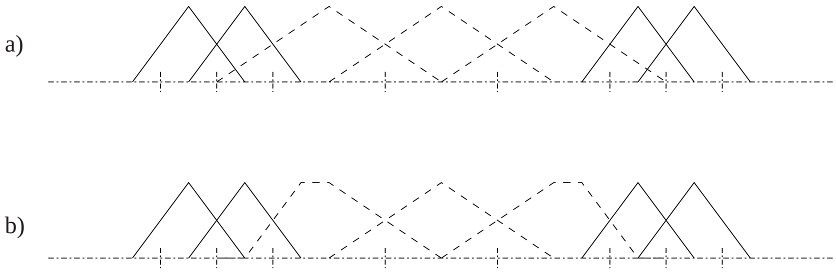


Figure 8.3: *a) Symmetric MDCT windows with maximum overlap are employed during optimization. b) Adapted transition windows are employed during coding to ensure perfect reconstruction.*

8.2.2 suboptimality of the existing approach

When the flexible time segmentation algorithm from Section 8.2.1 is applied in combination with windowing and overlap-add, a segmental cost $J_{k,l}$ is computed on a windowed time interval that can be larger than the segment $s_{k,l}$ under consideration. In the case of a fixed window overlap the aforementioned algorithm still provides the optimal solution [6]. However, in general, segments of the same length can be used with different window shapes, since the mutual overlap of the windows corresponding to two adjacent segments depends of the length of *both* segments. Hence, independent computation of costs for the individual segments is no longer possible. An exact solution to this dependency problem, when using the flexible time segmentation algorithm from Section 8.2.1, has been derived by the authors [16]. However, this solution increases the computational complexity of the flexible time segmentation algorithm significantly, without providing an equally significant cost reduction and is therefore not applied in the transform coder discussed in Section 8.3.

Instead, the dependency between costs and window overlap is neglected and during optimization a window overlap is selected that only depends on the length of the segment under consideration [13]. However, the cost thus obtained is, in general, not equal to the minimum cost achieved by the method in [16]. Moreover, the selection of a window overlap that only depends on the length of the segment under consideration results in a non-PR overlap-add system. As a result, in the subsequent coding stage all segments within the selected segmentation undergo an additional sequence of MDCT transformation, quantization and coding, where windows with the correct overlap are applied. This difference in window shape for the separate optimization and coding operations is depicted in Figure 8.3. Note that this particular solution to the dependency problem leads to time segmentations that can be suboptimal. We come back to this point when comparing the existing algorithm with time segmentations obtained by upfront segmentation in Section 8.4.

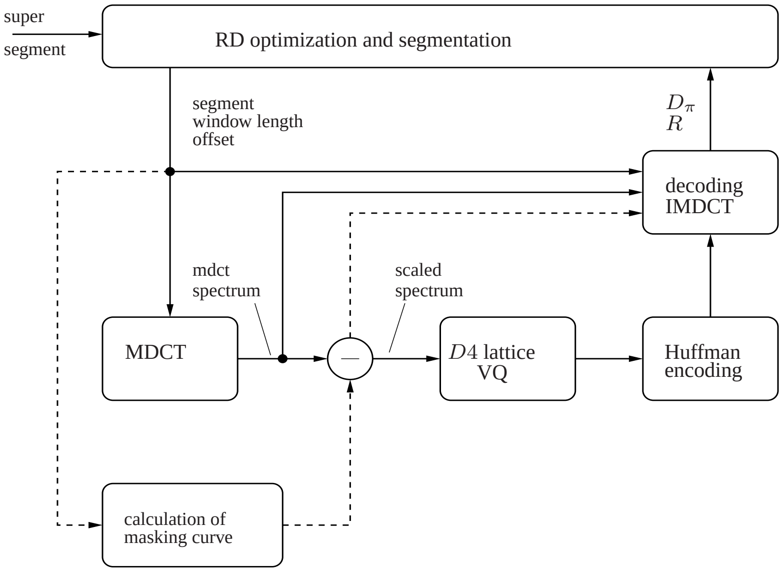


Figure 8.4: *Scheme of the proposed transform coder.*

8.3 The transform coder framework

In this section we discuss a transform audio coder that implements the flexible time segmentation algorithm from Section 8.2.1. Figure 8.4 presents a block scheme of the encoder part of the transform coder. First, the audio input signal is partitioned in large time intervals, called supersegments. Such a supersegment is initially divided into of a number of frames. The RD optimization block applies the flexible time segmentation on individual supersegments, to determine the optimal number and lengths of segments within a supersegment. As a results of this, the target rate constraint is satisfied on a per-supersegment basis.

Within the scope of this article, a framesize of 128 samples is considered. A supersegment consists of 144 frames and we allow segment lengths of $1, 2, 4$ and 8 frames, leading to windows of $256, 512, 1024$ and 2048 samples, respectively. At a sampling frequency of 48 kHz for the input signal, this corresponds to windows of 5.33, 10.67, 21.33 and 42.67 ms, respectively, and a supersegment of 384 ms. The overlap at supersegment boundaries is currently fixed to 64 samples, that is, the overlap corresponding to the smallest possible window length.

The idea behind the initial division of a signal in supersegments is to provide a trade-off between coding performance on one hand and complexity and coding delay on the other hand. Employing supersegments that consist of many frames results in high coding performance at the costs of increased complexity and a significant delay, since optimization is performed and bitstream information is returned every supersegment. Low complexity and delay values are obtained when using a small number of frames within a supersegment, which will result in a lower quality at similar rates.

However, several studies [17, 18] indicate that even for very short supersegments, flexible time segmentation can increase coding performance compared to a uniform segmentation within the supersegment.

To calculate the (R, D_π) pairs necessary for optimization, the signal data in each of the segments is transformed with an MDCT. Additionally, excitation patterns are calculated for all segments based on the model in [19] and according to the method in [12]. From such an excitation pattern a masking curve is derived, that is subsequently used for scaling the MDCT coefficients. The scaling can be seen as a perceptual whitening operation by substraction of the masking threshold from the MDCT spectrum in the sound pressure level domain. Next, the scaled MDCT coefficients are quantized with a D4-lattice vector quantizer (VQ) [20]. The resulting VQ codebook indices are mapped to Huffman codewords which are stored in a bitstream later on. The bit rate for each segment is then calculated as the sum of the Huffman codeword lengths. Additional tools to further reduce the bit rate are applied, such as a lowpass filter and separate codewords to indicate long runs of zero-valued high frequency coefficients. The total bit rate per segment also includes all side information such as the segment length and the masking curve offset factor, as explained later.

For a particular segment, the perceptual distortion D_π is defined as

$$\int_0^1 \hat{a}(f)|\hat{\varepsilon}(f)|^2 \mathrm{d}f,$$

where $\hat{a}(f)$ is the inverse of the masking curve and $\hat{\varepsilon}(f)$ denotes the Fourier transform of the time-domain reconstruction error $\varepsilon(n)$. Since the MDCT is not perfect reconstructing on a block basis, this error $\varepsilon(n)$ is computed as the difference between a segment (including overlap) synthesized from unquantized parameters and a segment that is reconstructed after quantization.

A trade-off between the bit rate required for coding the scaled MDCT coefficients and the resulting perceptual distortion can be achieved by shifting the global level of the masking curve with a single offset prior to the scaling of the MDCT coefficients. A positive offset, i.e. raising the masking threshold, will result in more MDCT coefficients falling below the masking threshold and, therefore, more coefficients being quantized to zero. The resulting bit rate is lower but perceptual distortion has increased. Likewise, a negative offset lowers the masking threshold, which results in a lower perceptual distortion at a higher rate. The masking curve offset can be varied in 30 steps of 1 dB. In order to perform the inverse shifting of the spectral coefficients in the decoder, the offset and a coded version of the excitation pattern are stored in the bitstream for each segment. In [11] a method for coding the excitation patterns at bit rates around 4 kbps is disclosed, which is applied here.

8.3.1 experimental results

The transform audio coder as described above has been compared in a MUSHRA test [21] with state-of-the-art MPEG-2/4 [22] audio codecs such as AAC [4, 7] at 48 kbps and SSC [23] at 24 kbps. In this test, 8 fragments from various musical genres were used, listed in Table 8.1, and 15 listeners participated in the test. The results are

Table 8.1: *The 8 test signals used in the MUSHRA test.*

1	Basketball commentary
2	Classic music
3	Pop music (Fool's Garden)
4	Pop music (Eric Clapton)
5	Jazz music
6	Castanets
7	Harpsichord
8	German male speech

presented in Fig. 8.5 and show that the proposed transform coder, denoted as RDTC, can compete with state-of-the-art codecs at multiple bit rates.

To investigate the effect of adaptive time segmentation, the same set of fragments was coded with the transform coder having only fixed window sizes of 2048 and 256 samples, respectively. Comparison of distortions for each supersegment revealed that the adaptive time segmentation algorithm made a dominant contribution to the reduction of the perceptual distortion.

8.4 Upfront time segmentation

Although the flexible time segmentation algorithm delivers a significant contribution to the overall performance of the system presented in Section 8.3, its complexity remains quite high. However, the main part of the complexity lies in the computation of (R, D_π) pairs as performed in the *initialization* phase, where MDCT transforms are performed for each segment, as well as multiple psychoacoustic analysis operations. Furthermore, repeated quantization with each of the coding templates (e.g. the 30 masking curve offset values) has to be performed. Therefore, we investigate the possibilities for a complexity reduction by applying upfront time segmentation.

Upfront time segmentation can be accomplished by separation of the optimization procedure in a stage where the segmentation is obtained and a stage where the coding templates are selected. For the stage where the segmentation is obtained, the RD cost functional $J_{\mathrm{RD}}(\mathbf{s}, \mathbf{c}, \lambda) = D_\pi(\mathbf{s}, \mathbf{c}) + \lambda R(\mathbf{s}, \mathbf{c})$ is replaced with a cost measure that is independent of distortion and rate. When switching to a cost function that is independent of quantized values, complexity can be significantly lowered. Moreover, only a single dynamic programming operation needs to be run, instead of one at every iteration of the bisection algorithm. Assume we have such a cost functional, say J_{UP}. Then the problem as stated in Eq. 8.3 is replaced by a sequential procedure, where the upfront segmentation results in a segmentation $\mathbf{s}^*$ of the signal by solving

$$\mathbf{s}^* = \arg\min_{\mathbf{s} \in S} J_{\mathrm{UP}}(\mathbf{s}),$$

and the subsequent search for the RD optimal coding templates, given $\mathbf{s}^*$, is performed

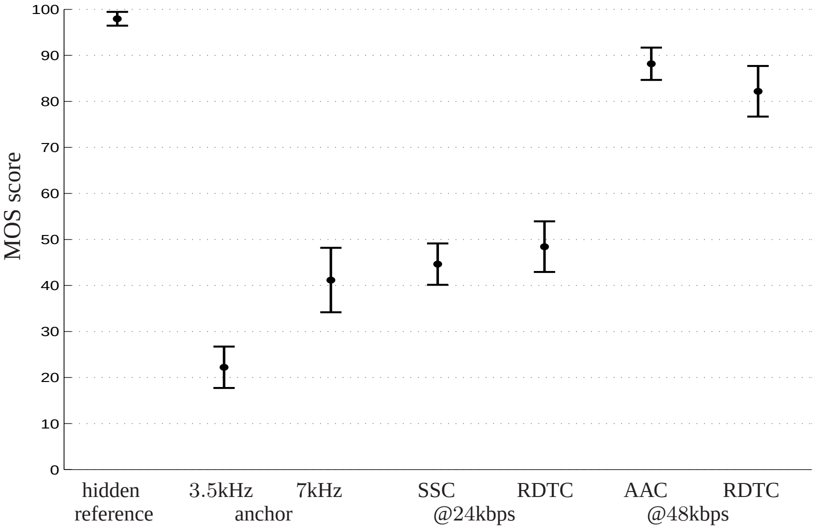


Figure 8.5: *MUSHRA listening test results with the proposed transform coding system. Mean scores and* 95% *confidence intervals over all listeners and signals are shown.*

by solving

$$\sum_{k=1}^{p_{\mathbf{s}*}} \min_{c_k} J_{\text{RD}}(s_k, c_k, \lambda).$$

Given the predefined cost J_{UP}, the upfront segmentation can be performed by applying the flexible time segmentation algorithm from Section 8.2.1. This has the advantage of retaining the dynamic programming based optimization procedure for obtaining a time segmentation, which provides a large dictionary of time segmentations and a fast search procedure. Given a segmentation of a supersegment, the subsequent search for the optimal scale factors is still performed using the RD cost measure J_{RD}, since this is the final cost measure that we are interested in. A drawback of this approach is that the resulting segmentation can not adapt to varying target rates.

8.4.1 cost measures for upfront segmentation

Several RD-independent cost measures known from literature can be used in a best basis search algorithm. Specifically for (audio) coding, we first study the Coifman-Wickerhauser entropy (CWE) cost measure [24]. This measure has been previously applied for audio and speech signal in several best basis algorithms similar to [13] and can be seen as an energy compaction measure. The CWE costs are typically computed for DCT coefficients. Given a set of DCT coefficients $\mathbf{X}$ computed for a segment of length M, the CWE cost J_{CWE} is defined as

$$J_{\text{CWE}} = -\sum_{k=0}^{M-1} X[k]^2 \log_2(X[k]^2),$$

Clearly, the computation of the CWE costs does not require a psychoacoustic analysis or any quantization operations.

Secondly, we investigate the use of the perceptual entropy (PE) cost measure [8]. Given a set of DFT coefficients $\mathbf{Y}$ computed for a segment of length M, the perceptual entropy J_{PE} is defined as

$$J_{\text{PE}} = \sum_{k=0}^{M-1} \log_2\Big(1 + \sqrt{\hat{a}[k]Y[k]^2}\Big),$$

where $Y[k]^2$ represents the signal intensity and $\hat{a}[k]$ denotes the relative intensity of the inverse of the masking threshold, both at frequency line k. It can be regarded as the minimum number of bits needed to code a segment transparently. As indicated before, this measure has been applied for time segmentation in various audio coding systems, e.g. in AAC where the perceptual entropy measured on long blocks is monitored for presence of transients. However, the combination with a dynamic programming based framework has not been studied.

The notion of perceptual entropy is derived from an *ideal* transform coding framework where quantization noise is injected at each frequency sample in the power spectrum $\mathbf{Y}^2$ up to the masking threshold. Separate quantizers are derived implicitly per component from the masking threshold and the perceptual entropy is computed as the information necessary to send reconstruction levels per component. No further side info (e.g. for sending the masking curve) is assumed. Since we do separate encoding of the masking threshold and, other than the quantized and coded spectral coefficients, little other side information is required at the decoder, the presented transform coding framework correlates well with the idea of perceptual entropy.

8.4.2 computational complexity

We can derive the complexity reduction from the complexity of the *initialization* phase, given as $\mathcal{O}(MN^3(\log_2 MN + |C|))$ in Section 8.2.1. Note that since we do not consider all possible window sizes, this is an upperbound to the actual complexity. We have the number of frames within a supersegment as $N = 144$, the number of coding templates $|C| = 30$ and the number of samples in a frame, including window overlap, as $M = 256$. With upfront segmentation, complexity is reduced to $\mathcal{O}(MN^3(\log_2 MN))$. Since $\log_2 MN \approx 15$, we see that upfront time segmentation can reduce the computational complexity of the initialization phase by 66%. Moreover, only a single dynamic programming operation needs to be run during the *optimization* phase, instead of one at every iteration of the bisection algorithm. Since typically 12 to 14 iterations are required to reach the target rate, this can result in additional savings. However, after having found a time segmentation $\mathbf{s}^*$, the $p_{\mathbf{s}^*}$ constituent segments have to be coded with the $|C|$ coding templates to generate a set of (R, D_π) pairs.

It remains difficult to determine the exact reduction in complexity that can be obtained with these upfront segmentation methods in the proposed transform coding framework. First of all, the existing framework has not been optimized for coding speed. Also, in the case of the PE cost measure, many of the psycho-acoustic analysis

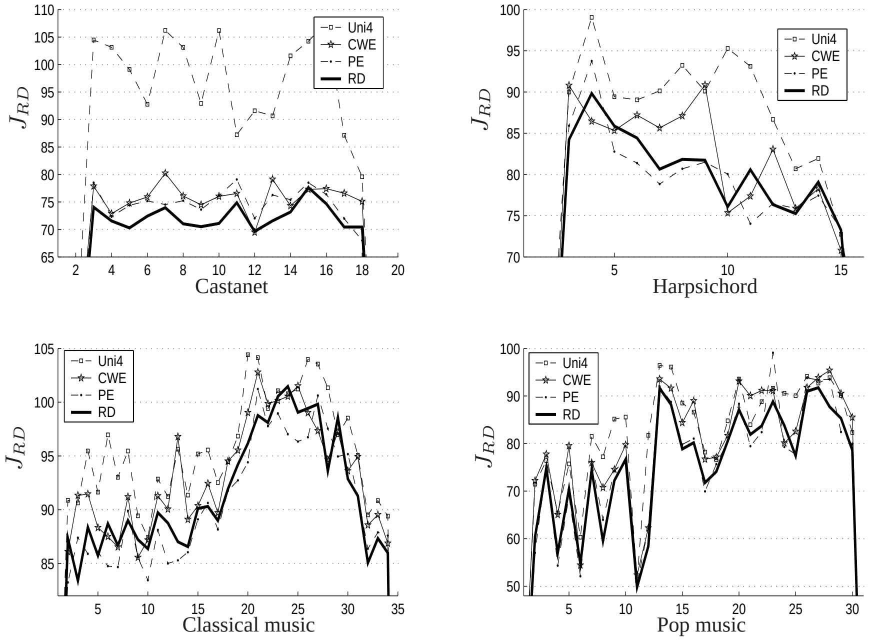


Figure 8.6: *Lagrangian costs obtained for each time segmentation method, for* 4 *audio fragments. For each fragment and method, the Lagrangian costs* J_{RD} *are shown per supersegment. The label* $Uni4$ *denotes uniform time segmentation with segments of* 4 *frames.*

results can be used later on during coding. This further reduction has not been implemented during our experiments. Execution times resulting from both RD optimal and upfront time segmentation algorithms were recorded for five fragments, showing an average reduction of 47%.

8.4.3 experimental results

The upfront segmentation algorithm was implemented in the transform coding system from Section 8.3. Both CWE and PE cost measures were employed and comparisons were made with the existing RD optimal segmentation method. Additionally, the fragments were coded using a uniform time segmentation, where segments of 512 samples were employed (denoted by the label $Uni4$). Experimental results for 4 fragments from Table 8.1 were generated in these experiments. Each signal was coded at a total bit rate of 24 kbps. This rate included the bit rate for sending the excitation patterns, around 4 kbps. The target rate for the optimization was therefore equal to 20 kbps.

Triplets of bit rate, perceptual distortion and λ were recorded for every supersegment. From the (R, D_π, λ) triplets, Lagrangian costs per supersegment were computed. These results are displayed in Figure 8.6, from which we make the following observations. First, the need for a time-varying segmentation of the signal is emphasized, as the Lagrangian costs for signals coded with any of the variable time segmentation methods are nearly always lower than those obtained with uniform time segmentation. This is for instance clearly noticeable with the castanet signal.

Secondly, it is seen that the difference between RD segmentation and CWE based upfront segmentation can be quite large for individual supersegments, as is the case in the harpsichord fragment. This has also been asserted by the authors through extensive prelistening. Surprisingly, failure of the CWE based method to select a correct segmentation does not necessarily occur during critical signal parts, i.e. signal parts where one can expect difficulties during coding, such as explicit transients. We therefore concluded that CWE based upfront time segmentation method was unpredictable and as such, the method was not included in the listening test with multiple listeners. On the other hand, the results obtained with PE based upfront segmentation are fairly close to those obtained with RD optimal segmentation. Indeed, for several supersegments, upfront segmentation based on the PE cost measure resulted in a lower Lagrangian cost than RD optimal segmentation. This comes from neglecting the dependency between window overlap in the optimization phase as noted in Section 8.2.2, which renders the existing algorithm suboptimal.

This effect can be seen more clearly in Figure 8.7, where a part of the harpsichord signal has been encoded. The resulting RD costs are given above each plot. In the upper plot 8.7a, a uniform time segmentation in segments of 8 frames has been applied. The next plot 8.7b shows results obtained with the RD optimized algorithm. Clearly, a cost reduction can be obtained through appropriate segmentation of the signal. However, we can also observe a rather excessive usage of short segments in the stationary part of the harpsichord fragment. Results with CWE based upfront segmentation, as displayed in Figure 8.7c, are slightly worse, mainly due to increased usage of short segments. PE based upfront segmentation provides the lowest cost, as seen above Figure 8.7d.

The discrepancy between PE upfront and RD based time segmentation can be explained as follows. From Figure 8.3 in Section 8.2.2, we see that during optimization the selection of a short segment size does not influence previously selected segment sizes. When applying RD optimal segmentation, the RD cost measure is affected by many aspects of the coding framework. Therefore, it can be easily the case that the selection of a short segment between two large segments results in a small cost reduction, e.g. due to a better alignment of the MDCT basis functions with the segment to be coded. In contrast, during coding of the signal, the windows for coding the large segments are adapted such that the spectral properties of these windows are severely compromised. As a result, coding efficiency is reduced. On the other hand, the PE cost measure relies only on the signal and its corresponding masking capabilities. Moreover, a DFT is applied for computation of the signal spectrum, which results in shift invariance. Clearly, the PE cost measure exhibits a high discriminative power for stationary signals.

To further investigate the PE based upfront segmentation method, an OAB preference listening test with 10 participants was performed on 5 fragments from Table 8.1. Listeners had to indicate their preference for either the existing RD based method,

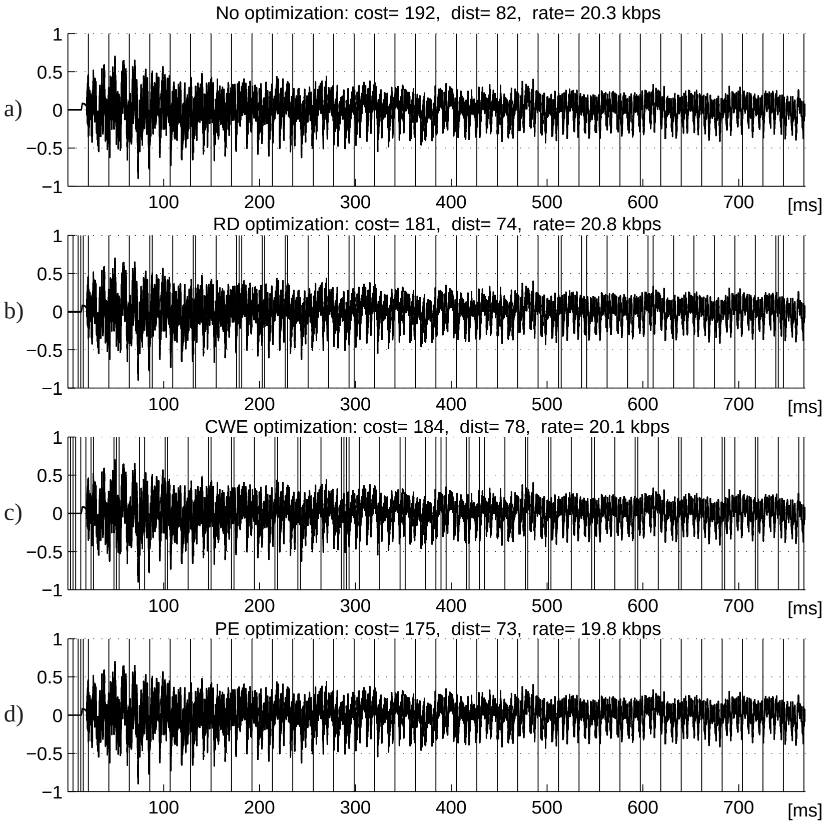


Figure 8.7: *Time segmentation of harpsichord fragment with a) uniform segmentation b) RD cost measure c) CWE cost measure and d) PE cost measure. The resulting RD costs are given above each plot.*

resulting in a score 1, and the new PE based method, corresponding to a score 0. Each listener had to score every fragment four times. The resulting set of four scores was transformed into new single scores of 0,25,50,75 or 100. This was done for each listener and fragment. The score distributions are displayed per fragment in Figure 8.8. Additionally, the scores over all fragments are shown. Given the number of listeners, fragments and test repetitions, we obtained a total of 200 scores for either the PE based or the RD based method. Given these 200 scores, exactly 100 scores were for the RD based method and 100 scores for the PE based method.

Furthermore, we assume that if listeners are not able to distinguish between the two methods, the median of the score distribution lies at 50. This then indicates that both time segmentation methods give coded fragments with a similar perceptual distortion. Therefore, a two-sided Wilcoxon signed-ranks test [25] was performed with a null hypothesis that the score distributions came from a population with a median of 50. Indicated above each plot in Figure 8.8 is the probability of observing the ob-

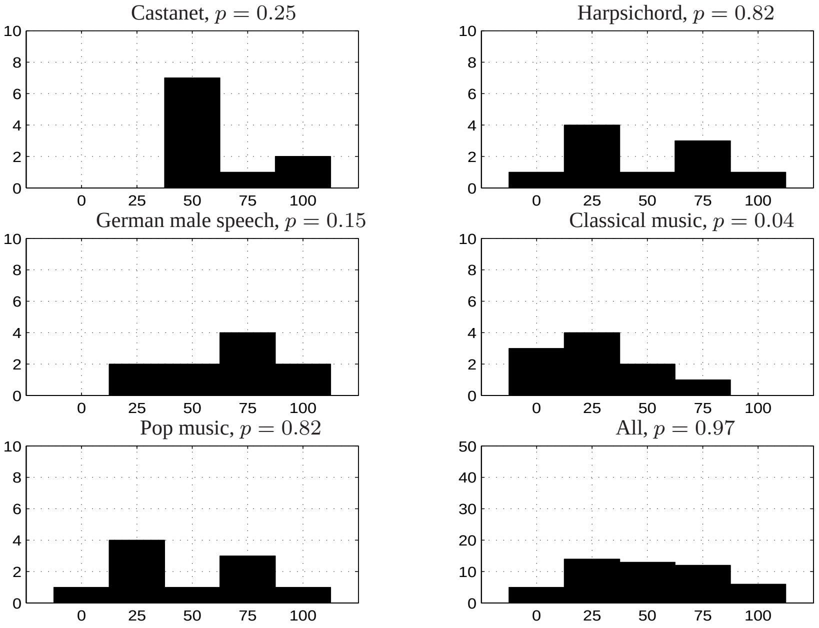


Figure 8.8: *OAB score distributions for the individual fragments and for all fragments. Higher scores indicate preference for RD based segmentation.*

tained score distribution by chance if the null hypothesis is true. Small values of the probability cast doubt on the validity of the null hypothesis. At a significance level of 0.1, the null hypothesis can only be rejected for the classical music fragment, where there is a significant preference for the PE based method.

8.5 Conclusions

In this contribution, we studied a transform coding framework that employs rate-distortion optimal time segmentation. We discussed the dynamic programming based times segmentation method and presented results from a MUSHRA test. Furthermore, as a low-complexity alternative, we proposed an upfront time segmentation method based on the perceptual entropy cost measure. Results of comparative test of these two methods, both in the form of perceptual distortion-vs-rate plots and listening tests showed that dynamic programming based upfront time segmentation for minimization of a perceptual entropy cost measure can be a viable alterative to rate-distortion optimal time segmentation. We showed that upfront time segmentation has a lower computational complexity since the number of (R, D_π) pairs to be computed is significantly lower. Future work will concentrate on exploiting the redundancies between the various psycho-acoustical analysis functions that are performed in the optimization and coding phases.

8.6 Acknowledgments

The research was conducted within the ARDOR project, supported by the E.U. grant no. IST-2001-34095. The authors would like to thank all members of the ARDOR project for their contributions.

Bibliography

[1] J.P. Princen and A.B. Bradley. Analysis/synthesis filter bank design based on time domain aliasing cancellation. *IEEE Transactions On Acoustics, Speech, And Signal Processing*, 34(5):1153–1161, October 1986.

[2] H. S. Malvar. *Signal Processing with Lapped Transforms*. Artech House, Boston, MA, 1992.

[3] International Standard ISO/IEC 11172-3 (MPEG). Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s. part 3: Audio, 1993.

[4] International Standard ISO/IEC 13818-7 (MPEG). Information technology - generic coding of moving pictures and associated audio, part 7: Advanced audio coding, 1997.

[5] B. Edler. Codierung von audiosignalen mit uberlappender transformation und adaptiven fensterfunktionen (in german). *Frequenz*, 43(9):252–256, 1989.

[6] J. Kovačević and M. Vetterli. Time-varying modulated lapped transforms. In *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*, pages 481–485, Pacific Grove, USA, November 1993.

[7] M. Bosi and et al. ISO/IEC MPEG-2 advanced audio coding. *Journal of the Audio Engineering Society*, 45:789–812, October 1997.

[8] J.D. Johnston. Estimation of perceptual entropy using noise masking criteria. In *Proceedings of the 1988 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'88)*, pages 2524–2527, New York, USA, April 1988.

[9] 3rd Generation Partnership Project. 3GPP TS 26.403 V6.0.0: Encoder specification AAC part (release 6). ftp://ftp.3gpp.org/specs/2004-12/Rel-6/26_series/26403-600.zip, November 2004.

[10] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions On Image Processing*, 2(2):160–175, April 1993.

[11] O. Niemeyer and B. Edler. Efficient coding of excitation patterns combined with a transform audio coder. In *Proceedings of the 118th AES Convention*, Barcelona, Spain, May 2005.

[12] S. van de Par, V. Kot, and N. H. van Schijndel. Scalable noise coder for parametric sound coding. In *Proceedings of the 118th AES Convention*, Barcelona, Spain, May 2005.

[13] O.A. Niamut and R. Heusdens. RD optimal time segmentations for the time-varying MDCT. In *Proceedings of the 12th European Signal Processing Conference (Eusipco'04)*, pages 1649–1652, Vienna, Austria, September 2004.

[14] R. Bellman. *Dynamic Programming*. Princeton University Press, New Jersey, 1957.

[15] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard. Flexible time segmentations for time-varying wavelet packets. In *Proceedings of the IEEE-SP Conference on Time-Frequency and Time-Scale Analysis*, pages 9–12, Philadelphia, USA, October 1994.

[16] O.A. Niamut and R. Heusdens. Optimal time segmentation for overlap-add systems with variable amount of window overlap. *IEEE Signal Processing Letters*, 12(10):665–668, October 2005.

[17] Nicolle H. van Schijndel and Grégory d'Ambrosio. On delay in parametric audio coding with adaptive segmentation. In *Proceedings of the 1st IEEE BENELUX/DSP Valley Signal Processing Symposium (SPS-DARTS'05)*, pages 37–40, Antwerp, Belgium, April 2005.

[18] C.A. Rodbro, J. Jensen, and R. Heusdens. Adaptive time-segmentation for speech coding with limited delay. In *Proceedings of the 2004 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'04)*, pages 465–468, Montreal, Canada, May 2004.

[19] S. van de Par, A. Kohlrausch, R. Heusdens, J. Jensen, and S. H. Jensen. A perceptual model for sinusoidal audio coding based on spectral integration. *to appear in EURASIP Journal on Applied Signal Processing (Special Issue on Anthropomorphic Processing of Audio and Speech)*, 2005.

[20] N. Meine and B. Edler. Improved quantization and lossless coding for subband audio coding. In *Proceedings of the 118th AES Convention*, Barcelona, Spain, May 2005.

[21] Radiocommunication Sector BS.1534 International Telecommunications Union. Method for the subjective assessment of intermediate quality level coding systems - general requirements (MUSHRA), 2001.

[22] International Standard ISO/IEC 14496-3 (MPEG). Information technology - coding of audio visual objects, part 3: Audio, 1999.

[23] Bert den Brinker, Erik Schuijers, and Werner Oomen. Parametric coding for high-quality audio. In *Proceedings of the 112st AES Convention*, Munich, Germany, May 2002.

[24] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718, March 1992.

[25] D. Sheskin. *Handbook of parametric and nonparametric statistical procedures*. CRC Press, Boca Raton, Florida, 2004.

Chapter 9

RD Optimal Temporal Noise Shaping

@ 2006 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE.

This chapter is a slightly modified version of “RD Optimal Temporal Noise Shaping”, by O.A. Niamut and R. Heusdens in the Proceedings of the *IEEE International Conference on Audio, Speech and Signal Processing,* May 2006.

Abstract

In this article we investigate rate-distortion optimal temporal noise shaping for transform audio coding. Temporal noise shaping, or TNS, is a technique for reshaping the quantization noise in the time domain through open-loop linear predictive coding of frequency domain coefficients. Traditionally, a selection mechanism based on prediction gain is employed to determine whether it is advantageous to apply TNS or not. Although this method is effective for reducing coding artifacts in transient and speech signals, critical adjustment of the prediction gain threshold is necessary to avoid excessive bit rate demands. We propose the use of TNS in a rate-distortion optimization framework. Within this framework a jointly optimal selection of the prediction filter order and the quantizer for coding the coefficients can be made, such that the perceptual distortion is minimized for a given target rate. Experimental results for an MDCT-based audio coding system are presented and it is shown that TNS within an RD optimization framework outperforms the existing TNS method.

9.1 Introduction

In audio coding applications the goal is to minimize the perceptual distortion introduced by the coding process whilst satisfying a bit rate constraint. This goal can be achieved by the application of an operational rate-distortion (RD) optimization approach [1, 2] that reaches the solution to the audio coding problem for a given coding environment and allows for optimal selection of the involved coding parameters.

Most audio coding schemes rely on a time-frequency analysis of the input signal and typically, an MDCT [3] is applied for this purpose. The MDCT has desirable properties, such as good channel separation, strong stopband attenuation, minimum blocking artifacts and the availability of fast algorithms. Additionally, there are various techniques available for efficient resolution switching to further enhance the performance. Temporal noise shaping (TNS) [4, 5] is such a technique that allows for block-continuous adaptation of the time-frequency resolution

In this paper, we propose an operational rate-distortion optimization framework for TNS in an MDCT-based audio coder. The paper is organized as follows. In Section 9.2 the traditional TNS technique is explained and some of its potential problems are highlighted. Next, we propose an RD optimal temporal noise shaping algorithm in Section 9.3. Finally, both quantitative and qualitative experimental results for encoding several audio fragments with an MDCT-based audio coding scheme are presented in Section 9.4.

9.2 Temporal noise shaping

TNS [4, 5] is a technique for reshaping and controlling the quantization noise in the time domain through open-loop linear predictive coding (LPC) of frequency domain coefficients. Given a block of signal samples that is transformed to the frequency domain, an LPC filter is applied to a (sub)set of transform coefficients and instead of direct quantization of the frequency domain coefficients, the filtered residual is quantized along with the LPC filter coefficients. Upon reconstruction of the time domain signal, the inverse LPC filter is applied to the quantized residual before the inverse signal transform. This inverse LPC filter acts as a temporal envelope that shapes the quantization noise according to the signal energy distribution over the segment, thereby permitting a coder to exercise control over the temporal structure of quantization noise within a set of frequency coefficients. Thus, most of the quantization noise will reside in signal regions with significant energy in the time domain, thereby avoiding temporal masking problems in coding transient and speech signals, such as pre-echos and reverberation.

TNS is part of the MPEG-2/4 AAC standard [6, 7] where it is applied on coefficients obtained from using a modified discrete cosine transform (MDCT) [3] on the input signal. The MDCT is a so-called 50% overlapped block transform, i.e. a transform where samples from consecutive 50% overlapping segments are windowed and transformed. Given a signal x divided into overlapping segments of length $2M$, a set of M transform coefficients X_i is computed from the ith segment x_i by applying the direct MDCT, which is defined as [3]

$$X_i(k) = \sum_{n=0}^{2M-1} x_i(n)w(n)\cos\Big[\frac{\pi}{4M}(2n+M+1)(2k+1)\Big],$$

where $k = 0, 1, \ldots, M–1$ and w an appropriate analysis window.

Assume that X_i has variance $\sigma^2_{X_i}$. A pth order linear prediction $\tilde{X_i}$ of X_i is given by

$$\tilde{X_i}(k) = \sum_{j=1}^{p} a_j X_i(k-j).$$

The prediction error signal $\Delta X_i(k) = X_i(k) - \tilde{X_i}(k)$ with variance $\sigma^2_{\Delta X_i}$ is then computed (in the Z-transform domain) as

$$\Delta X_i(z) = X_i(z)A(z),$$

with $A(z) = 1 - \sum_{j=1}^{p} a_j z^{-j}$. We want to find the filter $A(z)$ that minimizes $\sigma^2_{\Delta X_i}$ and thus maximizes the prediction gain $\sigma^2_{X_i}/\sigma^2_{\Delta X_i}$. This can be done efficiently with the well-known Levinson-Durbin recursion algorithm.

In many implementations of TNS in the AAC standard, the prediction gain is used to determine whether TNS should be applied or not. First, for a block of MDCT coefficients a high order (e.g. 12 or 20) LPC filter is computed with the autocorrelation method using the Levinson-Durbin algorithm. If the prediction gain is larger than a certain threshold T_{PG}, TNS is applied. The Levinson-Durbin algorithm generates a set of reflection coefficients r, ordered in decreasing magnitude. The final LPC filter order p' is determined by subsequently removing reflection coefficients having an absolute value lower than a threshold T_r from the reflection coefficient array. This procedure lowers the side information for sending the filter coefficients. A block scheme of a typical TNS implementation, which we shall refer to as TNS PG (prediction gain), is shown in Fig. 9.1. We can distinguish two separate quantizer blocks since

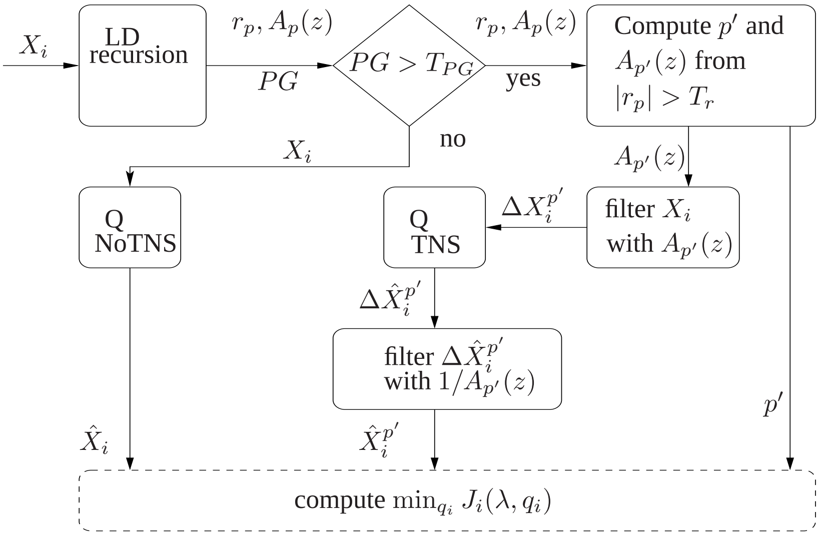


Figure 9.1: *Block scheme of a standard TNS implementation based on prediction gain. The dashed block indicates the RD control block that is employed in the experiments.*

it is generally efficient to have separate quantizers for unfiltered and filtered MDCT coefficients.

It has been recently recognized that TNS can cause several undesirable coding artifacts [8]. For example, the quantization noise increases with the LPC filter order and is amplified around attacks, which might result in unmasked quantization noise. Moreover, the prediction gain does not always accurately represent the coding performance resulting from TNS usage. If TNS is applied, but the actual coding performance turns out to be insignificant, a large portion of the available bit rate unnecessarily goes to the filter coefficients. Clearly, the choice of the thresholds on prediction gain T_{PG} and reflection coefficients T_r is a delicate one, since it determines both the side information for sending the filter and the occurrence of undesired artifacts. Therefore, the thresholds should be made dependent on the target bit rate, which requires a difficult tuning process. The method in [8] proposes a perceptual entropy measure for determining the usage of TNS. This is basically a one-sided bit rate cost measure, since perceptual entropy can be seen as the minimal bit rate for transparent audio coding. We extend this notion to a two-sided cost measure of both bit rate and perceptual distortion.

9.3 RD optimal TNS

In an operational RD optimization framework [1, 2], the distortion D is minimized subject to a bit rate constraint R_T for a given coding environment. In the TNS case the coding environment consists of an MDCT-based audio coding system for coding a signal consisting of, say N, overlapping segments. Given a set of LPC filter orders $\mathbb{P}$ and a set of quantizer stepsizes $\mathbb{Q}$ for coding the MDCT coefficients, either directly or

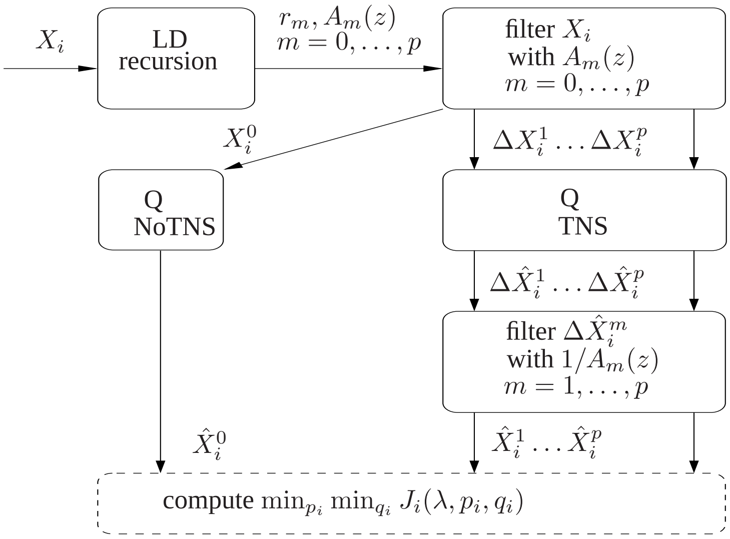

X_i
LD recursion
$r_m, A_m(z)$ $m = 0, \ldots, p$
filter X_i with $A_m(z)$ $m = 0, \ldots, p$
X_i^0
$\Delta X_i^1 \ldots \Delta X_i^p$
Q NoTNS
Q TNS
$\Delta \hat{X}_i^1 \ldots \Delta \hat{X}_i^p$
filter $\Delta \hat{X}_i^m$ with $1/A_m(z)$ $m = 1, \ldots, p$
$\hat{X}_i^0$
$\hat{X}_i^1 \ldots \hat{X}_i^p$
compute $\min_{p_i} \min_{q_i} J_i(\lambda, p_i, q_i)$

Figure 9.2: *Block scheme of the proposed TNS implementation in an RD optimization framework.*

after LPC filtering, we want to select for the ith segment x_i the LPC filter order $p_i \in \mathbb{P}$ and the quantizer stepsize $q_i \in \mathbb{Q}$ that jointly minimize the perceptual distortion, denoted by D_π for a given target rate R_T. Note that a zero-order LPC filter, i.e. $p_i = 0$, indicates direct coding of the MDCT coefficients. Fig. 9.2 shows a block scheme of the RD-based TNS algorithm, referred to as TNS RD.

Let $\mathbf{p}$ be the N-dimensional vector consisting of the selected filter orders for all N segments and let $\mathbf{q}$ denote the vector containing the N quantizer stepsizes. The rate-constrained problem of minimizing D_π subject to R_T can then be written as

$$\min_{\mathbf{p}} \min_{\mathbf{q}} D_\pi(\mathbf{p}, \mathbf{q}) \quad \text{s.t.} \quad R(\mathbf{p}, \mathbf{q}) \leq R_T,$$

which can be converted into the unconstrained problem

$$\min_{\mathbf{p}} \min_{\mathbf{q}} J(\lambda, \mathbf{p}, \mathbf{q}), \tag{9.1}$$

where $J(\lambda, \mathbf{p}, \mathbf{q}) = D_\pi(\mathbf{p}, \mathbf{q}) + \lambda R(\mathbf{p}, \mathbf{q})$ is a Lagrangian cost function of perceptual distortion D_π and bit rate R for coding the complete signal. The parameter $\lambda \geq 0$ has to be chosen such that the target rate is met, i.e. $R(\mathbf{p}, \mathbf{q}) = R_T$. The bit rate can be split up into contributions from the coded -and possibly filtered- MDCT coefficients and side information, such as the LPC filter order, the quantizer stepsize and the coded LPC filter coefficients. Under the assumptions of additivity of the cost measure and independent coding over segments (or, equivalently, independence of the cost measure over segments), the problem from Eq. (9.1) can be formulated as

$$\min_{\mathbf{p}} \min_{\mathbf{q}} \sum_{i=1}^{N} J_i(\lambda, p_i, q_i) = \sum_{i=1}^{N} \min_{p_i \in \mathbb{P}} \min_{q_i \in \mathbb{Q}} J_i(\lambda, p_i, q_i), \tag{9.2}$$

where $J_i(\lambda, p_i, q_i) = D_{\pi,i}(p_i, q_i) + \lambda R_i(p_i, q_i)$ denotes the cost for segment x_i. As previously mentioned, λ has to be chosen such that the target rate is met. Hence, if $R(\mathbf{p}, \mathbf{q}) \neq R_T$, we have to modify λ and solve Eq. (9.2) for the new λ. This is a convex problem in λ and fast algorithms exist to solve the problem, e.g. the bisection algorithm in [1].

Given that the MDCT coefficients obtained for segment x_i are filtered with an LPC filter of order p and quantized with stepsize q, the perceptual distortion $D_{\pi,i}(p, q)$ is derived as a perceptually weighted squared sum of the difference between the original MDCT coefficients X_i before TNS and the quantized coefficients $\hat{X}_i^p$ after the inverse TNS operation, that is,

$$D_{\pi,i}(p, q) = \sum_{k=0}^{M-1} \alpha_i(k) \big[X_i(k) - \hat{X}_i^p(k)\big]^2. \tag{9.3}$$

The set of M perceptual weighting coefficients α_i is taken as the inverse masking curve for segment x_i evaluated at the MDCT center frequencies, i.e. $\alpha_i(k) = msk_i^{-1}(\frac{\pi}{M}(k + \frac{1}{2})),\ k = 0, 1, \ldots, M{-}1$. This has the desired effect that spectral distortions in frequency regions with strong masking power are de-emphasized. Furthermore, TNS and subsequent quantization is performed on the weighted set of coefficients, which typically has a lower variance and flatter spectrum than the unweighted set. This is similar to computation of a weighted MDCT spectrum in [7] where the weighting coefficients are derived from inverse energy levels in scalefactor bands.

9.4 Experimental results

In our experiments, a simple implementation of TNS in an MDCT-based audio coding system was considered. The LPC filters were applied on the complete set of MDCT coefficients. Although the quantization of the LPC filter coefficients can be taken into account, errors due to imperfect modelling of the LPC filter were neglected in this study. As discussed in the previous section, for every segment of the input signal both the LPC filter order and the coding template were selected that minimized the perceptual distortion over all segments, subject to a target rate constraint for coding the complete signal. We compared this implementation with both a system not employing TNS and the existing TNS method, i.e. based on prediction gains. In all three systems the selection of quantizers was performed in an RD optimal manner, as outlined in the previous section.

A 1024-channel MDCT was used, similar to the AAC long block operation, along with a 2048-sample Kaiser-Bessel derived window. Masking curves for each segment were computed according to the perceptual model in [9]. Based on settings in [7], the maximum LPC filter order was set to 20 and the prediction gain threshold T_{PG} for selection of TNS was set to 1.4 dB. The threshold for discarding high-order reflection coefficients was set to $T_r = 0.1$. For both direct and TNS filtered coefficients, eight quantizer stepsizes were available, including a stepsize for quantizing all coefficients to zero, and corresponding Huffman codebooks were designed. Additionally, efficient coding of long zero regions was applied. As a distortion measure, the perceptual distortion measure D_π from Eq. (9.3) was taken and the Huffman codewords were used to determine the bit rate. The side information consisted of the LPC filter order (5 bits), the filter coefficients (4 bits per filter coefficient) and the quantizer stepsizes (3 bits). The perceptual weighting coefficients, derived from the masking curve were assumed to be coded at 4 kbps, in line with results from [10]. A set of four audio fragments (48 kHz, 16-bit, mono) was used for evaluation of the three algorithms, consisting of the castanet, German male speech, bass guitar and English female speech signals.

9.4.1 results for single fragment

Fig. 9.3 presents coding results for 3 seconds of the German male speech fragment (upper plot), coded at 32 kbps. In the 2nd plot, the reconstruction error signals are displayed for the three algorithms. It can be seen that TNS localizes the quantization noise around glottal pulses and that the RD-based algorithm leads to reconstruction noise that is shaped similarly to the PG-based method. The 3rd plot shows the bit allocation over segments. The standard TNS method lowers the peak bit rate demand compared to the system without TNS, however, bit shortages still occur at several positions in the signal, mainly during segments with low energy signal content. This can be explained from the fact that at low bit rates, the existing schemes frequently run out of bits in various segments. Since selection of the stepsize that quantizes all coefficients to zero requires very few bits, this remains the only choice at these segments, thereby creating gaps in the reconstructed signal.

Although this gap artifact can not be attributed to the TNS algorithm directly, it is a clear example of inefficient use of the available bits when the TNS algorithm operates outside the rate-distortion control in this coding framework, or when not using TNS. In contrast, the RD-based method reduces the bit demand even further and facilitates a more continuous bit rate distribution over various signal parts. This allows for a more continuous signal modelling. The bit savings can be explained from the lower plot, where it is seen that the RD-based algorithm selects high order LPC filters only at critical signal parts. In contrast, the standard method uses a high order LPC filter almost exclusively, which in some cases requires an unnecessary high amount of bits.

9.4.2 results for multiple fragments

For the four signals, objective perceptual distortions for different target bit rates ranging from 18 to 64 kbps are presented in Fig. 9.4. It can be seen that the RD-based method results in the lowest perceptual distortion at every bit rate for all fragments. In order to subjectively evaluate our scheme, a MUSHRA [11] listening test was performed for evaluating the four fragments at a bit rate of 36 kbps. At this bit rate, no obvious time gap artifacts where present in the coded fragments where TNS was applied. A total of ten listeners participated (authors not included). The results averaged over all listeners are displayed per fragment in Fig. 9.5. For three fragments, the RD-based method significantly outperforms the PG-based method. Only for the German male speech fragment no significant improvement is observed. This can be explained as follows.

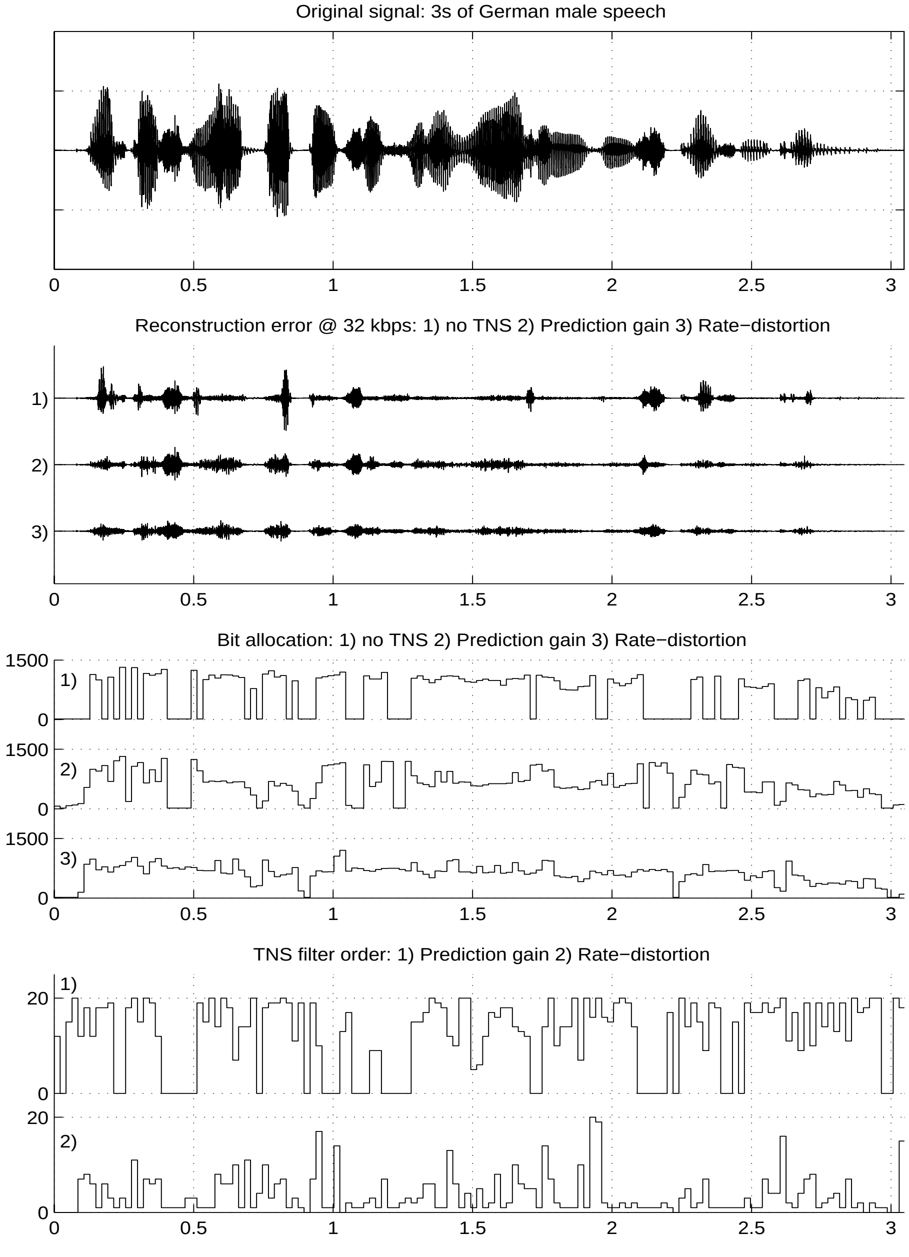


Figure 9.3: *Upper plot: 3s of German male speech signal. 2nd plot: Reconstruction error signals for the 3 methods. 3rd plot: Bit allocation over segments for the 3 methods. Lower plot: LPC filter order for (1)PG-based method and (2)RD-based method. Note that the upper two plots are displayed on a sample basis, while the lower two plots shows segment-based results.*

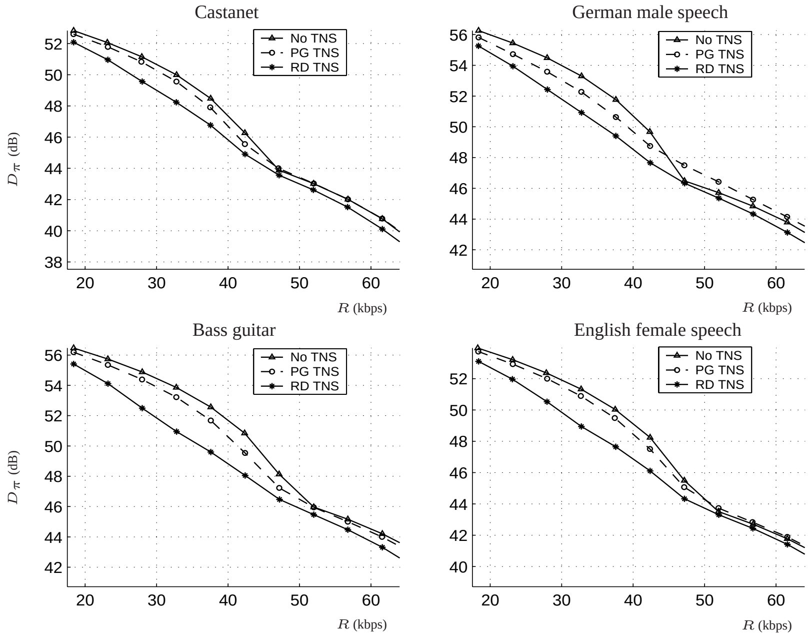


Figure 9.4: *Rate vs. perceptual distortion plots for the set of four audio fragments.*

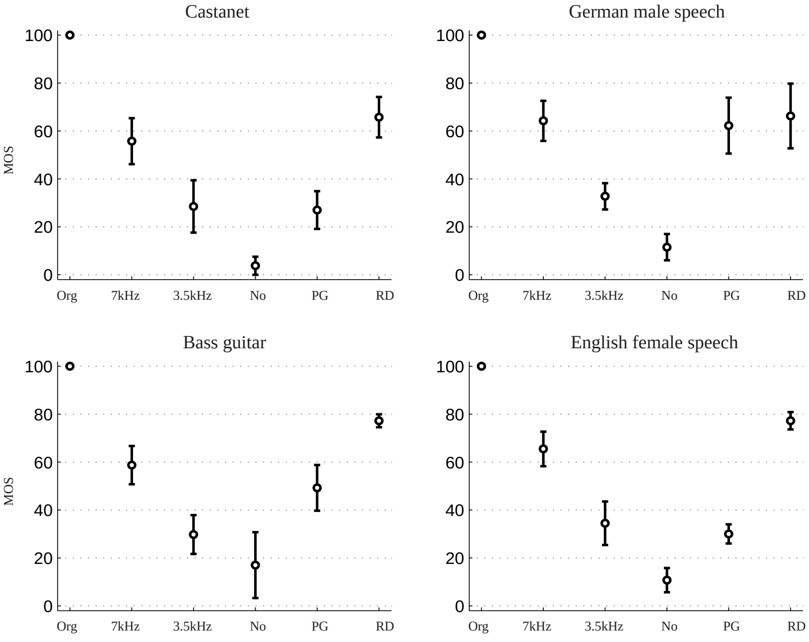


Figure 9.5: *MUSHRA test results for the original, the anchors and the coded versions without TNS, PG-based TNS and RD-based TNS.*

Since a perceptual model is used that accounted for simultaneous masking only, the RD-based method does not concentrate specifically on time domain coding artifacts such as speech reverberation. The PG-based method determines TNS usage outside the perceptual model, hence a larger reduction of these artifacts is obtained than with the RD-based method. Therefore, we expect improved performance when a perceptual model that incorporates temporal masking is applied. We conclude that the main contribution of the proposed algorithm lies in increased bit rate reduction compared to the existing method. This performance gain is obtained at a higher complexity, mostly determined by the repeated inverse IIR filtering. Therefore, complexity reductions will focus at estimating the perceptual distortion in the LPC filtered domain.

Bibliography

[1] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions On Image Processing*, 2(2):160–175, April 1993.

[2] P. Prandoni and M. Vetterli. R/D optimal linear prediction. *IEEE Transactions on Speech and Audio Processing*, 8(6):646–655, November 2000.

[3] J.P. Princen, A.W. Johnson, and A.B. Bradley. Subband/transform coding using filter bank designs based on tdac. In *Proceedings of the 1987 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'87)*, pages 2161–2164, Dallas, USA, April 1987.

[4] J. Herre and J.D. Johnston. Enhancing the performance of perceptual audio coders by using temporal noise shaping. In *Proceedings of the 101st AES Convention*, Los Angeles, USA, November 1996.

[5] J. Herre and J.D. Johnston. Continuously signal-adaptive filterbank for high-quality perceptual audio coding. In *Proceedings of the 1997 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'97)*, New Paltz, USA, October 1997.

[6] M. Bosi and et al. ISO/IEC MPEG-2 advanced audio coding. *Journal of the Audio Engineering Society*, 45:789–812, October 1997.

[7] 3rd Generation Partnership Project (3GPP). TS 26.403 V6.0.0: Encoder specification AAC part (release 6). ftp://ftp.3gpp.org/specs/2004-12/Rel-6/26_series/26403-600.zip, November 2004.

[8] C.M. Liu, W.C. Lee, and T.W. Chang. The efficient temporal noise shaping method. In *Proceedings of the 116th AES Convention*, Berlin, Germany, May 2004.

[9] S. van de Par, A. Kohlrausch, G. Charestan, and R. Heusdens. A new psychoacoustical masking model for audio coding applications. In *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'02)*, pages 1805–1808, Orlando, USA, May 2002.

[10] O. Niemeyer and B. Edler. Efficient coding of excitation patterns combined with a transform audio coder. In *Proceedings of the 118th AES Convention*, Barcelona, Spain, May 2005.

[11] Radiocommunication Sector BS.1534-1 International Telecommunications Union. Method for the subjective assessment of intermediate quality level coding systems - general requirements (MUSHRA), 2003.

Chapter 10

Results and Recommendations

The objective of the work presented in this thesis was to study the combination of an operational RD optimization framework with adaptive MDCT-based time-frequency decomposition techniques. The first part of this thesis has provided background material of the relevant concepts of operational RD optimization and best basis search, and on the properties of the MDCT and some available adaptive time-frequency composition techniques. In the second part, results have been presented for three distinct adaptive time-frequency decomposition techniques. This chapter summarizes the results as presented in this thesis. Furthermore, we make some recommendations for further research.

10.1 Summary of results

Firstly, we have investigated adaptive frequency decomposition and have developed an algorithm for flexible frequency decomposition. This algorithm employed subband merging to construct a nonuniform MDCT and dynamic programming for fast searching, techniques that allow for fast and simple adaptation of the frequency decomposition depending on the input signal. The algorithm was implemented in a basic audio coding scheme and experimental results have shown that an SNR gain of up to 4 dB can be obtained for some fragments. Furthermore, it was shown that on average, listeners have an 80% preference for a simple perceptual audio coding scheme that incorporates the new algorithm. However, when incorporating the algorithm in this perceptual audio coding scheme, we have observed that the description of the resulting decomposition puts severe demands on the lossless coding of the associated side information and that it is not efficient in terms of bit rate when averaged over a large set of audio fragments. Moreover, incorporation of the algorithm in a generic MDCT-based audio coding scheme has led to a highly increased computational complexity. We therefore conclude that this particular frequency domain approach in its current form does not provide us with a performance increase that can justify the increase in complexity. Further steps can be made by restricting the decomposition search space.

Secondly, we have proposed an extension to existing time segmentation techniques in the form of a flexible time segmentation algorithm. Again, dynamic programming has been employed for best basis search through the dictionary of time segmentations. We have devised three variations that can cover a large range of complexity trade-offs. Initially, a medium-complexity solution was constructed that neglected the dependencies resulting from varying window overlap. This algorithm was shown to outperform an existing time segmentation method, for the class of fixed overlap windows. We then proceeded to quantify the loss incurred by neglecting the dependencies in the case of varying window overlap. The average loss was determined to be 12% for the Coifman-Wickerhauser entropy cost measure and 0.5 dB for a rate-distortion cost measure. Furthermore, we showed that an optimal solution could be obtained in polynomial time. Next, we implemented the medium-complexity algorithm in an audio coding system. In a direct comparison with MPEG-4 standardized coding systems, this audio coding system incorporating our time segmentation scheme performed equally good or better over a large range of bit rates. In order to obtain a low-complexity algorithm, the RD cost measure was replaced by a cost measure based on perceptual entropy. This third scheme showed a negligible performance loss compared to the RD-based scheme, at an average complexity reduction of 47%. We conclude that the low-complexity time segmentation algorithm can be a viable alternative to the time segmentation methods employed in existing audio coding schemes.

Finally, we returned to frequency decomposition and concentrated on RD optimal temporal noise shaping. We have developed an algorithm in which the linear prediction filter order of the TNS method was optimized for a perceptual distortion measure. Compared to an earlier method, which employed a thresholding scheme based on prediction gain, the algorithm showed improved performance, both in terms of RD behavior, where on average a perceptually weighted SNR gain of 1 dB was obtained, and subjective test results, where a 1.5 point MOS difference was observed. On the other hand, the initial reduction of pre-echo and double speak artifacts that was obtained with standard TNS was diminished. While this loss of artifact reduction can be partly attributed to the relatively simple perceptual model that was employed, we believe that the operation of the TNS technique outside RD optimization remains an efficient way of reducing aforementioned artifacts. Nevertheless, the overall performance of RD optimal TNS is substantial and we foresee a promising combination of this algorithm with the flexible time segmentation method. Extensions of this method can be made by using a dynamic programming approach to compute the optimal combination of filters, filter orders and frequency ranges for every block of transform coefficients.

10.2 Recommendations

To conclude this chapter, we provide some recommendations for future research and for the general application of operational RD optimization in audio coding. As stated in section 1.2, we have mainly focussed on the signal processing aspects of the combination of an RD optimization framework with adaptive MDCT-based time-frequency decomposition. Hence, the proposed algorithms have not been implemented in fully functional audio coding schemes, apart from the work presented in chapter 8. As a first extension of our research, we recommend implementing the temporal noise shaping scheme from chapter 9 in the audio coding system employed in 8. Additionally, temporal masking can be incorporated by replacing the existing perceptual model with a model derived from Dau [3, 2]. In the author's opinion, these enhancements can lead to an audio coding scheme that can compete with state-of-the-art audio coding schemes. Optimally, RD optimization will enable a comparison of audio coding schemes similar to the benchmark test for video coding by Wiegand *et al* in [6].

We conclude this chapter with an outlook on new and ongoing research that will help in making RD optimization a standard tool for audio coding. First, computational complexity will be an issue for years to come. In the algorithms presented in this work, we incorporated best basis search methods to significantly lower computational complexity. However, in most cases the computation of the values for bit rate and distortion dominate the complexity figures. To lower computational complexity in this area, we can seek to employ high-rate quantization [9, 10] to analytically determine the RD costs or feature-based techniques [7, 4] to estimate the RD behavior. Furthermore, new and improved perceptual distortion models have to be devised for better modelling of the human auditory systems and more accurate computation of perceptual distortions such as done in [1, 8]. Lastly, practical solutions have to be provided for optimization over multiple coding strategies in a universal audio coding scheme. Within such a scheme, an MDCT-based subcoding module can be envisioned that employs the techniques described in this thesis. The challenge then lies in the investigation and derivation of a jointly optimal RD optimization scheme. Initial steps to this end have been made in the SiCAS [5] and ARDOR [11] projects.

Bibliography

[1] M.G. Christensen and S.H. Jensen. On perceptual distortion minimization and nonlinear least-squares frequency estimation. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1):99–109, January 2006.

[2] T. Dau, B. Kollmeier, and A. Kohlrausch. Modelling auditory processing of amplitude modulation. i. detection and masking with narrowband carriers. *Journal of the Acoustical Society of America*, 102(5):2892–2905, November 1997.

[3] T. Dau, D. Püschel, and A. Kohlrausch. A quantitative model of the effective signal processing in the auditory system. i. model structure. *Journal of the Acoustical Society of America*, 99(6):3615–3622, June 1996.

[4] C.A. Rødbro, M.G. Christensen, F. Nordén, and S.H. Jensen. Low complexity rate-distortion optimized time-segmentation for audio coding. In *Proceedings of the 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'05)*, pages 231– 234, New York, USA, October 2005.

[5] Heusdens et al. Bit-rate scalable intra-frame sinusoidal audio coding based on rate-distortion optimisation. *Journal of the Audio Engineering Society*, 1:1–2, March 2006.

[6] T. Wiegand et al. Rate-constrained coder control and comparison of video coding standards. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):688–703, July 2003.

[7] F. Nordén, M.G. Christensen, and S.H. Jensen. Open loop rate-distortion optimized audio coding. In *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'05)*, pages III 161–164, Philadelphia, USA, May 2005.

[8] Jan H. Plasberg and W. Bastiaan Kleijn. The sensitivity matrix: Using advanced auditory models in speech and audio processing. *to appear in IEEE Transactions on Audio, Speech and Language Processing*, January 2007.

[9] R. Vafin and W.B. Kleijn. Entropy-constrained polar quantization and its application to audio coding. *IEEE Transactions on Speech and Audio Processing*, 13(2):220–232, March 2005.

[10] R. Vafin and W.B. Kleijn. Rate-distortion optimized quantization in multistage audio coding. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1):311–320, January 2006.

[11] N.H. van Schijndel and S. van de Par. Rate-distortion optimized hybrid sound coding. In *Proceedings of the 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'05)*, pages 235–238, New York, USA, October 2005.

Appendix A

Perfect Reconstruction of the MDCT

In this appendix we look at the perfect reconstruction property of the MDCT and derive the proof for Theorem 2.

For the ith input signal block y_i depicted in Fig. 3.2 and an MDCT analysis window h, let the $2M$ MDCT analysis signals X_i be defined as

$$X_i(k) = \sqrt{\frac{2}{M}} \sum_{n=0}^{2M-1} h(2M-1-n) y_i(n) \cos\left[\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n+n_0\right)\right], \tag{A.1}$$

for $k = 0, 1, \ldots, M-1$. Furthermore, let $\hat{y}_i$, i.e. the time-aliased reconstruction of y_i, be obtained from X_i as

$$\hat{y}_i(n) = \sqrt{\frac{2}{M}} \sum_{k=0}^{M-1} X_i(k) \cos\left[\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n+n_0\right)\right], \tag{A.2}$$

for $n = 0, 1, \ldots, 2M-1$ and let f be an MDCT synthesis window. From Fig. 3.3 it is seen that the ith signal block $\hat{x}_i$ is reconstructed as

$$\hat{x}_i(n) = f(n+M)\hat{y}_{i-1}(n+M) + f(n)\hat{y}_i(n). \tag{A.3}$$

Let $h'(n) = h(2M-1-n)$ and let $\varphi_k = \frac{\pi}{M}\left(k+\frac{1}{2}\right)$. Then, by substituting (A.1) for

X_i in (A.2), $\hat{y}_i(n)$ can be written as

$$
\begin{aligned}
\hat{y}_i(n) &= \frac{1}{2M}\sum_{k=0}^{2M-1} X_i(k)\cos\Big[\varphi_k\Big(n+n_0\Big)\Big] \\
&= \frac{1}{2M}\sum_{k=0}^{2M-1}\sum_{m=0}^{2M-1} x_i(m)h'(m)\cos\Big[\varphi_k\Big(m+n_0\Big)\Big]\cos\Big[\varphi_k\Big(n+n_0\Big)\Big] \\
&= \frac{1}{2}\sum_{m=0}^{2M-1} x_i(m)h'(m)\Bigg(\frac{1}{2M}\sum_{k=0}^{2M-1}\cos\Big[\varphi_k\Big(m-n\Big)\Big]+ \\
&\qquad\qquad \frac{1}{2M}\sum_{k=0}^{2M-1}\cos\Big[\varphi_k\Big(m+n+2n_0\Big)\Big]\Bigg) \\
&= \frac{1}{2}\sum_{m=0}^{2M-1} x_i(m)h'(m)\Bigg(\frac{\cos\Big[\frac{\pi(m-n)}{2M}\Big]}{2M}\sum_{k=0}^{2M-1}\cos\Big[\frac{\pi k(m-n)}{M}\Big]- \\
&\qquad\qquad \frac{\sin\Big[\frac{\pi(m-n)}{2M}\Big]}{2M}\sum_{k=0}^{2M-1}\sin\Big[\frac{\pi k(m-n)}{M}\Big]+ \\
&\qquad\qquad \frac{\cos\Big[\frac{\pi(m+n+2n_0)}{2M}\Big]}{2M}\sum_{k=0}^{2M-1}\cos\Big[\frac{\pi k(m+n+2n_0)}{M}\Big]- \\
&\qquad\qquad \frac{\sin\Big[\frac{\pi(m+n+2n_0)}{2M}\Big]}{2M}\sum_{k=0}^{2M-1}\sin\Big[\frac{\pi k(m+n+2n_0)}{M}\Big]\Bigg) \\
&= \frac{1}{2}\sum_{m=0}^{2M-1} x_i(m)h'(m)\sum_{\ell=-\infty}^{\infty}(-1)^{\ell}\Big(\delta(m-n-2\ell M)+\delta(m+n+2n_0-2\ell M)\Big) \\
&= \frac{1}{2}\sum_{\ell=-\infty}^{\infty}(-1)^{\ell}\Bigg(x_i(n+2\ell M)h(2M-1-n-2\ell M)+ \\
&\qquad\qquad x_i(-n-2n_0+2\ell M)h(2M-1+n+2n_0-2\ell M)\Bigg).
\end{aligned}
\tag{A.4}
$$

for $n = 0, 1, \ldots, 2M-1$.

Combining the result from (A.4) with (A.3) and choosing $2n_0 = M+1$ leads to

$$\begin{aligned}
\hat{x}_i(n) &= \frac{1}{2}f(n+M)\sum_{\ell=-\infty}^{\infty}(-1)^{\ell}x_{i-1}(n+M+2\ell M)h(M-1-n-2\ell M) \\
&+ \frac{1}{2}f(n+M)\sum_{\ell=-\infty}^{\infty}(-1)^{\ell}x_{i-1}(-n-2M-1+2\ell M)h(4M+n-2\ell M) \\
&+ \frac{1}{2}f(n)\sum_{\ell=-\infty}^{\infty}(-1)^{\ell}x_{i}(n+2\ell M)h(2M-1-n-2\ell M) \\
&+ \frac{1}{2}f(n)\sum_{\ell=-\infty}^{\infty}(-1)^{\ell}x_{i}(-n-M-1+2\ell M)h(3M+n-2\ell M). \qquad \text{(A.5)}
\end{aligned}$$

Only a few terms in (A.5) contribute to the output over the range $n = 0, 1, \ldots, M-1$. Using the fact that $x_{i-1}(n+M) = x_i(n)$ for $n = 0, 1, \ldots, M-1$ leads to

$$\begin{aligned}
\hat{x}_i(n) &= \frac{1}{2}x_i(n)\Big(f(n+M)h(M-1-n) + f(n)h(2M-1-n)\Big) \\
&+ \frac{1}{2}x_{i-1}(2M-1-n)\Big(f(n+M)h(n) - f(n)h(n+M)\Big), \qquad \text{(A.6)}
\end{aligned}$$

and we see that (A.3) is equal to (A.6), that is,

$$\begin{aligned}
\hat{x}_i(n) &= f(n+M)\hat{y}_{i-1}(n+M) + f(n)\hat{y}_i(n) \\
&= \frac{1}{2}x_i(n)\Big(f(n+M)h(M-1-n) + f(n)h(2M-1-n)\Big) \\
&+ \frac{1}{2}x_{i-1}(2M-1-n)\Big(f(n+M)h(n) - f(n)h(n+M)\Big).
\end{aligned}$$

Now we can prove Theorem 2, which we repeat below for completeness.

Theorem 2 ***(perfect reconstruction property of the MDCT)*** *For the ith input signal block y_i and MDCT analysis window h, let the $2M$ MDCT analysis signals X_i be defined as*

$$X_i(k) = \sqrt{\frac{2}{M}}\sum_{n=0}^{2M-1}h(2M-1-n)y_i(n)\cos\Big[\frac{\pi}{M}\Big(k+\frac{1}{2}\Big)\Big(n+n_0\Big)\Big].$$

Furthermore, let $\hat{y}_i$ be obtained from X_i as

$$\hat{y}_i(n) = \sqrt{\frac{2}{M}}\sum_{k=0}^{M-1}X_i(k)\cos\Big[\frac{\pi}{M}\Big(k+\frac{1}{2}\Big)\Big(n+n_0\Big)\Big],$$

and let f be the MDCT synthesis window.

The ith signal block $\hat{x}_i$ can be perfectly reconstructed if

$$\begin{aligned} f(n) &= h(n), & n &= 0,\ldots,2M-1, & \text{(A.7)}\\ h(n) &= h(2M-1-n), & n &= 0,\ldots,M-1, & \text{(A.8)}\\ h^2(n)+h^2(n+M) &= 1, & n &= 0,\ldots,M-1, & \text{(A.9)} \end{aligned}$$

and $2n_0 = M+1$.

Proof: The $2M$ MDCT coefficients are not independent but satisfy

$$X(k) = -X(2M-k-1), \quad k = 0,1,\ldots,M-1.$$

Therefore, only M MDCT coefficients are required for reconstruction. The ith signal block $\hat{x}_i$ is reconstructed as (A.3)

$$\hat{x}_i(n) = f(n+M)\hat{y}_{i-1}(n+M) + f(n)\hat{y}_i(n).$$

From (A.6) it follows that for $2n_0 = M+1$, (A.3) is equivalent to

$$\begin{aligned} \hat{x}_i(n) \quad &= \quad x_i(n)\Big(f(n+M)h(M-1-n) + f(n)h(2M-1-n)\Big)\\ &+ \quad x_{i-1}(2M-1-n)\Big(f(n+M)h(n) - f(n)h(n+M)\Big). \end{aligned}$$

Hence, $\hat{x}_i = x_i \ \forall i \in \mathbb{N}$ if (A.7)-(A.9) are satisfied. $\Box$

Appendix B

Frequency Domain Linear Prediction

In this appendix we study frequency domain linear prediction. We start with introducing the Hilbert transformer. Next, we study the relation between the squared Hilbert envelope and the spectral autocorrelation function. We then proceed by employing the results in frequency domain linear prediction.

B.1 The Hilbert transformer

Let x be a real-valued discrete time signal and let the complex-valued signal c be defined as

$$c(n) = x(n) + j\mathcal{H}\{x(n)\}, \tag{B.1}$$

where $\mathcal{H}\{\centerdot\}$ denotes a *Hilbert transformer* [1]. Moreover, let $c_a(t)$ be an analytic band-limited continuous time signal that satisfies

$$C_a(j\omega) = \begin{cases} C(e^{j\omega}), & 0 \leq \omega < \pi, \\ 0, & \text{otherwise.} \end{cases}$$

If $c_a(t)|_{t=n} = c(n)$, we can say that c corresponds to an analytic signal.

The Hilbert transformer operation on x is given as

$$\mathcal{H}\{x(n)\} = \sum_{m=-\infty}^{\infty} h(m)x(n-m),$$

with

$$h(n) = \begin{cases} \frac{2sin^2(\pi n/2)}{\pi n}, & n \neq 0, \\ 0, & n = 0. \end{cases}$$

and

$$H(e^{j\omega}) = \begin{cases} -j, & 0 \leq \omega < \pi, \\ j, & -\pi \leq \omega < 0. \end{cases}$$

That is, the Hilbert transformer is a 90-degree phase shifter.

B.2 Relation Between Squared Hilbert Envelope And Spectral Autocorrelation

The derivations presented here were originally published in [2]. Let X be the K-point DFT of x, i.e.

$$X(k) = \mathcal{F}\{x\} = \sum_{k=0}^{K-1} x(n)e^{-j2\pi kn/K}, \quad k = 0, \ldots, K-1.$$

The DFT of the analytic signal c from (B.1) is given as

$$C(k) = \sum_{k=0}^{K-1} c(n)e^{-j2\pi kn/K} = \begin{cases} 2X(k), & 0 < k < K/2, \\ X(k), & k = 0, \\ 0, & -K/2 \leq k < 0. \end{cases} \tag{B.2}$$

Let e denote the squared envelope of c, that is,

$$e(n) = |c(n)|^2 = c(n)c^*(n).$$

The signal e is called the squared Hilbert envelope of x. The DFT of e is

$$E(k) = C(k) * C^*(k) = \sum_{\ell=0}^{K-1} C(\ell)C^*(k-\ell). \tag{B.3}$$

Clearly, E is a spectral autocorrelation sequence. Taking the IDFT of E gives

$$e(n) = \mathcal{F}^{-1}\Big\{\sum_{\ell=0}^{K-1} C(\ell)C^*(k-\ell)\Big\}, \tag{B.4}$$

and it is observed that the squared envelope e of c, or equivalently, the squared Hilbert envelope of x, is the inverse Fourier transform of the spectral autocorrelation sequence E. This relation is the dual of the Wiener-Khinchin Theorem [3], which relates the power spectral density S_{xx} of a wide-sense stationary signal x to its autocorrelation function R_{xx} as

$$S_{xx}(f) = \mathcal{F}\{R_{xx}(\tau)\} = \mathcal{F}\Big\{\int_{\tau} x(\tau)x^*(\tau - t)\mathrm{d}\tau\Big\}.$$

B.3 Frequency Domain Linear Prediction

In the frequency domain, a pth order linear prediction $\tilde{X}$ of X is given by [4]

$$\tilde{X}(k) = \sum_{i=1}^{p} a_i X(k-i).$$

In general, if $p < K$, the prediction leads to a prediction error R, i.e.

$$R(k) = X(k) - \tilde{X}(k) = X(k) - \sum_{i=1}^{p} a_i X(k-i). \tag{B.5}$$

The set of p prediction filter coefficients $\{a_i\}$ is obtained by minimizing the squared prediction error as

$$\{a_i\} = \arg\min \Big| X(k) - \sum_{i=1}^{p} a_i X(k-i) \Big|^2,$$

which can be done efficiently using standard methods, e.g. the Levinson-Durbin algorithm [5, 6].

Linear prediction in the time domain results in a spectrally flat or white prediction error [4]. Dual to this, frequency domain linear prediction leads to a flattening of the temporal envelope of the prediction error r. This can be seen from (B.3) and (B.4). Linear prediction leads to decorrelation of X and, according to (B.2), of C. If C is maximally decorrelated, $E = 0$ for $k \neq 0$ and e is constant, i.e. has a flat envelope.

From (B.5) and (B.2), the squared Hilbert envelope e^p of the prediction error r is obtained by taking the IDFT of R as

$$e^p(n) = x(n) \Big(1 - \sum_{i=1}^{p} a_i e^{-j2\pi i n/K} \Big).$$

A pth order estimate of the envelope of x is now given as the (complex) inverse or synthesis filter h,

$$h(n) = \frac{x(n)}{e^p(n)} = \frac{1}{1 - \sum_{i=1}^{p} a_i e^{-j2\pi i n/K}}.$$

Bibliography

[1] A.V. Oppenheim and R.W. Schafer. *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1975.

[2] J. Herre and J.D. Johnston. Enhancing the performance of perceptual audio coders by using temporal noise shaping. In *101st AES Convention*, *Preprint 4384*, Los Angeles, USA, November 1996.

[3] M.B. Priestley. *Spectral Analysis and Time Series*. Academic Press, New York, 1981.

[4] N.S. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice-Hall, Englewood Cliffs, NJ, 1984.

[5] N. Levinson. The wiener rms (root mean square) error criterion in filter design and prediction. *Journal of Mathematical Physics*, 25:261–278, 1947.

[6] J. Durbin. The fitting of time-series models. *Rev. Inst. Int. Statist.*, 28:233–243, 1960.

Appendix C

Subband Merging

In this appendix we provide the proof for Theorem 3, which we repeat below for completeness.

Theorem 3 ***(subband merging)*** *Let p_0 denote a real-coefficient linear-phase low-pass prototype filter of length N for an M-channel PR uniform CMFB, satisfying*

$$|P_0(e^{j\omega})| = 0 \quad \textit{for } |\omega| \geq \frac{\pi}{2M} + \varepsilon,\ \varepsilon < \frac{\pi}{2M}, \tag{C.1}$$

and let $b_k = e^{j\varphi_k},\ k = 0, \ldots, M-1$. Furthermore, let the analysis filters h_k of the CMFB be defined as in (3.18).

Then

$$\Big|\sum_{i=0}^{p-1} b_{k+i} H_{k+i}(z)\Big|^2 = \sum_{i=0}^{p-1} \Big|H_{k+i}(z)\Big|^2, \tag{C.2}$$

for $1 \leq p \leq M$ and $0 \leq k \leq M-p$, if and only if $\alpha = (N-1) - M(2m+1),\ m \in \mathbb{Z}$, and $|\varphi_k - \varphi_{k+1}| = n\pi,\ n \in \mathbb{N}$.

Proof: We can write the $\mathcal{Z}$-transform of h_k from (3.18) as a sum of two filters U_k and V_k,

$$H_k(z) = a_k U_k(z) + a_k^* V_k(z), \tag{C.3}$$

with

$$\left\{ \begin{array}{lcl} U_k(z) & = & P_0(zW^{(k+\frac{1}{2})}), \\ V_k(z) & = & P_0(zW^{-(k+\frac{1}{2})}), \\ a_k & = & W^{(k+\frac{1}{2})\frac{\alpha}{2}}, \end{array} \right. \tag{C.4}$$

and $W = e^{-j2pi/M}$. Furthermore, for $\alpha = (N-1) - M(2m+1),\ m \in \mathbb{Z}$,

$$\sum_{i=0}^{p-1} \Big|H_{k+i}(z)\Big|^2 = \sum_{i=0}^{p-1} \Big|U_{k+i}(z)\Big|^2 + \Big|V_{k+i}(z)\Big|^2. \tag{C.5}$$

Let $H_{p,k}(z)$ be defined as in (3.19). From (C.3) and the linear-phase property of $P_0(z)$, we can write $|H_{p,k}(z)|^2$ as

$$\begin{aligned}|H_{p,k}(z)|^2 =& \sum_{i=0}^{p-1}\sum_{\ell=0}^{p-1} z^{N-1}\Bigg(b_{k+i}b^*_{k+\ell}a_{k+i}a_{k+\ell}c^{*^2}_{k+\ell}U_{k+i}(z)V_{k+\ell}(z)+ \\ & b^*_{k+i}b_{k+\ell}a^*_{k+i}a^*_{k+\ell}c^2_{k+i}U_{k+i}(z)V_{k+\ell}(z)\Bigg) \\ &+\sum_{i=0}^{p-1}\sum_{\ell=0}^{p-1} z^{N-1}\Bigg(b_{k+i}b^*_{k+\ell}a_{k+i}a^*_{k+\ell}c^2_{k+\ell}U_{k+i}(z)U_{k+\ell}(z)+ \\ & b^*_{k+i}b_{k+\ell}a^*_{k+i}a_{k+\ell}c^{*^2}_{k+\ell}V_{k+i}(z)V_{k+\ell}(z)\Bigg),\end{aligned} \tag{C.6}$$

with $c_k = W^{(k+\frac{1}{2})\frac{N-1}{2}}$ and $b_k = e^{j\varphi_k}$. Eq.(C.6) can be divided into four parts as

$$|H_{p,k}(z)|^2 = \sum_{i=0}^{p-1}\Big(|U_{k+i}(z)|^2 + |V_{k+i}(z)|^2\Big) \tag{C.7a}$$

$$\begin{aligned}&+\sum_{i=0}^{p-2} z^{N-1}\Bigg(b_{k+i}b^*_{k+i+1}a_{k+i}a^*_{k+i+1}c^2_{k+i+1}U_{k+i}(z)U_{k+i+1}(z)+ \\ & b^*_{k+i}b_{k+i+1}a^*_{k+i}a_{k+i+1}c^2_{k+i}U_{k+i}(z)U_{k+i+1}(z)+ \\ & b^*_{k+i}b_{k+i+1}a_{k+i}a^*_{k+i+1}c^{*^2}_{k+i}V_{k+i}(z)V_{k+i+1}(z)+ \\ & b^*_{k+i}b_{k+i+1}a^*_{k+i}a_{k+i+1}c^{*^2}_{k+i}V_{k+i}(z)V_{k+i+1}(z)\Bigg)\end{aligned} \tag{C.7b}$$

$$\begin{aligned}&+\sum_{i=0}^{p-1}\sum_{\ell=0}^{p-1} z^{N-1}\Bigg(b_{k+i}b^*_{k+\ell}a_{k+i}a_{k+\ell}c^{*^2}_{k+\ell}U_{k+i}(z)V_{k+\ell}(z)+ \\ & b^*_{k+i}b_{k+\ell}a^*_{k+i}a^*_{k+\ell}c^2_{k+i}U_{k+i}(z)V_{k+\ell}(z)\Bigg)\end{aligned} \tag{C.7c}$$

$$\begin{aligned}&+\sum_{i=0}^{p-1}\sum_{\ell=0}^{p-1} z^{N-1}\Bigg(b_{k+i}b^*_{k+\ell}a_{k+i}a^*_{k+\ell}c^2_{k+\ell}U_{k+i}(z)U_{k+\ell}(z)+ \\ & b_{k+i}b^*_{k+\ell}a^*_{k+i}a_{k+\ell}c^{*^2}_{k+\ell}V_{k+i}(z)V_{k+\ell}(z)\Bigg),\end{aligned} \tag{C.7d}$$

where (C.7d) is only considered for $|i-\ell| \geq 2$.

In the remainder of this proof we show that the terms in (C.7b)-(C.7d) vanish *if and only if* $\alpha = (N-1) - M(2m+1),\ m \in \mathbb{Z}$ and $|\varphi_k - \varphi_{k+1}| = n\pi,\ n \in \mathbb{N}$. The proof is by induction to p.

Let $p = 2$. Then $H_{2,k}(z) = b_k H_k(z) + b_{k+1} H_{k+1}(z)$, hence

$$\begin{aligned}
|H_{2,k}(z)|^2 &= |b_k H_k(z) + b_{k+1} H_{k+1}(z)|^2 \\
&= |U_k(z)|^2 + |V_k(z)|^2 + |U_{k+1}(z)|^2 + |V_{k+1}(z)|^2 \\
&+ z^{N-1}\Big(b_k b_{k+1}^* a_k a_{k+1}^* c_{k+1}^2 + b_k^* b_{k+1} a_k^* a_{k+1} c_k^2\Big) U_k(z) U_{k+1}(z) \\
&+ z^{N-1}\Big(b_k^* b_{k+1} a_k a_{k+1}^* {c_k^*}^2 + b_k b_{k+1}^* a_k^* a_{k+1} {c_k^*}^2\Big) V_k(z) V_{k+1}(z) \\
&+ z^{N-1}\Big(a_k^2 {c_k^*}^2 + {a_k^*}^2 c_k^2\Big) U_k(z) V_k(z) \\
&+ z^{N-1}\Big(b_k b_{k+1}^* a_k a_{k+1} {c_{k+1}^*}^2 + b_k^* b_{k+1} a_k^* a_{k+1}^* c_k^2\Big) U_k(z) V_{k+1}(z) \\
&+ z^{N-1}\Big(b_k^* b_{k+1} a_{k+1} a_k {c_k^*}^2 + b_k b_{k+1}^* a_{k+1}^* a_k^* c_{k+1}^2\Big) U_{k+1}(z) V_k(z) \\
&+ z^{N-1}\Big(a_{k+1}^2 {c_{k+1}^*}^2 + {a_{k+1}^*}^2 c_{k+1}^2\Big) U_{k+1}(z) V_{k+1}(z).
\end{aligned}$$

Now

$$\Big(b_k b_{k+1}^* a_k a_{k+1}^* c_{k+1}^2 + b_k^* b_{k+1} a_k^* a_{k+1} c_k^2\Big) = 0,$$

if

$$e^{2j(\varphi_k - \varphi_{k+1})} e^{-j\frac{\pi}{M}[-\frac{\alpha}{2} + (k+1)(N-1)]} = e^{-j\pi(2m+1)} e^{-j\frac{\pi}{M}[\frac{\alpha}{2} + k(N-1)]},\ m \in \mathbb{Z},$$

which is the case *if and only if* $\alpha = (N-1) - M(2m+1) - (\varphi_k - \varphi_{k+1}) 2M/\pi,\ m \in \mathbb{Z}$.

Similarly,

$$\Big(b_k^* b_{k+1} a_k a_{k+1}^* {c_k^*}^2 + b_k b_{k+1}^* a_k^* a_{k+1} {c_k^*}^2\Big) = 0,$$

if

$$e^{2j(\varphi_{k+1} - \varphi_k)} e^{-j\frac{\pi}{M}[-\frac{\alpha}{2} - k(N-1)]} = e^{-j\pi(2m+1)} e^{-j\frac{\pi}{M}[\frac{\alpha}{2} - (k+1)(N-1)]},\ m \in \mathbb{Z},$$

which occurs *if and only if* $\alpha = (N-1) - M(2m+1) - (\varphi_{k+1} - \varphi_k) 2M/\pi,\ m \in \mathbb{Z}$.

Moreover, if $P_0(z)$ satisfies (C.1), $U_{k1}(z) V_{k2}(z) = 0\ \forall\ k1, k2$, except for two situations. If $k1 = k2 = 0$ then

$$(a_0^2 {c_0^*}^2 + {a_0^*}^2 c_0^2) = 0 \text{ for } \alpha = (N-1) + M(2m+1),\ m \in \mathbb{Z}.$$

Let $m = -(m' + 1), m' \in \mathbb{Z}$, then $\alpha = (N-1) - M(2m' + 1)$.

If $k1 = k2 = M-1$ then it follows that

$$(a_{M-1}^2 c_{M-1}^{*^2} + a_{M-1}^{*^2} c_{M-1}^2) = 0 \text{ for } \alpha = (N-1) + \frac{M(2m+1)}{2M-1},\ m \in \mathbb{Z}.$$

Let $m = (1-2M)m'' - M, m'' \in \mathbb{Z}$, then $\alpha = (N-1) - M(2m''+1)$.

The conditions that need to be satisfied such that

$$|H_{2,k}(z)|^2 = |U_k(z)|^2 + |V_k(z)|^2 + |U_{k+1}(z)|^2 + |V_{k+1}(z)|^2,$$

are

$$\alpha = (N-1) - M(2m+1) - (\varphi_k - \varphi_{k+1})2M/\pi,\ m \in \mathbb{Z} \tag{C.8a}$$
$$\alpha = (N-1) - M(2m+1) - (\varphi_{k+1} - \varphi_k)2M/\pi,\ m \in \mathbb{Z}, \tag{C.8b}$$
$$\alpha = (N-1) - M(2\ell+1),\ \ell \in \mathbb{Z}. \tag{C.8c}$$

It is quickly derived that (C.8a)-(C.8c) are equal if

$$|\varphi_k - \varphi_{k+1}| = n\pi,\ n \in \mathbb{N}. \tag{C.9}$$

From (C.5) it is given that if (C.8c) is satisfied,

$$|H_k(z)|^2 + |H_{k+1}(z)|^2 = |U_k(z)|^2 + |V_k(z)|^2 + |U_{k+1}(z)|^2 + |V_{k+1}(z)|^2.$$

Therefore,

$$|b_k H_k(z) + b_{k+1} H_{k+1}(z)|^2 = |H_k(z)|^2 + |H_{k+1}(z)|^2,$$

if and only if (C.1), (C.8c) and (C.9) are satisfied and hence the induction hypothesis is true.

Let $p \geq 2$. Then $H_{p+1,k}(z) = \sum_{i=0}^{p} b_{k+i} H_{k+i}(z)$, hence

$$|H_{p+1,k}(z)|^2 =$$

$$\text{(C.7a)} + |U_{k+p}(z)|^2 + |V_{k+p}(z)|^2 + \tag{C.10a}$$

$$\begin{aligned}\text{(C.7b)} + z^{N-1}\Bigg(& b_{k+p-1} b^*_{k+p} a_{k+p-1} a^*_{k+p} c^2_{k+p} U_{k+p-1}(z) U_{k+p}(z) + \\ & b^*_{k+p-1} b_{k+p} a^*_{k+p-1} a_{k+p} c^2_{k+p-1} U_{k+p-1}(z) U_{k+p}(z) + \\ & b^*_{k+p-1} b_{k+p} a_{k+p-1} a^*_{k+p} {c^*}^2_{k+p-1} V_{k+p-1}(z) V_{k+p}(z) + \\ & b_{k+p-1} b^*_{k+p} a^*_{k+p-1} a_{k+p} {c^*}^2_{k+p} V_{k+p-1}(z) V_{k+p}(z) \Bigg) + \end{aligned} \tag{C.10b}$$

$$\begin{aligned}\text{(C.7c)} + \sum_{\ell=0}^{p} z^{N-1}\Bigg(& b_{k+p} b^*_{k+\ell} a_{k+p} a_{k+\ell} {c^*}^2_{k+\ell} U_{k+p}(z) V_{k+\ell}(z) + \\ & b^*_{k+p} b_{k+\ell} a^*_{k+p} a^*_{k+\ell} c^2_{k+p} U_{k+p}(z) V_{k+\ell}(z) \Bigg) \\ + \sum_{i=0}^{p-1} z^{N-1}\Bigg(& b_{k+i} b^*_{k+p} a_{k+i} a_{k+p} {c^*}^2_{k+p} U_{k+i}(z) V_{k+p}(z) + \\ & b^*_{k+i} b_{k+p} a^*_{k+i} a^*_{k+p} c^2_{k+i} U_{k+i}(z) V_{k+p}(z) \Bigg) + \end{aligned} \tag{C.10c}$$

$$\begin{aligned}\text{(C.7d)} + \sum_{\ell=0}^{p-2} z^{N-1}\Bigg(& b_{k+p} b^*_{k+\ell} a_{k+p} a^*_{k+\ell} c^2_{k+\ell} U_{k+p}(z) U_{k+\ell}(z) + \\ & b_{k+p} b^*_{k+\ell} a^*_{k+p} a_{k+\ell} {c^*}^2_{k+\ell} V_{k+p}(z) V_{k+\ell}(z) \Bigg) \\ + \sum_{i=0}^{p-2} z^{N-1}\Bigg(& b_{k+i} b^*_{k+p} a^*_{k+i} a_{k+p} c^2_{k+p} U_{k+i}(z) U_{k+p}(z) + \\ & b_{k+i} b^*_{k+p} a^*_{k+i} a_{k+p} {c^*}^2_{k+p} V_{k+i}(z) V_{k+p}(z) \Bigg). \end{aligned} \tag{C.10d}$$

By the induction hypothesis, the terms (C.7b)-(C.7d) reduce to zero *if and only if* $\alpha = (N-1) - M(2m+1),\ m \in \mathbb{Z}$ and $|\varphi_k - \varphi_{k+1}| = n\pi,\ n \in \mathbb{N}$.

First, consider (C.10b). Since

$$\left(b_{k+p-1}b_{k+p}^{*}a_{k+p-1}a_{k+p}^{*}c_{k+p}^{2} + b_{k+p-1}^{*}b_{k+p}a_{k+p-1}^{*}a_{k+p}c_{k+p-1}^{2}\right) = 0,$$

if

$$e^{2j(\varphi_{k+p-1}-\varphi_{k+p})}e^{-j\frac{\pi}{M}[-\frac{\alpha}{2}+(k+p)(N-1)]} = e^{-j\pi(2m+1)}e^{-j\frac{\pi}{M}[\frac{\alpha}{2}+(k+p-1)(N-1)]},$$

which holds if (C.8a) is true. Similarly,

$$\left(a_{k+p-1}a_{k+p}^{*}c_{k+p-1}^{*^2} + a_{k+p-1}^{*}a_{k+p}c_{k+p}^{*^2}\right) = 0,$$

if

$$e^{2j(\varphi_{k+p}-\varphi_{k+p-1})}e^{-j\frac{\pi}{M}[-\frac{\alpha}{2}-(k+p-1)(N-1)]} = e^{-j\pi(2z+1)}e^{-j\frac{\pi}{M}[\frac{\alpha}{2}-(k+p)(N-1)]},$$

which is the case if (C.8b) is satisfied. Again, these conditions can be reduced to (C.8c) and (C.9), respectively.

Next, consider (C.10c). If $P_0(z)$ satisfies (C.1) and (C.9) is true, $U_{k1}(z)V_{k2}(z) = 0 \;\forall\; k1, k2$.

Finally, consider (C.10d). It is observed that $U_{k1}(z)U_{k2}(z) = 0$ and $V_{k1}(z)V_{k2}(z) = 0 \;\forall\; k1, k2$ and $|k1 - k2| \geq 2$ if (C.1) is satisfied.

This completes the proof. $\square$

Samenvatting

Tegenwoordig is perceptuele audiocodering de *de facto* oplossing om met het efficiënt opslaan en versturen van digitale audio om te gaan. Wereldwijd worden aan consumenten gestandaardiseerde oplossingen aangeboden, die naar tevredenheid werken, indien ze op de juiste manier worden toegepast. Echter, de recente convergentie tussen consumentenelektronica en mobiele communicatie en het opkomen van alomtegenwoordige heterogene netwerkomgevingen met tijdvariërende bandbreedte -en vertragingsbeperkingen, leiden tot strenge eisen aan de mogelijkheden van de bestaande oplossingen en aan de gebruiker die moet kiezen uit een breed scala aan oplossingen. Dit kan gemakkelijk leiden tot situaties waarbij niet goed wordt aangesloten op de applicatie, zodat een audiocoderingssysteem buiten het bedoelde bereik wordt gebruikt. Nieuwe systemen zijn daarom nodig, die zich kunnen aanpassen aan de voorwaarden en beperkingen zoals die door de gebruiker en het netwerk worden opgelegd.

In dit proefschrift bestuderen we verscheidene technieken en combinaties daarvan, die we beschouwen als geschikte kandidaten voor integratie in nieuwe audiocoderingssystemen. In plaats van een volledig audiocoderingssysteem te ontwikkelen, concentreren we ons op de aspecten van de signaalverwerking en de interactie tussen deze technieken. In het eerste deel van dit proefschrift wordt een overzicht gegeven van twee technieken die we reeds aantreffen in verscheidene digitale signaalcoderingssystemen. Deze technieken dienen als ingrediënten voor de algoritmen die gepresenteerd worden in het tweede deel.

Eerst kijken we naar operationele *rate-distortion* (RD) optimalisering. Met operationele RD optimalisering proberen we de best haalbare prestaties te bereiken voor het coderen van een audiosignaal, gegeven de keuze voor het compressiekader of de codeeromgeving. In dit proefschrift kijken we opnieuw naar de literatuur over operationele RD optimalisering, formuleren we het bit allocatie probleem onder beperking van de *bit rate* en bestuderen we oplossingen voor dit probleem. We zijn daarbij voornamelijk geïnteresseerd in de interactie van een dergelijk RD optimaliseringskader met de tijd-frequentie decompositie van het signaal. Dit leidt tot een studie van *best basis* zoekalgoritmen en de combinatie daarvan met RD optimalisering.

In de meeste audiocoderingssystemen wordt de tijd-frequentie decompositie verkregen door de *modified discrete cosine transform*, of MDCT, toe te passen. Daarom onderzoeken we diverse eigenschappen van de MDCT, zoals de voorwaarden voor perfecte reconstructie, het ontwerp van vensters en snelle algoritmen. Voorts bekijken we drie verschillende adaptatietechnieken die beschikbaar zijn voor de MDCT om niet-uniforme tijd-frequentie decomposities te verkrijgen.

Het belangrijkste doel van het werk dat in dit proefschrift wordt gepresenteerd is het bestuderen van de combinatie van een operationeel RD optimaliseringskader met adaptieve MDCT-gebaseerde tijd-frequentie decompositie technieken. In het tweede deel worden nieuwe algoritmes en experimentele resultaten gepresenteerd voor de drie afzonderlijke decompositie technieken, in de vorm van wetenschappelijke artikelen.

We beginnen met een onderzoek naar adaptieve frequentie decompositie. Hierbij wordt *subband merging* gebruikt om een niet-uniforme MDCT te construeren en dynamisch programmeren wordt toegepast voor het snel zoeken naar de *best basis*. We tonen aan dat het ontworpen algoritme kan resulteren in een winst in SNR en tot hogere subjectieve luistertestresultaten kan leiden. Echter, we bemerken dat het verliesvrij coderen van de extra informatie gerelateerd aan de verkregen decomposities, leidt tot een hoge bit rate voor deze extra informatie en we concluderen dat deze specifieke frequentie domein benadering geen dusdanige prestatiewinst geeft die de stijging in complexiteit kan rechtvaardigen.

Vervolgens gaan we verder met adaptieve tijdsegmentatie, waar dynamisch programmeren wordt gebruikt voor de *best basis* zoektocht en *block switching* voor MDCT-gebaseerde tijdsegmentatie. Drie variaties van het basisalgoritme worden ontworpen die tezamen voorzien in een groot bereik aan uitruilmogelijkheden tussen complexiteit en prestatie. De gevolgen van een variërende vensteroverlap worden grondig bestudeerd en we laten zien dat een optimale oplossing kan worden verkregen binnen een polynomiale tijdsduur. Voorts maken we een rechtstreekse vergelijking tussen een nieuw audiocoderingssysteem dat ons tijdsegmentatie bevat met codeersystemen die zijn gestandaardiseerd binnen MPEG-4 en verkrijgen daarbij in een luistertest even goede of betere resultaten voor een groot bereik aan bit rates. Een *low complexity* variant van dit audiocoderingssysteem laat een verwaarloosbaar prestatieverlies zien.

Als laatste gaan we terug naar frequentiedecompositie en bestuderen we *temporal noise shaping*, waarbij lineaire predictie wordt toegepast in het frequentiedomein. We combineren *temporal noise shaping* met RD optimalisering om zo de orde van het predictiefilter en de selectie van kwantisatoren te controleren. Dit leidt tot een efficiënt algoritme dat beter presteert dan een bestaande methode om *temporal noise shaping* te controleren. Hoewel het algoritme de oorspronkelijke werking van *temporal noise shaping* deels teniet doet, is de prestatiewinst in termen van *rate-distortion* gedrag aanzienlijk.

Acknowledgements

Writing this thesis had so little to do with me alone and so much with the people surrounding me.

First, I would like to thank my supervisor Richard Heusdens for his support, confidence and inspiration that he has given me during all these years. Thank you for the just-in-time proofreading, for letting me shape my research as I saw fit and for setting an example of good research that I could live up to. Oh, and for obtaining the funding to equip the best listening room I ever set foot in, as well.

I thank professor Inald Lagendijk for using his sharp mind and broad knowledge to help me improve both this thesis and the accompanying propositions, and for providing a stimulating working environment within the ICT group.

Jesper Jensen, I have immensely enjoyed our collaborations and discussions, but most of all your Danish humor. Emile Hendriks, thanks for the support through the years, I've come a long way since I showed up at your door as a first-year student. Huib Lincklaen Arriëns, your practical assistance was of immense value to my experiments and our talks together were always entertaining.

If you spend most of your time sitting in a room doing research, you had better make sure you sit in there with people you like. I was incredibly fortunate to have the best room mates anyone could wish for.

Mathieu Charestan, it was great to embark together on the path of research. Ivo Shterev, you set an example of dedication to science. Pim Korten, it was a privilege to see you grow as a researcher and become confident with audio coding. Jan Østergaard, you are one of the smartest persons I ever met. It is beyond me how you combine that with also being one of the funniest guys that I know.

I want to thank all other members of the ICT group, among which Richard Hendriks and Ivo Batina. I owe much gratitude to the ICT support and rescue team; Ben, Hans, Janroel, Robbert, Annet and Anja. At the end of your thesis there is so much to take care of and you have so many questions. Kathy and Eugene, thanks for passing on your knowledge and experience in these matters. Special thanks to Charles, Harry and Danesh for the good times we had as students.

It was my privilege to collaborate with the large group of international researchers from KTH, France Telecom, Hannover University, Aalborg University and Philips Research, in the context of the SiCAS and ARDOR projects. I hope we meet again in the future, if only to drink some more beers.

I'd like thank B&W Group Nederland, represented by Rene Degens, for their financial support that enabled printing this thesis and for making the astounding B&W Nautilus$^{\text{TM}}$ 800 loudspeakers that I used in some of my listening tests.

The last year of writing was a tough one. I strongly feel that I was able to finish because of the support from my new colleagues at TNO, in particular the people from BVS, BAHN and PPQ. My room mates Dick van Smirren, Menno Bangma and Erik Boertjes deserve a special thank you, as well as Rixte Thomas.

My interest in audio coding followed from my love for music and I have played with some superb musicians. Erik, Olaf, Frits, Peter, Roger, Marco, I will always remember the great years and music with Sinister Street and I sincerely hope we can play together again some time. Jan, Marco, Marco, Erick, Reinier, Peter, Jeroen, Evelien, Frank, thanks for sharing the stage with me. Lennard, I look forward to our jam sessions. Amit and Eelco, we are about to record the ultimate guitar record. Besides being great musicians, everyone of you has been a great friend to me.

Speaking of friends, it felt so good to come back to Vlaardingen and to have all of you so close; Maaike & Simon, Marco & Lianne, Rik & Daphne, Robbert & Susan, Erica & Eelco, Ferry & Linda, Bernadette & Peter, thanks for your friendship.

Eelco de Vaal, you truly are a graphical design guru. I am absolutely thrilled about the cover design.

Ron Veldhoen, I am so glad that you have made your dreams come true, together with Janine. Thanks for being my friend throughout these years.

I know that I have forgotten to mention people who have contributed in some way to this thesis. Please allow me to make it up to you in the future.

None of this would have been possible without the support and love from my family.

I thank the family De Ridder for their interest in my work and for their help in building us a new home.

To my beloved cat Boris, thank you for bringing so much joy into my life and for staying with me deep into the night. Losing you has been difficult, yet I will always cherish the moments we had.

To my sister Tamar, I can but marvel at the strength you show while faced with so many burdens and I hope you continue to set an example for the rest of us.

To my brother Farid, thank you for not giving up and for finishing what you started. I wish you, Nynke and my first nephew Raoul all the best for the future.

To my parents, it is you who have made this thesis possible, with all the love, opportunities and advice you have given me.

Petra, thank you for loving me and for staying with me during the good and bad times. I know now that I can face any future challenge with you at my side.

Curriculum Vitae

Omar Aziz Niamut was born in Rotterdam, the Netherlands, on January 17, 1978. He obtained his VWO-diploma from the Stedelijk Gymnasium Schiedam in 1996. He then went on to study Electrical Engineering at the Delft University of Technology. In 2000, during a three-month internship at Philips Research, he worked on a detection system for multi-channel audio. In 2001, he graduated from the Information and Communication Theory Group, where he developed a new design method for nonuniform filter banks.

In 2001, he started as a Ph.D. student at the Information and Communication Theory Group, Department of Mediamatics, at the Delft University of Technology. In the first year, he worked in the SiCAS project funded by STW and Philips Research. The project was aimed at developing a generic audio coding system that could compete with application-optimized systems. In 2002, he joined the ARDOR research project funded by the European Union. The focus of this project was to develop a universal audio codec. During his Ph.D., he served as a teaching assistant and lecturer for various courses in the fields of digital signal processing and audio coding. He developed multiple toolboxes and software for demonstrations and workshops. From 2004 to 2005, he served as the representative of DITOO. In 2005, he contributed to the development of the MP3 project, an inter-disciplinary education project for first year students of the EEMCS faculty.

Since 2006, he has been working as a scientist in the Broadband and Voice Solutions department, business unit Wireline, of TNO Information and Communication Technology in Delft. Here, his work focusses on digital video broadcast techniques, IPTV and speech quality assessment.

He is a member of the Audio Engineering Society and the IEEE Signal Processing and Circuits and Systems Societies. He is also an accomplished guitarist and recorded the album Trust (Musea Records 2002) with progressive rock band Sinister Street. Since 2005, he regularly performs live with rock cover band 2Big4Words.