Ye Wang

Selected Advances in Audio Compression and Compressed Domain Processing

Abstract

Error resilient audio coding is an essential enabling technology for on-line music delivery in wireless networks. Three crucial requirements for the transmission of audio over mobile networks are compression efficiency, computational simplicity and error resilience. This dissertation concentrates on the development of novel solutions in these three areas.

The first contribution of this dissertation is the development of an equivalent rectangular band (ERB) based masking model and its incorporation into an audio encoder. Most currently employed psychoacoustic models are based on the Bark frequency scale. The proposed model enables performance comparison of models based on ERB and Bark scales. The investigation reported in this dissertation has shown that ERB based masking models work at least as well as the Bark based models. They improve the relatively poor performance of current perceptual audio coding technology when applied to speech signals at very low bitrates. Superior performance of the new model with speech signals may also suggest that models based on the Bark scale may have introduced distortions at frequencies below 500 Hz, although this does not necessarily reduce speech intelligibility. Additionally it is shown how excess masking can be exploited to further improve coding efficiency in systems using perceptual models based on the ERB scale.

The second contribution of this dissertation is the study of the modified discrete cosine transform (MDCT) and its mismatch with the discrete Fourier transform (DFT) based psychoacoustic model. Presentation of a signal in the MDCT domain has emerged as the dominant tool in high quality audio coding because of the special properties of MDCT. In addition to the energy compaction capability similar to the discrete cosine transform (DCT), the MDCT simultaneously provides critical sampling, reduction of block effects and flexible window switching. However, perceptual models of the auditory system often use a Fourier transform implemented by a DFT. Using a masking curve calculated with a DFT based psychoacoustic model to quantize MDCT coefficients could present problems in certain special cases. This dissertation provides a first step toward solving this mismatch and thereby simplifying the encoder structure. A comparative study of the energy compaction properties of some relevant transforms is presented. The integer-to-integer DCT can implement a lossless scheme preserving the spatial structure of quantization errors, thus preventing possible binaural unmasking effects. A novel method is presented to remove inter-channel redundancy in multichannel audio using the integer-to-integer DCT.

The third contribution of this dissertation focuses on compressed domain audio processing for the purpose of error concealment and improvement of existing audio coding technologies such as MPEG-1 Layer 3 (MP3) and MPEG-2/4 advanced audio coding (AAC). A novel compressed-domain beat-pattern based error concealment algorithm is proposed to tackle packet loss in streaming music over error prone channels such as Mobile Internet. Finally, schemes to recompress MP3 audio bitstreams are studied for applications such as messaging, browsing and storage applications in mobile terminals.

Preface

The work contained in this dissertation was carried out during the years 1998 - 2001 at the Speech and Audio Systems Laboratory, Nokia Research Center (NRC), Finland and during a Foreign Research Fellowship in the spring of 2001 at the Department of Experimental Psychology, University of Cambridge, UK.

I would like to express my gratitude to my dissertation supervisor, vice president of Nokia Mobile Phones (former head of the Speech and Audio Systems Laboratory), Prof. Petri Haavisto for his encouragement, which enabled me to cope with the tempo of work at Nokia and to finish my Ph.D. program, finally, as a byproduct. Prof. Brian C. J. Moore is thanked for arranging my research visit to the University of Cambridge where I have enjoyed the beautiful Cambridge campus and learnt a lot from many inspiring discussions on sound perceptions, etc. I would like to thank my pre-examiners, Prof. Anibal Ferreira (University of Porto) and Dr. Bernd Edler (University of Hannover) for their critical and constructive comments to this dissertation.

My special thanks go to my mentor and collaborator Prof. Leonid Yarovslavsky (Tel Aviv University) with whom it has been an honor and a pleasure to work and to learn from. I hope that our fruitful collaboration and discussions will continue in the future. I am grateful to my friend and mentor Dr. Jilei Tian (Nokia Research Center) for the thought-provoking discussions and many valuable suggestions. Jilei's wisdom and smile have always been an inspiration, especially during the long, dark and cold Finnish winter.

I am indebted to all my colleagues at Nokia Research Center, Speech and Audio Systems Laboratory for providing a pleasant research environment. In particular, I would like to thank my superior, Mr. Mauri Väänänen for supporting my study and for many valuable discussions and suggestions. My collaboration with Mr. Miikka Vilermo has always been a pleasure, which proves at least experimentally that culture differences do not have to be an obstacle for excellent cooperation. Mr. Juha Ojanperä's programming skill has helped me to accelerate my research. Mr. David Isherwood's participation in arranging subjective listening tests and his proofreading of this dissertation have been very helpful. Dr. Leo Kärkkäinen's insightful comments on some of the attached publications have been of great value. I would like to acknowledge my other colleagues, Mr. Markus Vaalgamaa, Dr. Jyri Huopaniemi, Dr. Nick Courtis, Mr. Jarno Seppänen, Mr. Kalervo Kontola, Mr. Bogdan Moldoveanu, Mr. Arto Lehtiniemi, Mr. Nick Zacharov and Mr. John Cozens for assistance in various aspects.

I have been fortunate to study with many experts at the Department of Information Technology, Tampere University of Technology (TUT). I wish to thank Prof. Tapio Saramäki who supervised my Licentiate thesis, Prof. Karen Egiazarian, Prof. Jaako Astola, Prof. Moncef Gabbouj for giving me useful advice and practical help. I am grateful to Mr. Anssi Klapuri for many stimulating discussions.

The members of the Psychoacoustics Group at the Department of Experimental Psychology, Cambridge University, especially Dr. Michael Stone, Dr. Brian Glasberg, Dr. Tom Baer, Dr. José I. Alcántara, Dr. Chin-Tuan Tan, Mr. Thomas Stainsby, Mr. Geoffrey Moore and Ms. Sheila Flanagan are thanked for all their hospitality and for a stimulating research environment.

I would like to mention three professional organizations (Audio Engineering Society and its technical committee of audio coding, IEEE signal processing and communication societies, and SIGMM of Association for Computing Machinery) from which I have benefited a lot. Some of my research works originated from the enlightening discussions with some of the best experts in the field. In particular, I

would like to thank Mr. James Johnston (AT&T Labs), whose genius and insights into audio coding have always been an inspiration. I also would like to thank Prof. Karlheinz Brandenburg (FhG), Dr. Thomas Sporer (FhG), Dr. Jurgen Herre (FhG), Dr. Schuyler Quackenbush (AT&T Labs), Dr. Chin-Hui Lee (Bell Labs), Dr. Fred Juang (Bell Labs), Prof. K.J. Ray Liu (University of Maryland), Prof. Tsuhan Chen (Carnegie Mellon University), Prof. San Yuan Kung (Princeton University), Prof. Ralf Steinmetz (Darmstadt University of Technology), Prof. Matti Karjalainen (Helsinki University of Technology), Dr. Simon Dixon (Austrian Research Institute for Artificial Intelligence) and Dr. Masataka Goto (Japanese National Institute of Advanced Industrial Science and Technology) for their advice, suggestions and encouragement. It has been indeed a privilege to meet the brightest minds.

The journey to this dissertation has been long (a cross-millennium project) but rewarding. The list of people who deserve my thanks is long as well. In particular, I would like to thank my M.Sc. thesis and former Ph.D. dissertation supervisor, Prof. Ulrich Reimers (Braunschweig University of Technology) for helping me to make an important and difficult decision – leaving Germany for Finland and starting a new career in what at the time was a less celebrated Nokia in digital audio technology – totally uncharted territory for me in 1993. After several years of struggling in the cold water, I have finally managed to swim nicely without being swallowed by the huge waves of daunting challenges. Dr. Aki Mäkivirta (currently with Genelec OY) is thanked for introducing me to the interesting and challenging digital audio world. His friendship and assistance helped me a great deal to overcome the initial culture shock when I started my job at Nokia Research Center in the spring of 1994. The encouragement from Dr. Harri Raittinen (TUT) has helped me to continue research in audio coding in spite of some frustrations.

I would like to thank all my friends worldwide for their friendships, help and support. In particular, I would like to mention some of my Finnish friends, Kaarina Melkas, Matti Hämäläinen, Kari Laurila, Tuomo & Katja Hammer, and Jari Puranen. Like most Finns, they don't have many glossy words with accompanying duplicity, but they gave me their kindly hands whenever it was necessary. This had the effect of keeping me in Finland until this memorable moment for nearly eight years, which is still a record for a foreign researcher working at NRC Tampere. During this period, I have not only come to understand Finnish culture (e.g. *sisu*) better but also found an answer to an often-asked question; how a tiny country like Finland, with a population of 5 million, can create hi-tech giants such as Nokia.

A special word of thanks goes to my wife Dr. Ning Xiang (Optoelectronics Research Center, TUT) for her support, encouragement and patience in the course of my research work, especially in the past two years. It was quite a challenge for her to find an optimal balance between a good housewife, a caring mother and a full-time profession as a senior researcher. It is fair to say that this dissertation would have not been possible without her full support at home, although we are working with competing technologies at work (optical fiber versus wireless).

Last but not least, I would like to thank my mother Meiying Wu and my father Pucai Wang for their never ending love, support and in taking care of Tina Wang, our lovely daughter.

The Academy of Finland and Nokia Foundation are gratefully acknowledged for providing me with scholarships, which enabled me to conduct an important part of my research at Cambridge University, UK.

Tampere, August, 2001

Ye Wang

Contents

List of Publications

This dissertation includes the following eight publications.

[P1] Wang, Y., Vilermo, M. “Exploiting Excess Masking for Audio Compression”, AES 17th International Conference on High Quality Audio Coding, September 2 – 5, 1999, Florence, Italy, pp. 216-219

[P2] Wang, Y., Vilermo, M. “An Excitation Level Based Psychoacoustic Model for Audio Compression,” The 7th ACM International Multimedia Conference, October 30 to November 4, 1999 Orlando, Florida, USA, pp. 401-404

[P3] Wang, Y., Vilermo, M., Yaroslavsky, L. “Energy Compaction Property of the MDCT in Comparison with other Transforms”, AES109th International Convention, September 22-25, 2000, Los Angeles, California, USA, preprint 5178

[P4] Wang, Y., Vilermo, M., Isherwood, D. “The Impact of the Relationship Between MDCT and DFT on Audio Compression: A Step Towards Solving the Mismatch”, The First IEEE Pacific-Rim Conference on Multimedia (IEEE-PCM2000), December 13-15, 2000, Sydney, Australia, pp. 130-138

[P5] Wang, Y., Vilermo, M., Väänänen, M., Yaroslavsky, L. “A Multichannel Audio Coding Algorithm for Inter-Channel Redundancy Removal”, AES110th International Convention, May 12-15, 2001, Amsterdam, The Netherlands, preprint 5295

[P6] Wang, Y. “A Beat-Pattern based Error Concealment Scheme for Music Delivery with Burst Packet Loss”, IEEE International Conference on Multimedia and Expo (ICME2001), August 22-25, 2001, Tokyo, Japan, pp. 73-76

[P7] Wang, Y., Vilermo, M. “A Compressed Domain Beat Detector using MP3 Audio Bitstream”, The 9th ACM International Multimedia Conference (ACM Multimedia 2001), September 30 – October 5, 2001, Ottawa, Ontario, Canada, pp. 194-202

[P8] Wang, Y., Ojanperä, J., Vilermo, M., Väänänen, M. “Schemes for Re-Compressing MP3 Audio Bitstreams”, accepted by the AES111th International Convention, November 30 - December 3, 2001, New York, USA

List of Supplementary Publications

These publications are not included as part of the dissertation. However, they describe the research work conducted by the author in the field of error resilient audio coding with applications in error-prone channels such as wireless Internet.

[S1] Mäkivirta, A., Väänänen, M., Sydänmaa, M., Wang, Y., "Error Performance and Error Concealment Strategies for MPEG Audio Coding", 1994 Australian Telecommunication Network & Applications Conference, December, 1994, Melbourne, Australia, pp. 505-510

[S2] Wang, Y. "Assessment System of Psychoacoustic Models", NATO ASI workshop on Computational Hearing, July 1 - 12, 1998, II Ciocco, Italy, pp. 195-197

[S3] Wang, Y. "A New Watermarking Method of Digital Audio Content for Copyright Protection", 4th IEEE International Conference on Signal Processing, October 12-16, 1998, Beijing, China, pp. 1420-1423

[S4] Wang, Y., Vilermo, M. "Audio Signal Representation and Processing in Time-Frequency Domain", International Computer Music Conference 1999, October 22-27, 1999, Beijing, China, pp. 264-267

[S5] Wang, Y., Yaroslavsky, L., Vilermo, M., Väänänen, M. "Restructured Audio Encoder for Improved Computational Efficiency", AES 108th International Convention, February 19-22, 2000, Paris, France, preprint 5103

[S6] Wang, Y., Yaroslavsky, L., Vilermo, M. "On the Relationship between MDCT, SDFT and DFT", WCC2000 – 16th IFIP World Computer Congress/ICSP 2000 – 5th IEEE International Conference on Signal Processing, August 21 – 25, 2000, Beijing, China, pp. 44-47

[S7] Wang, Y., Yaroslavsky, L., Vilermo, M., Väänänen, M. "Some Peculiar Properties of the MDCT", WCC2000 – 16th IFIP World Computer Congress/ICSP 2000 – 5th IEEE International Conference on Signal Processing, August 21 – 25, 2000, Beijing, China, pp. 61-64

[S8] Yaroslavsky, L., Wang, Y., "DFT, DCT, MDCT, DST and signal Fourier spectrum analysis", EUSIPCO 2000 - 10th European Signal Processing Conference, September 5-8, 2000, Tampere, Finland, pp. 1065-1068

[S9] Wang, Y., Streich, S., "A Drumbeat-Pattern based Error Concealment Method for Music Streaming Applications", submitted to 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2002), May 13-17, 2002, Orlando, Florida, USA.

List of Abbreviations

3G	3rd Generation Mobile Communications
AAC	MPEG Advanced Audio Coding
AC-3	Audio Coding Technique from Dolby Laboratories Inc.
ARQ	Automatic Repeat Request
ASPEC	Adaptive SPectral Entropy Coding
BER	Bit Error Rate
BMLD	Binaural Masking Level Difference
BSAC	MPEG-4 Bit Sliced Arithmetic Coding
CB	Critical Band
CD	Compact Disk
C/I	Carrier-to-Interference
CPEs	Channel-Pair-Elements
DAB	Digital Audio Broadcasting
DFT	Discrete Fourier Transform
DiffServ	Differential Services
DVD	Digital Video Disc
DWPT	Discrete Wavelet Packet Transform
EPAC	Enhanced Perceptual Audio Coding from Lucent Technologies
ERB	Equivalent Rectangular Band
FEC	Forward Error Correction Coding
FIR	Finite Impulse Response
FFT	Fast Fourier Transform
FV	Feature Vector
GA	General Audio in MPEG-4
GSM	Global System for Mobile Communications
HDTV	High Definition Television
Hi-Fi	High-Fidelity
IBI	Inter-Beat Interval
INT-DCT	Integer-to-Integer DCT
IP	Internet Protocol
ISP	Internet Service Provider
KLT	Karhunen-Loeve Transform
LAN	Local Area Network
LC	(AAC) Low-Complexity Profile
LFE	Low Frequency Enhancement
LSB	Least Significant Bit

LSF	Low Sampling Frequency
LTP	Long Term Prediction
Mbone	Internet Multicast Backbone
M-commerce	Mobile Commerce
MDCT	Modified Discrete Cosine Transform
MIDI	Musical Instrument Digital Interface
MLT	Modulated Lapped Transform
MPEG	(ISO/IEC) Moving Pictures Expert Group
MP3	MPEG-1 Layer 3
MUSICAM	Masking pattern adapted Universal Subband Integrated Coding And Multiplexing
NMT	Noise Masking Tone
OA	Overlap-Add
PAC	Perceptual Audio Coding from Lucent Technologies
PCM	Pulse Code Modulation
PNS	Perceptual Noise Substitution
PQMF	Pseudo Quadrature Mirror Filterbank
PQF	Polyphase Quadrature Filterbank
PR	Perfect Reconstruction
QoS	Quality of Service
RSVP	Resource reServation Protocol
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
SDFT	Shifted Discrete Fourier Transforms
SFB	Scale-factor Band
SMR	Signal-to-Masking Ratio
SNR	Signal-to-Noise Ratio
SPL	Sound Pressure Level
SSR	(AAC) Scalable Sampling Rate profile
TCP	Transport Control Protocol
TDAC	Time Domain Alias Cancellation
TMN	Tone Masking Noise
TNS	Temporal Noise Shaping
UEP	Unequal Error Protection
UDP	User Datagram Protocol
VoIP	Voice Over IP
WMA	Windows Media Audio

Chapter 1 Introduction

In the past couple of years, an explosive growth in the use of the Internet and mobile telephones has been experienced. The convergence of these two technologies will open a wide range of new opportunities for the already flourishing multimedia market [1].

Wideband audio is an important element of multimedia. The Internet transmission of compressed digital audio, such as MP3, has already shown a profound effect on the traditional process of music distribution.

With increasing channel capacity available in the new generation of mobile networks, it is logical to envision an interesting scenario that would bring music to a mobile terminal via the Internet. For example, music or radio programs can be ordered for immediate or later listening; web-based services can be accessed via the mobile network; music can be distributed from peer to peer; and interactive audio-related games can be played with friends. These applications can be implemented within the different technical requirements of the communication systems. Depending on the constraints on delay, three types of communication modes can be employed. These are non-real-time messaging, near real-time browsing, and two-way real-time rich call. Messaging does not have any constraint on delay. Browsing has some constraint on delay to the degree that is not very annoying to the customers. Two-way real-time rich call has the strictest constraint on delay, which should not exceed 250 ms [61].

These scenarios could provide added value to consumers and become an important form of mobile-commerce (m-commerce) in the near future. However, the characteristics of mobile networks pose special problems to making this vision a reality. This dissertation addresses some of the relevant technical aspects and reports some advances.

1.1 Context and Motivation

The digital coding of high fidelity (Hi-Fi) audio has been commercialized since the 1970s in the form of the compact disk (CD). However, the amount of data needed for a faithful digital representation of audio signals is enormous. For example, the net bitrate for pulse code modulation (PCM) recordings on CDs is 705.6 kbps (= 44.1 kHz · 16 bits) for each monophonic channel. This has become a tremendous obstacle for many applications, especially for band limited transmission systems. The necessity for digital audio coding/compression is obvious.

Most of the pilot research on Hi-Fi audio coding has been conducted during the development of digital audio broadcasting (DAB) system [2], where near CD quality digital audio should be sent via radio channels. This effort had an important impact on the first international standard (MPEG-1 audio) [3] for the digital compression of Hi-Fi audio.

Along with deployment of new generations of mobile communication networks, the transmission of wideband audio is becoming more feasible. However, the price of channel capacity in the wireless network is still significantly higher than its wired counterpart such as optical fibers, thus making wireless channels a very scarce commodity. In addition, a mobile terminal usually has quite limited computing and memory capacity. All these factors make it a challenging task to compress and to transmit wideband audio over wireless networks.

Speech and music are two important classes of audio signals. For wireless audio content delivery, the following features have to be considered carefully. 1) Coding efficiency is very important. Since lossless coding techniques alone are inadequate for this application, lossy coding techniques become a natural choice. 2) Computational complexity and memory consumption are critical for mobile devices. 3) Error resilience is crucial to cope with the adverse conditions of different wireless systems.

With current audio compression technologies the bitrate and its associated price per bit could still limit widespread acceptance of high quality music delivery in cellular networks, even in the 3rd generation mobile communications (3G) systems. Improvements in compression efficiency are thus desired to foster on-line music business in wireless networks. The second major problem in current audio coding technologies is the poor performance with speech signals at low bitrates. Improved speech quality at low bitrates would be a much-desired achievement in generic audio coding technology.

It becomes evident from this that current audio coding technologies do not provide all the answers to the daunting task of high quality audio delivery over wireless networks. It is thus desirable to have more efficient and error resilient coding algorithms optimized for mobile networks. This dissertation presents some improvements and solutions within this perspective. The emphasis here is on coding efficiency, reduction of computational complexity, memory consumption and error concealment.

1.2 Outline of the Dissertation

This dissertation consists of eight publications and an introductory review of the relevant audio coding techniques and the error resilient transmissions. The introductory part is organized into five chapters. In Chapter 2, the fundamentals of audio compression are reviewed. Chapter 3 focuses on compressed domain audio processing with applications to error resilient transmissions and to enhance coding efficiency of existing technologies. Chapter 4 summarizes the eight publications and presents the author's contribution to publications. Chapter 5 concludes the findings of the dissertation and outlines future research perspectives.

Chapter 2 Fundamentals of Audio Compression

2.1 Overview of Audio Coding Technologies

The problem of signal compression or source coding is to achieve a low bitrate in the digital representation of an input signal with minimum perceived loss of signal quality [6]. There are two fundamental ways to compress digital audio signals. The first method is to remove the *redundancy* that is not necessary for the reconstruction of the original signal. The operation of removing signal redundancy is traditionally called entropy coding or lossless coding. An encoder structure in Figures 1 and 3 without a psychoacoustic model represents a common configuration of such techniques. The configuration consists of a decorrelation module such as prediction or a transform, which serves to reduce the redundancy of the audio signal (due to the memory of the audio source [7]), as a primary stage, followed by a quantizer. The quantized data usually exhibit some residual redundancy (due to a non-uniform probability density function of the quantized symbols [7]), which can be further reduced by a Huffman or arithmetic coding as a secondary stage. These types of methods are capable of compressing audio signals by a factor of two or three. The merit is that they can reproduce the quantized signal exactly, not just approximately. It should be noted, however, that the decorrelation module as well as the quantizer has to be designed carefully to ensure lossless coding.

In order to achieve a more compact representation of digital audio, a second method is introduced to remove the *irrelevancy*. This class of methods is based on the incorporation of the limited time and spectral resolution capability of the human auditory system. The irrelevant parts of the signal inaudible to the human ear, need not be transmitted [5]. Applying knowledge of auditory perception leads to hearing-specific codecs that perform remarkably well. This second approach needs a psychoacoustic model, which is usually a simple estimation of the masking effect of the human auditory system. This type of method is capable of compressing audio signals by a factor of ten without perceptible loss of quality. Extensive reviews of perceptual coding can be found in [6][8].

2.1.1 Historical Development

Essentially, all state-of-the-art low-bitrate audio coding technologies are based on the combination of the two basic operations discussed above. The groundbreaking achievement in perceptual audio coding was marked by the first international standard – MPEG-1 audio-coding standard [3] in 1992. In spite of some proprietary technologies such as AC-3 from Dolby Laboratories, WMA from Microsoft, EPAC from Lucent Technologies and ATRAC-3 from Sony in recent years, MPEG audio standards seem to remain the mainstream of technologies. Since there is more information available regarding the MPEG Audio

international standard, only the development of MPEG's family of audio coders is briefly reviewed in this dissertation.

The MPEG-1 audio [3] was developed for one or two audio channels (mono, stereo, or dual channel), sampled at 32, 44.1 or 48 kHz. MPEG-1 Layer 3, commonly known as MP3, has become very popular in the Internet world. A good tutorial on MPEG-1 audio can be found in [4]. In 1994 MPEG-2 audio [9] was developed for low bitrate coding of multichannel audio, exemplified by the common 5.1 surround sound format (front left, front right, center, two rear channels, and an optional low frequency enhancement channel). Furthermore, MPEG-2 audio also provides the low sampling frequency (LSF) extension to MPEG-1 audio. Due to the backward compatibility to MPEG-1, the MPEG-2 audio coding algorithm is very similar to its predecessor. The backward compatibility, however, has limited the coding performance of MPEG-2.

In 1994 it became obvious that, by giving up the backward compatibility and introducing new technologies, much better quality at lower bitrates could be achieved [11]. As a result of the new efforts, the MPEG-2 advanced audio coding (AAC) algorithm [12] was finalized in 1997 as the third generation MPEG audio coder, which is formally an extension to the MPEG-2 standard. MPEG-2 AAC provides monophonic, 2-channel and multichannel coding capabilities.

In parallel, the MPEG-4 audio standard [14] started its development in 1994/1995 and was finalized in 2000. An important aspect of the overall MPEG-4 audio functionality is covered by the so-called "General Audio" (GA) part, *i.e.* coding of arbitrary natural audio signals. MPEG-4 general audio coding is built around the coder kernel provided by MPEG-2 AAC, which is extended by additional coding tools and coder configurations [11]. AAC has a very flexible bitstream syntax that supports multiple audio channels, subwoofer channels, embedded data channels, and multiple programs consisting of multiple audio, subwoofer, and embedded data channels. AAC combines the coding efficiencies of a high-resolution filter bank, backward-adaptive prediction, joint channel coding, and Huffman coding with a flexible coding architecture to permit application-specific functionality while still delivering excellent signal compression [15].

2.1.2 Basic Audio Coding Tools

Modern perceptual audio encoders are conceptually similar in the sense that they consist of four basic building blocks (see Figure 1): a transform or filterbank, perceptual model, requantization and coding, and bitstream formatting.

The concept of perceptual audio coding (bitrate reduction) described from the viewpoint of quantization noise shaping therefore turns into the following: initially a PCM signal, such as music on a commercial CD, has the quantization noise uniformly distributed across the whole frequency band. A transform or filterbank creates a frequency domain representation of this signal. A perceptual model usually uses the original signal to estimate a time and frequency dependent masking threshold indicating the maximum quantization noise inaudible in the presence of this audio signal. By requantization, a quantizer then reduces the number of bits used to represent this signal resulting in an increase and shaping of quantization noise to the limit of the masking threshold. This explains the significance of masking in perceptual audio coding technologies.

The transform or filterbank and the perceptual model connect at the quantizer. For orthogonal transforms, the coding reconstruction error variance equals that introduced by a set of coefficient quantizers [7]. Therefore, a coarser quantization of spectral coefficients corresponds an increased reconstruction error, reducing the signal-to-noise ratio (SNR) of a reconstructed audio signal.

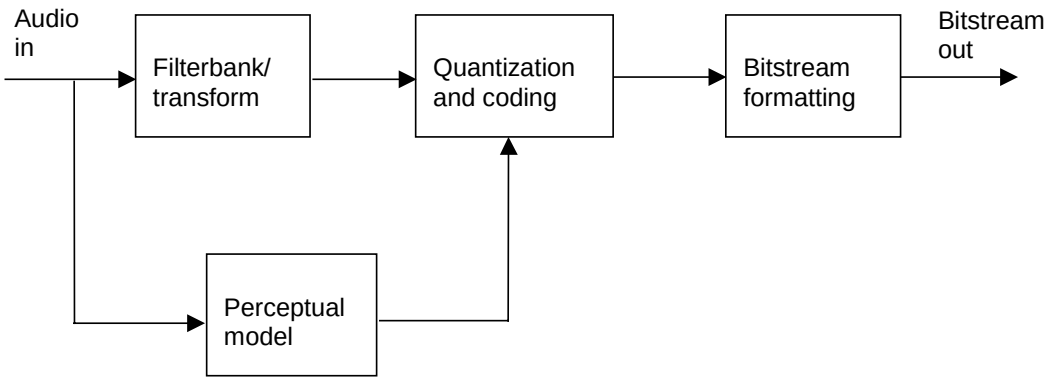

Figure 1. Sketch of the basic structure of a perceptual encoder.

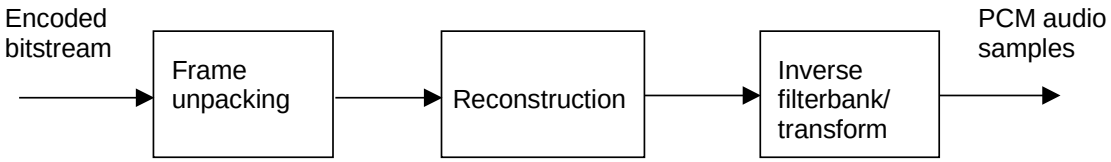

Figure 2. Sketch of the basic structure of a perceptual decoder.

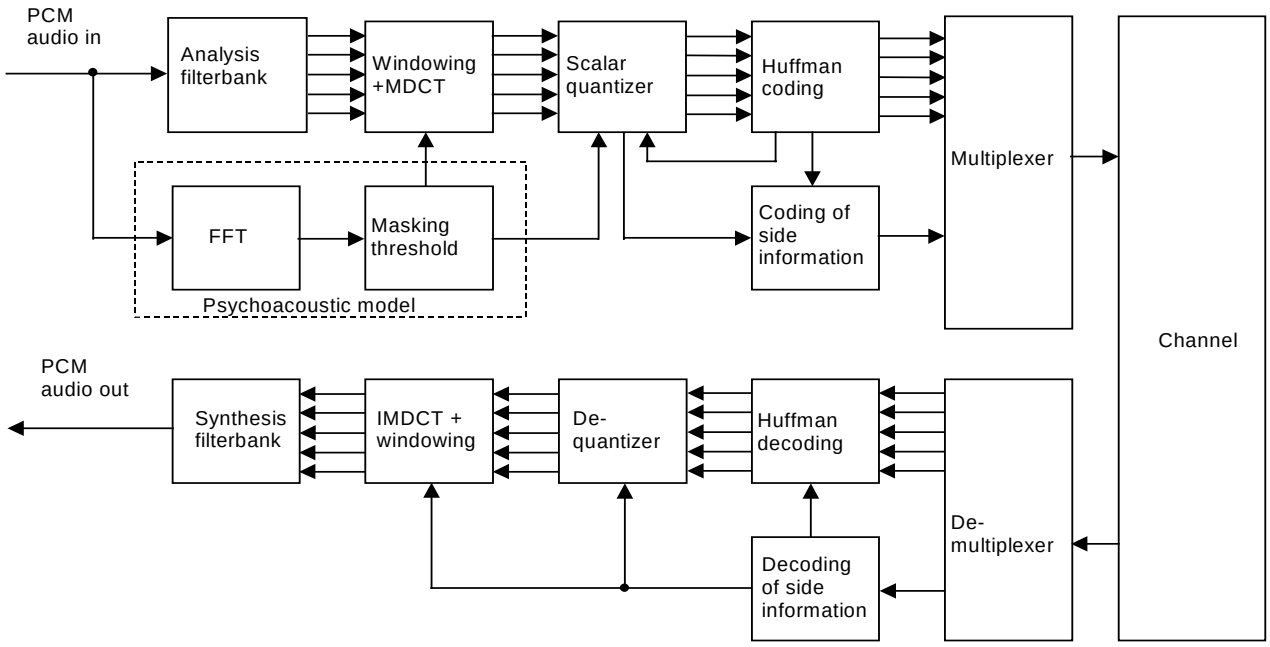

Figure 3. Block diagram of the MP3 codec.

In the case of non-orthogonal transforms, such as the modified discrete cosine transform (MDCT), the relationship between the spectral quantization noise and the time domain reconstruction error is not so straightforward. However, the same dependency exists, i.e., a coarser quantization of spectral coefficients, consuming less bits, leads to reduced SNR of a reconstructed audio signal.

The general design philosophy is that the decoder is significantly simpler than the encoder (see Figure 2). Without loss of generality, the focus of this dissertation is put on MPEG audio coding tools. In this chapter, a review of the four basic tools is first presented and is then followed by a presentation of some other complementary coding tools. In order to show how these coding tools are used in actual coding systems, the MP3 codec structure is illustrated in Figure 3.

2.2 Psychoacoustic Models

A perceptual model is the heart of all perceptual coders. The single most important feature of this is the exploitation of masking effects within the human auditory system such that the audibility of quantization noise can be reduced. The masking threshold is computed using rules known from psychoacoustics.

Two key concepts are that the ear uses frequency division as a first step in the hearing process and that the masking effect of stimuli presented to the ear can be understood from a relatively small number of experiments using simple stimuli. One basic approach used in the design of these coders follows this principle and uses the extension of the masking characteristics for simple signals to complex ones. This is a bold assumption because typical music signals are often very complex, with many tonal or noiselike components, and they are not at all like the simple stimuli used in basic psychoacoustic experiments [10].

It should also be noted that all psychoacoustical experiments are conducted with a group of people, who are supposed to be “normal listeners” with “normal ears”. However, everyone has different hearing characteristics. The results of psychoacoustical tests are usually presented as a statistical average. Therefore, in applying theory to practical coding applications, a careful trade-off has to be chosen between computational complexity and effectiveness.

2.2.1 The Concept of a Critical Band

When a sine tone excites the ear, a region of the basilar membrane oscillates around its equilibrium position. This region is fairly broad; however, there is a rather sharp point of maximum displacement. The distance of this maximum from the end of the basilar membrane is directly related to the frequency [24]. In other words, frequency is mapped onto a particular place along the membrane with a limited frequency resolution. This limit is closely related to an important characteristic of the perceptual mechanism known as the *critical band* [25][26]. The critical band was first discovered in masking experiments by Fletcher [27]. He measured the minimum level (“threshold”) at which a sinusoidal signal could be detected as a function of the bandwidth of a bandpass noise masker. The noise was always centered at the signal frequency, and the noise power density was held constant. Thus the total noise power increased as the bandwidth increased.

This experiment has been repeated several times since then. An example of such experiments from Moore *et al*. was presented in [28]. The threshold of the signal increases at first as the noise bandwidth increases, but than flattens off so that further increases in noise bandwidth do not change the threshold significantly [29].

To account for these results, Fletcher [27] suggested that the peripheral auditory system behaves as if it contained a set of bandpass filters with continuously overlapping passbands. These filters are now called

the auditory filters. When detecting a signal centered on a noise masker, the listener is assumed to focus his attention on the output of the auditory filter centered at the signal frequency. Increases in noise bandwidth result in more noise passing through the auditory filter provided the noise bandwidth is less than the filter bandwidth. However, once the noise bandwidth exceeds the filter bandwidth, further increases in noise bandwidth do not increase the noise passing through the filter. Fletcher called the bandwidth at which the signal threshold ceased to increase the critical bandwidth (CB) [29].

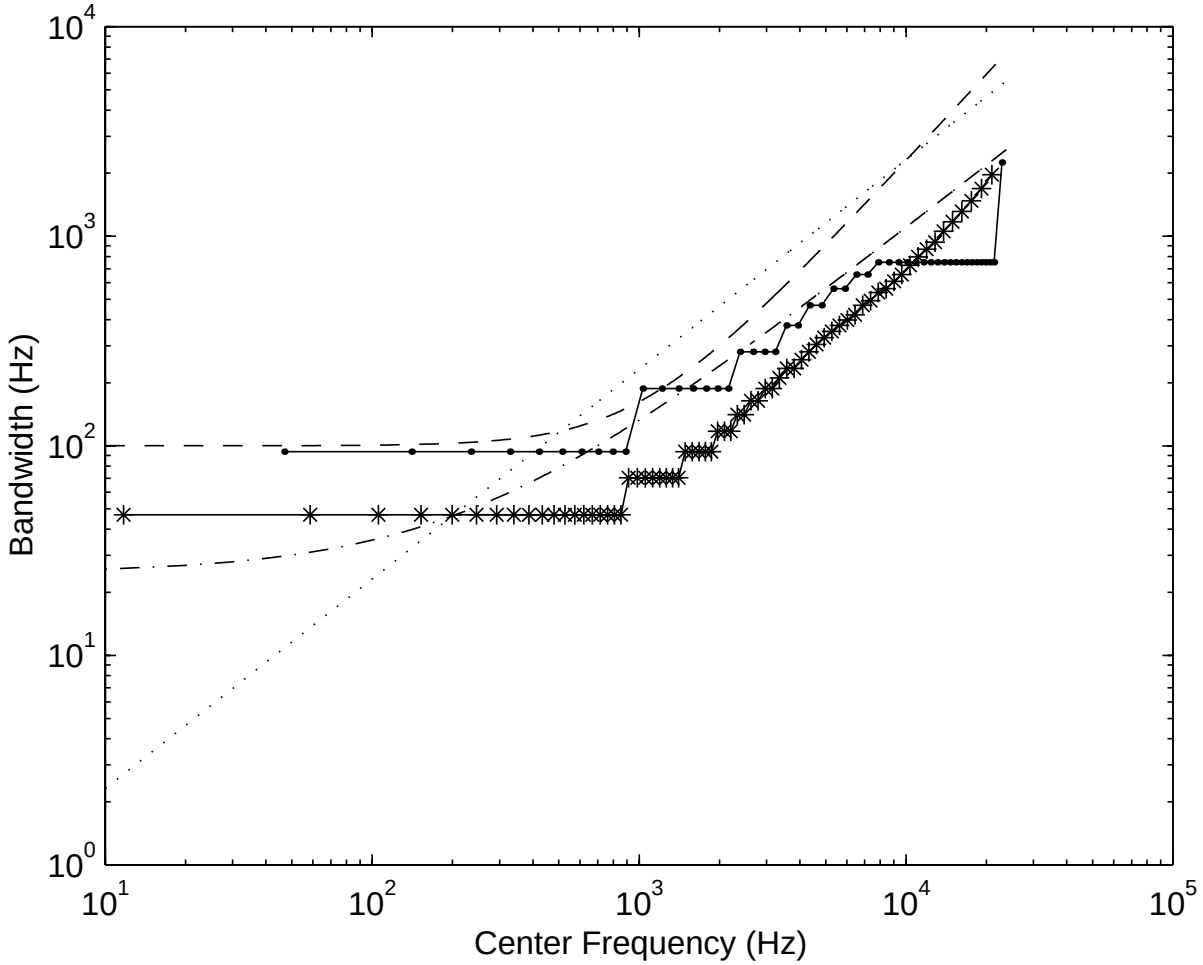

Figure 4. Estimated auditory filter bandwidths versus center frequencies. The one-third octave, the Bark and the ERB scale are shown as a dotted line, dashed line and dash-dotted line respectively. The scale factor bandwidth in AAC is shown as a solid line marked with dots. The one-third critical bandwidth in the psychoacoustic model of AAC is shown as a solid line marked with stars.

The critical-band concept is based on the well-proven assumption that our auditory system analyzes a broad spectrum in parts that correspond to critical bands [30]. Critical bands reflect the frequency resolving power of the ear as a function of the center frequency. The ear blurs the various signal components within a critical band. Empirical results show that our ears have a limited, frequency dependent acuity. This acuity (critical band) is approximately 100 Hz at the lowest audible frequencies and about 4 kHz at the highest. It is described by the following equation [31]:

$$CB = 25 + 75\left(1 + 1.4F^2\right)^{0.69} \quad (1)$$

where F is the frequency in kilohertz and CB is the critical bandwidth. The critical bandwidth increases monotonically with increasing frequency in a non-linear manner.

Adding one critical band to the next, so that the upper limit of the lower critical band corresponds to the lower limit of the next higher critical band, produces the scale of the *critical-band rate* [31].

Because the critical-band concept became a well-established theory applied in so many models, a unit for the critical-band rate was defined, which is one critical band wide. It is the *bark*, in memory of Barkhausen, a scientist from Dresden, Germany, who introduced the *phon*, a unit describing the loudness level for which the critical band plays an important role [30]. Zwicker *et al*. developed a Bark-scale loudness model [32][33][34]. Most of the perceptual models in perceptual audio coding and quality measurement systems are based on Zwicker's model and therefore the Bark scale.

Moore and Glasberg have presented a summary of experiments measuring auditory filter shapes using symmetric/asymmetric notched noise maskers [35]. The equivalent rectangular bandwidth (ERB) of the filters is defined by

$$ERB = 24.7(4.37F + 1) \tag{2}$$

where F is the frequency in kilohertz. The ERB and Bark scales are depicted with one-third octave bandwidth as a reference in Figure 4. It should be noted that the ERB function differs somewhat from the traditional critical band function, which flattens off below 500 Hz at a value of about 100 Hz.

For comparison, also scale-factor bandwidths and bands of one-third critical bandwidth used in AAC are shown in the same figure as a function of center frequency.

In AAC using one-third of a critical bandwidth is significant as it accounts for the fact that we use fixed bands. The human auditory system centers a critical band on the frequency of a masking component, extending the maximum masking effect to plus and minus one-half the critical bandwidth from the masking component frequency. Analysis methods employing fixed bands do not have this capability to center their frequency of operation. Using one-third of the critical bandwidth in the AAC psychoacoustic model is an engineering solution, which balances computational complexity with performance. For accurate quantization using the masking threshold, a higher resolution than the critical bandwidth is generally required in calculations. If fixed rather than adaptively setting bands are used, the characteristic bandwidth in a coder has to be reduced to improve modeling resolution. Values of typically one-half to one-third of a critical bandwidth are used. This helps to cope with the situation where a masking component's frequency lies at the high frequency boundary of a fixed band and the noise component to be masked is at its low frequency boundary [10].

On the other hand, AAC scale-factor bandwidth is used when quantizing the MDCT coefficients. The design is a good compromise between several constraints. Firstly, the bandwidth should reflect the critical bandwidth, so that the quantization noise can be set up to the masking threshold for an individual critical band. Secondly, a finer bandwidth than critical bandwidth will enable more accurate quantization noise tuning to the masking threshold, thus theoretically saving bits. However, this saving of bits in representing frequency components may be offset by the increase in associated side information, such as bit-allocation information. Finally, it is necessary to cope with other limiting factors such as computational complexity and the requirement of the Hufman coding in MP3 or AAC. As a result, the subband division presented in Figure 5 is codec dependent. For example, MPEG-1 Layer 1 and 2 employ a polyphase filterbank dividing audio signal into 32 equal-width subbands, while MP3 and AAC employ scale-factor bands that are more close to the critical bandwidth.

It has been suggested [36] that Bark-scale based perceptual models perform better than ERB based models. Our results [P2] cannot confirm these suggestions. Preliminary investigations performed in this dissertation suggest that an ERB based model can be applied in audio coding with good results, especially with speech signals.

2.2.2 Masking

Masking is an important phenomenon for perceptual coding. Masking is a complex result of the transducing and neural components of perception. It is highly adaptive and refers to the perceptibility of one signal in the presence of another in its time or frequency vicinity. In other words, when one sound makes another sound harder or impossible to hear, the former sound is masking the latter. Among the huge amount of literature on masking are two good introductory books [31][47].

The effect of masking is normally classified into *simultaneous* and *non-simultaneous masking* [31] that are respectively frequency and time domain phenomena. Simultaneous masking has been studied traditionally in the *critical-band-rate* scale. An example for the simultaneous condition would be the case where we have a conversation with a neighbor while a loud truck passes by. In this case, our conversation is severely disturbed. To continue our conversation successfully we have to raise our voices to produce more speech power and greater loudness. In music, similar effects take place. The different instruments can mask each other and softer instruments become audible only when the loud instrument ceases.

In the following sections, a more detailed description is given on issues related to the masking.

2.2.2.1 Masking in the Frequency Domain

Simultaneous masking is a frequency domain phenomenon where a low-level signal, *e.g.*, a pure tone (the maskee) can be made inaudible (masked) by a simultaneously occurring stronger signal (the masker), *e.g.*, narrowband noise, if the masker and maskee are close enough to each other in frequency. A masking threshold can be estimated below which any signal is likely to be inaudible. The masking threshold depends on the sound pressure level (SPL) and on the time-frequency characteristics of the masker and maskee.

Figure 5 shows the masking pattern of a pure tone at 1 kHz, where critical-band wide noise is masked by the tone. It can also be seen that loud low-frequency sounds mask weaker high-frequency sounds much more strongly than vice versa. The figure includes the absolute masking threshold (threshold in quiet) as a baseline and illustrates the associated signal-to-masking ratio (SMR). To a large extent, existing perceptual audio coders depend primarily on the masking of noiselike sounds (quantization noise) by tonelike sounds (speech or music) [29].

A sound signal below the masking threshold of the masking sound will not be perceived and is therefore irrelevant as far as the ear is concerned. This effect of “mutual masking” is particularly evident in broadband sound signals with well-defined formant structures. If mutual masking is taken into consideration for the coding of an audio signal, then it follows that the portion of the signal lying beneath the masking threshold does not need to be coded and therefore does not need to consume transmission capacity [5].

The concept of noise-masking tone, on the other hand, is a significant departure from models used earlier in perceptual coding, where it was usual to test the coder with a tone input and model the coding distortion as noise. In adaptive perceptual coding, we recognize that the signal in a critical band can be

noiselike, while a distortion component could be very localized and tonelike. Results for tone-masking noise tend to be more complicated [6].

The results from masking experiments are sometimes summarized by the following equations:

$$TMN = ET - 14.5 - B \ (dB) \quad (3)$$
$$NMT = EN - K \ (dB) \quad (4)$$

where *TMN* and *NMT* represent respectively tone-masking-noise and noise-masking-tone, which estimate the maximum energy of the masked signal in both cases. *ET* and *EN* are tone and noise energies, *B* is the critical band number which is given in the form of a table in MPEG standards. Various values in the range of 3 to 6 dB for the parameter *K* have been proposed [6].

Simultaneous masking is most widely exploited in audio coding schemes. A fairly common practice is to calculate the signal energy and masking level within each subband. Then the SMR is used to control the quantization of the transform coefficients. This concept is illustrated with a single sinusoid as the signal in Figure 5. The distance between the masker energy and the masking threshold is the SMR, which is used to control the quantizer. The data used to depict the masking threshold are from [31].

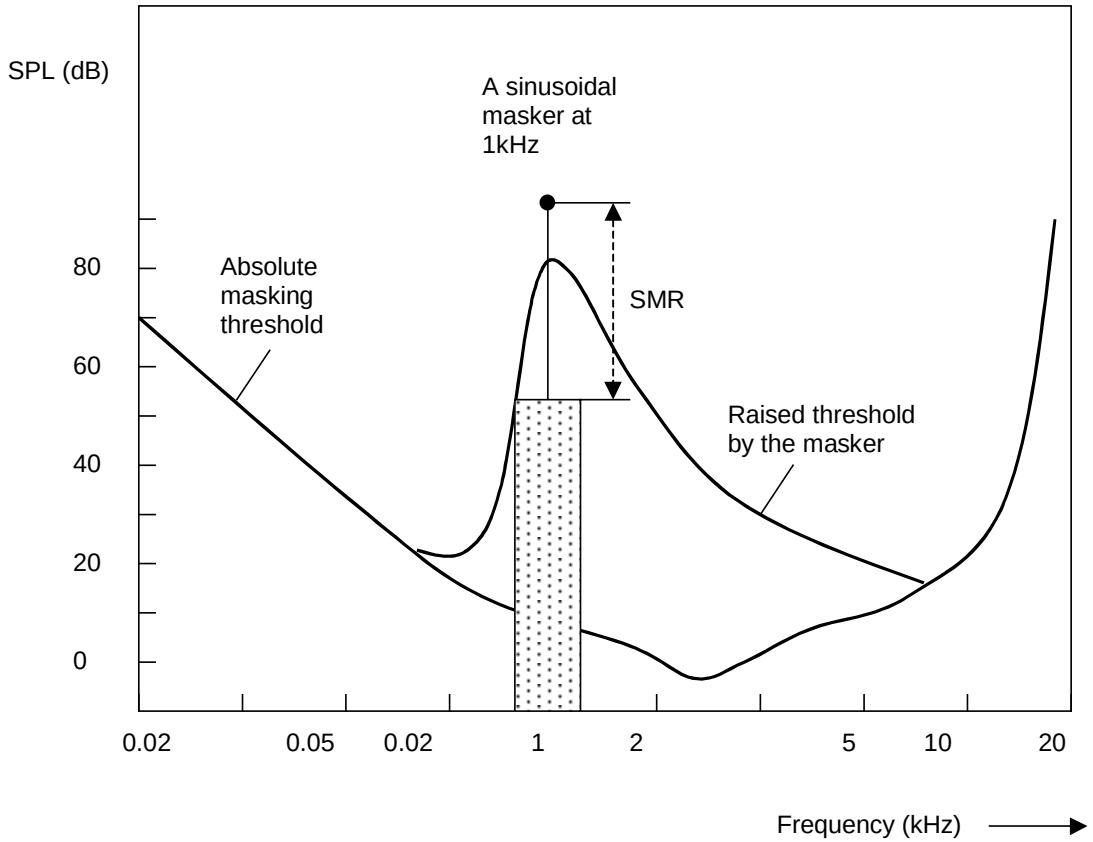

Figure 5. Simultaneous masking threshold and the signal-to-masking ratio (SMR). The rectangle filled with dots represents a frequency subband and the minimum masking threshold within the subband.

2.2.2.2 Masking in the Time Domain

Masking effects can also be measured when the masker and the test sound are not simultaneously present. When a short test signal is present before the masker stimulus is switched on, pre-masking/backward-masking can be measured. If the test sound is present after the masker is switched off, post-masking/forward-masking can be measured. In comparison with post-masking, pre-masking is weak and short of less than 20 ms in duration. By contrast, post-masking is much more obvious and long. Its duration can be as long as 200 ms. Figure 6 qualitatively describes pre- and post-masking effects. Both pre- and post-masking are frequency-dependent [31].

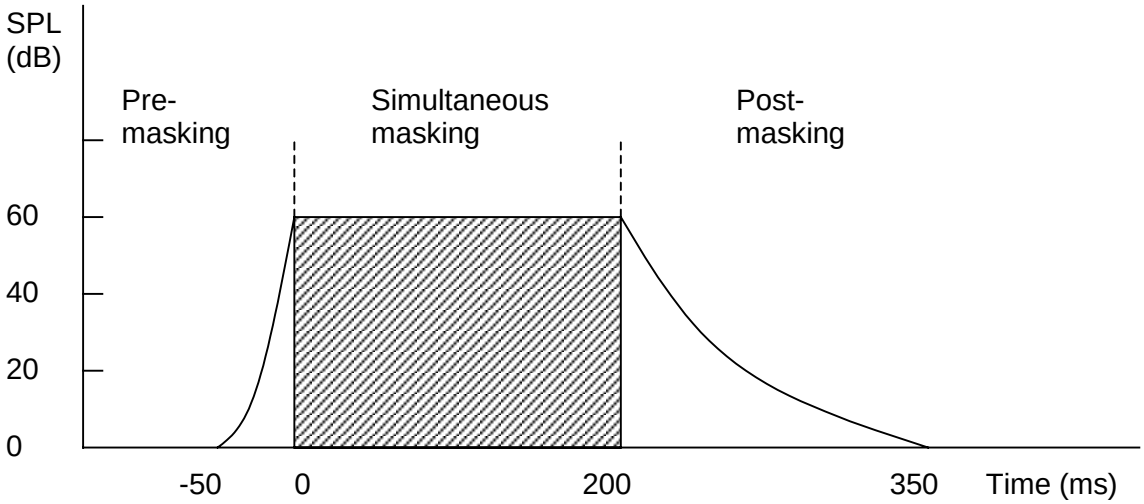

Figure 6. Illustration of temporal masking (pre- and post-masking). The rectangle filled with upward diagonals represents the duration of the masker [31].

2.2.2.3 Excess Masking

In psychoacoustical studies, normally only simple maskers have been used to determine the masking threshold. What about the more realistic situation with complex maskers? Can the masking threshold produced by combining simple maskers (sinusoids or band-limited noise) be predicted from their individual masking thresholds? These questions have not so far been answered clearly in published literature. As a simple example of implementation, the two psychoacoustic models presented in the informative part of the MPEG-1 standard take the simple sum of the individual masking thresholds.

However, several studies [43][44][45] have shown that the combined masking effect of two equally effective simultaneous maskers is 3 to 15 dB greater than the masking predicted by the linear addition of masker energies. This “additional” amount of masking is defined as excess masking. In order to take advantage of these characteristics, some investigations were performed and are reported in this dissertation. Excess masking exists not only in the frequency domain but also in the time domain [47]. However, time domain excess masking has not been studied in this dissertation.

2.2.3 MPEG Psychoacoustic Models

As a starting point, the two psychoacoustic models presented in MPEG-1 audio are briefly reviewed as references, since there is very little publically available information about the state-of-the-art in this aspect.

2.2.3.1 MPEG-1 Psychoacoustic Model 1

A high frequency resolution in the lower frequency region and a lower resolution in the higher frequency region should be the basis for an adequate calculation of the masking thresholds in the frequency domain. This would lead to a tree structure of the filterbank. However, the uniform 32-band polyphase filterbank of the MPEG-1 Audio standard, which is used for the subband filtering, has a parallel structure that does not provide subbands of different widths. Nevertheless, one major advantage of subband filtering is given by its high temporal resolution so that the quantization noise can be controlled with sufficient temporal resolution. This helps to prevent audible pre-echo. The small delay and complexity give the second major advantage.

To compensate for the lack of accuracy of the spectrum analysis of the filterbank, a 512-point fast Fourier transform (FFT) for Layer 1, and a 1024-point FFT for Layer 2 are used in parallel to filtering the audio signal into 32 subbands [37]. The output of the FFT is used to determine the relevant tonal, i.e. sinusoidal, and nontonal, i.e. noise maskers, of the actual audio signal. It is well known from psychoacoustic research that the tonality of both masker and maskee has an influence on the masking threshold. For this reason it is worthwhile to discriminate between tonal and nontonal components [38].

The basic idea of the psychoacoustic model 1 is to divide the auditory spectrum into tonal and non-tonal components. The total masking function is calculated by summing up the masking functions of these components and the absolute hearing threshold in power domain [3]. The output of the psychoacoustic model is the signal-to-masking ratio for each subband. The following sections explain how the psychoacoustic model is implemented.

The calculation of the model is performed in every data block, which are 384 input PCM samples for Layer 1 and 1152 input PCM samples for Layer 2 and 3. The input data is first windowed with a Hanning window, followed by a FFT routine to calculate the signal spectrum.

For simplicity, psychoacoustic model 1 is designed to distinguish tonal and noise components in the frequency domain in a rather simple way. In order to identify the tonal components, a list of all the local maxima in the spectrum is compiled and then pruned by applying a set of searching rules. All the remaining spectral lines are used for calculating the non-tonal components (*i.e.,* noise maskers). They are grouped into critical bands and within each critical band, a single non-tonal component, representing the effect of these lines is computed.

The next step is the so-called decimation: The tonal and noise components which are below the absolute hearing threshold or are less than one half of a critical bandwidth from a stronger neighboring component are removed. In this operation a 0.5 barks sliding window is used and only the component with the highest power is retained within the window.

Now the masking threshold of a tonal or a non-tonal component are calculated according to the following formulas respectively:

$$LT_{tm}(j,i) = X_{tm}(j) + av_{tm}(z(j)) + vf[z(i) - z(j), X_{tm}(j)] \quad (5)$$

$$LT_{nm}(j,i) = X_{nm}(j) + av_{nm}(z(j)) + vf[z(i) - z(j), X_{nm}(j)] \quad (6)$$

where *LT(j,i)* is the masking threshold at frequency index *i*, which is caused by a component at index *j* with strength *X*, and *z* is a function for mapping frequency indices to the Bark scale. It is the sum of three terms: the strength of the component X (on a linear scale), the masking index (av) and the masking function (vf). The masking index is an attenuation term, which depends on the critical band rate of the component, and whether it is tonal or non-tonal. The masking function is another attenuation factor, which depends on both the displacement of the component from the neighboring frequency and the component's signal strength. Since the masking function has infinite attenuation beyond -3 barks and +8 barks, the component has no masking effect on frequencies beyond those ranges.

The global masking threshold is computed for all spectral frequencies by adding the masking thresholds computed above for all the neighboring tonal and non-tonal components with the absolute hearing threshold in the power spectral domain.

In the next step, the minimum masking threshold is determined for each subband from the global masking threshold. Then the SMR is calculated for the bit allocation.

2.2.3.2 MPEG-1 Psychoacoustic Model 2

The frequency domain representation of the data is calculated via a FFT with a window length of 1024 samples. The calculation is done for every 576 samples in parallel to the hybrid filterbank of Layer 3, which is explained later in this chapter. The separate calculation of the frequency domain representation is necessary because the hybrid filterbank values cannot easily be used to get a magnitude-phase representation of the input sequence. The magnitude-phase representation is necessary to calculate the tonality of the maskers of the current input block.

The tonality estimation works using a simple polynomial predictor, as described in [39]. The basic idea is to use the predictability of the signal as an indicator for tonality. The prediction is done in the magnitude-phase domain. The values from the last two blocks are used to predict the magnitude and phase of each frequency line for the current block. The Euclidean distance between estimated and actual values in the complex FFT domain (*i.e.* real and imaginary part) is normalized to maximum possible distance. The normalized value is called the "chaos measure" and can assume values between 0 and 1. A logarithmic mapping is used to map the chaos measure range between 0.5 and 0.05 to tonality values between 0 and 1.

The magnitude values of the frequency domain representation are converted to a one-third critical band energy representation. A convolution of these values with the cochlea-spreading function follows. The next step in the threshold estimation is the calculation of the just-masked noise level in the cochlea domain using the tonality index and the convolved spectrum. A correction for the DC gain of the convolution has to be applied. The last step to get the preliminary estimated threshold is the adjustment for the absolute threshold. As the sound pressure level of the final audio output is not known in advance, the absolute threshold is assumed to be somewhat below the least significant bit (LSB) for the frequencies around 4 kHz. A more detailed description of the estimation of the masking threshold using spreading convolution can be found in [40].

The final step in the calculation of the threshold is pre-echo control. Pre-echo is audible if the backward masking of the signal is not sufficient to mask the error signal, which was spread in time due to the limited time resolution of the synthesis filterbank. This is only possible if there is a sudden increase in signal energy, at least for part of the signal bandwidth. From this a necessary (but not sufficient) condition

for the absence of audible pre-echo can be derived. The estimated masking threshold is restricted not to exceed the estimated threshold of the previous block. This condition on the final estimated threshold may reduce the estimated threshold by a large amount. To keep the actual quantization noise below this modified threshold, additional bits need to be available to the quantization and coding loop.

The masking threshold function has to be transformed back to the linear frequency scale. This is done by spreading it evenly over all the spectral lines corresponding to the partition domain defined in the MPEG-1 standard. The partitions are approximately one-third of a critical band. Finally, the SMR is computed for the subbands (in Layer 1 or 2) or the scale-factor bands (SFBs) (in Layer 3) to control the quantization.

2.2.4 The New Perceptual Models

One of the main contributions in this dissertation is the development of a new perceptual model based on the ERB concept. In addition, the excess masking effect is also incorporated into the proposed model. This combination seems to give quite satisfactory results in audio coding.

2.2.4.1 An ERB-based Model

The incorporation of a psychoacoustic model into audio coding has significantly improved coding efficiency. However, the psychoacoustic models used in established perceptual coders such as MPEG audio are based on greatly simplified assumptions, which may compromise the accuracy of the approximated masking thresholds. The MPEG audio standards give some examples in the informative part showing how a psychoacoustic model can be implemented. They use a DFT of successive blocks of the audio signal, which gives the associated spectral components of the blocks. For each spectral component an individual masking threshold is generated. The overall masking threshold follows from superposition of the individual thresholds, which is carried out by simply adding up the threshold at the corresponding frequencies [3][9][12]. This masking threshold determines the maximum quantization noise energy that can be added to the original signal so as to keep the noise inaudible. These models give a rather rough approximation, when a complex target (quantization noise) has to be masked by a complex masker comprising multiple spectral components (either speech or musical sounds) [48]. Further bit rate reduction heavily depends on the accurate estimation of the masking threshold both in the time and frequency domains.

For a better estimation of the masking threshold, some ear models have been developed [35][47][49][50]. The new model presented in this dissertation is based on Moore and Glasberg's excitation level calculation [35]. This is somewhat different from published psychoacoustic models in audio coding, and it leads to some advantages in masking threshold estimation. The proposed psychoacoustic model (see Figure 7) has been integrated into an audio coder similar to MPEG-2 AAC, which contains only the basic coding tools. The model performs better than or as well as the psychoacoustic model described in the MPEG-2 AAC audio coding standard for all the test signals. Almost transparent quality was achieved with bitrate below 64 kbps for most of the monophonic critical test signals. Significant improvements have been achieved with speech signals, which are always difficult for transform audio coders.

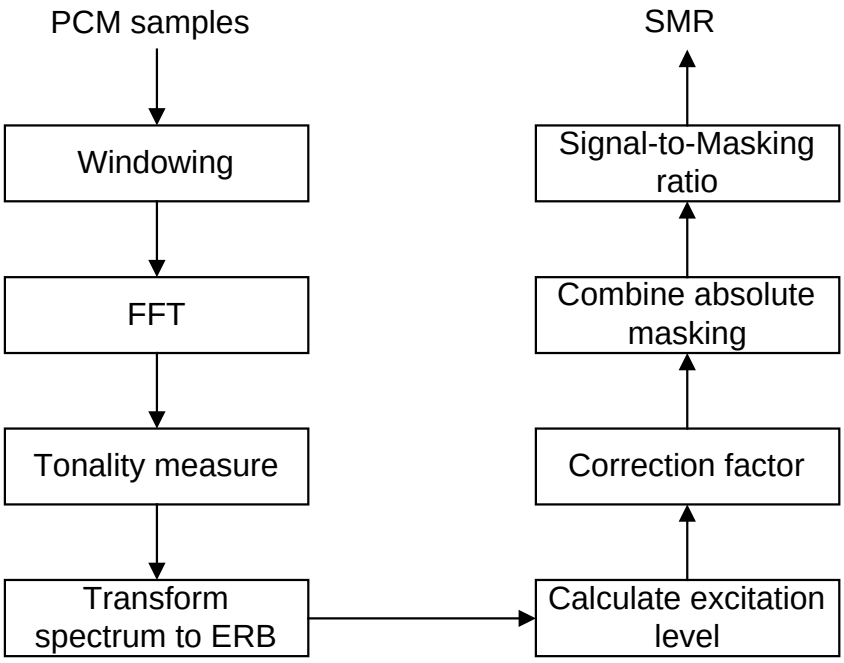

Figure 7. Block diagram of the proposed ERB-based perceptual model

A possible explanation for the good performance of the new model with speech signal is that the human ear has much better resolution in the low frequency range and the ERB approximates the ear better in lower frequency bands than the traditional Bark [47]. Speech quality with a Bark based model is affected by distortion at frequencies below 500 Hz, while speech intelligibility is not. This better match between the ERB and the ear might explain the improved subjective quality of coded speech signal using ERB-based model developed by Moore *et al* [35].

2.2.4.2 An Excess Masking Model

A psychoacoustic model in a coder calculates the masking threshold to determine the maximum allowable noise injection level without audible distortion. Such models simulate masking effects from psychoacoustic studies. There is a major challenge however: Only simple stimuli such as sinusoids and bands of noise have been used in most psychoacoustical studies. In audio coding we are dealing with real life audio signals. That is, a multi-component complex masker (coded audio signal) must mask the spectrally complex target (quantization noise).

The excitation-pattern model seems to underestimate the combined masking effects of multiple-component maskers [41][42]. More specifically, it underestimates the combined effects of two maskers both when the masker frequency components fall within the maskee auditory-filter bandwidth, and when they fall outside this bandwidth [42]. Therefore, some initial work was conducted to exploit the excess masking of two-tone maskers within the equivalent rectangular bandwidths (ERBs) [35] for audio compression. The stepwise masking thresholds (estimated within each AAC scale-factor band) with and without excess masking are shown together with the power spectrum of the signal in Figure 8. The AAC scale-factor bandwidths are indicated with the stepwise masking thresholds.

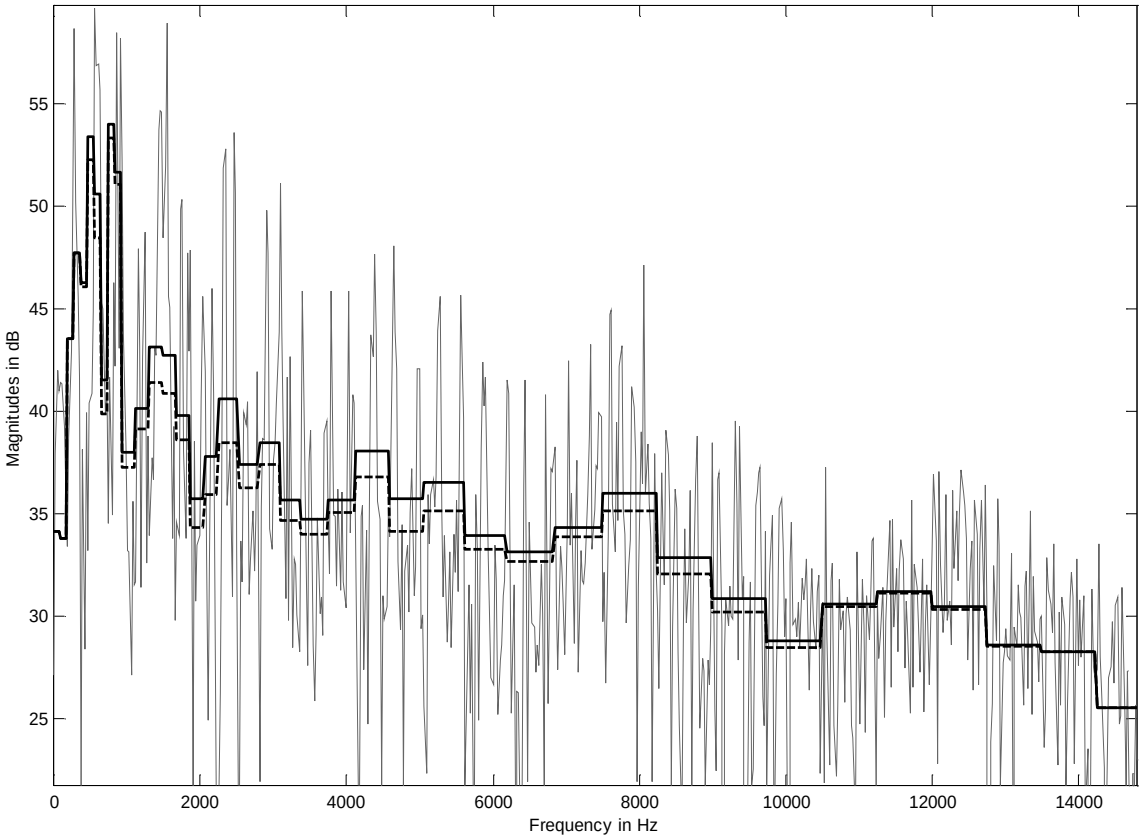

Figure 8. Spectrum (halftone) and stepwise masking thresholds estimated in AAC-SFBs. Masking thresholds with and without excess masking are shown as a solid and dashed line respectively.

2.3 Time-Frequency Analysis: Transforms and Filterbanks

Analysis-synthesis filterbanks or transforms are of particular importance in audio coding. They are used to decompose the audio signal into a set of compact time-frequency components that are representatives of the partition of audible spectrum as performed by the human auditory system. Given such a set, it is possible to discriminate between the perceptually relevant and irrelevant elements when used in conjunction with a perceptual model. Then various quantization techniques can be applied to represent the relevant time-frequency components with as little precision as possible, without introducing perceptible distortion.

The properties of the filterbanks should be matched to the characteristics of the incoming signal. This is, however, a very challenging task, because the characteristics of the music signals can be very different and may change abruptly. Therefore, some compromise must be found within engineering constraints. A few types of filterbanks are commonly used in established perceptual audio coding technologies, among which MDCT has played a dominant role. Three types of filterbanks are reviewed in the subsequent sections.

The historical development in employing filterbank/transform into audio coding is illustrated in Figure 9.

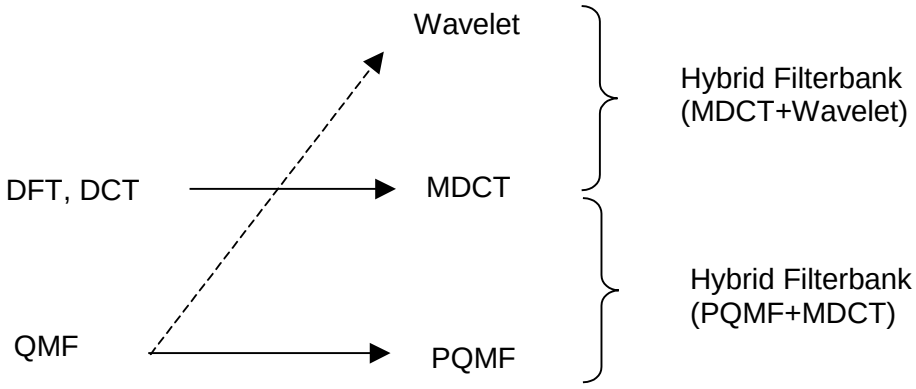

Figure 9. Development of filterbank/transform schemes applied in audio coding.

Historically, perceptual coding systems working in the frequency domain have been called either subband coders or transform coders. Subband coders normally use a low number of frequency-selective channels, processing samples, which are adjacent in time. Transform coders use a large number of channels and the simultaneous processing of samples that are adjacent in frequency.

Fundamentally, filterbanks/transforms in audio coding can be classified into two major classes. These are the quadrature mirror filterbank (QMF) and the modified discrete cosine transform (MDCT). The wavelet approach is closely related to QMF.

QMFs were the first filters to be used for low-bit-rate coding of music signals. The QMF uses the concept of frequency domain alias cancellation, while the MDCT uses the concept of time domain alias cancellation. This can be described as the duality of QMF and MDCT. However, it should be noted that MDCT also cancels frequency domain aliasing, while the QMF does not cancel time domain aliasing. In other words, MDCT is designed to achieve perfect reconstruction (PR), while QMF is not.

Early examples of transform coding used DFT and DCT. Commonly used window functions are rectangular and sine-taper functions. With a rectangular window the analysis/synthesis system is critically sampled, *i.e.*, the overall number of the transformed domain samples is equal to the number of time domain samples, but the system suffers from poor frequency resolution and block effects, which are introduced after quantization or other manipulation in the frequency domain. Overlapped windows allow for better frequency response functions but carry the penalty of additional values in the frequency domain, thus not critically sampled. MDCT is currently the best solution, achieving the three important requirements simultaneously. Those requirements are critical sampling, reduction of block effects and flexible window switching. The concept of the window switching was introduced to tackle possible pre-echo problems in the case of insufficient time resolutions [21][22].

To achieve backward compatibility and better coding performance with both stationary and transient signals, various hybrid filterbanks have been introduced. Well-known examples of hybrid structures are PQMF+MDCT in MP3 and Wavelet+MDCT in EPAC.

2.3.1 Pseudo Quadrature Mirror Filterbank (PQMF)

A 32-channel PQMF filterbank, also known as polyphase quadrature filterbank (PQF) [16] has been employed in all layers of MPEG-1 audio. It combines the filter design flexibility of generalized QMF banks with low computational complexity. The polyphase filterbank used in the MPEG audio coding system is described in [4]. For increased frequency resolution, Layer 3 employs a cascaded MDCT after PQMF to form a hybrid filterbank.

2.3.2 Modified Discrete Cosine Transform (MDCT)

The MDCT [18], also known as the modulated lapped transform (MLT) [19], was first proposed as a transform coding scheme performing time domain aliasing cancellation (TDAC). The time window for MDCT is constructed such that perfect reconstruction condition is satisfied [19]. The MDCT can be viewed as a dual of a QMF approach performing aliasing cancellation in frequency domain.

In addition to good energy compaction property MDCT combines simultaneously critical sampling, reduction of block effects and adaptive window switching capabilities. Therefore it has been widely applied in perceptual audio coding. However, mismatch of the MDCT with DFT based psychoacoustic models may be one reason behind a poor coding performance for some test signals.

This mismatch can be illustrated with a practical example of an MP3 audio coder. The output of a psychoacoustic model is the signal-to-masking ratio (*SMR*) calculated in the DFT domain. The maximal inaudible quantization error is calculated according to $MAX\_ERROR = \frac{ES}{SMR}$, where *ES* is the MDCT domain signal energy. Using a sinusoid as a test signal, the *SMR* is stable over time because DFT is an orthogonal transform. However, *ES* can fluctuate over time because it does not obey Parseval's theorem, thus causing an undesirable fluctuation of the MAX_ERROR over time. This phenomenon is referred to as the MDCT-DFT mismatch phenomenon in [P4].

The direct and inverse MDCT are defined as [17][18]:

$$\alpha_r = \sum_{k=0}^{2N-1} \tilde{a}_k \cos\left[\pi \frac{(k+(N+1)/2)(r+1/2)}{N}\right],\ r = 0,...,N-1 \tag{7}$$

$$\hat{a}_k = \frac{1}{N}\sum_{r=0}^{N-1} \alpha_r \cos\left[\pi \frac{(k+(N+1)/2)(r+1/2)}{N}\right],\ k = 0,...,2N-1 \tag{8}$$

where $\tilde{a}_k = h_k a_k$ is the windowed input signal, a_k is the input signal of $2N$ samples. h_k is a window function. $\hat{a}_k$ in (8) is the IMDCT coefficients of α_r, which contain time domain aliasing:

$$\hat{a}_k = \begin{cases} \frac{1}{2}\tilde{a}_k - \frac{1}{2}\tilde{a}_{N-1-k}, & k = 0,...,N-1 \\ \frac{1}{2}\tilde{a}_k + \frac{1}{2}\tilde{a}_{3N-1-k}, & k = N,...,2N-1 \end{cases} \tag{9}$$

If an identical analysis-synthesis time window is assumed, the constraints of perfect reconstruction are [19][20]:

$$h_k = h_{2N-1-k} \tag{10}$$

$$h_k^2 + h_{k+N}^2 = 1 \tag{11}$$

A sine window is widely used in audio coding because it offers good stop-band attenuation, provides good attenuation of the block edge effect and allows perfect reconstruction. Other optimized windows can be applied as well [20]. The sine window is defined as:

$$h_k = \sin\left[\pi\left(k+1/2\right)/2N\right], \quad k = 0,...,2N-1 \tag{12}$$

The relationship between MDCT and DFT can be established via Shifted Discrete Fourier Transforms (SDFT). The direct and inverse SDFTs are defined as [23]:

$$\alpha_r^{u,v} = \sum_{k=0}^{2N-1} a_k \exp\left[i2\pi\left(k+u\right)\left(r+v\right)/2N\right], \tag{13}$$

$$a_k^{u,v} = \frac{1}{2N}\sum_{r=0}^{2N-1} \alpha_r^{u,v} \exp\left[-i2\pi\left(k+u\right)\left(r+v\right)/2N\right], \tag{14}$$

where u and v represent arbitrary time and frequency domain shifts respectively. SDFT is a generalization of DFT that allows a possible arbitrary shift in position of the samples in the time and frequency domain with respect to the signal and its spectrum coordinate system.

It has been proven that the MDCT is equivalent to the SDFT of a modified input signal [S5][S6].

$$\alpha_r = \sum_{k=0}^{2N-1} \hat{a}_k \exp\left[i2\pi \frac{\left(k+\left(N+1\right)/2\right)\left(r+1/2\right)}{2N}\right] \tag{15}$$

The right side of (15) is $SDFT_{(N+1)/2,1/2} = \left(\alpha_r^{(N+1)/2,1/2}\right)$ of the signal $\hat{a}_k$ formed from the initial windowed signal $\tilde{a}_k$ according to (9). Physical interpretation of (9) and (15) is straightforward. MDCT coefficients can be obtained by adding the $SDFT_{(N+1)/2,1/2}$ coefficients of the initial windowed signal and the alias.

Another important physical interpretation of (15) is that the MDCT coefficients of a signal represent the real part of the corresponding SDFT spectrum. This has a useful implication for the design of an MDCT based audio encoder. That is, the frequency analysis in the psychoacoustic model and the MDCT can be implemented in one single block, thus reducing the computational complexity of the encoder [P4].

As in (16), the matrix of the MDCT for transforming 2N input samples to N spectral components is of size $N \times 2N$ and therefore cannot be orthogonal. However, the underlying basis functions of MDCT (corresponding to the rows of the matrix) are orthogonal.

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & \cdot & \cdot & \cdot & \cdot & p_{1,2N} \\ p_{2,1} & p_{2,2} & p_{2,3} & \cdot & \cdot & \cdot & \cdot & p_{2,2N} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ p_{N,1} & p_{N,2} & p_{N,3} & \cdot & \cdot & \cdot & \cdot & p_{N,2N} \end{bmatrix} \tag{16}$$

In the case of a continuous input stream, a block-diagonal matrix T can be made of MDCT matrices P on the diagonal and zeros elsewhere (see (17)). This block-diagonal matrix T for transforming $(n+1)\cdot N$ input samples to $n \cdot N$ spectral components is of size $(n \cdot N)\times[(n+1)\cdot N]$. T becomes an orthogonal and square matrix if $n \to \infty$.

$$X_{n\cdot N} = \begin{bmatrix} P & & & 0 \\ & P & & \\ & & \ddots & \\ & & & P \\ 0 & & & P \end{bmatrix}_{(n\cdot N)\times[(n+1)\cdot N]} \cdot x_{(n+1)\cdot N} = T \cdot x_{(n+1)\cdot N} \qquad (17)$$

where x is the input vector of the signal, X is the output vector of the MDCT coefficients. The orthogonality of T implies

$$T^T \cdot T = T \cdot T^T = I \qquad (18)$$

However, in the case of finite-length input signals, T is not anymore orthogonal. In order to illustrate this scenario in an intuitive way, let us observe a simple example with N=2 and n=5. In this case, the block-diagonal matrix appears as follows:

$$T = \left(\begin{array}{cccccc} \overbrace{P}^{3N/2=3} & & & & & \\ & P & & & 0 & \\ & & P & & & \\ & & & P & & \\ & 0 & & & P & \end{array} \right)_{10\times 12} \qquad (19)$$

That is,

$$T \cdot T^T = \begin{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & & & & & & 0 \\ & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & & & & & \\ & & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & & & & \\ & & & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & & & \\ & & & & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & & \\ 0 & & & & & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \end{bmatrix} = I \tag{20}$$

and

$$T^T \cdot T = \begin{bmatrix} \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} & & & & & & 0 \\ & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & & & & & \\ & & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & & & & \\ & & & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & & & \\ & & & & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & & \\ 0 & & & & & \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} & \end{bmatrix} \tag{21}$$

It is clear from (21) that the MDCT matrices of the first and last blocks are not orthogonal, though this usually does not cause a serious problem in audio coding applications. However, one should keep this effect in mind when manipulating audio signals in the compressed domain, such as editing, error concealment, etc.

With a regular block transform, we simply require that the number of samples in the signal is a multiple of the block size N. With MDCT, however, we need a slight modification in the computation of the transforms of the first and last blocks. Otherwise, the corresponding MDCT basis function would extend outside the region of support of the signal [19].

For a finite-length input vector, the infinite matrix T in (17) would be replaced by the finite matrix

$$T = \begin{bmatrix} P_a & & & 0 \\ & P & & \\ & & \ddots & \\ & & & P \\ 0 & & & P_b \end{bmatrix}_{(n \cdot N) \times (n \cdot N)} \qquad (22)$$

Matrices of the first and last blocks are denoted as P_a and P_b respectively. The block-diagonal matrix T becomes square only after some special handling of P_a and P_b, which would generate basis functions of length 3N/2, because there can be no overlapping outside the signal region of support [19]. N is an even number. After this handling, the dimension of both P_a and P_b becomes $N \times \left(\frac{3N}{2} \right)$ as illustrated in (19), thus T becomes a square matrix as shown in (22). It corresponds to the square matrix shown in dashed line in (19).

This has some implications for practical applications such as compressed domain audio processing discussed in the later part of this dissertation.

The following are some conclusions based on the author's investigations of MDCT:

- MDCT becomes asymptotically an orthogonal transform with increasing length of the signal. If the signal length is infinite, MDCT is an orthogonal transform. This is different from the traditional definition of orthogonality, which must satisfy (18) and requires a square transform matrix.
- The MDCT spectrum of a signal is the Fourier spectrum of the signal mixed with its alias. This compromises the performance of MDCT as a Fourier spectral analyser and leads to possible mismatch problems between MDCT and DFT based perceptual models [P3][P4]. Nevertheless, MDCT has been successfully applied to perceptual audio compression without major problems if a proper window such as a sine window is employed.
- The TDAC of an MDCT filterbank can only be achieved with an overlap-add (OA) process in the time domain. Although MDCT coefficients are quantized in an individual data block (*e.g.* granule in MP3), it is usually analyzed in the context of a continuous stream. In the case of discontinuity such as editing or error concealment, the aliases of the two neighboring blocks in the overlapped area are not able to cancel each other. This will be further discussed in Chapter 3.
- MDCT can achieve perfect reconstruction only without quantization, which is never the case in coding applications. If we model the quantization as a superposition of quantization noise to the MDCT coefficients, then the time domain alias of the input signal is still cancelled, but the noise components will be extended as additional "noise alias". In order to have 50% window overlap and critical sampling simultaneously, the MDCT time domain window is twice as long as that of ordinary orthogonal transforms such as DCT. Because of the increased time domain window length, the quantization noise is spread to the whole window, thus making pre-echo more likely to be audible. Well-known solutions to this problem are window switching [21][22] and temporal noise shaping (TNS) [52].
- In very low bitrate coding, the high frequency components are often removed. This corresponds to a very steep lowpass filter. Due to the increased window size, the ringing effect caused by high frequency cutting is longer.

2.3.3 Discrete Wavelet Packet Transform (DWPT)

A significant part of recent audio coding research has been work on time-varying signal adaptive filterbanks constructed from DWPTs. A comprehensive review of this subject is given in [8] with many relevant references. The DWPT based coding methods have two major attractive properties: 1) flexible time-frequency tiling so that it is possible, for example, to approximate the critical band auditory filterbank utilizing a wavelet packet approach; 2) flexible scalability can be achieved with proper formatting. Due to its characteristics of multiresolution decomposition, DWPT is widely adopted in scalable image coding. Considering the tree structure of DWPT, the low resolution components here can be viewed as the low frequency components that normally represent the most important information, while the high resolution components represent the less important details.

If we consider the scenario of delivering audio content over networks with different bandwidths, it would be very advantageous to store or transmit only one high resolution bitstream, since it automatically contains all possible lower resolution bitstreams to match the available network bandwidth. Theoretically, DWPT based algorithms provide a framework with attractive features such as the possibility of almost continuous bitrate scalability. However, the high computational complexity and a relatively moderate coding performance have prevented widespread applications of DWPT in audio coding. It is worth noting that MPEG-4 bit sliced arithmetic coding (BSAC) has also provided a framework for scalable audio coding based on MDCT [14]. For these reasons, DWPT has not been a focus of this dissertation.

2.4 Quantization and Entropy Coding

The reduction in bitrate takes place in the quantization and coding of the audio signal. The spectral components are quantized and coded with the aim of keeping the quantization noise below the masking threshold. Depending on the algorithm, this step is done in very different ways, from simple block companding to analysis-by-synthesis systems using additional lossless coding. As an example of this latter approach, the quantization and entropy coding of MPEG-1 Layer 3 is explained as follows [51]. MPEG2/4 AAC uses a similar approach.

A system of two nested iteration loops is the common solution for quantization and coding in MPEG-1 Layer 3. Quantization is done via a power-law quantizer. In this way, larger values are automatically coded with less accuracy and some noise shaping is already built into the quantization process.

The noise shaping is accomplished in the quantization process with the objectives to control the bitrate and to keep the quantization noise below the masking threshold. In the established coding schemes such as MP3 and AAC, a global gain value (determining the quantization step size) and the scale-factors (determining noise level for each scale-factor band) are applied before actual quantization. The process of finding the optimum gain and scale-factors for a given block, bitrate and output from the perceptual model is usually done by two nested iteration loops in an analysis-by-synthesis way. The inner iteration loop is designed to control the overall bitrate by adjusting the global gain. The outer iteration loop is designed for shaping quantization noise according to the masking threshold by adjusting the scale-factors.

The quantized values are coded by an entropy-coding scheme. Huffman coding has been employed in MPEG audio. To adapt the coding process to different local statistics of music signals, the optimum Huffman table is selected from a number of choices. To achieve even better adaptation to signal statistics, different Huffman tables can be selected for different parts of the spectrum. The Huffman coding works on pairs and quadruples (only in the case of very small numbers to be coded) of quantized MDCT coefficients.

2.5 Frame Formatting

Frame formatting is used to assemble the bitstream, which typically consists of the quantized and coded mapped samples and some side information, such as bit allocation information [38]. In an audio coding standard, frame formatting is always in the normative part to ensure interoperability. Different audio formatting directly affects its error resilience and adaptivity to channel conditions.

2.6 Other Coding Tools

To further improve coding efficiency of perceptual coding schemes including only basic tools, some additional tools have been developed. Some of these tools are briefly summarized here.

2.6.1 Temporal Noise Shaping (TNS)

The TNS tool is used prior to quantization/coding of the spectral values to exercise some control over the temporal fine structure of the quantization noise within each filterbank window [52][53]. TNS is effective if we have distinct temporal structures in tonal signals such as speech. In this case, instead of switching to short windows too often, thus sacrificing coding efficiency, we can employ TNS to control the temporal envelope of the quantization noise. The basic concept of TNS is the application of a forward predictor in the frequency domain (*i.e.* along the frequency axis). This leads to a correlation of neighboring quantization error values. From time-frequency duality it follows that the corresponding convolution in the frequency domain is equivalent to a multiplication in the time domain. The application of TNS makes the correlations of the quantization noise similar to those of the signal spectral components, thus also making their temporal envelopes similar in the time domain.

2.6.2 Perceptual Noise Substitution (PNS)

The PNS tool [54] allows for a very compact representation of noise-like signal components and in this way further increases compression efficiency for certain types of input signals. The concept of the PNS technique can be described as follows:

- In the encoder, noise-like components of the input signal are detected on a scale-factor band basis.
- The groups of spectral coefficients belonging to scale-factor bands containing noise-like signal components are not quantized and coded as usual but omitted from the quantization/coding process.
- Instead, only a noise substitution flag and the total power of the substituted spectral coefficients are transmitted for each of these bands.
- In the decoder, pseudo random vectors with the desired total noise power are inserted for the substituted spectral coefficients.

2.6.3 Long Term Prediction (LTP)

LTP is a technique that is well-known in speech coding and has been used to exploit redundancy in the speech signal, which is related to the signal periodicity as expressed by the speech pitch. While use of long term prediction in common speech coders happens in a time-domain coder framework, the MPEG-4 audio LTP tool [55] has been integrated into the framework of a generic perceptual audio coder where quantization and coding is performed on a spectral representation of the input signal.

As can be expected due to the underlying principle, the LTP tool provides considerable coding gain for stationary harmonic tonal signals such as pitch pipe as well as some gain for non-harmonic tonal signals.

2.6.4 MPEG-2 AAC Codec Structure

After a discussion of most of the relevant audio coding tools, the MPEG-2 AAC codec structure (see Figures 10 and 11) is illustrated here to show how different coding tools can be combined to perform efficient audio compression.

In order to allow a tradeoff between the quality and the memory and processing power requirements, the AAC system offers three profiles [12]:

- Main profile, which provides the best audio quality at any given data rate at the cost of the highest computational complexity and memory requirement;
- Low-complexity (LC) profile, which provides very good quality performance, while having the lowest memory and processing power requirements; This feature makes the LC profile attractive for many applications;
- Scalable sampling rate (SSR) profile, which can provide frequency scalability with four equally divided bandwidths.

Based on the analysis so far, we can conclude that the four basic tools play a central role in perceptual audio coding. Other tools are introduced to improve coding gain with specific profiles or signals. For example, the gain control tool is only required in the SSR configuration. The preprocessing performed by the gain control tool consists of a PQF, gain detectors and gain modifiers. The PQF splits the input signal of each audio channel into four frequency bands of equal width, which are critically decimated. The output of each filterbank has gain modification as necessary and is processed by the MDCT tool to produce 256 spectral coefficients, for a total of 1024 coefficients. Gain control can be applied to each of the four bands independently [13].

Intensity stereo coding/coupling and M/S stereo coding have been employed in AAC for coding multichannel audio. Coupling is adopted based on psychoacoustic evidence that at high frequencies (above approximately 2 kHz) the human auditory system localizes sound based primarily on the envelopes of critical-band-filtered versions of the signals reaching the ears, rather than on the signals themselves. MS stereo coding encodes the sum and difference of the signals in two symmetric channels instead of the original signals in left and right channels [13]. Both MS stereo and intensity stereo coding operate on channel-pair-elements (CPEs).

MPEG-2 AAC is a good representative of the current state-of-the-art technology in high quality audio coding. It is also used as the kernel in the MPEG-4 General Audio coding, which is extended by additional coding tools and coder configurations [11].

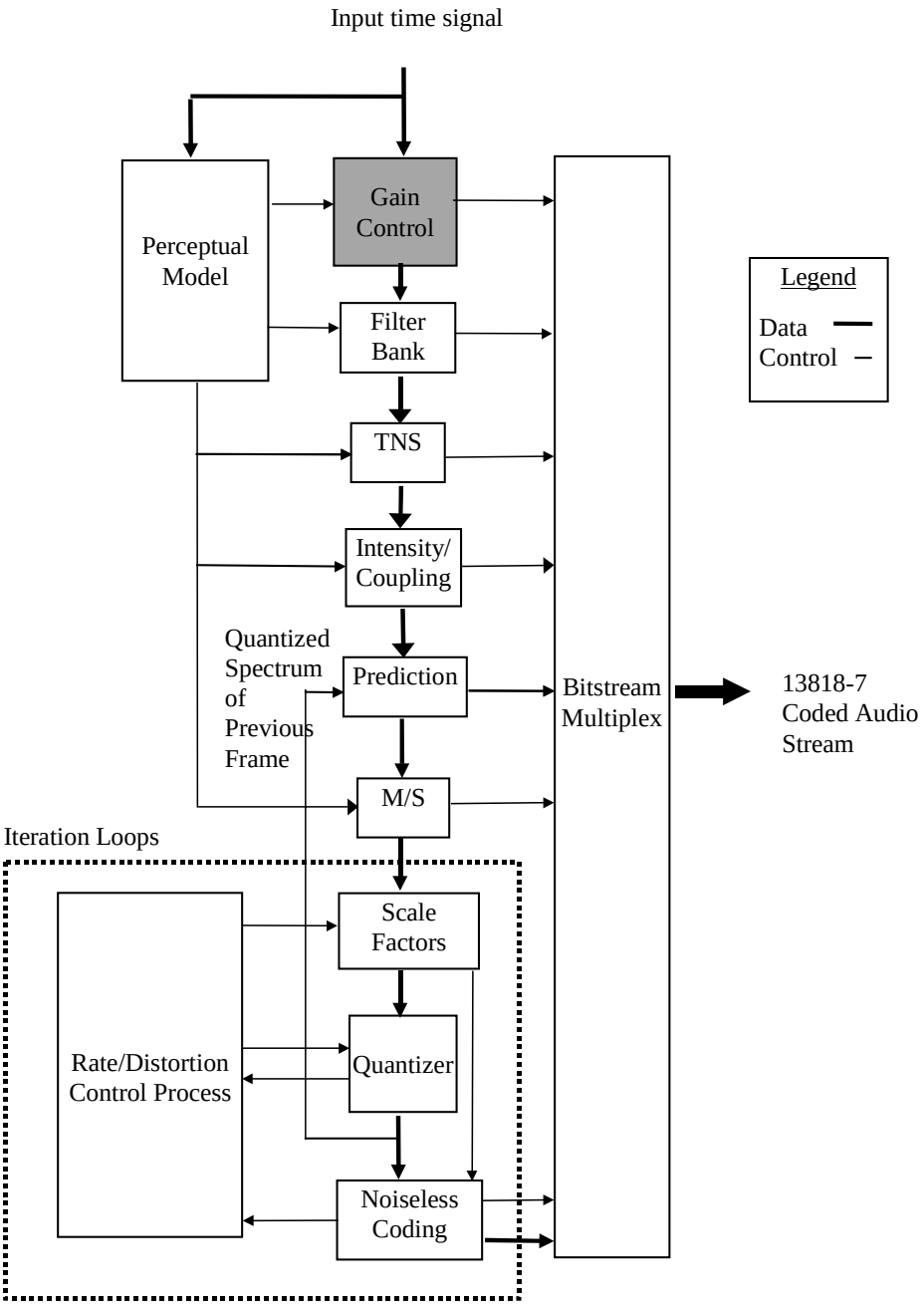

Figure 10. Block Diagram for the MPEG-2 AAC Encoder [12].

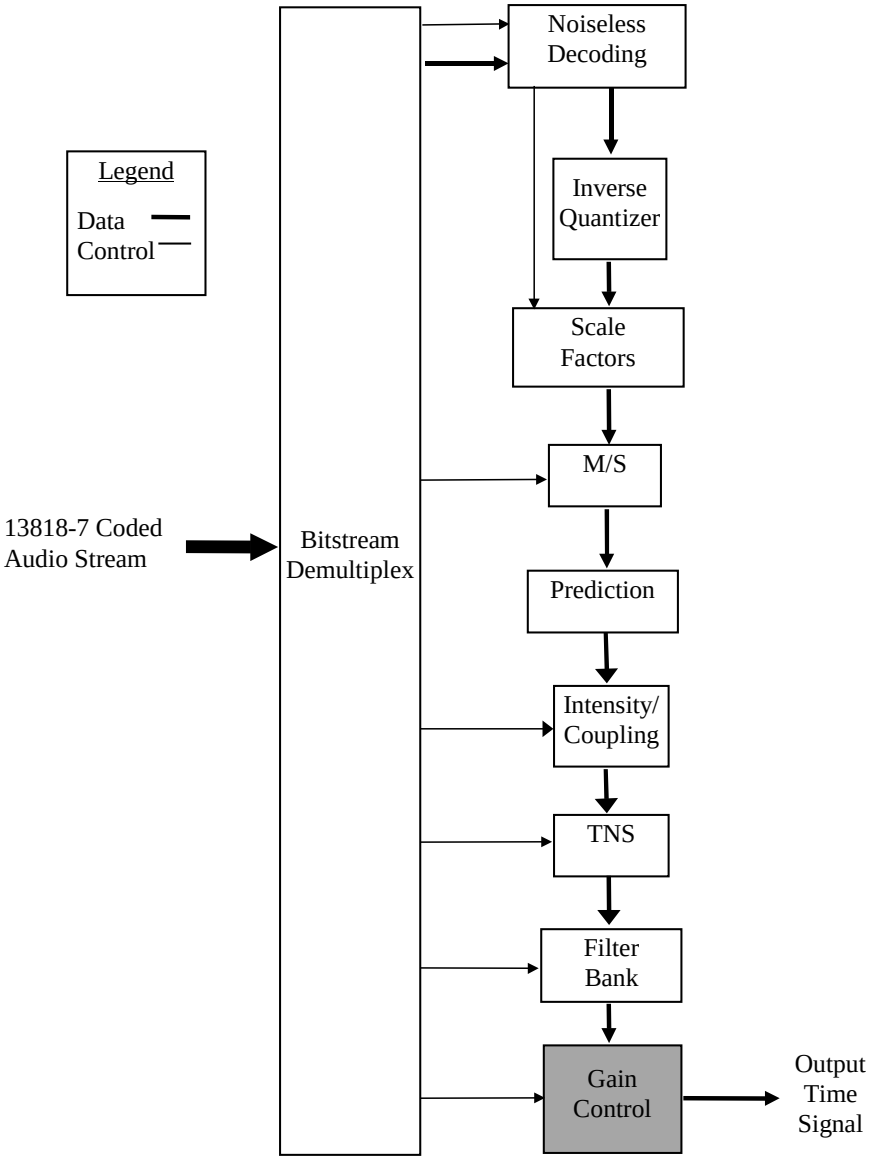

Figure 11. Block Diagram for the MPEG-2 AAC Decoder [12].

2.7 Integer-to-Integer DCT (INT-DCT)

In comparison with the existing coding tools discussed previously, the INT-DCT tool has not been included in any specific standard. Rather, it represents a relevant contribution of the author's research.

The motivation for lossless coding of previously quantized multichannel audio spectral components is to avoid the so-called "binaural unmasking" effect due to binaural masking level difference (BMLD) [46][47]. This is characterized by a decreasing masking threshold when the masker is spatially separated from the maskee, i.e., when the masker and maskee arrive at the ears from different directions.

An important feature of the proposed approach is that the spatial structures of the quantization errors are not modified, thus it does not suffer from any binaural unmasking effect.

The DCT is a well-known and effective decorrelation transform when used e.g. with speech or image signals, since it exhibits an energy compaction property similar to that of the Karhunen-Loeve transform (KLT) [7]. An INT-DCT approximates the DCT and is computationally very efficient. More importantly, with the proposed INT-DCT tool the quantized multichannel MDCT coefficients can be perfectly reconstructed at the decoder. The work performed in [P5] was to investigate how well the INT-DCT performs interchannel decorrelation with real-life multichannel audio signals in terms of net bit-reduction with regard to the associated side information. The scheme has been implemented based on an AAC codec.

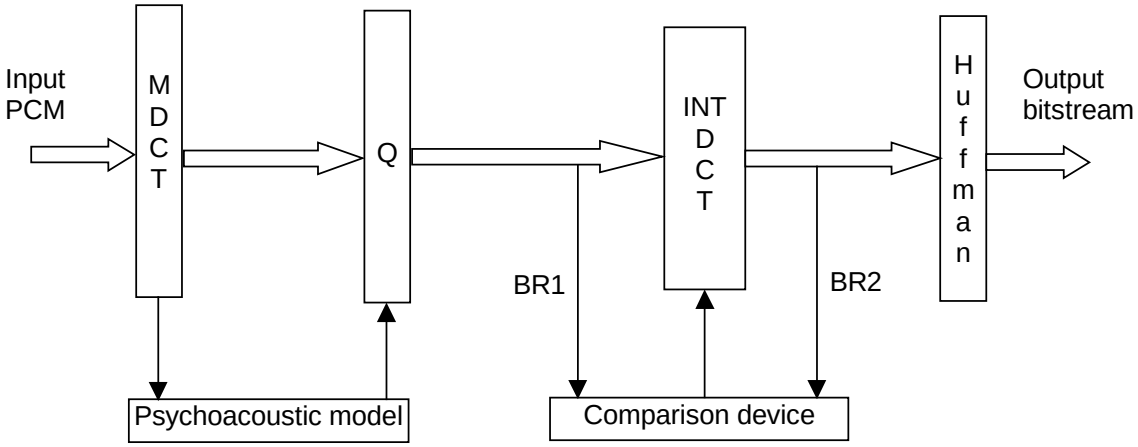

Figure 12. A simplified block diagram of the INT-DCT based multichannel-coding scheme.

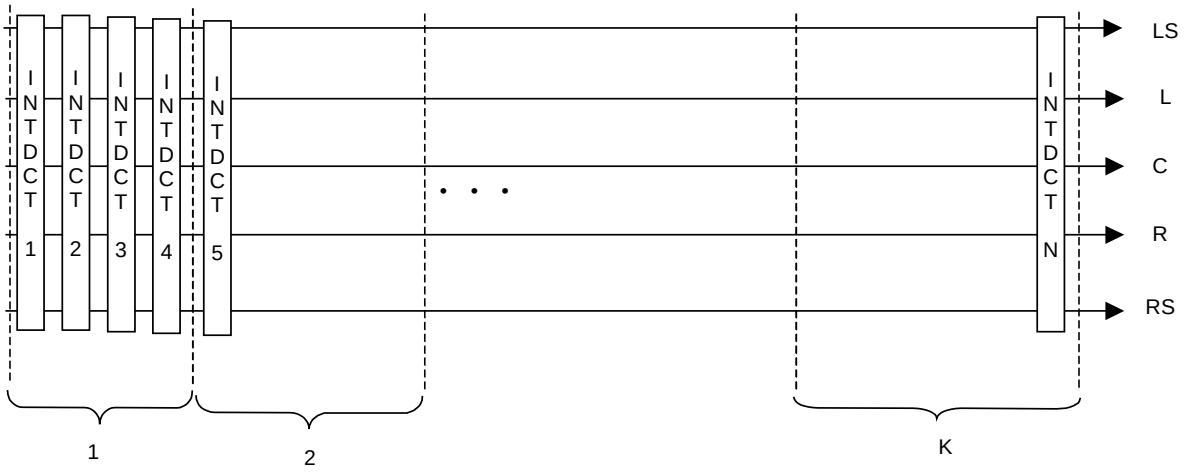

Figure 13. Block diagram illustrating actual INT-DCT operation. K indicates the SFB index.

The scheme is illustrated in Figure 12. The input signals to the comparison device are the total bits (BR1 and BR2) needed to represent the MDCT coefficients within each SFB for all channels before and after the INT-DCT. The output of the comparison device is the control signal to indicate whether INT-DCT is performed in that SFB. Therefore, the side information is only one bit for each SFB. The control signal

needs to be transmitted as side information. The rationale for performing INT-DCT within each SFB as illustrated in Figure 13 is to reduce side information.
The most widely adopted 5.1 multichannel configuration has been employed in the experiments. In Figure 13, the horizontal lines represent the MDCT coefficients of all channels, which refers to left (L), center (C), right (R), left surround (LS), right surround (RS). The optional low-frequency-enhancement (LFE) channel has not been used in the experiments. The dashed lines represent the SFB division.

Because of the energy compaction property of the MDCT, the MDCT coefficients in high frequencies are mostly zeros. Therefore, the INT-DCT has been limited to low and middle frequencies for a reduced computational complexity and side information.

Based on the relatively small scale of experiments, the net bitrate reduction is rather moderate by employing INT-DCT tool. It will be interesting to investigate the coding performance of INT-DCT with an increased number of channels, *e.g*. 48 channels.

Chapter 3 Compressed Domain Audio Processing

3.1 Overview of Compressed Domain Audio Processing

With rapid deployment of audio compression technologies, more and more audio content is stored and transmitted in compressed formats. The transmission of audio signals in compressed digital packet formats, such as MP3, has revolutionized the process of music distribution. Consequently, compressed domain audio processing is becoming a subject of study.

Compressed domain audio processing can be defined as the ability to perform modifications to coded digital audio or to extract features directly in its compressed form. Examples are dynamic range compression, time scale modifications, pitch scale modifications, audio segmentation, etc.

The focus of this chapter is on error concealment in streaming music applications. A compressed domain beat detector has been developed as a part of the error concealment scheme. Re-compression of compressed audio has also been investigated.

3.2 Error Resilient Delivery of Compressed Audio

In the transmission of compressed audio, one of the most significant challenges today is the need to handle errors in lossy channels. Lossy channels can arise in many different forms. In packet networks, such as the Internet, packets can be dropped due to congestion at switches, they can be misrouted, or they can arrive with such a long delay as to be useless. In wireless channels, a host of different mechanisms including fading, interference, and additive white noise can cause bits to arrive in error [56]. For real-time or near real-time audio transmission, some quality degradation is often acceptable, allowing a wider range of solutions.

In the battle against errors, there are many different methods available to the system designer. For example, retransmission of lost packets is an obvious means by which loss may be remedied. It is clearly of value in non-interactive applications, with relaxed delay bounds, but the delay imposed means that it does not typically perform well for interactive use. In addition to the possible high latency, there is a potentially large bandwidth overhead for re-transmission, thus increasing the network congestion probability and hence packet loss, leading to exacerbation of the problem the scheme was intended to solve.

A suitable network transport system is vital for delivery of audio with acceptable quality of service (QoS). A preferred architecture for delivering audio and other multimedia content over Mobile Internet is the

TCP/IPv6 protocol suite with its recent improvements, especially in QoS aspect. These new features are RTP/RTCP, UDP as application transport part and DiffServ and RSVP as QoS system part. A well-engineered application performing audio streaming on top of IPv4 or IPv6, is obviously necessary but not sufficient to provide QoS. The system aspects and the network characteristics also have an important role to play. It is worth mentioning that transforming the traditional best-effort Internet into a QoS Mobile Internet is not so straightforward because of many practical reasons, such as the heterogeneous nature of the Internet.

In this chapter, the channel error characteristics are presented. Then some primary methods to handle channel errors such as interleaving; error detection/correction, error resilience, and error concealment are reviewed. Error handling measures have costs in terms of complexity, and in terms of added bitrate. Based on the channel characteristics and the cost/benefit analysis, an optimal combination of these methods can be designed.

Error resilient audio transmission is of particular importance in wireless networks due to its error-prone channel characteristics.

3.2.1 Overview of Channel Error Characteristics

If the objective is error resilient transmission of compressed audio over error-prone channels, it is necessary to have some knowledge of the error characteristics likely to be encountered. In this section, errors in three types of channels, namely mobile network, Internet and Mobile Internet are briefly reviewed.

3.2.1.1 Channel Error Characteristics of Mobile Networks

Channel errors in wireless networks typically consist of statistically independent random errors and burst errors. The quality of the received signal depends on many possibly interdependent factors, which, however, are usually independent of the used system. These factors include channel coding methods, modulation schemes, system bandwidth, antenna type and directivity, effective transmission power, surrounding noise level, weather conditions, slow/fast fading, multipath propagation due to scattering, reflection, refraction, or diffraction, hand-off handling, propagation loss, etc. *Carrier-to-Interference (C/I)* is a commonly used term to describe the minimum ratio of the desired signal levels (C) to the interference levels (I).

Handover is a particularly important process in a cellular mobile network when a mobile terminal is traveling from one cell to another. Improper handover can result in dropped calls and crosstalk.

The bit error rate (BER) in a realistic mobile network such as GSM can reach 1% or even higher with possible burst errors in bad situations. The dominant errors in mobile networks can generally be assumed to be random errors, especially when an interleaving scheme is employed. There are many good references on mobile radio propagation and channel analysis [58][59][60].

3.2.1.2 Channel Error Characteristics of the Internet

The dominant errors in the Internet are packet loss and excessive delay mainly due to congestion. Quite extensive reviews on error characteristics in different networks can be found in [61][62][63]. The error characteristics of different application scenarios/networks are significantly different. It is beyond the scope of this dissertation to cover such a vast and fast developing area. For real-time audio streaming, IP Multicast seems to be the choice and it is therefore outlined in this section.

A number of studies have been conducted [64][65] on the loss characteristics of the Internet multicast backbone (Mbone) and although the results vary somewhat, the broad conclusion is clear: in a large conference it is inevitable that some receivers will experience packet loss [61][62]. Packet traces broadly show a session in which most receivers experience loss in the range of 2-5%, with a somewhat smaller number experiencing significantly higher loss rates.

It has also been shown that the vast majority of losses are of single packets. Burst losses of two or more packets are around an order of magnitude less frequent than single packet loss, although they do occur more often than would be expected from a purely random process. Longer burst losses (of the order of tens of packets) occur infrequently. These results are consistent with a network where small amounts of transient congestion cause the majority of packet loss. In a few cases, a network link is found to be severely overloaded and a large amount of loss results.

The primary focus of a repair scheme must, therefore, be to correct single packet loss, since this is by far the most frequent occurrence. It is desirable that losses of a relatively small number of consecutive packets may also be repaired, since such losses represent a small but noticeable fraction of observed losses. The correction of large burst loss is of considerably less importance.

It is necessary to define some terminology and protocol framework for consistency. A *unit* is defined to be a timed interval of media data, typically derived from the workings of the media coder, such as an MP3 granule or an AAC frame. A packet typically comprises one or more units encapsulated for transmission over the network. For example, many audio coders operate on 20 ms units, which are typically combined to produce 40 ms or 80 ms packets for transmission. The framework of Real-time Transport Protocol (RTP) is assumed. This implies that packets have a sequence number and timestamp. The sequence number denotes the order in which packets are transmitted, and is used to detect losses. The timestamp is used to determine the playout order of units. Most loss recovery schemes tackle primarily the scenario that units are received out of order, so an application must use the RTP timestamp to schedule playout [62].

It should be noted that many published works on streaming audio over various networks have concentrated on PCM coded speech signals. If the unit length is fixed in bytes, the duration of the audio then depends on the actual coding scheme with a certain bitrate and sampling frequency. For example, assume that the payload of a packet is 800 bytes. If we use a PCM-based mono speech sample with 8 bits/sample and 8 kHz sampling frequency, the duration of the payload is then 100 ms. Keeping the payload length of 800 bytes unchanged, we compress a stereo music sample from a commercial CD (16 bits/sample and 44.1 kHz sampling frequency) using MP3 with a bitrate of 128 kbps, and thus the duration of the payload would be 50 ms. However, the duration would be about 4.5 ms, if the music sample had not been compressed.

3.2.1.3 Channel Error Characteristics of the Mobile Internet

This is a situation in which mobile terminals via Wireless LAN, 3G or Bluetooth access e.g. the Internet. Since Mobile Internet is a rather vague term, it is considered here only as a mathematical model, in which the channel errors are modeled as a simple combination of two independent error sources. In wireless networks, losses are more commonly due to external factors such as those discussed in section *3.2.1.1*, rather than congestion in the Internet. It is evident from this that the errors in Mobile Internet will be more severe than those in mobile networks or Internet alone, thus making it a challenging task to provide QoS in such hybrid networks. Only a system level solution including a proper network protocol and a joint sender-receiver based error recovery will be able to meet the challenge.

3.2.2 Sender-based Error Recovery

Sender-based error recovery techniques can be classified into two classes; these are active sender-based methods such as retransmission and passive sender-based methods such as interleaving, forward error correction coding (FEC) and other error resilient tools. The material in 3.2.2 and 3.2.3 is mainly borrowed from [61].

3.2.2.1 Retransmission

This is an active sender-based repair technique employing automatic repeat request (ARQ). Due to potentially long delay, retransmission-based recovery is not the first choice for lost packets in real-time applications such as media streaming. If large end-to-end delays can be tolerated, the use of retransmission to recover from loss becomes a possibility [61].

3.2.2.2 Interleaving

Interleaving is a process of rearranging the order of a sequence of binary or non-binary symbols in some unique one-to-one deterministic manner. The reverse of this process is de-interleaving to restore the sequence to its original ordering [57].

Interleaving is often employed between the channel encoder and the modulator to enhance the error correction performance especially in wireless communications [57] as well as in the Internet [66]. Since channel coding techniques are designed to combat random independent errors, interleaving is deployed to disperse burst errors and to reduce the concentration of the errors that must be corrected by the channel code. Thus interleaving effectively makes the channel appear like a random error channel to the decoder [57].

A major advantage of interleaving is that it does not increase the bandwidth requirements. An obvious disadvantage of interleaving is that it increases delay/latency. Therefore, a tailored tradeoff between error performance and interleaving delay has to be made for a specific application.

Since the focus of this dissertation is on packet networks, the interleaving scheme is extended to packet interleaving. For streaming audio, the data unit size is usually smaller than the packet size. The units (instead of symbols) are re-sequenced before transmission so that originally adjacent units are separated by a guaranteed distance in the transmitted stream and returned to their original order at the receiver. Packet interleaving disperses the effect of burst packet losses.

3.2.2.3 Error Detection/Correction

In the FEC approach, extra (redundant) bits are added to a coded audio bitstream. If some bits arrive in error, the redundancy usually allows the decoder to detect that a portion of the stream is erroneous, and it may allow the decoder to correct the errors. In a digital audio bitstream, especially compressed domain bitstream, bits are normally of unequal importance. For example, side information is perceptually more important than the main data. Therefore, unequal error protection (UEP) schemes are developed to give more important bits more protection. UEP is an efficient method to improve error robustness of compressed audio bitstreams and is widely used in various speech and audio coding systems designed for error-prone channels such as mobile communication networks and digital audio broadcasting (DAB).

3.2.2.4 Error Resilience

In error resilience, we are concerned with mechanisms that do not directly correct errors, but which limit the extent of the damage these errors cause. In the worst case, one might imagine a system in which a single error in a bitstream can render the entire stream useless from beginning to end. With a resilient system, an error is confined to just one or several data units. This may involve resynchronization words or resynchronization points, so that the decoder can get back on track after an error occurs. It may also involve the design of transform/quantization schemes and source reshaping such that the coded bitstream becomes more robust to channel errors [56]. For example, MPEG-4 has error resilience tools, which are designed for error prone channels.

3.2.3 Error Concealment

Error concealment is an important method to mitigate the degradation of audio quality in real-time streaming applications. Error concealment usually refers to post-processing error concealment methods, or those methods where the decoder, recognizing that an uncorrected error has occurred, seeks to hide or minimize the distortion from the listener so that a subjectively more pleasant rendition of the decoded audio can be obtained. For digital audio, post-processing error concealment can consist of time domain methods, in which one typically tries to interpolate across missing samples, and frequency domain methods, in which one tries to estimate lost transform coefficients. It is also possible to combine the time and frequency domain methods. Existing methods are briefly reviewed in this section. A few commonly used conventional error concealment schemes are depicted in Figure 14.

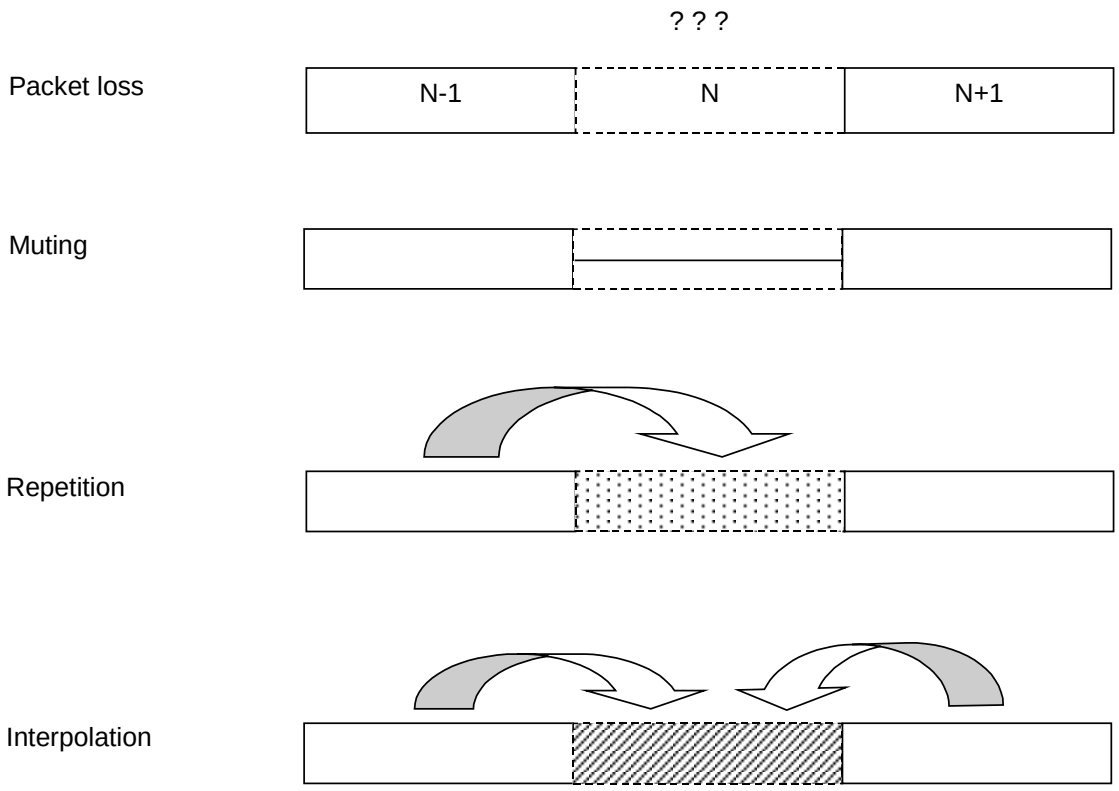

Figure 14. Examples of a few commonly used conventional error concealment schemes

It should be noted that existing methods were tested extensively with uncompressed speech signals only. To verify their effectiveness with compressed generic audio bitstreams, especially music bitstreams, further research is clearly needed. One major contribution of this dissertation is some advances in this direction.

3.2.3.1 Insertion-based Schemes

Insertion based repair schemes derive a replacement for a lost packet by inserting a simple fill-in. The simplest case is splicing, where a zero-length fill-in is used; an alternative is silence substitution, where a fill-in with the duration of the lost packet is substituted to maintain the timing of the stream. Better results are obtained by using noise or a repetition of the previous packet as the replacement [61].

The distinguishing feature of conventional insertion-based repair techniques is that the characteristics of the signal are not used to aid reconstruction, thus making these methods simple to implement but resulting in generally poor performance.

A major contribution of this dissertation is to improve the performance of the insertion-based scheme by exploiting beat-pattern characteristics of music signals. This will be further discussed in a subsequent section.

- Splicing

Lost units can be concealed by splicing together the audio on both sides of the loss; no gap is left due to a missing packet, but the timing of the stream is disrupted. This technique has been evaluated by Gruber and Strawczynski [67] and shown to perform poorly. Low loss rates and short clipping length (4-16 ms) faired best, but the results were intolerable for losses above 3 percent.

The use of splicing can also interfere with the adaptive playout buffer required in a packet audio system, because it makes a step reduction in the amount of data available to buffer. The adaptive playout buffer is used to allow for the reordering of misordered packets and removal of network time jitter, and poor performance of this buffer can adversely affect the quality of the entire system.

It is clear, therefore, that splicing together audio on both sides of a lost unit is not an acceptable repair technique [61].

- Silence Substitution/Muting

Silence substitution fills the gap left by a lost packet with silence in order to maintain the timing relationship between the surrounding packets. It is only effective with short packet length (<4 ms) and low loss rates (< 2%) [68], making it suitable for interleaved audio over low-loss paths.

The performance of silence substitution degrades rapidly as packet sizes increase, and quality is unacceptably bad for the 40 ms packet size in common use in network audio conference tools [69]. Despite this, the use of silence substitution is widespread, primarily because it is simple to implement.

- Noise Substitution

Since muting has been shown to perform poorly, an obvious next choice is noise substitution, where, instead of filling in the gap left by a lost packet with silence, background noise is inserted instead.

A number of studies of the human perception of interrupted speech have been conducted, *e.g.* [71]. These have shown that phonemic restoration, the ability of the human brain to subconsciously repair the missing

segment of speech with the correct sound, occurs for speech repair using noise substitution but not for silence substitution.

In addition, when compared to silence, the use of white noise has been shown to give both subjectively better quality and improved intelligibility. It is therefore recommended as a replacement for silence substitution [70][71].

As an extension to this, a proposed future revision of the RTP profile for audio-video conference allows for the transmission of comfort noise indicator packets. This allows the communication of the loudness level of the background noise to be played, allowing for better fill-in information to be generated.

- Repetition

Repetition replaces lost units with copies of the unit arriving immediately before the loss. It has low computational complexity and performs reasonably well. The subjective quality of repetition can be improved by gradually fading repeated units. The GSM system, for example, advocates the repetition of the first 20 ms with the same amplitude followed by fading the repeated signal to zero amplitude over the next 320 ms [72].

The use of repetition with fading is a good compromise between the other poorly performing insertion-based concealment techniques and the more complex interpolation-based and regenerative concealment methods.

3.2.3.2 Interpolation-based Schemes

Interpolation-based error concealment techniques utilize packets from both sides of the lost segment to produce a replacement for the lost packet. The advantage of these types of schemes over insertion-based techniques is that they account for the changing characteristics of a signal. Most of the reported works concentrate on speech signals.

- Waveform Substitution

Waveform substitution uses audio before, and, optionally, after the failure to find a suitable signal to cover the loss. The use of waveform substitution in packet voice systems has been studied by Goodman *et al*. [73]. They examined one- and two-sided techniques that use templates to locate suitable pitch patterns on either side of the loss. In the one-sided scheme the pattern is repeated across the gap, but with the two-sided schemes interpolation occurs. The two-sided schemes generally performed better than one-sided schemes, and both work better than silence substitution and packet repetition.

- Pitch Waveform Replication

Wasem *et al*. [74] presented a refinement on waveform substitution by using a pitch detection algorithm on either side of the loss. Losses during unvoiced speech segments are repaired using packet repetition and voiced losses repeat a waveform of appropriate pitch length. The technique, known as pitch waveform replication, was found to work marginally better than waveform substitution.

- Time Scale Modification

Time scale modification allows the audio on either side of the loss to be stretched across the loss. Sanneck *et al*. [75] present a scheme that finds overlapping vectors of pitch cycles on either side of the loss, offsets them to cover the loss, and averages them where they overlap. Although computationally demanding, the technique appears to work better than both waveform substitution and pitch waveform replication.

3.2.3.3 Regeneration-based Schemes

Regenerative repair techniques use knowledge of the audio compression algorithm to derive codec parameters, such that audio in a lost packet can be synthesized. These techniques are necessarily codec-dependent but perform well because of the large amount of state information used in the repair. Typically, they are also somewhat computationally intensive.

- Interpolation of Transmitted State

For codecs based on transform coding or linear prediction, it is possible for the decoder to interpolate between states. For example, the ITU G.723.1 speech coder [77] interpolates the state of the linear predictor coefficients from both sides of short losses and uses either a periodic excitation the same as in the previous frame, or a gain-matched random number generator, depending on whether the signal was voiced or unvoiced. These types of methods have the potential advantage of producing little boundary effects.

- Model-Based Recovery

In model-based recovery the speech on one or both sides of the loss is fitted to a model that is used to generate speech to fill the gap. This type of approaches also assume that the lost block is short enough to ensure that the speech characteristics of neighboring blocks are similar. An example of such methods is presented in [76].

3.3 Drumbeat-Pattern based Error Concealment Scheme

For speech communications in a packet network, the use of repetition is recommended as offering a good compromise between achieved quality and excessive complexity [61]. There are considerably fewer publications on error concealment schemes for compressed music streams over packet networks, which is one of the major focus areas of this dissertation. An MP3 and an AAC codec have been employed to perform our investigations.

The rationales for conducting this reseach are as follows:

1) Music streaming over error prone channels such as Mobile Internet is becoming an important application scenario. And music with a quite regular drumbeat structure such as pop, dance, rap, etc. represents an important class of music in streaming applications.
2) Percussive sounds such as drums and hi-hats are very common in such music and represent the transients. The transients in music are usually associated with beats, which are fairly stable and repetitive. The beat characteristic is one of the most important features that makes the music flow unique and differentiates it from other audio signals.
3) Examination of the MP3 and AAC audio bitstreams produced from pop music signals in commercial CDs has shown that 20-50% of data units in our test signals represent the population of the transients. These transients often appear in the form of window switching in most bitstreams of the state-of-the-art audio codecs. Therefore, a proper handling of packet loss during transients poses a real problem and greatly affects overall quality.

3.3.1 Limitations of Existing Methods

As a starting point, we have tested repetition and some other simple error concealment schemes, employing MP3 or AAC compressed music bitsreams. The quality of simple repetition has been used as a reference. During our investigations we have discovered a fundamental limitation of existing methods, which is the assumption that the audio signals are short-term stationary. This assumption is not always

valid for music signals, especially for music with percussive sounds such as drums, hi-hats, etc. As a result, if the lost unit includes or is close to a short transient signal, such as a drumbeat in music, simple repetition will not be able to produce satisfactory results.

Possible problems of the simple repetition approach are illustrated with the help of Figure 15.

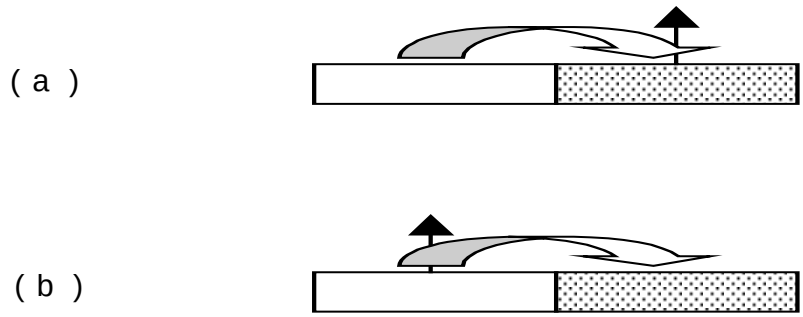

Figure 15. Illustrating possible problems with the packet repetition. Shaded rectangles represent corrupted packets. Light rectangles represent error-free ones. The thin arrows indicate the drumbeats and the thick arrows indicate packet repetition operations. (a) drumbeat eliminated, (b) double-drumbeat created.

1) If the drumbeat is replaced with other signals such as singing, the drumbeat is simply removed as in Figure 15 (a).
2) If the drumbeat is copied to its following packet, it may result in a subjectively very annoying distortion defined as a *double-drumbeat effect* by the author, as shown in Figure 15 (b). The degree of annoyance of the double-drumbeat effect depends on the time-frequency structure of the drumbeat. And it also depends on the distance between the original drumbeat and that generated due to packet repetition.

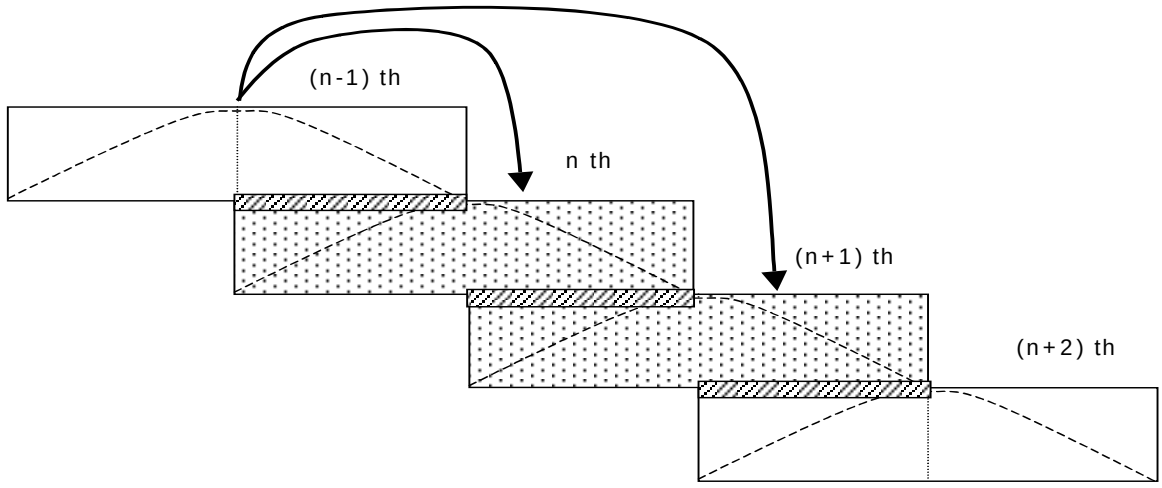

Figure 16. Illustration of a special problem with repetition scheme in the MDCT domain. Shaded rectangles represent corrupted data units. Blank rectangles represent error-free ones. Dashed lines indicate the window shape. The arrows indicate packet repetition operations. Heavily shaded rectangles represent the uncancelled alias. n is an integer number that represents the data unit index.

3) Due to the special property of MDCT discussed in section *2.3.2*, the repetition violates the TDAC conditions. Consequently, the alias distortions in the overlapped parts cannot cancel each other out (see Figure 16). However, the MDCT window functions enable a natural fade-in and fade-out in the overlap-add operation in the time domain. The uncancelled alias is normally not perceptible if the signal is stationary and the lost data unit is short enough.
4) Simple repetition does not consider the window switching commonly used in state-of-the-art audio codecs (see Figure 17). Therefore, it leads to a possible *window type mismatch phenomenon* [P7], which is illustrated with the help of Figure 18.

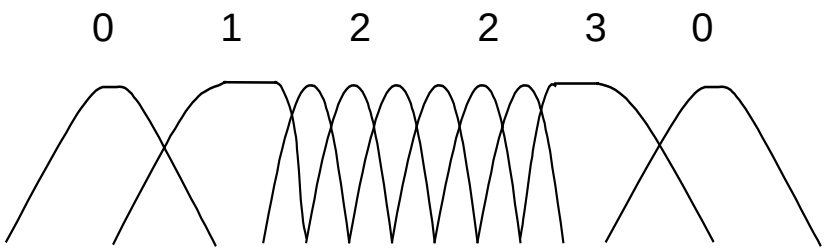

Figure 17. Illustration of the window switching pattern decoded from an MP3 audio bitstream. The four window types (long, long-to-short, short and short-to-long) are indexed with 0, 1, 2, 3 respectively.

As discussed in section *2.3.2*, window switching is an important concept to reduce pre-echo in an MDCT based audio codec such as MP3 and AAC. Both MP3 and AAC use four different window types: long, long-to-short, short and short-to-long which are indexed with 0, 1, 2, 3 respectively. The short window is introduced to tackle transient signals better. In the case of MP3, the long window length is 36 subband samples and the short window length is 12 subband samples. Therefore, the duration of a long window is three times that of a short window as shown in Figure 17. 50% window overlap is used with the MDCT. The data unit here is an MP3 granule. An MP3 frame consists of two granules where each granule consists of 576 frequency components.

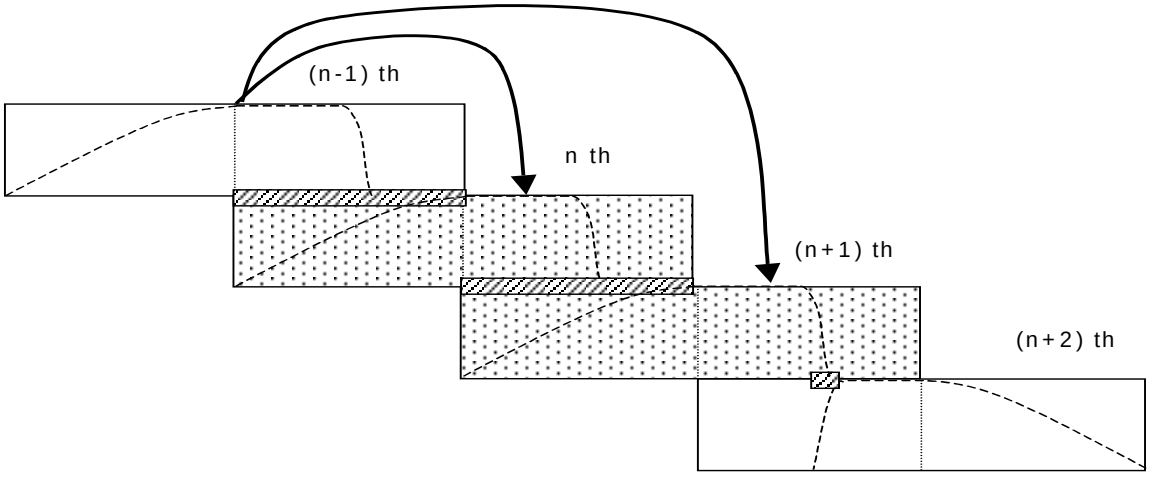

Figure 18. An example of the window type mismatch problem

If the two consecutive short window granules indexed as 22 in a window-switching sequence of 1223 are lost in a transmission channel, it is easy to deduce their window types from their neighboring granules. A

previous short window granule-pair indexed as 22 could be used to replace them to mitigate the subjective degradation. However, if we disregard the window-switching information available from the audio bitstream and perform simple repetition, it will result in window-switching patterns of 1113 (see Figure 18). In this case, not only are the TDAC conditions violated in the window overlapped areas, but also we will have some undesired energy fluctuation, since the squares of the two overlapping window functions do not add up to a constant [21]. This may create annoying artifacts. This phenomenon is defined by the author as *window type mismatch phenomenon.*

These problems cannot be solved with existing methods. To overcome these shortcomings, two drumbeat-pattern based error concealment schemes have been developed by the author, and are presented in the subsequent sections.

The underlining principle of the drumbeat-pattern based error concealment is fairly simple and straightforward: a segment around a drumbeat is subjectively more similar to a segment around a previous drumbeat than to its immediate neighboring segment. Therefore, an erroneous audio segment around beat (k+1) should be replaced with a corresponding segment from a previous beat as indicated by the thick arrow (see Figure 19). k is a positive integer determined by the employed level of beat information (e.g. quarter-note or half-note level). IBI stands for inter-beat interval. Rectangles filled with dots indicate corrupted MP3 granules. Blank rectangles indicate error-free ones.

To avoid the *window type mismatch phenomenon*, window types of the neighboring data units should be checked to ensure window type consistency after the replacement.

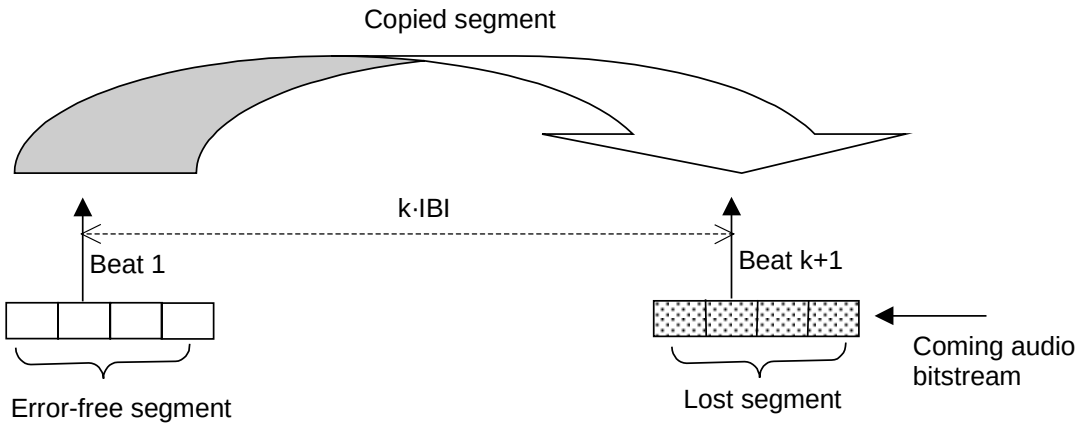

Figure 19. Concept of the drumbeat-pattern based error concealment method.

3.3.2 A Receiver-based Solution

A simplified block diagram of the system proposed in [P6] is illustrated in Figure 20. The principle of the scheme is that we store both the beat information and a segment (e.g. an IBI) of already decoded data units (history) in a buffer. If a burst packet loss occurs, we can use the buffered data to fill the gap sensibly in a musical sense.

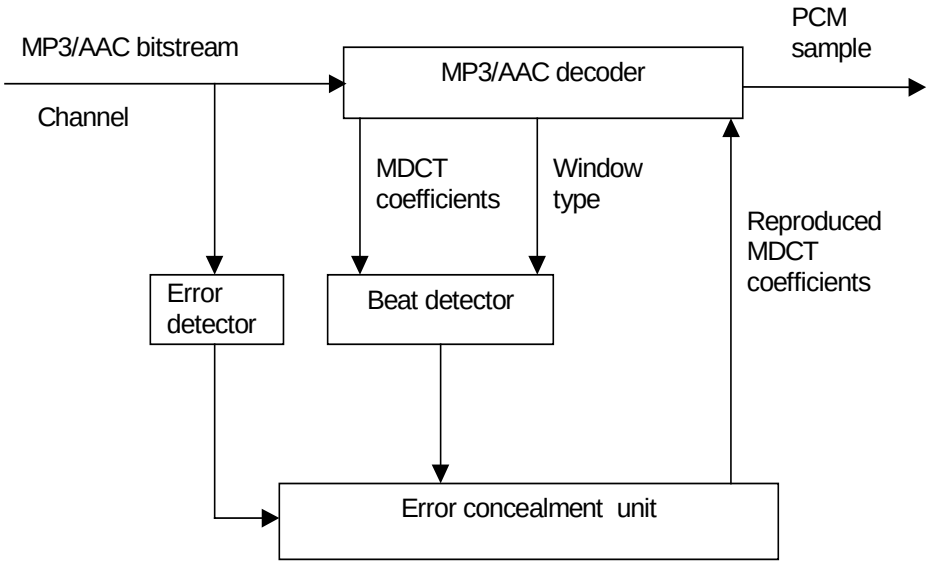

Figure 20. Block diagram of a receiver-based solution.

This scheme initially produced very promising results with music signals without singing, where regular drumbeats are the single dominant component with weak accompanying instruments, but rather disappointing results with more realistic situations, where drumbeat, singing and other musical instruments are combined. Further investigations have revealed some serious limitations of the receiver-based approach, especially if we focus on the single packet loss, which represents the majority loss in streaming applications.

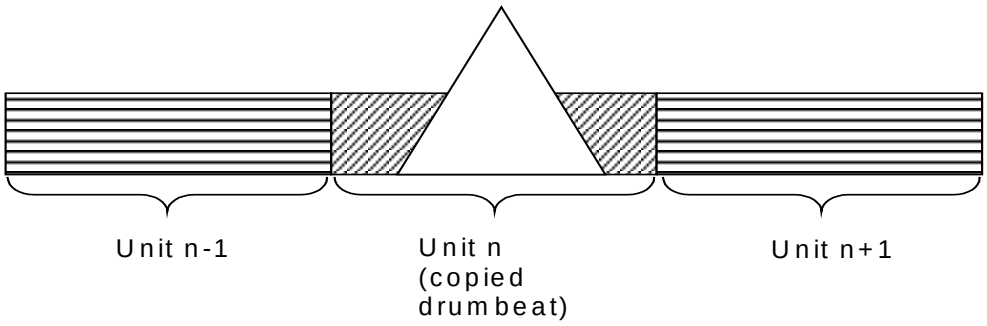

Figure 21. Illustration of the *spectral fine structure disruption effect.* The rectangle filled with upward diagonals represents the harmonic structure around the copied drumbeat. The triangle represents the drumbeat. The rectangles filled with horizontal lines represent the harmonic structures in the neighboring units around the missing unit.

The first problem is illustrated with the help of Figure 21. When a lost unit n is replaced by a previous drumbeat, it is likely to create a distortion, which is defined by the author as a *spectral fine structure disruption effect*. The spectral fine structure is particularly important for the harmonic and melodic parts of the music, such as singing. Although a typical drumbeat lasts about 100 ~ 200 ms, it is not reasonable to assume that a drumbeat in the entire duration of unit n is always loud enough to mask other signals such as singing. The masking effect depends on the relative strength of the drumbeat and the singing. Thus, if there is singing around the drumbeat, we can experience such discontinuity on the unit

boundaries with the receiver-based fullband approach proposed in [P6]. This problem becomes even more evident as we move from MP3 to AAC with the unit duration almost doubled from 576 to 1024 MDCT (modified discrete cosine transform) coefficients, which affect about 26 ms and 46 ms time domain samples respectively if the sampling frequency is 44.1 kHz.

Second, when a data unit containing a beat is lost, it will be difficult for the decoder to guess whether the lost data contains a beat. In the case of a single packet loss, we may rely on the window type information from the neighboring units to detect a beat on the lost data unit. However, the window switching patterns are not a reliable cue for beat detection. It should be noted that AAC in general has much less window switching than MP3, since more advanced codecs tend to use less window switching for improved coding efficiency. This is an inherent problem of the receiver-based approach.

Third, in comparison to repetition the double-drumbeat effect is reduced but not eliminated with our receiver-based method. Our recent investigations have shown that another inherent weakness has contributed to this problem. That is, only MDCT coefficients are readily available on the decoder side. This MDCT based feature is not very reliable due to a special property of MDCT – it does not preserve time domain energy. This will be further discussed in a subsequent section.

Fourth, the *window type mismatch* problem has not been tackled in the receiver-based scheme. In addition, its computational complexity is significantly higher than a simple packet repetition due to the beat detection on the decoder side.

In order to achieve better error concealment performance, while reducing the decoder complexity to a level similar to a simple packet repetition, a joint-sender-receiver-based approach is proposed.

3.3.3 A Joint-Sender-Receiver-based Solution

This method detects the drumbeat-pattern of music signals on the encoder side and embeds the beat information as ancillary data in a preceding data unit in the compressed bitstream. The beat information in this method consists of only the beat position. As a result, it significantly reduces the computational complexity of the decoder. The embedded beat information is then used to perform an error concealment task on the decoder side. The proposed method was implemented using an MPEG-4 AAC codec as shown in Figure 22.

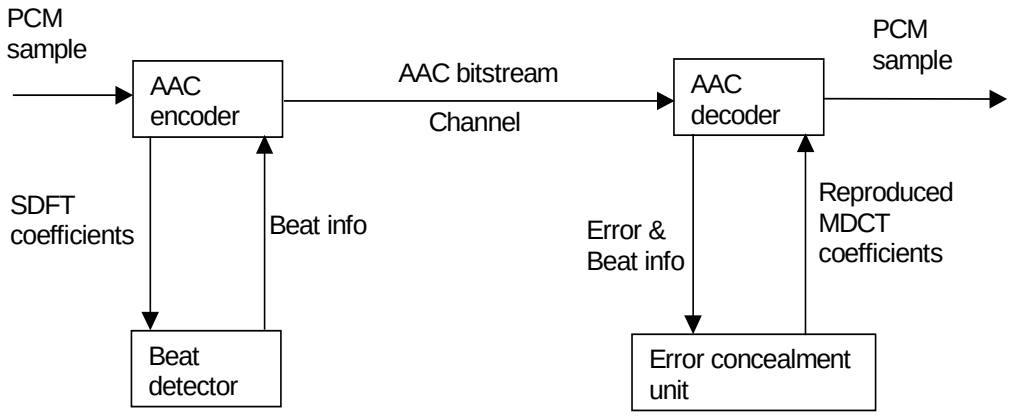

Figure 22. System overview.

The overall system comprises the blocks illustrated in Figure 22. An incoming musical signal in PCM format is fed to the AAC encoder. The AAC performs a frequency analysis in a form of shifted discrete Fourier transform (SDFT) [P4]. The beat detector uses SDFT based feature vector (FV) to detect beats and then embeds the beat information within the compressed bitstream as ancillary data in a preceding data unit. If the data unit of the beat is lost in the transmission channel, its position can still be determined by the beat information embedded in a previous data unit, since the probability of the simultaneous loss of two separate data units on the beats is low. The error concealment unit on the decoder side uses the embedded beat information and error information to reconstruct the lost MDCT coefficients. The reconstructed MDCT coefficients are then sent to the AAC decoder to produce the final PCM musical samples.

The beat detector used here is similar to that in [P7]. However, it has been improved in several ways. First, we have employed the SDFT coefficients alone as the input instead of both the window types and the MDCT coefficients. Second, we have used the subband energy slope (derivative) as FV.

The detected beat position is embedded in a previous data unit (AAC frame) for application in the decoder. This solution therefore implies an additional coding delay due to the necessity of a look-ahead. If we focus on single packet loss, only 1 bit is needed for each data unit of the audio stream to indicate whether the following data unit is on a drumbeat. If we want more protection of the beat position to tackle burst packet loss, the beat position can be embedded in two separate previous data units, which can be the preceding unit and a preceding drumbeat. In this case, some additional bits are needed to embed the position of drumbeat 3 as ancillary data in the frame at the position of drumbeat 1. Likewise, the position of drumbeat 4 is embedded in the frame at the position of drumbeat 2 as shown in Figure 23.

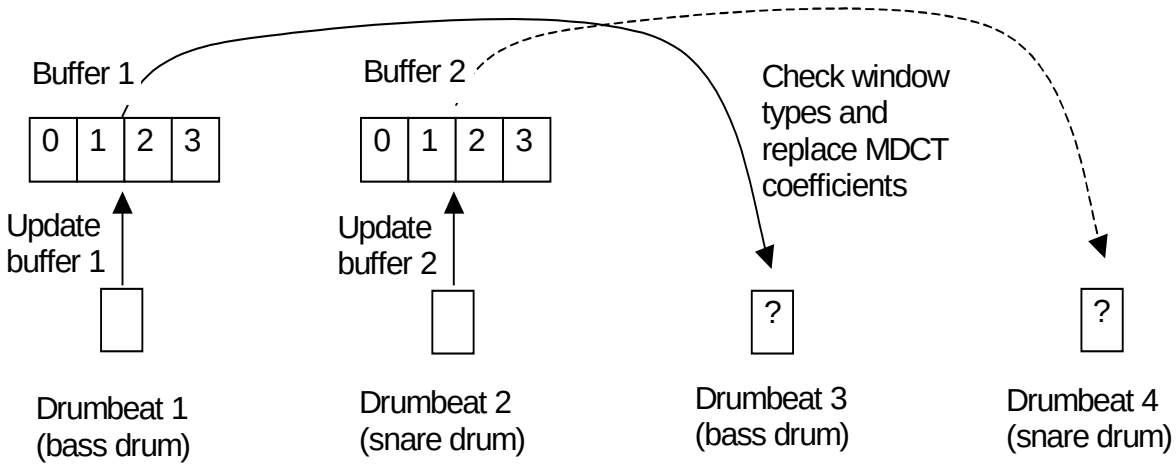

Figure 23. Illustration of the proposed error concealment operation based on drumbeat pattern. The numbers in the two drum buffers indicate the window types. The four window types (long, long-to-short, short and short-to-long) are indexed with 0, 1, 2, 3 respectively.

We assume a time signature of 4/4, which is common in most music signals such as pop, dance and rap music in streaming applications. According to their window types, the decoder saves the MDCT coefficients on the drumbeats in two drum-buffers for the bass drum and snare drum respectively (see Figure 23). The drum-buffers are updated if no error is detected on the drumbeat. When a packet loss is detected, the error concealment unit first checks the embedded beat information and the window types of the neighboring units. If the lost data unit (an AAC frame) is on the beat, it fetches the saved frame with a

correct window type from the corresponding buffer as shown in Figure 23. This effectively eliminates the *window type mismatch phenomenon* [P7].

In order to effectively reduce the *spectral fine structure disruption effect,* we have adopted a subband approach instead of the fullband approach in [P6]. The new subband approach is illustrated with the help of Figure 24.

The entire frequency band is divided into 3 parts. The frequency band between F1 and F2 represents the most relevant harmonic and melodic parts. The low and high frequency bands are more relevant for drumbeats. By copying the stochastic parts (drums) from a previous beat and copying the spectral fine structure from the neighboring data unit, we have achieved a very satisfactory overall subjective quality in the case of packet loss on the drumbeat.

F1 and F2 were about 344 Hz and 4500 Hz respectively. They were chosen empirically based on the spectrogram observation of the test signals and the constraints of the AAC standard. In the case of a long window, F1 corresponds to the 16^{th} MDCT coefficient, and F2 corresponds to the 208^{th} MDCT coefficient. In the case of the short window, F1 corresponds to the 2^{nd} MDCT coefficient, and F2 corresponds to the 26^{th} MDCT coefficient.

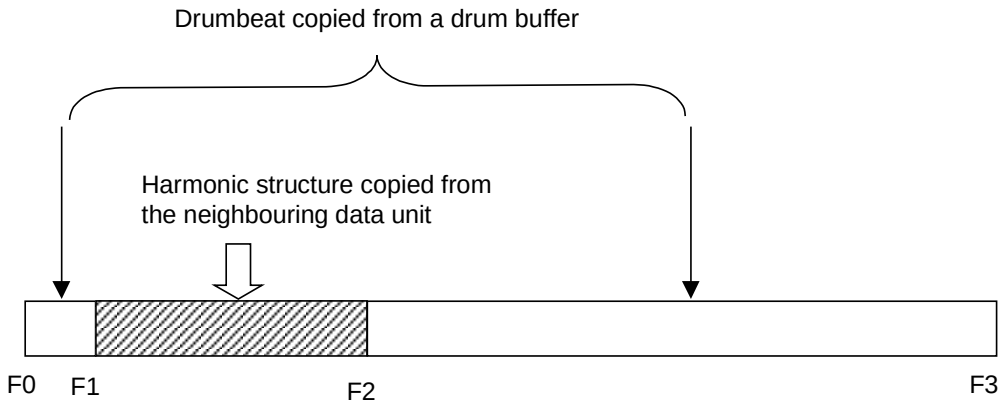

Figure 24. Illustration of the subband based error concealment. The rectangle filled with upward diagonals represents the fine spectral structure copied from a neighboring frame. The two blank rectangles represent the drum data copied from a drum buffer.

Informal evaluations performed by the author and several expert listeners have shown that in comparison with existing methods such as muting, simple repetition and frequency domain interpolation, the joint-sender-receiver based method was clearly preferred if the packet loss includes drumbeats. In comparison with the receiver-based method, this new method significantly reduced the *spectral fine structure disruption effect* and the *double-drumbeat effect.*

3.4 Compressed Domain Beat Detection

A block diagram of the compressed domain beat detector is shown in Figure 25. It uses the window type information and MDCT coefficients decoded from the bitstream to detect the beat in the compressed domain. The beat information is used for error concealment purposes.

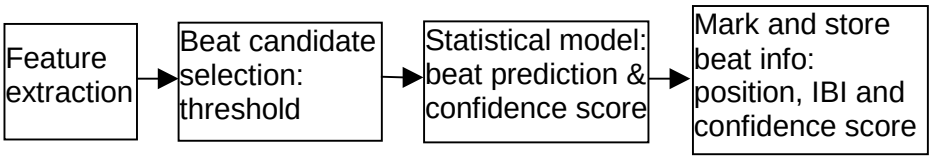

Figure 25. General structure of a beat detector

We have initially developed the compressed domain beat detector as an independent module. During the process of integrating the beat detector to the error concealment system, we have discovered a few problems with the method described in [P7].

As noted earlier, MDCT does not obey Parseval's theorem, *i.e.,* it does not preserve time domain energy [P4]. This compromises the FV quality in two ways. First, the MDCT based FV fluctuates excessively over time (see the dashed line in Figure 26(b)). This makes it difficult to set a proper threshold for selecting beat candidates. Second, the maximum positions of the MDCT based FV over time jitter around the real beats by about one AAC frame, while the SDFT based FV is far more stable and consistent with the position of the real beat (see Figure 26(b)).

In our implementation, the window switching mechanism is based on the SDFT domain information. Switching beat information from two different domains will compromise its time resolution. This was the rationale for us to use the SDFT based FV alone in the joint-sender-receiver based approach. As a result of the improved time resolution, the double-drumbeat effect is notably reduced.

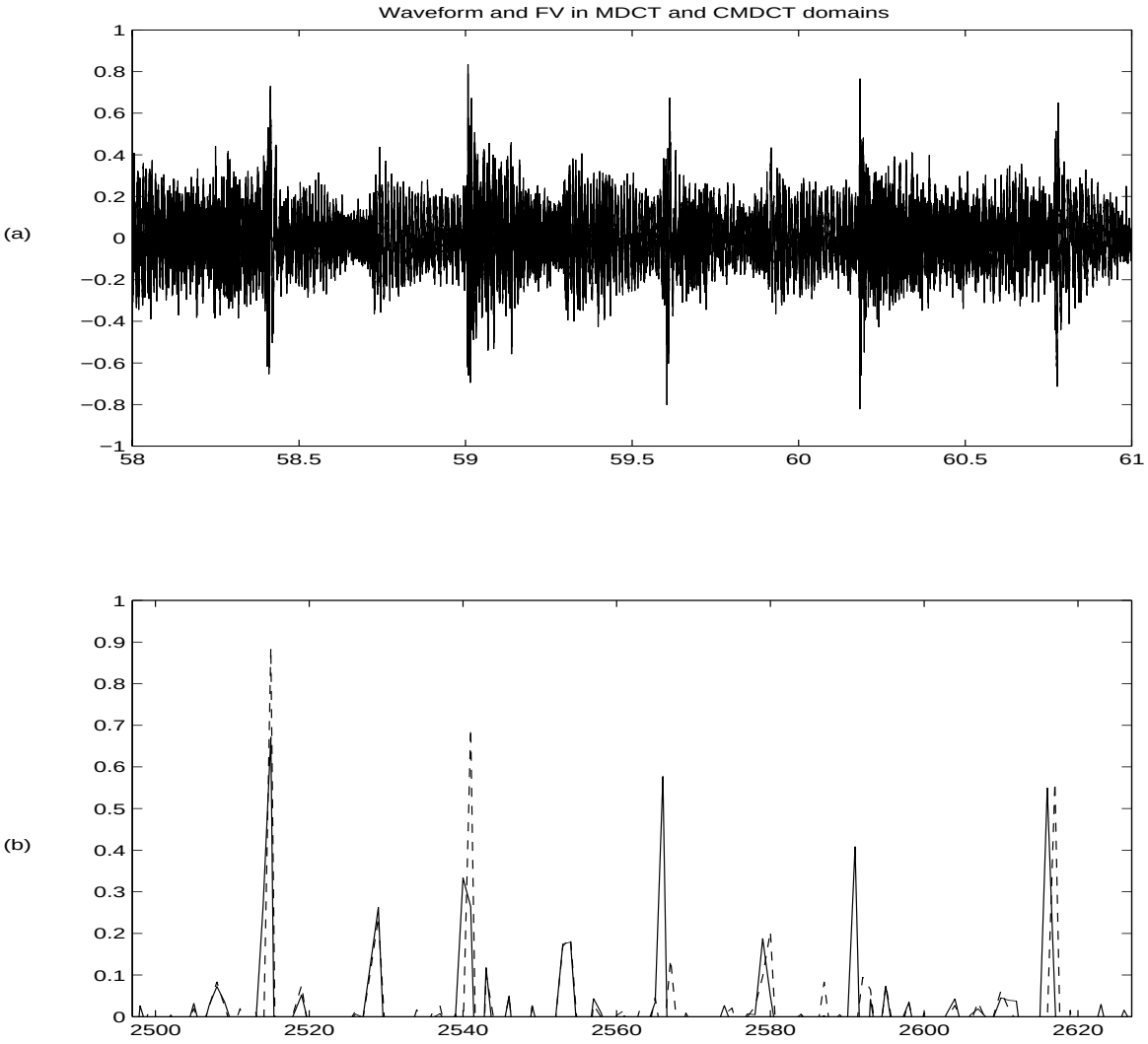

Figure 26. Music waveform and its corresponding AAC FV. (a) Music waveform versus time in seconds, (b) FVs versus AAC frame index. FVs in MDCT domain (dashed) and SDFT domain (solid).

3.5 Re-Compression of Compressed Audio

MP3 is a very popular audio format in both Internet and consumer products. But its coding performance is rather modest in comparison to state-of-the-art coding technologies. The motivation for re-compressing MP3 files losslessly is that it will be able to perfectly reconstruct the original MP3 files on the decoder side. This dissertation presents two lossless schemes to exploit the redundancy of the MP3 parameters such as scale-factors and MDCT coefficients in [P8]. The principle is very simple and straightforward; after a subtraction in the spectral domain, the residual will be small if there is redundancy in the MP3 main data. Since the performance improvement is so marginal, these schemes are not discussed here in detail.

The self-similarity-coding scheme represents a small scientific adventure, which is designed to exploit the similarity between consecutive drum patterns as illustrated in Figure 27 [P8].

As discussed in the drumbeat-pattern based error concealment method, music, especially pop music, is highly repetitive perceptually. What we have attempted in [P8] is to take two consecutive MP3 granules around all drumbeats, and then classify them into two classes – strong beats and weak beats, according to a fixed energy threshold. Generally, what we need to transmit is only four granule MDCT data to represent the drumbeats (two granule data for each class) in the whole piece of test music signal and one bit index merely to indicate whether this drumbeat is a strong or weak beat. To our surprise, the reproduced music is not so terribly bad, although it sounds a bit monotonous.

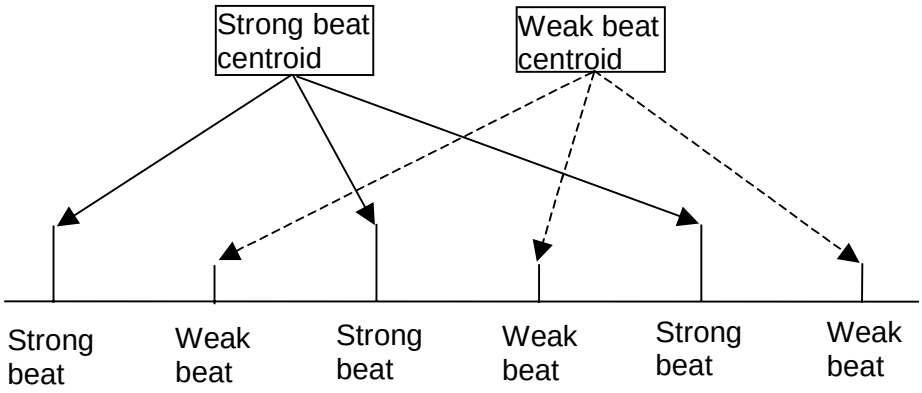

Figure 27. Concept of self-similarity based coding method

Chapter 4 Summary of publications

This chapter summarizes the published work incorporated in this dissertation and describes the contribution of the author. The publications are clustered into three modules. The first module consisting of P1 and P2 describes two new psychoacoustic models with applications in audio coding. The second module consists of P3, P4 and P5 and provides some new insights into a few relevant transforms and their impact on audio coding. The third module, which consists of P6, P7 and P8, focuses on compressed domain audio processing for the purpose of improving coding efficiency and of error concealment in mobile terminals.

4.1 Overview of Individual Publications

4.1.1 Publication 1

In order to improve audio coding performance, excess masking has been employed for the compression of complex audio signals. A new algorithm has been developed to classify and pre-process maskers. A ERB scale based psychoacoustic model is used to estimate the simultaneous masking threshold. This masking threshold is used for quantizing audio signal coefficients in the frequency domain. Preliminary test results show improved coding efficiency.

4.1.2 Publication 2

This paper describes an excitation level based psychoacoustic model to estimate the simultaneous masking threshold for audio coding. The system has the following stages: 1) a windowing function; 2) a time-to-frequency transformation; 3) an excitation level calculation block similar to that in Moore and Glasberg's loudness model; 4) a correction factor for estimating masking threshold; 5) the inclusion of the absolute masking threshold; 6) the output Signal-to-Masking ratio. We have evaluated the performance by integrating the proposed psychoacoustic model into an audio coder similar to MPEG-2 AAC, which contains only the basic coding tools. Our model performs better than, or as well as, the psychoacoustic model suggested in the MPEG-2 AAC audio coding standard for all the test signals. We can achieve almost transparent quality with bitrates below 64 kbps for most of the monophonic critical test signals. Significant improvements have been achieved with speech signals, which are always difficult for transform audio coders.

4.1.3 Publication 3

This paper presents an experimental study addressing spectrum estimation using various types of transforms and time-domain windows. The study concentrates in transform energy compaction properties. Transforms studied are DFT, DCT, SDFT, MDCT and discrete sine transform (DST). Transform energy compaction property is measured experimentally, and the influence of varying window size (256, 512 and 1024 samples) and window shape (rectangular and sine) is investigated. In addition to sinusoidal signals, classical and pop music excerpts were used as test material.

The results can partially be explained by the fact that the main lobe width of a rectangular window frequency response is $4\pi / N$ [7], while the main lobe width of the sine window is $6\pi / N$ [20]. This paper is an initial investigation, and further work is needed to more precisely establish factors influencing practical energy compaction properties of different transforms.

4.1.4 Publication 4

Most state-of-the-art audio encoders have two basic coding tools: an MDCT and DFT based psychoacoustic model. The MDCT coefficients are quantized according to the masking threshold calculated by the psychoacoustic model. However, this kind of encoder structure can fail for some test signals. Research has been undertaken to find the reasons behind this failure, during which it has been found that the failure may be caused by the peculiar properties of MDCT and the mismatch between MDCT and a DFT based psychoacoustic model. We have established a direct and compact formulation of the MDCT with the help of a SDFT. This formulation has a clear physical interpretation. It enables us i) to clarify the symmetric properties of MDCT, ii) to illustrate the concept of the time domain alias cancellation in a very intuitive and illustrative way, and iii) to show some peculiar properties of MDCT which may affect the coding performance of an MDCT based audio codec. Based on these new interconnections we propose a new encoder structure as a first step towards solving the mismatch. A small, formal listening test was initiated to verify the relative performance of our optimized codec. The mismatch between the two basic coding tools is relevant for multimedia codec design, watermark embedding, etc. The improvement in computational efficiency discussed in this paper is essential for hand-held devices such as mobile phones.

4.1.5 Publication 5

This paper presents a novel lossless multichannel audio-coding algorithm to remove inter-channel redundancy. We employ an integer-to-integer discrete cosine transform (INT-DCT) to perform inter-channel decorrelation after quantization of modified discrete cosine transform (MDCT) coefficients of individual channels. When compared with a Karhunen-Loeve transform (KLT) based approach our new method has three major advantages: 1) it avoids quantization noise spreading to other channels; 2) computational simplicity; 3) it uses less overhead information (a quantized covariance matrix or eigenvector is avoided in our algorithm), while maintaining a similar decorrelation capability.

4.1.6 Publication 6

Error concealment is an important method to mitigate the degradation of the audio quality when compressed audio packets are lost in error prone channels, such as Mobile Internet and digital audio broadcasting. This paper presents a novel error concealment scheme exploiting the beat and rhythmic pattern of music signals. Preliminary simulations show significantly improved subjective sound quality in comparison to conventional methods in the case of burst packet losses. The new scheme is proposed as a

complement to prior solutions. It can be adopted to essentially all existing perceptual audio decoders such as an MP3 decoder for streaming music.

4.1.7 Publication 7

This paper presents a novel beat detector that processes MPEG-1 Layer 3 (known as MP3) encoded audio bitstreams directly in the compressed domain. Most previous beat detection or tracking systems dealing with signals in the formats of musical instrument digital interface (MIDI) or PCM are not directly applicable to compressed audio bitstreams, such as MP3 bitstreams. We have developed the beat detector as a part of a beat-pattern based error concealment scheme for streaming music over error prone channels. Special effort was used to obtain a tailored trade-off between performance, complexity and memory consumption for this specific application. A comparison between the machine-detected results to human annotation has shown that the proposed method correctly tracked beats in 4 out of 6 popular music test signals. The results were analyzed.

4.1.8 Publication 8

This paper presents three schemes for re-compressing MP3 (MPEG-1 Layer 3) audio bitstreams. The first two schemes are lossless and exploit the inter-frame redundancies of the main data (the scale-factors and the quantized MDCT coefficients) of the MP3 bitstream. The third scheme is a lossy approach exploiting the redundancies between consecutive beat-patterns. The aim is to study the potential of the new coding schemes. Preliminary results are reported in this paper.

4.2 Author's Contribution to the Publications

The work in the first module was accomplished at Nokia Research Center with support from Prof. Brian Moore, Cambridge University. The transform-related research in the second module is the result of fruitful collaboration with Prof. Leonid Yaroslavsky, Tel Aviv University. The compressed domain audio processing related work was conducted at Nokia Research Center in Finland and Cambridge University in UK. Miikka Vilermo's creative implementation of the proposed psychoacoustic models and the multichannel audio coding algorithm proved very successful. Many of his suggestions have contributed to improving the proposed algorithms. Juha Ojanperä's fast implementation of the MP3 bitstream re-compression schemes has been of great value in this and later studies. David Isherwood organized the subjective listening tests for [P4]. However, the author's contribution to all publications has been essential. Moreover, in each case the present author prepared the manuscripts for publication and performed a major part of research, simulations and experiments.

In particular, the author's contribution to the publications is as follows: In Publication 1, the author initiated the idea of exploiting the excess masking effect in the frequency domain for improving the coding efficiency. The implementation and subsequent evaluations were performed jointly by both authors.

In Publication 2, the author proposed the concept of employing an ERB based perceptual model for audio coding. The implementation and integration of the model into an audio encoder similar to the MPEG-2 AAC were performed jointly by the authors. This proved experimentally that an ERB based perceptual model can be used in audio coding applications with good results especially with speech signals.

In Publication 3, the author proposed a thorough study of the energy compaction property of the MDCT in comparison with other transforms and performed almost all the experiments. The study showed that MDCT could be used as a frequency analyzer in general, although it does not fulfil Parseval's theorem.

In Publication 4, the author initiated the study for reducing the computational complexity of an MDCT-based audio encoder. The relationship between MDCT, SDFT and DFT was derived by the author in cooperation with Prof. Leonid Yaroslavsky, which has led to a direct and compact formulation of the MDCT with the help of a SDFT. This new formulation has a clear physical interpretation and provides some new insights into MDCT. The subjective listening test was organized by David Isherwood.

In Publication 5, the author proposed a cascaded MDCT-DCT multichannel perceptual audio coding scheme. In particular, the author initiated the idea of using an Integer-to-Integer DCT to remove the inter-channel correlation. The implementation was mainly programmed by Miikka Vilermo.

In Publication 6, the author proposed a novel beat-pattern based error concealment scheme for streaming music over error prone channels and performed all the simulations. This algorithm significantly improved the subjective sound quality in the case of burst packet loss in comparison with existing methods. The author was the sole writer of this paper.

In Publication 7, the author proposed a novel compressed domain beat detector as a part of the error concealment scheme described in Publication 6 using MP3 audio bitstream. The implementation was a joint effort by both authors.

In Publication 8, the author proposed and designed three new schemes for re-compressing MP3 audio bitstreams and performed almost all the experiments. The implementation was mainly programmed by Juha Ojanperä.

Chapter 5 Conclusion

Non-real-time downloading of compressed audio such as MP3 has boosted the application of audio coding technologies, not only in the Internet world but also in consumer products. Near real-time applications such as streaming audio have become a reality in the Internet and are penetrating mobile networks. Delivery of high quality multimedia content such as music via future Mobile Internet will become an important form of mobile service. Nevertheless, it is important to realize that many technical problems to make this vision a reality are still unsolved. Several technical obstacles need to be overcome before we can deliver mono/stereo, even surround sound music with sufficient quality of service.

Current audio coding technologies are not optimized for delivery of audio content in Mobile Internet environment. The audio quality of low bitrate streams (*e.g.* 14.4 - 64 kbps) must be significantly improved. New coding methodologies such as new transforms, new perceptual models, and fully embedded and scalable bitstream formatting are needed to achieve the compression objective.

To combat errors on the channel, both sender and receiver-based methods must be deployed in an optimal way in the sense of performance/complexity. Interleaving has been shown to be an effective tool in applications such as VoIP. Its effectiveness should be tested in high quality music streaming. Error detection/correction is generally necessary. In the context of audio streaming, a media-specific FEC is recommended. Some error resilience tools are already available in MPEG-4 for example, and can be deployed in a wireless audio content delivery system. However, it should be noted that MPEG-4 error resilience tools are mostly designed for a circuit switched network with random bit errors, and therefore are not necessarily effective for packet losses.

Error concealment, as the last resort, plays an important role in mitigating the degradation of subjective audio quality in the case of packet loss. In order to keep the decoder structure simple, a simple and effective error concealment method such as packet replacement, is recommended.

In this dissertation, three elementary technologies have been addressed. The first module is the development of two improved perceptual models for audio coding. The second module is the MDCT based audio codec optimization. The third module focuses on the compressed domain audio processing.

Current and future work will be system level integration. That is, selecting/developing the right elementary technologies to build a system to meet the challenge – deliver high quality audio and multimedia content to mobile terminals, which will further enrich people’s lives.

References

[1] Haavisto, P., Castagno, R., Honko, H., "Multimedia Standardization for 3G Systems", Proc. of 16th IFIP World Computer Congress (WCC2000)/5th International Conference on Signal Processing, Beijing, August, 2000, pp. 32-39

[2] Plenge, G., "DAB – A new Sound Broadcasting System, Status of the development – Routes to its Introduction", EBU Review – Technical, No.246, April 1991

[3] ISO/IEC 11172-3 International Standard, "Information Technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s", 1993

[4] Shlien, S., "Guide to MPEG-1 Audio Standard", IEEE Transactions on Broadcasting, Vol. 40, No.4, December 1994

[5] Theile, G., Stoll, G., Link, M., "Low bit-rate coding of high-quality audio signals – An introduction to the MASCAM system", EBU Review – Technical, No. 230, August 1988

[6] Jayant, N., Johnston, J., Safranek, R., "Signal Compression Based on Models of Human Perception", Proceedings of the IEEE, Vol.81, No.10, October 1993, pp. 1385-1422

[7] Jayant, N., Noll, P., "Digital Coding of Waveforms – Principles and Applications to Speech and Video", Prentice-Hall, Englewood Cliffs, NJ, 1984

[8] Painter, T., Spanias, A., "Perceptual Coding of Digital Audio," Proceedings of the IEEE, Vol.88, No.4, April 2000, pp. 451-513

[9] ISO/IEC 13818-3, "Information Technology – Generic Coding of Moving Pictures and Associated Audio, Part 3: Audio", 1997

[10] Fielder, L.D., Bosi, M., Davidson, G.A., Davis, M., Todd, C., Vernon, S., "AC-2 and AC-3: Low Complexity Transform-Based Audio Coding," in Geilchrist, N. and Grewin, C. (ed.), Collected Papers on Digital Audio Bit-Rate Reduction, AES, 1996, pp. 54-72

[11] Herre, J., Grill, B., "Overview of MPEG-4 Audio and its Applications in Mobile Communications", Proc. of 16th IFIP World Computer Congress (WCC2000)/5th International Conference on Signal Processing, Beijing, August, 2000, pp. 11-20

[12] ISO/IEC 13818-7, "Information Technology – Generic Coding of Moving Pictures and Associated Audio, Part 7: Advanced Audio Coding", 1997

[13] Bosi, M., Brandenburg, K., Quackenbush, S., Fielder, L., Akagiri, K., Fuchs, H., Dietz, M., Herre, J., Davidson, G., Oikawa, Y. "ISO/IEC MPEG-2 Advanced Audio Coding", J. Audio Eng. Soc., Vol. 45, No. 10, October 1997

[14] ISO/IEC 14496-3, "Coding of Audio-Visual Objects: Audio", 1999

[15] Johnston, J.D., Quackenbush, S.R., Herre, J., Grill, B., "Review of MPEG-4 General Audio Coding," in Puri, A., Chen, T. (ed), Multimedia Systems, Standards, and Networks, Marcel Dekker, Inc. New York, USA, 2000, pp. 131-155

[16] Rothweiler, J. H., "Polyphase Quadrature Filters – A New Subband Coding Technique", Proc. ICASSP 1983

[17] Princen, J. P., Bradley, A. B., “Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation,” IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No. 5, October 1986.

[18] Princen, J. P., Johnson, A. W., Bradley, A. B., “Subband/Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation,” IEEE International Conference on Acoustics, Speech, and Signal Processing, 1987, Dallas, USA, pp. 2161-2164

[19] Malvar, H., “Signal Processing with Lapped Transforms,” Artech House, Inc., 1992

[20] Ferreira, A., “Spectral Coding and Post-Processing of High Quality Audio,” Ph.D. thesis http://telecom.inescn.pt/doc/phd_en.html, 1998

[21] Edler, B., “Coding of Audio Signals with Overlapping Block Transform and Adaptive Window Functions,” (in German), *Frequenz*, vol.43, pp.252-256, 1989

[22] Edler, B., “Äquivalenz von Transformation und Teilbandzerlegung in der Quellencodierung,” Ph.D. thesis, Universität Hannover, 1995

[23] Yaroslavsky, L., Eden, M., “Fundamentals of Digital Optics,” Birkhauser, Boston, 1996

[24] Roads. C., “The Computer Music Tutorial,” The MIT Press, Cambridge, Massachusetts, 1998

[25] Scharf, B., “Complex Sounds and Critical Bands,” Psychological Bulletin 58, 1961, pp. 205-217

[26] Scharf, B., “Critical Bands,” In J. Tobias (ed.), Foundations of Modern Auditory Theory, Academic Press, Orlando, 1970

[27] Fletcher, H., “Auditory Patterns”, Rev. Mod. Phys., vol. 12, 1940, pp. 47-65

[28] Moore, B. C. J., Shailer, M. J., Hall, J. W., and Schooneveldt, G. P., “Comodulation Masking Release in Subjects with Unilateral and Bilateral Cochlea Hearing Impairment,” J. Acoust. Soc. Am., vol. 93, 1993, pp. 435-451

[29] Moore, B. C. J., “Masking in the Human Auditory System”, in Geilchrist, N. and Grewin, C. (ed.), Collected Papers on Digital Audio Bit-Rate Reduction, AES, 1996, pp. 9-19

[30] Zwicker, E., Zwicker, U. T., “Audio Engineering and Psychoacoustics: Matching Signals to the Final Receiver, the Human Auditory System”, J. Audio Eng. Soc., Vol.39, No.3, March 1991

[31] Zwicker, E., Fastl, H., “Psychoacoustics, Facts and Models”, Springer-Verlag, Berlin Heidelberg, Germany, 1990

[32] Zwicker, E., Feldtkeller, R., “Das Ohr als Nachrichtenempfänger,” Hirzel, Stuttgart, Germany, 1967

[33] Zwicker, E., “Procedure for Calculating Loudness of Temporally Variable Sounds,” J. Acoust. Soc. Am., vol.62, 1977, pp. 675-682

[34] Paulus, E., Zwicker, E., “Programme zur automatischen Bestimmung der Lautheit aus Terzpegeln oder Frequenzgrouppenpegeln,” ACOUSTICA, vol. 27, Heft 5, 1972, pp. 253

[35] Moore, B. C. J., Glasberg, B. R., Baer, T., “A Model for the Prediction of Thresholds, Loudness, and Partial Loudness”, J. Audio Eng. Soc., Vol. 45, No. 4, 1997

[36] Colomes, C., Schmidmer, C., Thiede, T., Treurniet, W. C., “Perceptual Quality Assessment for Digital Audio: PEAQ – The New ITU Standard for Objective Measurement of the Perceived Audio Quality”, Proc. of the AES 17th International Conference, Florence, Italy, September 1999

[37] Wiese, D., Stoll, G., “Bitrate Reduction of High Quality Audio Signals by Modeling the Ears Masking Thresholds”, 89th AES Convention, Los Angeles, CA, September 1990

[38] Braundenburg, K., Stoll, G., “ISO-MPEG-1 Audio: A Generic Standard for Coding of High-Quality Digital Audio”, J. Audio Eng. Soc., vol.42, October 1994

[39] Brandenburg, K., Johnston, J. D., “Second Generation Perceptual Coding: The Hybrid Coder,” 88th AES Convention, May 1990

[40] Johnston, J. D., “Transform Coding of Audio Signals Using Perceptual Noise Criteria”, IEEE J. Selected Areas in Communications, vol. 6, 1988, pp. 314-323

[41] Van der Heijden M., Kohlrausch A., “Using an Excitation-pattern Model to Predict Auditory Masking”, Hearing Research 80, 1994, pp. 38-52.

[42] Espinoza-Varas B., Cherukuri S. V., “Evaluating a model of auditory masking for applications in audio coding”, IEEE ASSP Workshop on Application of Signal Processing to Audio & Acoustics. New Paltz, New York, 1995.

[43] Green D. M., “Additivity of Masking”, J. Acoust. Soc. Am., 41, 1967, pp. 1517-1525.

[44] Lutfi, R. A., “Additivity of simultaneous masking”, J. Acoust. Soc. Am., 73, 1983, pp. 262-267.

[45] Humes, L. E. and Jesteadt, W., “Models of the additivity of masking”, J. Acoust. Soc. Am., 85, 1989, pp. 1285-1294.

[46] Blauert, J., “Spatial Hearing”, MIT press, Cambridge, MA, USA, 1983.

[47] Moore B. C. J., “An Introduction to the Psychology of Hearing”, 4. Edition, Academic Press, London, 1997.

[48] Espinoza-Varas, B., Cherukuri, S. V., “Evaluating a model of auditory masking for applications in audio coding”, proc. IEEE ASSP Workshop on Application of Signal Processing to Audio & Acoustics. New Paltz, New York, 1995

[49] Beerends, J. G., Stemerdink, J. A., “A Perceptual Audio Quality Measure Based on a Psychoacoustic Sound Representation”, J. Audio Eng. Soc., Vol. 40, No. 12, 1992

[50] Baumgarte, F., “A Physiological Ear Model for Auditory Masking Applicable to Perceptual Coding”, 103rd AES Convention, New York, NY, September 1997

[51] Brandenburg, K., “MP3 and AAC explained”, Proc. of the AES 17th International Conference, Florence, Italy, September 1999

[52] Herre, J., Johnston, J.D., “Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS),” 101st AES Convention, Los Angeles 1996, preprint 4384

[53] Herre, J., Johnston, J.D., “Exploiting Both Time and Frequency Structure in a System that Uses an Analysis/Synthesis Filterbank with High Frequency Resolution,” 103rd AES Convention, New York, 1997, preprint 4519

[54] Herre, J., Schulz, D., “Extending the MPEG-4 AAC Codec by Perceptual Noise Substitution,” 104th AES Convention, Amsterdam 1998, Preprint 4720

[55] Ojanperä, J., Väänänen, M., "Long Term Predictor for Transform Domain Perceptual Audio Coding," 107th AES Convention, New York 1999, Preprint 5036

[56] Chen, C.W., Cosman, P., Kingsbury, N., Liang, J., Modestino, J.W., "Guest Editorial Error-Resilient Image and Video Transmission," IEEE Journal on Selected Areas in Communications, June 2000

[57] Wong, K.H.H., Hanzo, L., "Channel Coding," in Raymond Steele (ed.), Mobile Radio Communications, Pentech Press, London, 1992, pp. 348-357

[58] Steele, R., "Mobile Radio Communications," Pentech Press, London, 1992

[59] Correia, L., Prasad, R., "An Overview of Wireless Broadband Communications," IEEE Communication Magazine, January 1997, pp. 28-33

[60] Radha, H., Ngo, C.Y., Sato, T., Balakrishnan, M., "Multimedia Over Wireless," in Atul Puri and Tsuhan Chen (ed.), Multimedia Systems, Standards, and Networks, Marcel Dekker, Inc., New York, 2000

[61] Perkins, C., Hodson, O., Hardman, V., "A Survey of Packet-loss Recovery Techniques for Streaming Audio," IEEE Network, Sept/Oct 1998.

[62] Perkins, C., Hodson, O., "Options for Repair of Streaming Media", http://www.ietf.org/rfc/rfc2354.txt under www.ietf.org: RFC 2354.

[63] Carle, G., Biersack, E.W., "Survey of Error Recovery Techniques for IP-Based Audio-Visual Multicast Applications," IEEE Network, Nov/Dec 1997.

[64] Bolot, J.C., Vega-Garcia, A., "Control Mechanisms for Packet Audio in the Internet," Proc. IEEE INFOCOM'96, 1996.

[65] Yajnik, M., Kurose, J., Towsley, D., "Packet Loss Correlation in the Mbone Multicast Network," Proc. IEEE Global Internet Conference, Nov. 1996.

[66] Ramsey, J.L., "Realization of Optimum Interleavers," IEEE Transactions on Information Theory, May 1970, IT-16, pp. 338-345.

[67] Gruber, J.G., Strawczynski, L., "Subjective Effects of Variable Delay and Clipping in Dynamically Managed Voice Systems," IEEE Trans. Commun., vol. COM-33, no. 8, Aug. 1985, pp. 801-808.

[68] Jayant, N.S., Christenssen, S.W., "Effects of Packet Losses in Waveform Coded Speech and Improvements due to an Odd-Even Sample Interpolation Procedure," IEEE Trans. Commun., vol. COM-29, no. 2, Feb. 1981, pp. 101-109.

[69] Hartman, V. et al., "Reliable Audio for Use over the Internet," Proc. INET'95, 1995

[70] Miller, G.A., Licklider, J.C.R, "The Intelligibility of Interrupt Speech," J. Acoust. Soc. Amer., vol. 22, no. 2, 1950, pp. 167-173

[71] Warren, R.M., "Auditory Perception," Pergamon Press, 1982.

[72] ETSI Rec. GSM 6.11, "Substitution and Muting of Lost Frames for Full Rate Speech Signals," 1992.

[73] Goodman, O.J. *et al*., "Waveform Substitution Techniques for Recovering Missing Speech Segments in Packet Voice Communications," IEEE Trans. Acoustics, Speech, and Sig. Processing, vol. ASSP-34, no. 6, Dec. 1986, pp. 1440-1448.

[74] Wasem, O.J. *et al*., "The Effect of Waveform Substitution on the Quality of PCM Packet Communications," IEEE Trans. Acoustics, Speech, and Sig. Processing, vol. 36, no. 3, Mar. 1988, pp. 342-348.

[75] Sanneck, H. *et al*., "A New Technique for Audio Packet Loss Concealment," IEEE Global Internet 1996, Dec. 1996, pp. 48-52.

[76] Chen, Y.L., Chen, B.S., "Model-based Multirate Representation of Speech Signals and its Application to Recovery of Missing Speech Packets," IEEE Trans. Speech and Audio Processing, vol. 15, no. 3, May 1997, pp. 220-231.

[77] ITU Rec. G723.1, "Dual Rate Speech Coder for Multimedia Communications transmitting at 5.3 and 6.3 kbit/s," Mar. 1996.

Errata

- In ([P2], p.402) the sentence “which are similar to traditional critical bands at low frequencies (see below).” should read “which are similar to traditional critical bands except at low frequencies (see below)”.
- In ([P5], p.4) the sentence “Givens rotations are factorized into 3 matrices each, resulting the total of 15 matrix multiplications. However the internal structure of these matrices guarantees that only 15 multiplications and 15 rounding operations are needed in total.” should read “Givens rotations are factorized into 3 matrices each, resulting the total of 30 matrix multiplications. However the internal structure of these matrices guarantees that only 30 multiplications and 30 rounding operations are needed in total.”
- In ([P6], p.74) the sentence “In the case of packet-based network, the time stamp of the packet is a reliable cue for missing packets.” should read “In the case of packet-based network, the sequence number of the packet is a reliable cue for missing packets.”

Publications

[P1] Wang, Y., Vilermo, M. “Exploiting Excess Masking for Audio Compression”, AES 17th International Conference on High Quality Audio Coding, September 2 – 5, 1999, Florence, Italy, pp. 216-219

EXPLOITING EXCESS MASKING FOR AUDIO COMPRESSION

YE WANG AND MIIKKA VILERMO

Nokia Research Center
Speech and Audio Systems Lab.
Tampere, Finland

ye.wang@research.nokia.com
miikka.vilermo@research.nokia.com

In order to improve audio coding performance, excess masking has been employed for the compression of complex audio signals. A new algorithm is developed to classify and pre-process maskers. A psychoacoustic model is used to estimate simultaneous masking threshold. This masking threshold is used for quantizing audio signal coefficients in the frequency domain. Preliminary test results show improved coding efficiency.

INTRODUCTION

Auditory masking plays a major role in audio coding, because all coding algorithms engender a certain level of undesirable low-level quantization noise that occurs simultaneously with the desired coded signal. All perceptual audio encoders have a psychoacoustic model, which calculates the masking threshold to determine the maximum allowable noise injection level without audible distortion. These models simulate masking effects from psychoacoustic studies. There is a major challenge however: Only simple stimuli such as sinusoids and bands of noise have been used in most psychoacoustical studies. In audio coding we are dealing with real life audio signals. That is, a multi-component complex masker (coded audio signal) must mask the spectrally complex target (quantization noise).

In our previous paper [1], we have applied an excitation-pattern model to estimate the simultaneous masking threshold for audio coding. This model performs fairly well for narrow-band-noise masking, but may overestimate the masking produced by tonal components [2][3]. We have introduced a weighting function, which includes the tonality measure to solve this problem [1].

On the other hand, the excitation-pattern model seems to underestimate the combined masking effects of multiple-component maskers [4][5]. More specifically, it underestimates the combined effects of two maskers both when the masker frequency components fall within the maskee auditory-filter bandwidth, and when they fall outside this bandwidth [5]. We hereby present some initial work that we have done to exploit the excess masking of two-tone maskers within the equivalent rectangular bandwidths (ERBs) [3] for audio compression.

Excess masking has been discussed in many publications since 1960's. In essence, the masking produced by the combination of simple maskers (sinusoids or bands of noise) is not a simple summation of the masking produced by the individual maskers. Several studies [6][7][8] have shown that the combined masking effect of two equally-effective simultaneous maskers is 3 to 15 dB greater than the masking predicted by the linear addition of masker energies. This "additional" amount of masking is defined as excess masking. Excess masking exists not only in frequency domain but also in time domain [9]. But the time domain excess masking will not be covered in this paper.

1. MODEL DESCRIPTION

The model includes the following stages: 1) time-to-frequency domain transformation, which is a FFT in our case; 2) masker classification and pre-processing, in which maskers are classified by their types and spectral structure; 3) masking threshold estimation, including excess masking as well as the absolute masking threshold as employed in the MPEG-2 AAC standard; 4) SMR (Signal-to-Masking Ratio) calculation, as the output of the model, used to control the quantizer in the audio encoder.

Because this is a modification of the model described in [1], the basic structure is essentially the same. Difference happens in stage 2 and 3. In stage 2, the algorithm searches for components that are subject to the following criteria: 1) The components must be local maxima; 2) They have to be tonal (predictable) i.e. the

unpredictability measure has to be under a certain threshold; 3) They have to be greater than 10.0 dB. Then the algorithm finds the first component that meets the criteria. Afterwards it searches for other components within one ERB that fulfill the criteria and that differ in amplitude less than 3 dB. If such components are found, then these components, together with the first one, are marked to cause excess masking (refer to the circles on top of some spectral lines in Figures 1, 2, 3). For every marked component, a 6-dB excess masking is introduced by modifying the weighting function described in [1].

The weighting function is introduced to integrate the tonality measure to the excitation-pattern model. From psychoacoustical experiments, the masking threshold is about 18 dB below the masker excitation level for a tonal masker, but about 6 dB below for a narrow band noise masker. For tonal components with excess masking we have lifted the masking threshold by 6 dB. That is, the masking threshold is about 12 dB below the masker excitation level for a tonal masker with excess masking. We have introduced this difference before excitation level calculation. The weighting function is described by

$$Spectrum\_weighted = 10^{(12(1-CW))/10} Spectrum \quad (1)$$

if there is no excess masking for this component,

$$Spectrum\_weighted = 10^{(12(1-CW-0.5))/10} Spectrum \quad (2)$$

if excess masking occurs for this component,

where CW is the unpredictability measure. The weighting function requires further optimization. The weighting function differs a bit from [1], because the spectrum is an amplitude spectrum in that case, a power spectrum in this paper.

2. EXPERIMENTAL RESULTS

For preliminary test purposes, we have used a pitchpipe signal, which contains rich sinusoidal harmonics. Figure 1 shows its amplitude spectrum. Then we have produced a second signal from the previous one by shifting all frequency components upward one semitone. By mixing the above two signals together, we produce a signal, which has equal amplitude component pairs that are close in frequency (see Figure 2). In addition, we have created a major triad in root position with a similar approach using the same pitchpipe signal. These kinds of signals are supposed to produce quite obvious excess masking.

We have evaluated the performance by integrating the modified excitation-pattern model into an MPEG-2 AAC type audio encoder, which contains only the basic coding tools. We first code these mixed signals with the original masking curve calculated with the excitation-pattern model and then with the modified one (exploiting excess masking). Without degrading the subjective audio quality, the average bitrate can be reduced by 5% for both pitchpipe and bagpipe signals, 10% for both two-pitchpipe-mixed signal and the major triad in root position of pitchpipe signal. For many other audio signals such as speech, harpsichord, castanets, glockenspiel, plucked strings, trumpet concerto, symphony orchestra and contemporary pop music, this model seems to have little effect in bitrate and causes no audible degradation in sound quality. Tests were performed informally by the authors and two young colleagues in the same lab.

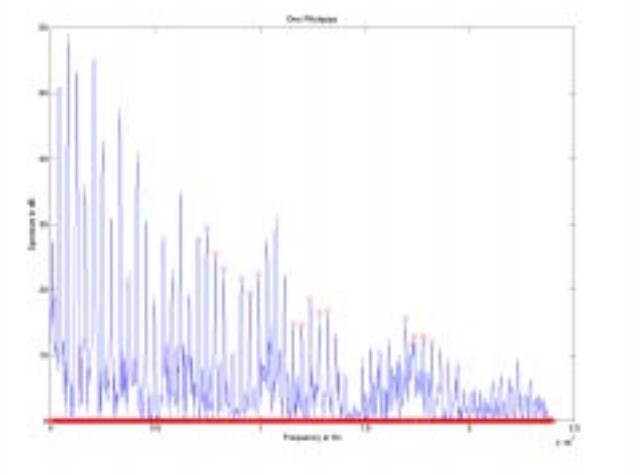

Figure 1. Spectrum of a piece of pitchpipe signal. Components that cause excess masking are marked with circles on top of them.

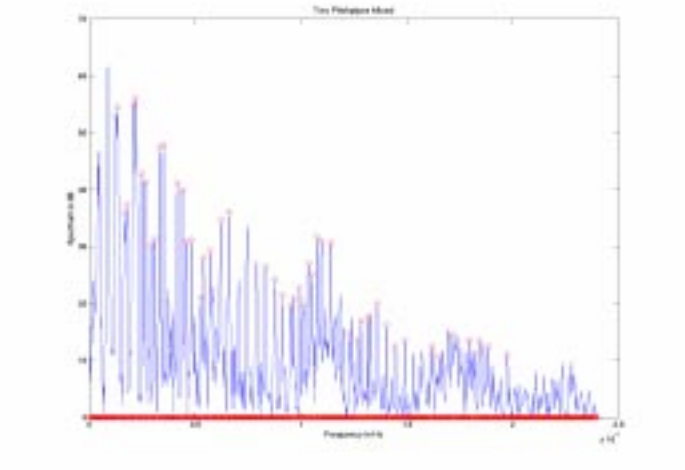

Figure 2. Spectrum of a piece of two-pitchpipe-mixed signal. Components that cause excess masking are marked with circles on top of them.

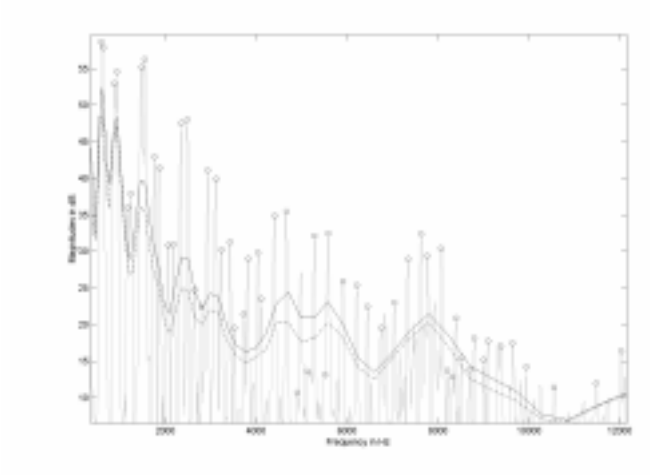

Figure 3. Spectrum of a piece of two-pitchpipe-mixed signal (dotted line), circles indicate components that cause excess masking, masking threshold without excess masking (dashed line), with excess masking (solid line). The masking thresholds are lifted parallel upwards for better visibility.

3. DISCUSSION

This work is only an initial work that utilises excess masking for audio coding applications. We have used only excess masking produced by pairs of sinusoids within one ERB. The algorithm that identifies components, which produce excess masking, is not optimised. It helps to reduce bitrate only for a few special audio signals such as pitchpipe, bagpipe and the mixed signals described earlier.

In addition to sinusoidal pairs, maskers can be two nearby narrow bands of noise, sinusoid combined with a narrow band of noise, etc. Excess masking of 8 dB was found for all masker configurations [7]. Even a pair of maskers outside the maskee auditory-filter bandwidth also produces some excess masking [5]. In addition, excess masking has been found in the time domain as well. That is, if the maskers are close enough in the time domain, the combined masking effect in the arithmetic center of the pair of maskers is not a linear combination of forward and backward masking [9]. In principle, all these excess masking phenomena can be utilised in audio coding. It is however very difficult to find a computationally efficient way to combine all excess masking into a practical audio encoder. It is worthwhile to point out that the maskees in almost all psychoacoustical studies [6][7][8] were sinusoids. To what extent these results can be utilised in audio coding is still an open question, since the maskee of an audio encoder is always the quantization noise, not sinusoids. Essentially what we are looking for is the optimal shaping of quantization noise according to the auditory masking.

In most of the publications, excess masking was measured at one particular point, most commonly in the middle of the pairs of maskers. How about the overall shape of excess masking (excess masking pattern) in the nearby frequency region or time span between the forward and backward maskers? This kind of overall shape would be much more useful in practical applications such as audio coding.

So far we have modified the weighting function to cope with the masking of both tonal components [1] and pairs of sinusoids. We have modified the amplitudes of these components before excitation-pattern model calculation. It is not obvious if this is the optimal way to solve these problems, since the excitation-pattern model is level dependent. In the case of reducing the amplitude of a tonal component, the corresponding auditory filter shape has been changed as well. More research is needed to answer these questions.

4. CONCLUSIONS

This preliminary test result proves that excess masking of sinusoidal pairs within one ERB can be exploited to compress at least some subclasses of audio signals more efficiently, especially for low bitrate applications. However, it is a challenging task to find technically feasible algorithms to include all excess masking into audio coding algorithms.

REFERENCES

[1] Wang Y., Vilermo M., (1999) “An Excitation Level Based Psychoacoustic Model for Audio Compression”, The Seventh ACM International Multimedia Conference, Florida, USA.

[2] Moore B. C. J., (1996) "Masking in the Human Auditory System", Collected Papers On Digital Audio Bit-Rate Reduction, special publication of AES.

[3] Moore B. C. J., Glasberg B. R., Baer T., (1997) “A Model for the Prediction of Thresholds, Loudness, and Partial Loudness”, J. Audio Eng. Soc., Vol. 45, No. 4.

[4] Van der Heijden M., Kohlrausch A., (1994) “Using an Excitation-pattern Model to Predict Auditory Masking”, Hearing Research 80, 38-52.

[5] Espinoza-Varas B., Cherukuri S. V., (1995) "Evaluating a model of auditory masking for

applications in audio coding", IEEE ASSP Workshop on Application of Signal Processing to Audio & Acoustics. New Paltz, New York.

[6] Green D. M. (1967). “Additivity of Masking”, J. Acoust. Soc. Am., 41, 1517-1525.

[7] Lutfi, R. A. (1983). “Additivity of simultaneous masking”, J. Acoust. Soc. Am., 73, 262-267

[8] Humes, L. E. and Jesteadt, W. (1989). “Models of the additivity of masking”, J. Acoust. Soc. Am., 85, 1285-1294.

[9] Moore B. C. J., (1997) "An Introduction to the Psychology of Hearing", 4. Edition, Academic Press, London.

[P2] Wang, Y., Vilermo, M. “An Excitation Level Based Psychoacoustic Model for Audio Compression,” The 7th ACM International Multimedia Conference, October 30 to November 4, 1999 Orlando, Florida, USA, pp. 401-404

An Excitation Level Based Psychoacoustic Model for Audio Compression

Ye Wang
Nokia Research Center
Speech and Audio Systems Lab
Tampere, Finland
Tel.: +358 3 272 5609
E-mail: ye.wang@nokia.com

Miikka Vilermo
Nokia Research Center
Speech and Audio Systems Lab
Tampere, Finland
Tel.: +358 3 272 5826
E-mail: miikka.vilermo@nokia.com

ABSTRACT

This paper describes an excitation level based psychoacoustic model to estimate the simultaneous masking threshold for audio coding. The system has the following stages: 1) a windowing function; 2) a time-to-frequency transformation; 3) an excitation level calculation block similar to that in Moore and Glasberg's loudness model; 4) a correction factor for estimating masking threshold; 5) the inclusion of the absolute masking threshold; 6) the output Signal-to-Masking ratio. We have evaluated the performance by integrating the proposed psychoacoustic model into an audio coder similar to MPEG-2 AAC, which contains only the basic coding tools. Our model performs better than or as well as the psychoacoustic model suggested in the MPEG-2 AAC audio coding standard for all the test signals. We can achieve almost transparent quality with bitrate below 64 kbps for most of the critical test signals. Significant improvements have been achieved with speech signals, which are always difficult for transform audio coders.

Keywords

Psychoacoustic model, excitation level, masking threshold, audio compression.

1. INTRODUCTION

Combining psychoacoustic models into audio coders significantly improves the coding efficiency. However, the psychoacoustic models used so far in perceptual coders are based on very simplified assumptions, which may result in much less accurate approximations of masking thresholds. For example, the psychoacoustic models suggested in the audio parts of MPEG-1 and MPEG-2 use a DFT of successive blocks of the audio signal, which gives the associated spectral components of the blocks. For each spectral component an individual masking threshold is generated. The overall masking threshold follows from superposition of the individual thresholds, which is carried out by simply adding up the threshold at the corresponding frequencies [1]. This masking threshold determines the maximum quantization noise energy that can be added to the original signal to keep the noise inaudible. These models are quite approximate, when a complex target (quantization noise) has to be masked by a complex masker comprising multiple spectral components (either speech or musical sounds) [11]. Further bit rate reduction heavily depends on the accurate estimation of the masking threshold both in the time and frequency domains.

To simulate the human ear better, some ear models have been developed [4][5][6][10]. Our model is based on Moore and Glasberg's excitation level calculation. This is quite different from psychoacoustic models commonly used, and it leads to some advantages in masking threshold estimation.

2. MODEL DESCRIPTION

Figure 1 shows the block diagram of our method. The following steps are performed:

A windowing function is first applied to the input audio signal. We apply the same window function as specified in MPEG-2 AAC. Depending on the signal, the model changes the time/frequency resolution by using two different windows: LONG_WINDOW = 2048 and SHORT_WINDOW = 256. We have applied two different transition windows LONG_START_WINDOW and LONG_STOP_WINDOW in case of switching between long and short windows. The transition windows have not been used in the psychoacoustic model suggested in MPEG-AAC. Using the exactly same window switching in both the psychoacoustic model and in the MDCT (Modified Discrete Cosine Transform) helps to reduce some coding artifacts.

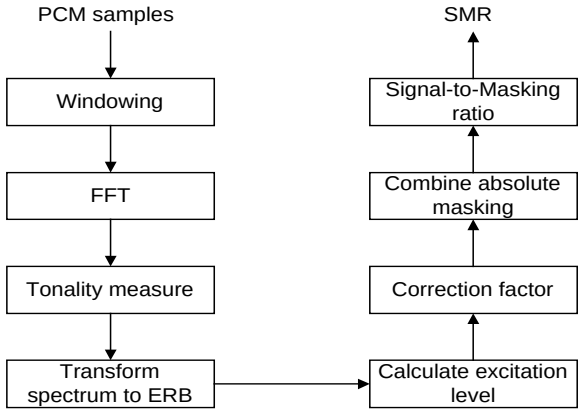

Figure 1. Block diagram of sequence of stages in the model

The reason for window switching is that Moore's loudness model is designed for steady sounds. It can not cope with transient signals well, and at the moment, we solve this problem by introducing window switching.

The FFT has been chosen for the time-to-frequency transformation. The transform block length is 32768 ($=2^{15}$) for practical reasons: Moore and Glasberg's model uses the equivalent rectangular bandwidths (ERBs), which are similar to traditional critical bands at low frequencies (see below). To ensure that each ERB has at least one frequency line, the FFT block length has to be increased by padding with zeros after the actual data, which are 2048 points for the long window and 256 for the short window. This increases the number of frequency lines, while preserving the shape of the spectrum.

Because tonal and non-tonal components have very different masking properties, we introduce the tonality measure as a weighting function of the frequency components. Currently we use unpredictability as a tonality measure similar to the method specified in MPEG-2 AAC. However, our model predicts from both the past and the future two frames. We choose the one with less prediction error for calculating the unpredictability measure. This remarkably improves the coding efficiency for some signals.

A critical problem is how to integrate the tonality measure with the masking threshold. From psychoacoustical experiments, the masking threshold is about 18 dB below the masker excitation level for a tonal masker, but about 6 dB below for a narrow band noise masker. We have introduced this difference before excitation level calculation. The weighting function is described by

$$\text{Spectrum\_weighted} = 10^{-(12(1-CW))/20}\text{Spectrum}, \qquad (1)$$

where CW is the unpredictability measure. The weighting function requires further optimization.

At moderate sound levels, the ERB width is described by

$$\text{ERB} = 24.7(4.3F+1), \qquad (2)$$

where the ERB is in hertz and the center frequency F is in kilohertz. This function is similar to the "traditional" critical bandwidth (CB) function at medium to high frequencies, but gives markedly lower values than the CB function at center frequencies below 500 Hz. [5]

The next step is to transform from the frequency domain to the ERB scale, which is described by

$$\text{Number of ERBs} = 21.4 \log10(4.37F+1), \qquad (3)$$

where the frequency is in kilohertz [5].

In our model we have not used the outer and middle ear transfer function, because the final masking threshold for coding must be transformed back to free field sound pressure level. We assume that the forward and backward transfer function of outer and middle ear cancel each other.

The excitation pattern for a given spectrum is calculated being the pattern of outputs from the auditory filters. Each auditory filter is assumed to be quasi-linear at a given level, but to change shape with frequency and with level in a way similar to that described by Moore and Glasberg [8].

It is assumed that the masking pattern should be parallel to the excitation pattern of the masker, but shifted vertically downwards by a small amount [9], we have introduced the *CORRECTION FACTOR* to represent that shift and tried to find out the optimal correction factor experimentally. For all test materials used, 6 dB is a suitable correction factor. We have also modified the correction factor below 500 Hz according to [5]. The influence on bitrate versus audio quality seems to be minimal.

Because Moore's model does not cover the whole audible frequency range up to 20 kHz, we combine the calculated masking threshold with the absolute masking threshold as the global masking threshold. Choosing the higher of the two thresholds approximates this combination. Finally we output the Signal-to-Masking ratio (SMR) for each scalefactor band.

3. EXPERIMENTAL RESULTS AND DISCUSSION

The psychoacoustic model is built into a codec similar to MPEG-2 AAC. The test materials are provided by MPEG and commonly used in audio coding evaluation. These include English and German speech spoken by male and female, female singing in English without instrumental accompaniment, harpsichord, castanets, pitchpipe, bagpipe, glockenspiel, plucked strings, trumpet concerto, symphony orchestra and contemporary pop music. Our model performs better than the MPEG-2 AAC psychoacoustic model for all signals. To achieve the same audio quality, we can save 10-20% bits.

Moore and Glasberg's loudness model is intended for stationary signals, but we have used it for real audio signals, which sometimes have strong transients. The transients should be tackled by using e.g., window switch, more accurate detection of transients, better exploiting temporal masking, short window grouping etc. Preliminary tests show that an additional 10% reduction in bit rate can be achieved through combining simultaneous masking and forward masking. It should be noted that we have not used any prediction for our test. Backward adaptive prediction would improve coding efficiency for some signals, such as the pitchpipe and bagpipe. However, it does not help very much for other test signals.

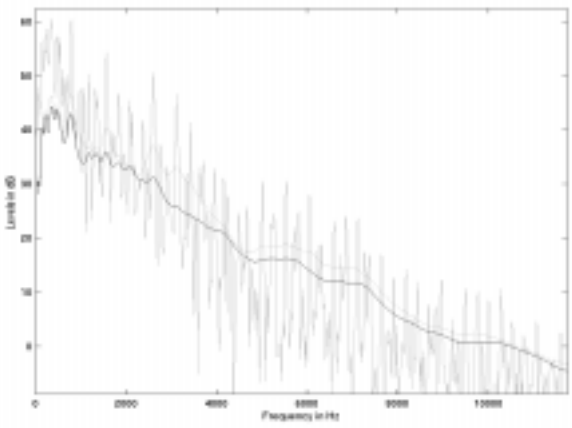

Figure 2 Masking threshold with (solid) and without (dotted) the tonality measure calculation for a noise-like signal

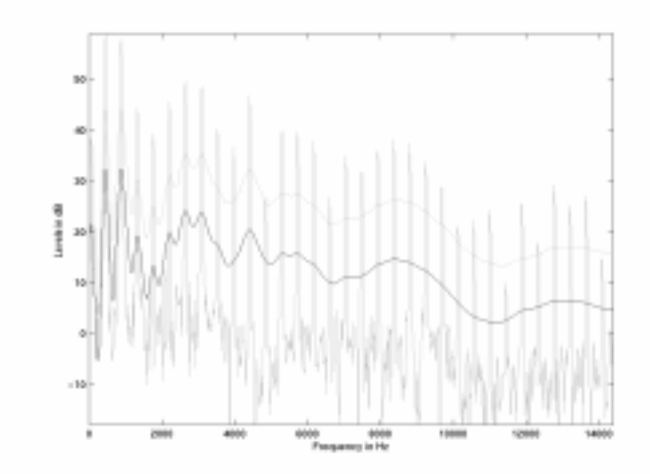

Figure 3 Masking threshold with (solid) and without (dotted) the tonality measure calculation for a signal with significant sinusoidal components

Figure 2 and 3 show the effect of the tonality measure. Figure 2 shows the spectrum of a piece of symphony orchestra signal and its masking thresholds calculated with and without the tonality measure. Figure 3 shows the spectrum of a section of pitchpipe signal and its masking thresholds calculated with and without the tonality measure. The symphony orchestra signal is more noise-like and the difference between the two masking thresholds is rather small. The pitchpipe contains rich sinusoidal harmonics and the difference between the two masking thresholds is more significant.

Figure 4 shows the spectrum of a section of the symphony orchestra signal and its masking thresholds from the MPEG2-AAC model (dotted) and our model (solid). Our model shows a different distribution of the allowed quantisation noise compared to the MPEG2-AAC model.

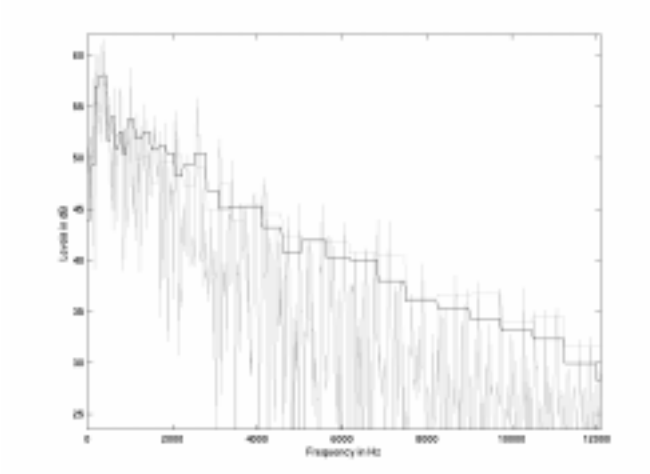

Figure 4 Masking threshold from the MPEG2-AAC model (dotted) and our model (solid)

4. CONCLUSION AND FUTURE WORK

The proposed psychoacoustic model can predict the masking threshold quite well for most test signals. Particularly, the performance with speech signals makes it very promising for a future hybrid speech and audio coders. Based on experimental codes in MATLAB, we have implemented our model in C language with some optimization for real-time applications.

What could be done in the future is:

- To find some other tonality measure which is more reliable than the unpredictability measure;
- In order to tackle transient signals better, the window switch mechanism has to be improved;
- In order to squeeze the bitrate further, short window grouping can be tested.

5. ACKNOWLEDGMENT

The authors wish to thank their superior, Mr. Mauri Väänänen (Nokia Research Center) for supporting this research; Prof. Brian C. J. Moore (University of Cambridge) for his careful reading of the manuscript and critical comments on improving the technical content and presentation of this paper, especially regarding the outer and middle ear transfer function; Dr. Jilei Tian (Nokia Research Center) for very inspiring discussion.

6. REFERENCES

[1] ISO/IEC JTC1/SC29/WG11, “Coding of moving pictures and audio- MPEG-2 Advanced Audio Coding AAC”, ISO/IEC 13818-7 International Standard, 1997.

[2] N. S. Jayant, P. Noll, “Digital Coding of Waveforms”, Prentice-Hall, Englewood Cliffs, NJ0732, U.S.A, 1984.

[3] E. Zwicker, H. Fastl: “Psychoacoustics, Facts and Models”, Springer-Verlag, Berlin Heidelberg, Germany, 1990.

[4] J. G. Beerends, J. A. Stemerdink, “A Perceptual Audio Quality Measure Based on a Psychoacoustic Sound Representation”, J. Audio Eng. Soc., Vol. 40, No. 12, 1992.

[5] B. C. J. Moore, B. R. Glasberg, T. Baer, “A Model for the Prediction of Thresholds, Loudness, and Partial Loudness”, J. Audio Eng. Soc., Vol. 45, No. 4, 1997.

[6] F. Baumgarte, “A Physiological Ear Model for Auditory Masking Applicable to Perceptual Coding”, 103rd AES Convention, New York, NY, September 1997.

[7] B. C. J. Moore, "An Introduction to the Psychology of Hearing", 4. Edition, Academic Press, 1997.

[8] B. R. Glasberg and B. C. J. Moore, "Derivation of Auditory Filter Shapes from Notched-Noise Data", Hear. Res., Vol. 47, pp.103-138 (1990).

[9] B. C. J. Moore, "Masking in the Human Auditory System", Collected Papers On Digital Audio Bit-Rate Reduction, special publication of AES, 1996.

[10] C. Colomes, M. Lever, J. B. Rault, Y. F. Dehery, "A Perceptual Model Applied to Audio Bit-Rate Reduction", J. Audio Eng. Soc., Vol. 43, No. 4, 1995 April.

[11] B. Espinoza-Varas, S. V. Cherukuri, "Evaluating a model of auditory masking for applications in audio coding", proc. 1995 IEEE ASSP Workshop on Application of Signal Processing to Audio & Acoustics. New Paltz, New York.

[P3] Wang, Y., Vilermo, M., Yaroslavsky, L. “Energy Compaction Property of the MDCT in Comparison with other Transforms”, AES109th International Convention, September 22-25, 2000, Los Angeles, California, USA, preprint 5178

ENERGY COMPACTION PROPERTY OF THE MDCT IN COMPARISON WITH OTHER TRANSFORMS

Ye Wang[1], Miikka Vilermo[1], Leonid Yaroslavsky[2]

[1]Nokia Research Center,
Speech and Audio Systems Laboratory,
FIN-33720 Tampere,
Finland
[2]Department of Interdisciplinary Studies,
Tel Aviv University,
Ramat Aviv 69978,
Israel
e-mail: ye.wang@nokia.com, miikka.vilermo@nokia.com, yaro@eng.tau.ac.il

Abstract - This paper focuses on the energy compaction properties of five different transforms: DFT, DCT, SDFT((N+1)/2, 1/2), MDCT and DST. Energy compaction properties of these transforms are compared experimentally. In addition to sinusoidal signals, sixteen classical and pop music pieces are used for the experiments. The influence of different window sizes (256, 512 and 1024 samples) and different window shapes (rectangular and sine) are investigated. The results of the experiments are presented and analyzed.

I. INTRODUCTION

Signal Fourier spectrum analysis is one of the major tools of signal processing. For real-life continuous signals such as audio signals and images, it is associated with signal integral Fourier transformation. In digital signal processing, integral Fourier transformation is approximated by Discrete Fourier Transforms implemented via Fast Fourier Transform algorithms. On the other hand, it has been found that in image and audio coding, restoration and similar applications other transforms such as Discrete Cosine Transform (DCT), Discrete Sine Transform (DST), and Modified Discrete Cosine Transform (MDCT) may be more suitable than DFT [1][2]. However, it is often necessary to establish interrelations between DFT signal spectra and those of DCT, MDCT, and DST to evaluate their applicability for signal Fourier analysis. Based on these interrelations, the energy compaction properties of these transforms are investigated in this paper. This work is an extension of our previous paper [3].

In most state-of-the-art audio encoders, MDCT [4] is used to compress signals in the frequency domain. In this context, it is necessary to examine how well the transform approximates the Fourier spectrum and why the MDCT exhibits an energy compaction property. Although MDCT fails in some special situations [5] for spectral analysis, it is commonly used in audio coding applications.

Transform energy compaction capability means the capability of the transform to redistribute signal energy into a small number of transform coefficients. It can be characterized by the fraction of the total number of signal transform coefficients that carry a certain (substantial) percentage of the signal energy. The lower this fraction is for a given energy percentage, the better the transform energy compaction capability is.

There are different approaches to studying the energy compaction property of different transforms, because the spectral discretization interval of the transforms may be different. In the case of stationary signals, the conventional solution is to use different time-domain window sizes so that the spectral discretization intervals of different transforms remain the same. However, if the signal is nonstationary it may be more reasonable to use the same time-domain window size for all transforms, because time domain windows of different sizes may contain significantly different frequency components. In addition there are certain constraints on the window size in different applications. In our approach, we employ interpolation and normalization to align all transform spectra in the same coordinate for a fair comparison.

Purely analytical evaluation of the transform energy compaction capability is problematic since it is only feasible for limited mathematical models of signals. Another option is to evaluate the energy compaction property experimentally with a large number of test signals. In this paper, we first study the energy compaction property of different transforms by comparing the spectral resolution of individual sinusoids. Then we present the results of such an evaluation for a set of 8 pieces of classical music and 8 pieces of pop music. For these signals, the energy compaction capability of transforms is investigated with different window sizes (256, 512 and 1024 samples) and with different window functions (rectangular and sine) over 60*44100 samples. The sampling frequency was 44.1 kHz. The results are illustrated in frequency coordinates normalized to [0-1] by the Nyquist frequency.

II. INTERRELATION BETWEEN INTEGRAL FOURIER TRANSFORM, DFT, DCT, MDCT, DST

Discrete representation of signal integral transforms parallels that of signals. For a signal $a(x)$ and its Fourier spectrum $\alpha(f)$ represented in a discrete form by means of sequences of their samples $\{a_k\}$ and $\{\alpha_r\}$ taken at sets of equidistant points $\{(k+u)\Delta x\}$ and $\{(r+v)\Delta f\}$, $k = \ldots,-2,-1,0,1,2,\ldots$; $r = \ldots,-2,-1,0,1,2,\ldots$ such that

$$a(x) = \sum_k a_k \varphi_x\left(x-(k+u)\Delta x\right), \tag{1}$$

$$\alpha(f) = \sum_r \alpha_r \varphi_f\left(f-(r+v)\Delta f\right), \tag{2}$$

where Δx and Δf are discretization intervals and u and v are shifts (in a fraction of the corresponding discretization interval) of sample positions from the origin of the corresponding coordinates, discrete representation of the Fourier integral

$$\alpha(f) = \int_{-\infty}^{\infty} a(x)\exp(i2\pi fx)dx \tag{3}$$

takes the form of "Shifted Discrete Fourier Transforms" (SDFT) [2]:

$$\alpha_r = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} a_k \exp\left(i2\pi\frac{kv}{N}\right)\exp\left(i2\pi\frac{(k+u)r}{N}\right) \tag{4}$$

the most wide known special case of which (for zero shifts u and v) is DFT:

$$\alpha_r = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} a_k \exp\left(i2\pi\frac{kr}{N}\right) \tag{5}$$

Other well-known transforms in digital signal processing such as DCT, MDCT and DST are also special cases of SDFT. From derivation it can be seen that signal spectra obtained by DCT, MDCT, DST are identical to Shifted Discrete Fourier Transform spectra of certain permutation modifications of the original signal:

$$\text{DCT: } \alpha_r = \sum_{k=0}^{N-1} a_k \cos\left[\pi\frac{\left(k+\frac{1}{2}\right)r}{N}\right] = \frac{1}{2}\sum_{k=0}^{2N-1}\tilde{a}_k \exp\left[i2\pi\frac{\left(k+\frac{1}{2}\right)r}{2N}\right], \tag{6}$$

where $\tilde{a}_k = \begin{cases} a_k, & k = 0,\ldots,N-1 \\ a_{2N-1-k}, & k = N,\ldots,2N-1 \end{cases}$

$$\text{MDCT: } \alpha_r = \sum_{k=0}^{N-1} a_k \cos\left[2\pi \frac{\left(k+\frac{N+2}{4}\right)\left(r+\frac{1}{2}\right)}{N}\right] = \frac{1}{2}\sum_{k=0}^{N-1} \tilde{a}_k \exp\left[i2\pi \frac{\left(k+\frac{N+2}{4}\right)\left(r+\frac{1}{2}\right)}{N}\right], \quad (7)$$

where $\tilde{a}_k = \begin{cases} a_k - a_{N/2-1-k}, & k = 0,...,N/2-1 \\ a_k + a_{3N/2-1-k}, & k = N/2,...,N-1 \end{cases}$, and N is even;

$$\text{DST: } \alpha_r = \sum_{k=0}^{N-1} a_k \sin\left[\pi \frac{(k+1)(r+1)}{N+1}\right] = \frac{1}{2}\sum_{k=0}^{2N} \tilde{a}_k \exp\left[i2\pi \frac{(k+1)(r+1)}{2(N+1)}\right], \quad (8)$$

Where $\tilde{a}_k = \begin{cases} a_k, k = 0,...,N-1 \\ 0, k = N, 2N+1 \\ -a_{2N-k}, k = N+1,..,2N \end{cases}$

These relationships mnemonically illustrated in Fig. 1 lucidly explain the interrelations between the above trigonometric bases and their similarity and dissimilarity.

III. COMPARISON OF TRANSFORM SPECTRAL RESOLUTION POWER AND ENERGY COMPACTION CAPABILITY

In this section, we compare the above trigonometric bases in terms of their energy compaction capability and their resolution power in Fourier spectrum analysis. This property is relevant for many applications, such as signal compression/coding.

DFT, DCT, DST and MDCT all have different spectral discretization intervals. For an N-samples long real sequence the independent DFT bins represent frequencies $2k/N, k = 0,...,\lfloor N/2 \rfloor$. The frequency ordinate is normalized to the Nyquist frequency for simplicity. DCT bins represent frequencies $k/N, k = 0,...,N-1$, DST bins represent frequencies $(k+1)/(N+1), k = 0,...,N-1$, and MDCT bins represent frequencies $(2k+1)/N, k = 0,...,N/2-1$. For MDCT we assume N to be even since MDCT is a Lapped Orthogonal Transform (LOT). The discretization interval Δf of DFT is the basis of all comparisons described in this paper. Note that the discretization intervals of DFT and MDCT are twice as long as that of DCT and DST.

Transform Resolution Power with Sinusoids

The transform resolution power in signal spectral estimation characterizes the sharpness of spectral peaks of sinusoidal signals. It can be evaluated numerically as the width, in proportion to the discretization interval, of the spectral peak within which a given (substantial) percentage of the energy of a sinusoidal signal is contained. From sampling theory it follows that the width of the spectral peaks in the signal discrete spectrum is, in general, proportional to the discretization interval in the frequency domain. However, the proportionality is different for different discrete trigonometric transforms discussed in previous section.

Evaluation of transform spectral resolution power requires testing the spectral peak width of sinusoidal signals having arbitrary frequencies within the frequency range defined by the signal discretization rate. Although the evaluation can be carried out analytically in principle, the same results can be obtained by numerical simulation of the transforms. As initial numerical simulations, sine test signals with frequencies uniformly distributed within the corresponding frequency discretization interval were selected and the results of spectrum estimation for each central frequency were published in [3]. Those results were averaged in such a way that the spectra within a discretization interval were added and the resulting spectrum was used to measure the resolution power. 100 realizations were used for each spectral discretization interval.

In this paper, we have chosen an improved method to estimate the resolution power. Instead of adding the 100 spectra together, we measure the individual spectral width of the 100 realizations, and then average them within each frequency discretization interval. In addition, we have tested cosine signals and cosine signals with random phase shifts. Some new results are reported in this paper.

The principle of our method is illustrated in Fig. 2 and Fig. 3 using only 18 time domain samples for clarification. The dashed lines in Fig. 2 represent the actual spectral lines of a sinusoid $\sin(2\pi ft)$ whose frequency changes within one frequency discretization interval Δf. DFT and DCT spectra (black and white respectively in Fig. 2) of $f_r - \Delta f/2$, $f_r - \Delta f/4$, f_r, $f_r + \Delta f/4$, $f_r + \Delta f/2$ are illustrated in (a)-(e), where f_r corresponds to one DFT sampling point. In order to have the same spectral discretization interval, DFT is interpolated by a factor of 2 using the lowpass interpolation algorithm described in [6]. Obviously these two spectra are different representations of the same sinusoid. For more precision, both DFT and DCT spectra are further interpolated by a factor of 5 in Fig. 3. Then we set the energy threshold at 50% and measure the normalized width of each spectrum and then take the averaged value within each frequency discretization interval Δf.

By increasing the realization from 5 to 25, the development of the DFT spectral shape of a cosine signal with a frequency changing from $f_r - \Delta f/2$ to $f_r + \Delta f/2$ is illustrated in Fig. 4. Similarly, the development of the DCT spectral shapes of the same cosine signal is illustrated in Fig. 5-7, MDCT spectral shapes in Fig. 9-11. Interestingly, the pattern of the DFT spectral shapes within each Δf remains the same in all frequency regions, while the patterns of DCT and MDCT spectral shapes within each Δf change with frequency.

The averaged frequency resolutions of the transforms are illustrated in Fig. 12-16 using sine, cosine and cosine with a random phase shift respectively. On average in the whole frequency range [0-1] with rectangular window as in Fig. 12, 14, 15, the frequency resolutions are $0.6635\Delta f$ for DFT, $0.7171\Delta f$ for MDCT, $0.525\Delta f$ for DCT and $0.5286\Delta f$ for DST. However, applying sine window has changed the frequency resolution landscape as illustrated in Fig. 13, 16.

From (6) (8) it can be seen that the phase of a_k has a direct impact on the DCT and DST coefficients. If a_k is in phase with the basis function, the frequency resolution of the transform is optimal. Conversely, a phase shift of $\frac{\pi}{2}$ corresponds to the most sub-optimal frequency resolution. This is verified by the experiments. To explain the frequency-dependent resolution power of DCT, we take the DCT basis function from (6) and change the expression to:

$$\cos\left[\pi\frac{\left(k+\frac{1}{2}\right)r}{N}\right] = \cos\left[\frac{2\pi kr/2}{N} + \frac{\pi r}{2N}\right] \tag{9}$$

The second term in the right hand side bracket is the phase of the basis function $\varphi = \frac{\pi r}{2N}$, $r = 0,...N-1 \Rightarrow$

$$\varphi = 0,...,\frac{\pi(N-1)}{2N} \tag{10}$$

This phase shift of the basis functions explains the frequency dependent resolution power of DCT in Fig 12, 14.

Similarly, we take the DST basis function from (8) and change the expression to:

$$\sin\left[\pi\frac{(k+1)(r+1)}{N+1}\right] = \sin\left[\frac{2\pi k(r/2+1/2)}{N}\frac{N}{N+1} + \frac{\pi(r+1)}{N+1}\right] \tag{11}$$

The second term in the right hand side bracket is the phase of the basis function $\varphi = \frac{\pi(r+1)}{N+1}$, $r = 0,...N-1$, $\Rightarrow$

$$\varphi = \frac{\pi}{N+1},..,\frac{\pi N}{N+1} \quad (12)$$

If N is big (512 in our experiments), the range of the phase is between 0 and π. This explains the DST frequency dependent resolution power in Fig. 12, 14. Fig 13 illustrates the effect of the window function.

The basis function of DFT is an exponential function, which can be split into sine and cosine basis functions. The sine and cosine basis functions complement each other in the resolution power. This explains why DFT frequency resolution is not sensitive to the phase of the signal as illustrated in Fig. 12-16.

From Fig. 2 to 14 the sinusoids used are all of fixed phase. Fig. 15 and 16 show the frequency resolution of sinusoids having random phase shifts. Because of the special properties of the MDCT [5], it is difficult to explain its frequency dependent resolution analytically, and therefore it is omitted in this paper.

Transform Energy Compaction with Real-life Audio Signals

This section discusses the transform energy compaction property with real-life audio signals. 16 pieces of pop and classic music signals were used in the experiment. Fig. 17 shows the energy compaction property with 8 pieces of pop music signals. The window size is 256-1024 samples for all transforms. Rectangular and sine windows are used. The length of the audio signals are 60*44100 samples and the sampling frequency is 44.1 kHz.

Fig. 18 shows a zoomed version of the comparison, but with a window length of 256 samples. In general, the DCT performs better than other transforms, and the DST performs poorest. This is not very consistent in comparison with the frequency resolution of sinusoidal test signals.

Fig 19 shows the comparison when a sine window is applied. This shows the effect of the window function. As in Fig. 19 the energy compaction property gets more unified with the sine window. Similar comparison with classic music signals is shown in Fig. 20 and 21.

We have also taken the conventional approach to compare the energy compaction property with different time domain windows. The results are shown in Fig 22 - 23. Interestingly, DCT with a window length of 512 performs better than MDCT with a window length of 1024, if rectangular windows are used. However, when keeping the window size unchanged and applying a sine window to both DCT and MDCT or to MDCT only, the energy compaction performance of MDCT is better than DCT. This comparison has been considered to be useful in audio coding applications. Fig. 24 and 25 show the comparison for the case that DCT has a rectangular window length of 512 and MDCT has a sine window length of 1024. However, the different time window may contain significantly different frequency components as noted earlier.

IV. CONCLUSION

All above-mentioned transforms can be used for signal Fourier analysis.

The transforms exhibit different Fourier spectrum analysis resolution power and energy compaction property; the resolution power is not uniform over the entire frequency range for DCT, DST and MDCT using sine and cosine test signals. The averaged resolution power of DFT is uniform within the whole frequency range. On average, over the whole frequency range, DCT and DST have the best resolution power, and MDCT has the poorest resolution power using rectangular window. All these transforms have almost the same resolution power when a sine window is used.

For real-life audio signals, DCT, MDCT and DST exhibit, on average, over large signal sequences, effectively similar energy compaction capabilities. More then 90% energy is concentrated within 10% of the normalized frequency scale for most of the test signals for all transforms concerned. The energy compaction property of different transforms gets more unified with increased window size.

V. ACKNOWLEDGEMENT

Ye Wang wishes to thank Prof. Peter J. Sherman (Iowa State University, USA), Dr. Jilei Tian (Nokia Research Center, Finland), Dr. Bernd Edler (Hannover University, Germany) and Dr. Juergen Herre (FhG-IIS, Germany) for

helpful discussions during ICASSP2000 in Istanbul. The financial support from Nokia Foundation and the Academy of Finland is greatly acknowledged. Leonid Yaroslavsky wishes to thank Tampere International Center for Signal Processing for supporting this research.

REFERENCES

[1] Malvar, H., "Signal Processing with Lapped Transform", Artech House, Boston, 1991

[2] Yaroslavsky, L., Eden, M., "Fundamentals of Digital Optics", Birkhauser, Boston, 1996.

[3] Yaroslavsky, L., Wang, Y., "DFT, DCT, MDCT, DST and Signal Fourier Spectral Analysis", X European Signal Processing Conference (EUSIPCO 2000), Tampere, Finland, September 4-8, 2000.

[4] Princen, J. P., Bradley, A. B., "Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No. 5, October 1986.

[5] Wang, Y., Yaroslavsky, L., Vilermo, M., Väänänen, M. "Some Peculiar Properties of the MDCT", WCC2000 – 16th IFIP World Computer Congress/ICSP 2000 – 5th International Conference on Signal Processing, August 21 – 25, 2000, Beijing, China.

[6] Wiley John & Sons, "Programs for Digital Signal Processing", IEEE Press, New York, 1979.

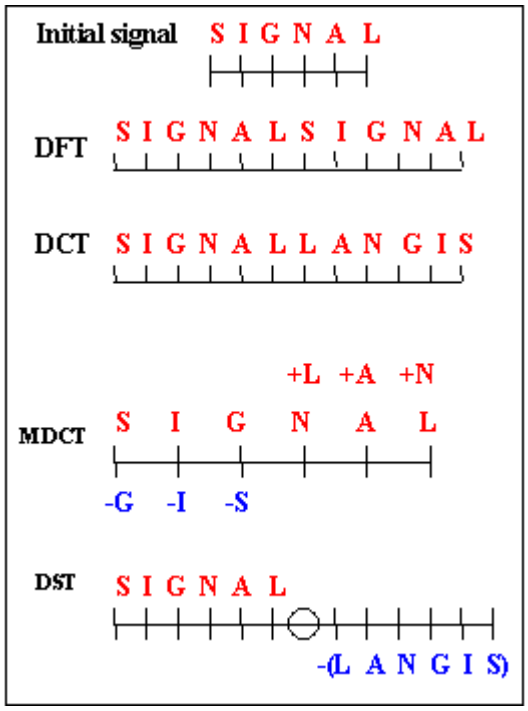

Fig. 1. Signal and its corresponding representations of DFT, DCT, MDCT and DST

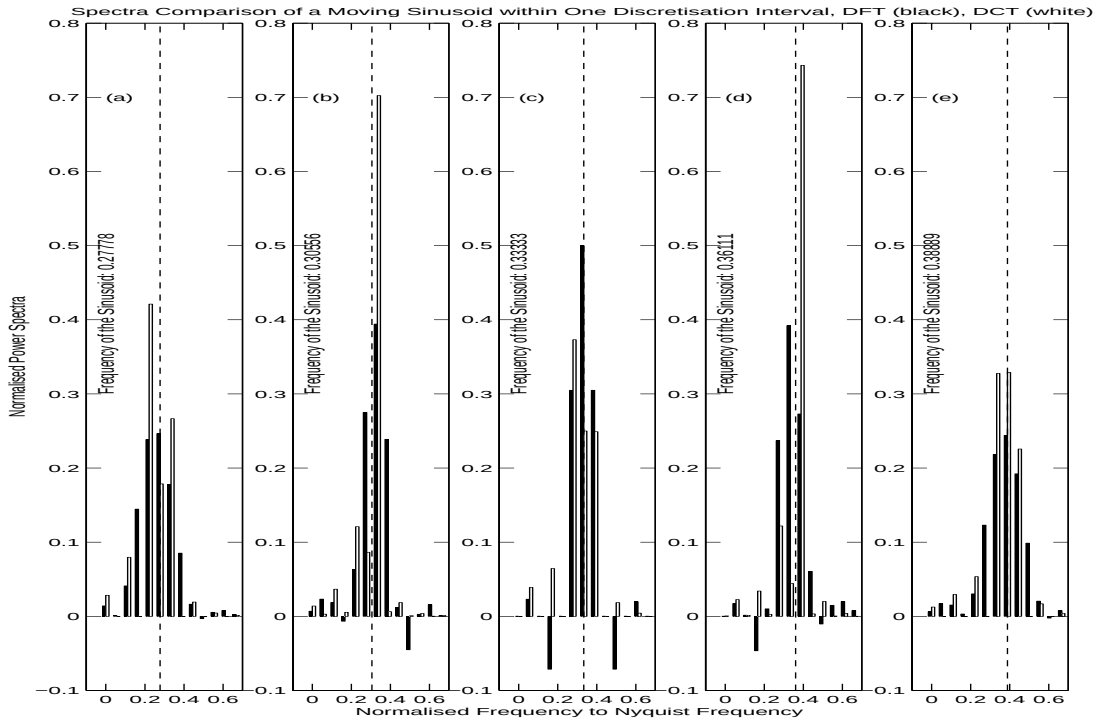

Fig. 2. DFT (black) and DCT (white) power spectra comparison using a sine signal whose frequency (indicated as the dashed lines) changes within one DFT discretisation interval Δf. DFT spectrum is interpolated by factor 2 to have the same spectral discretization interval as DCT. The negative values of DFT power spectra are caused by interpolations.

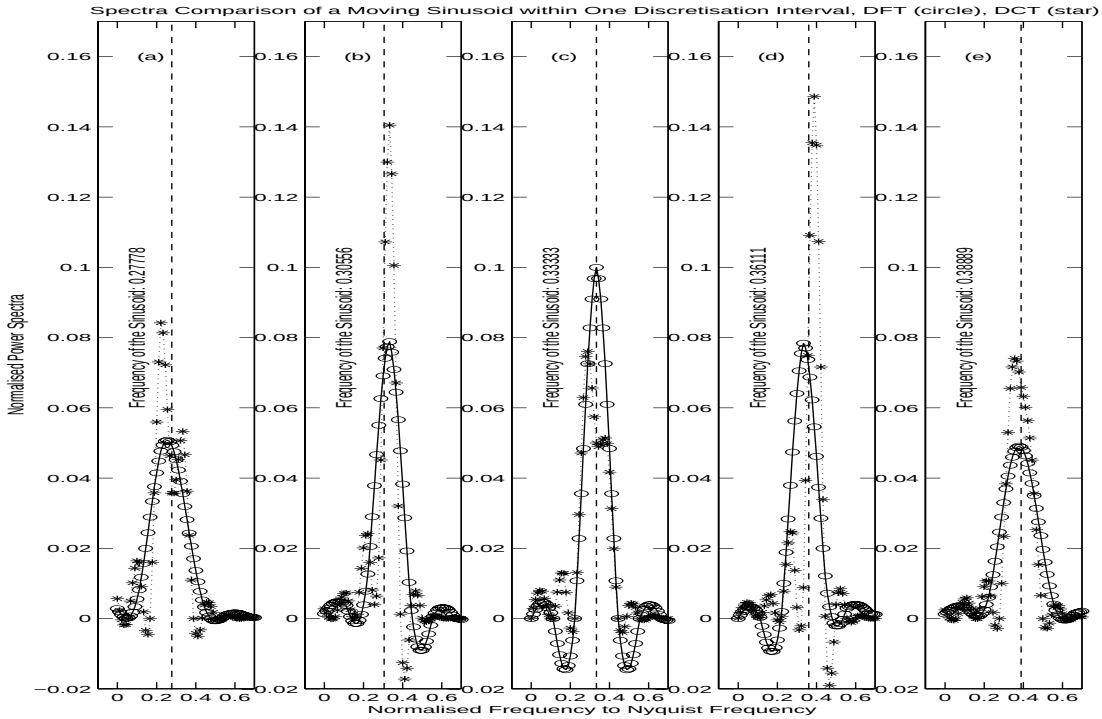

Fig. 3. DFT and DCT power spectra comparison using a sine signal whose frequency (indicated as the dashed lines) changes within one DFT discretisation interval Δf. Both DFT and DCT spectra are further interpolated by factor 5 from Fig.2.

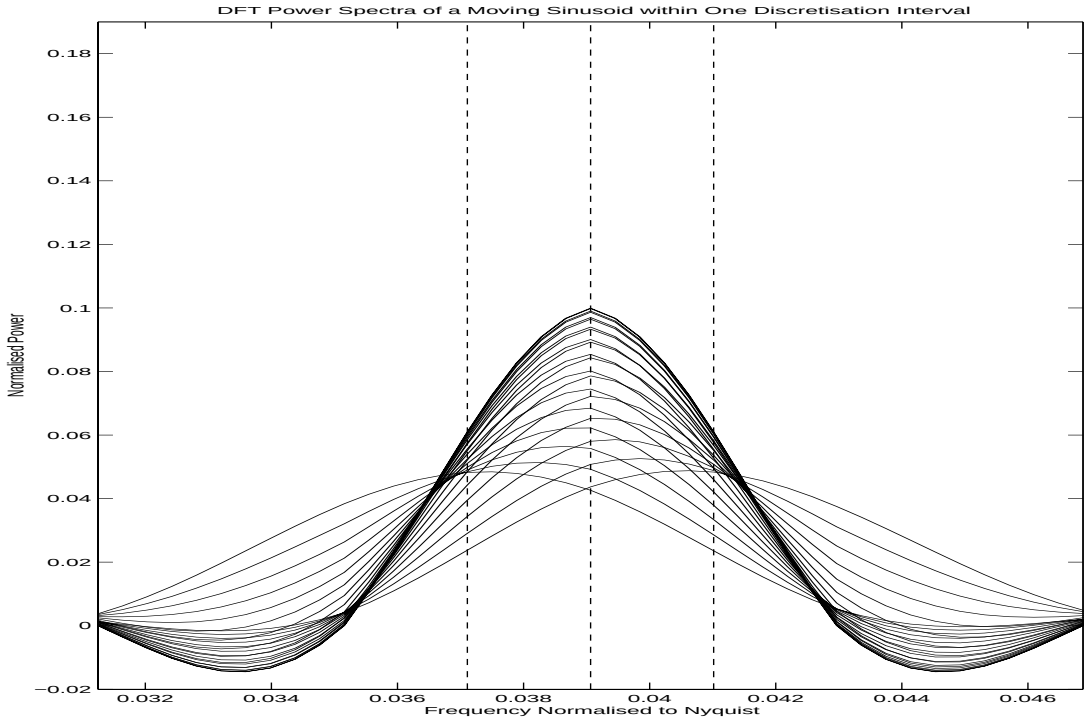

Fig. 4. DFT power spectra of a cosine signal whose frequency changes within one discretisation interval Δf. The 3 dashed lines from left to right represent $f_r - \Delta f/2$, f_r, $f_r + \Delta f/2$.

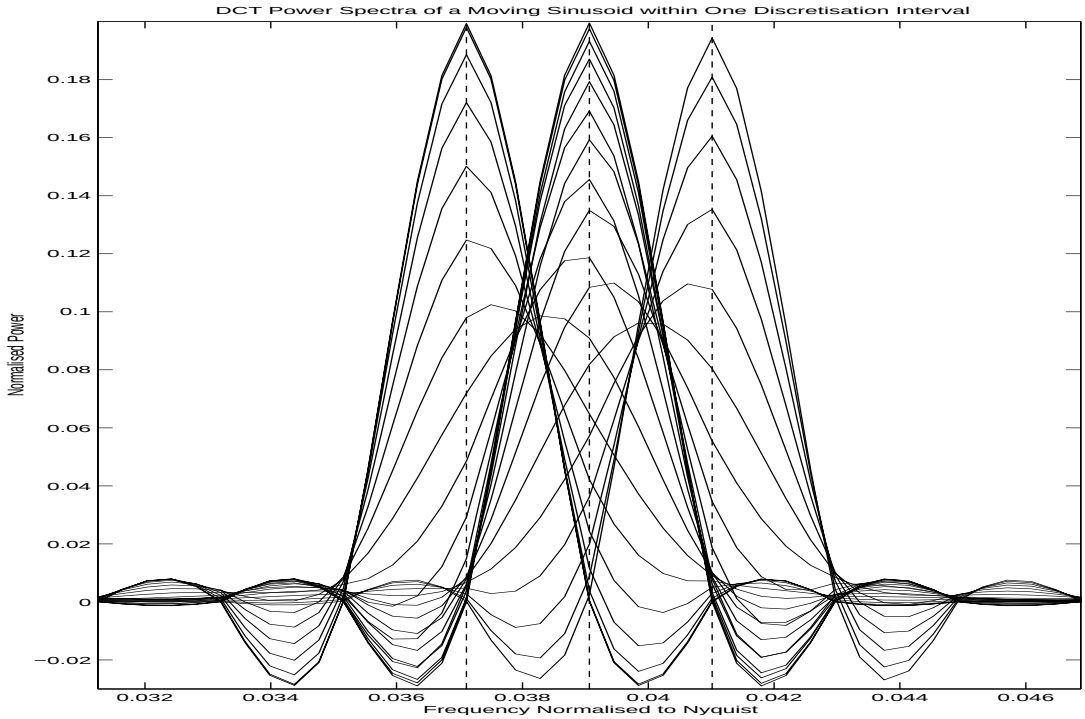

Fig. 5. DCT power spectra of a cosine signal whose frequency changes within Δf in low frequency range. Note that Δf is twice as long as DCT discretization interval. The dashed lines correspond to the DCT spectral sampling points.

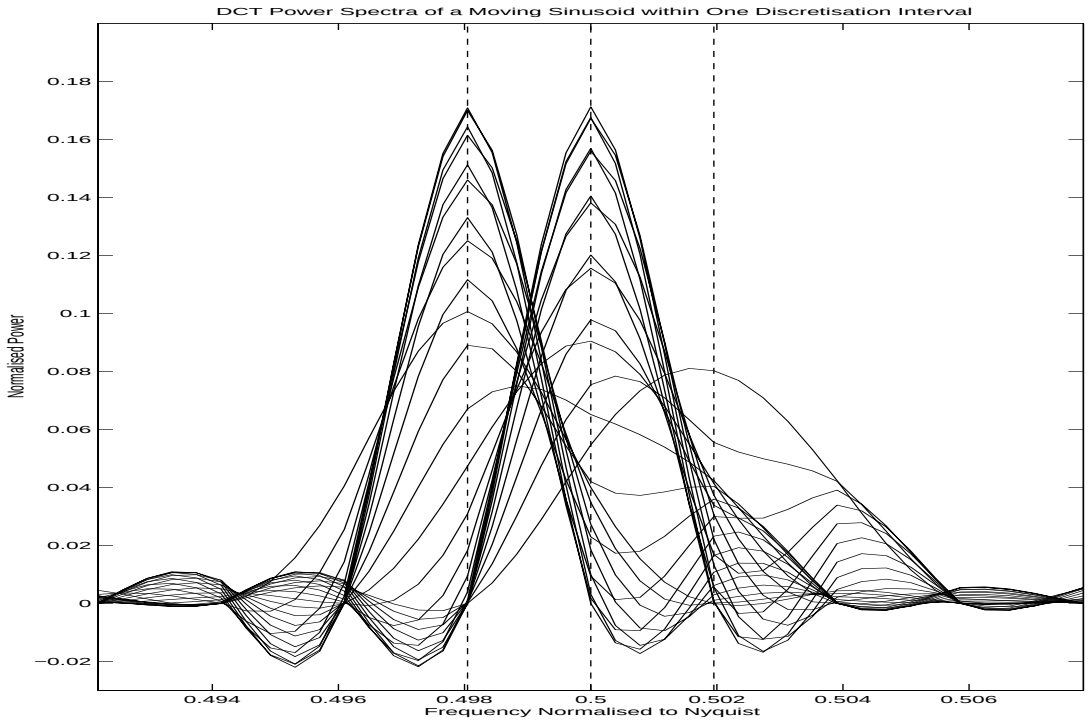

Fig. 6. DCT power spectra of a cosine signal whose frequency (around half of the Nyquist frequency) changes within Δf .

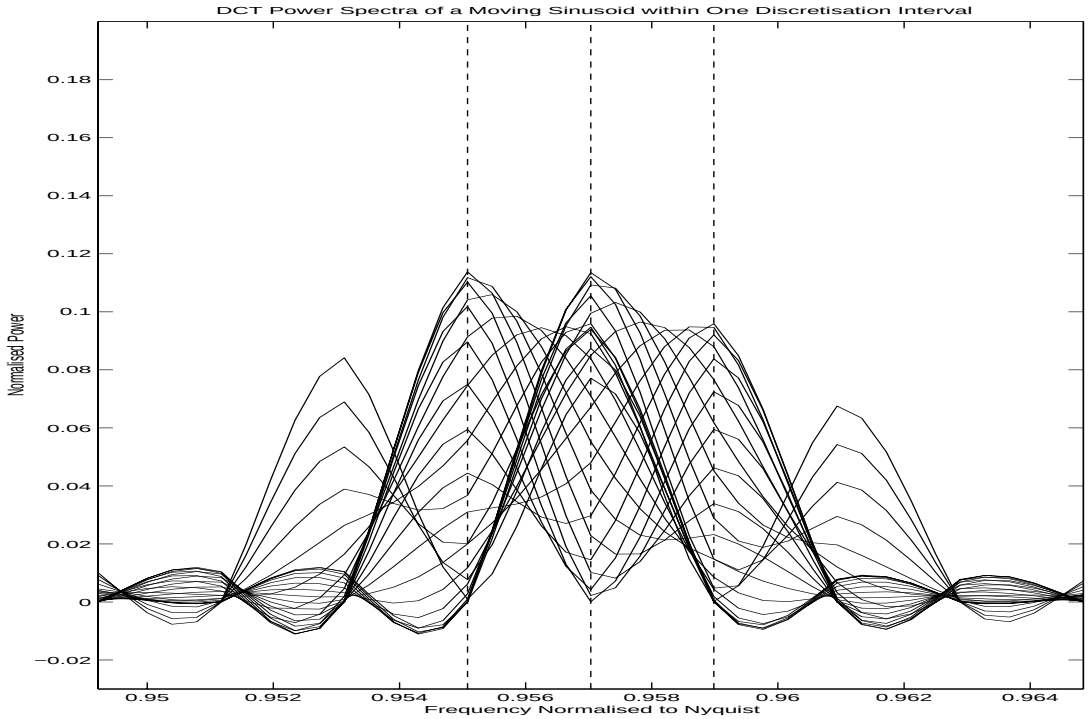

Fig. 7. DCT power spectra of a cosine signal whose frequency (slightly below the Nyquist frequency) changes within Δf . This figure clearly shows a rather poor frequency resolution due to the phase shift of the DCT basis function.

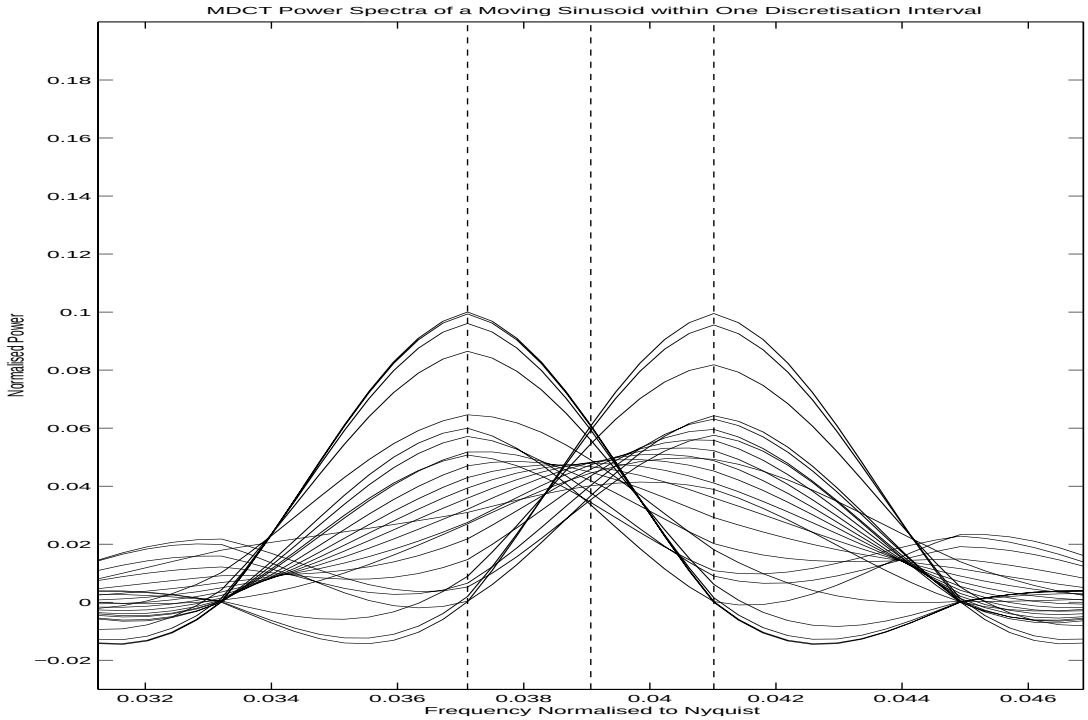

Fig. 9. MDCT power spectra of a cosine signal whose frequency (in the low frequency range) changes within Δf .

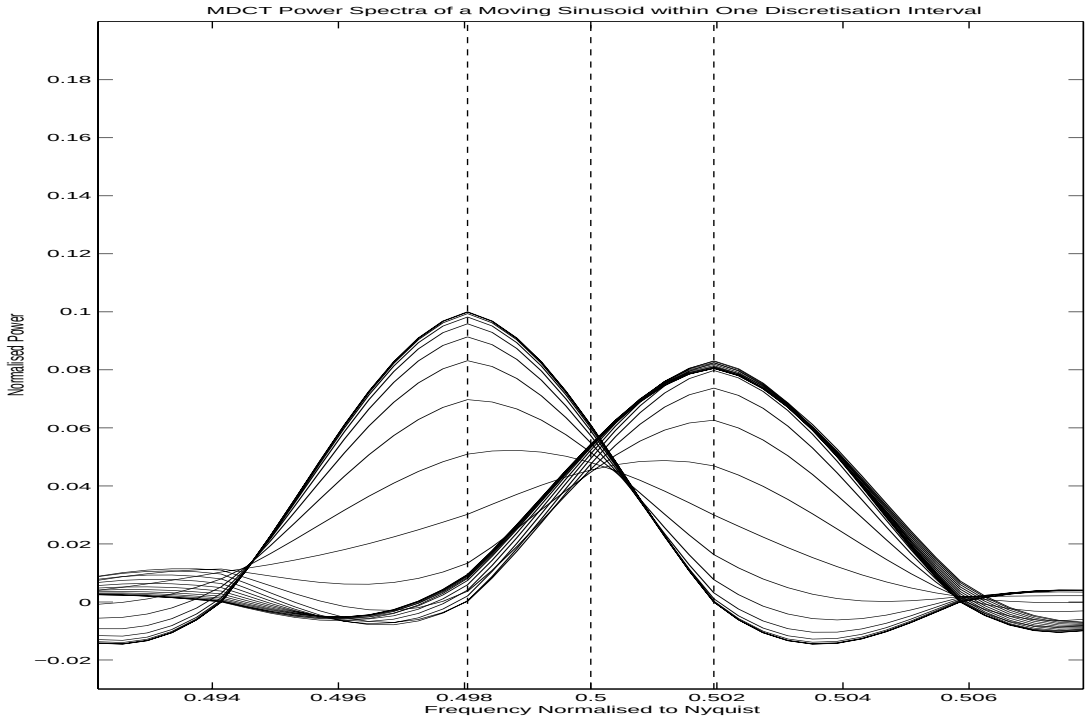

Fig. 10. MDCT power spectra of a cosine signal whose frequency (around half of the Nyquist frequency) changes within Δf .

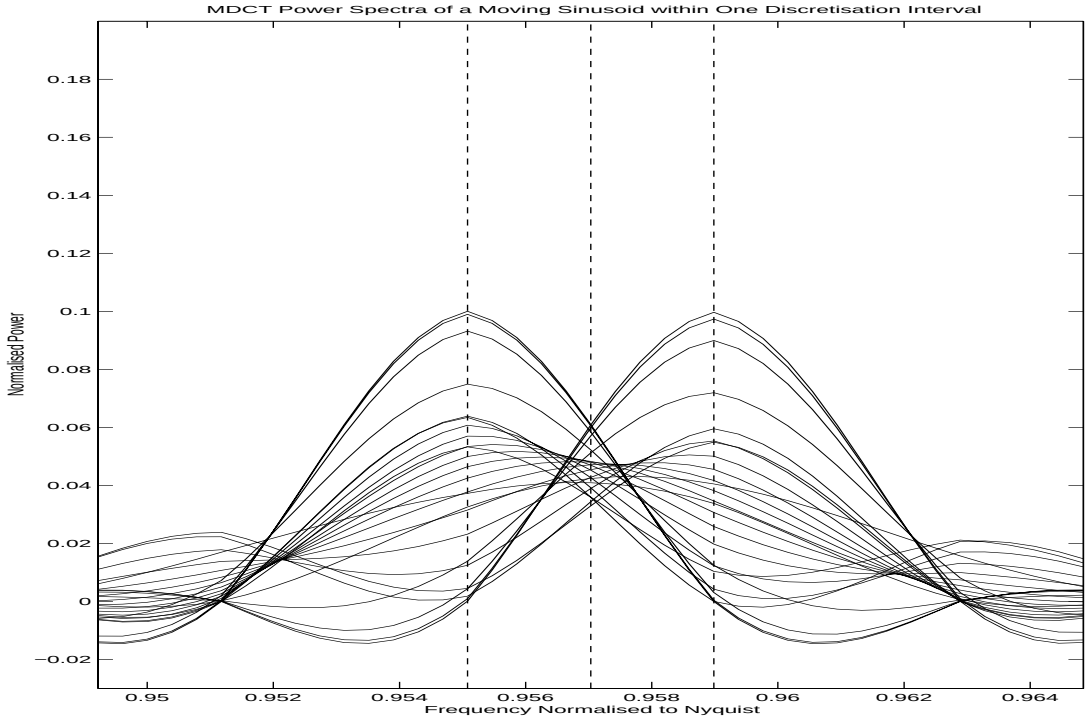

Fig. 11. MDCT power spectra of a cosine signal whose frequency (slightly below the Nyquist frequency) changes within Δf .

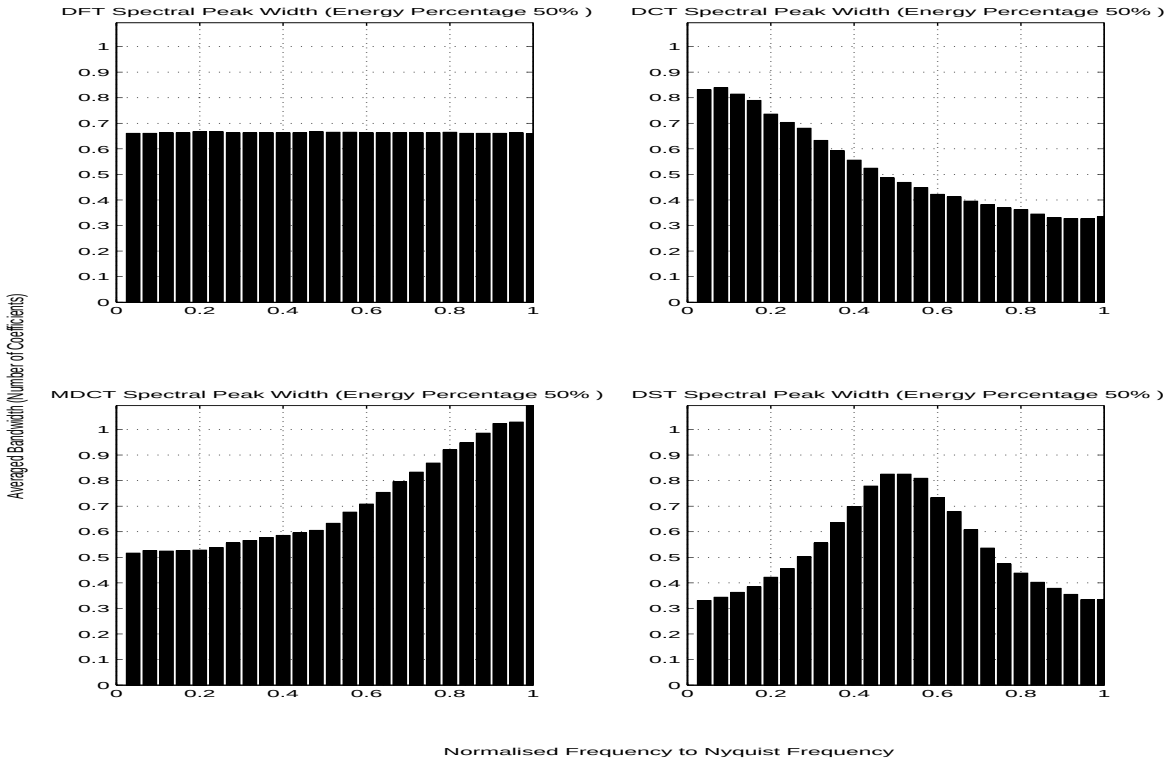

Fig. 12. Comparison of spectral resolutions (spectral peak width) of DFT, DCT, MDCT and DST using sine signals of 512 samples (rectangular window) as a function of signal frequency. The normalized power spectral threshold is 0.5, which corresponds to 50% energy within the spectral peak width.

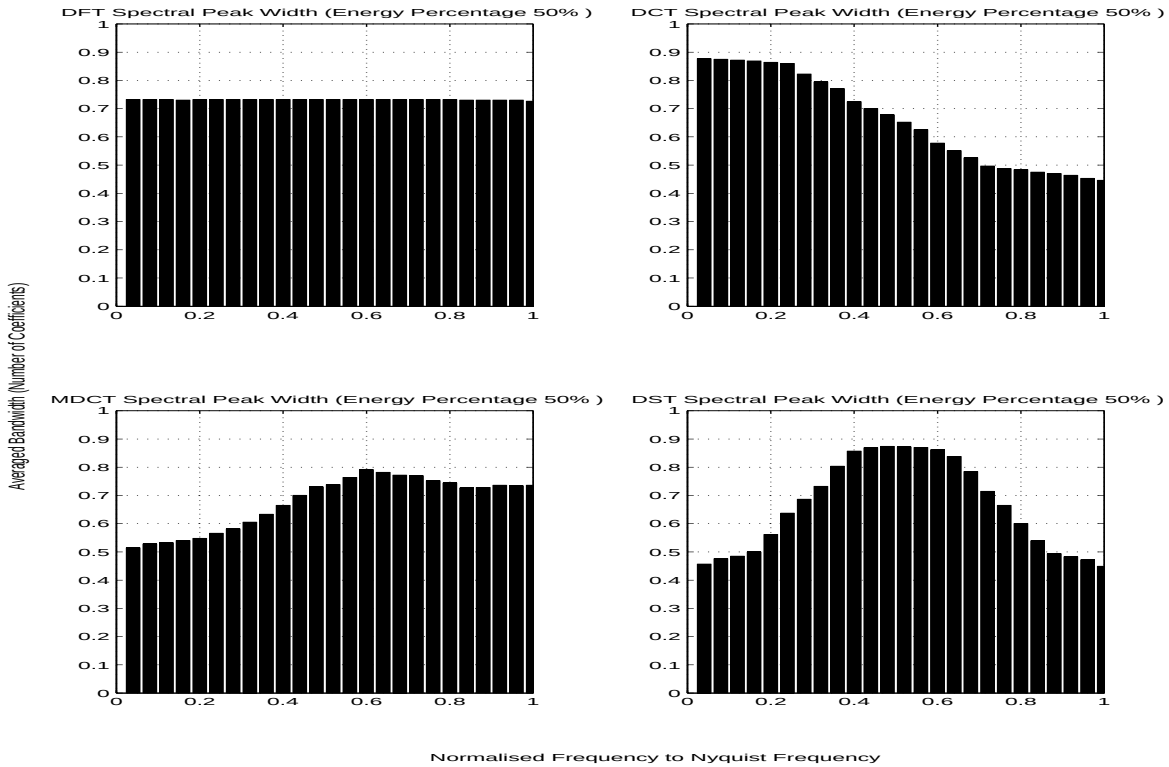

Fig. 13. Comparison of spectral resolutions of DFT, DCT, MDCT and DST using sine signals of 512 samples (sine window) as a function of signal frequency.

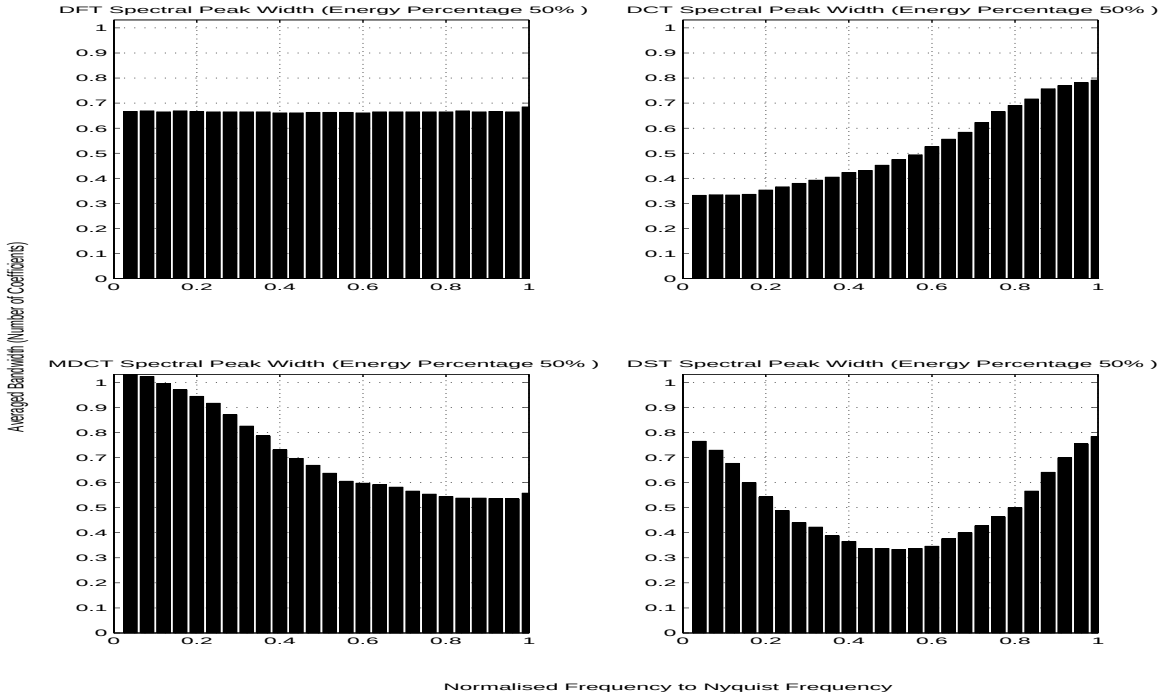

Fig. 14. Comparison of spectral resolutions (spectral peak width) of DFT, DCT, MDCT and DST using cosine signals of 512 samples (rectangular window) as a function of signal frequency.

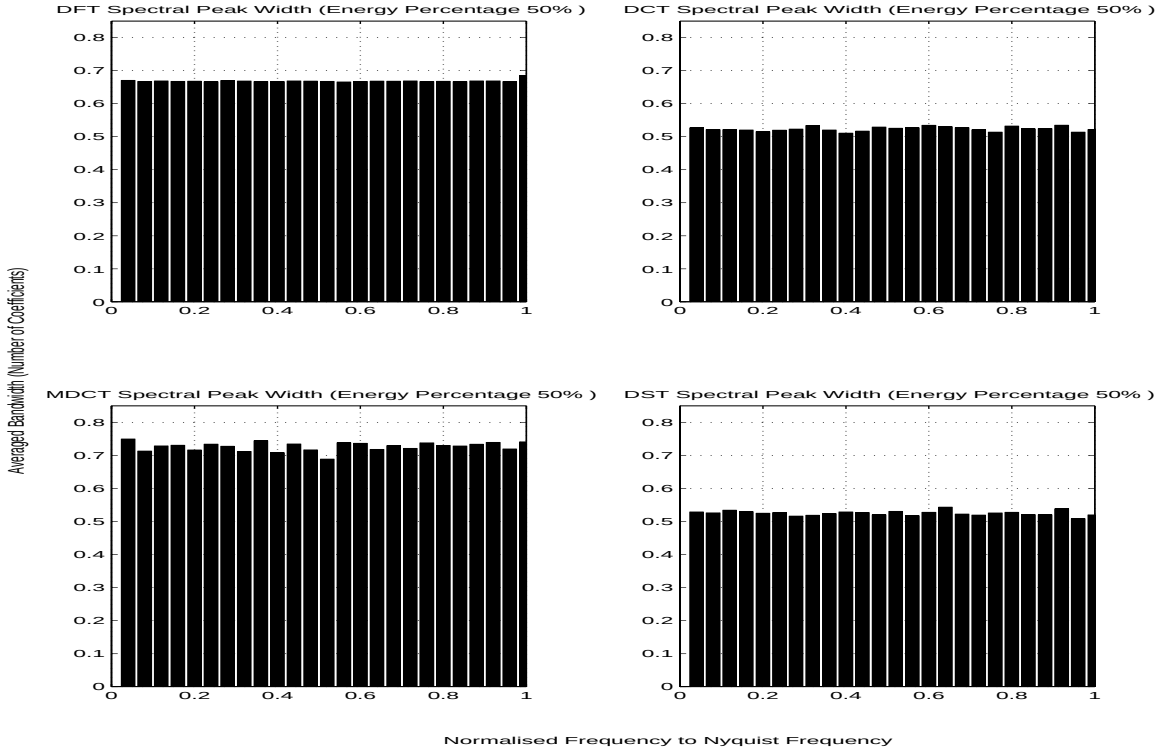

Fig. 15. Comparison of spectral resolutions (spectral peak width) of DFT, DCT, MDCT and DST using cosine signals with random phase shifts (rectangular window length = 512 samples) as a function of signal frequency.

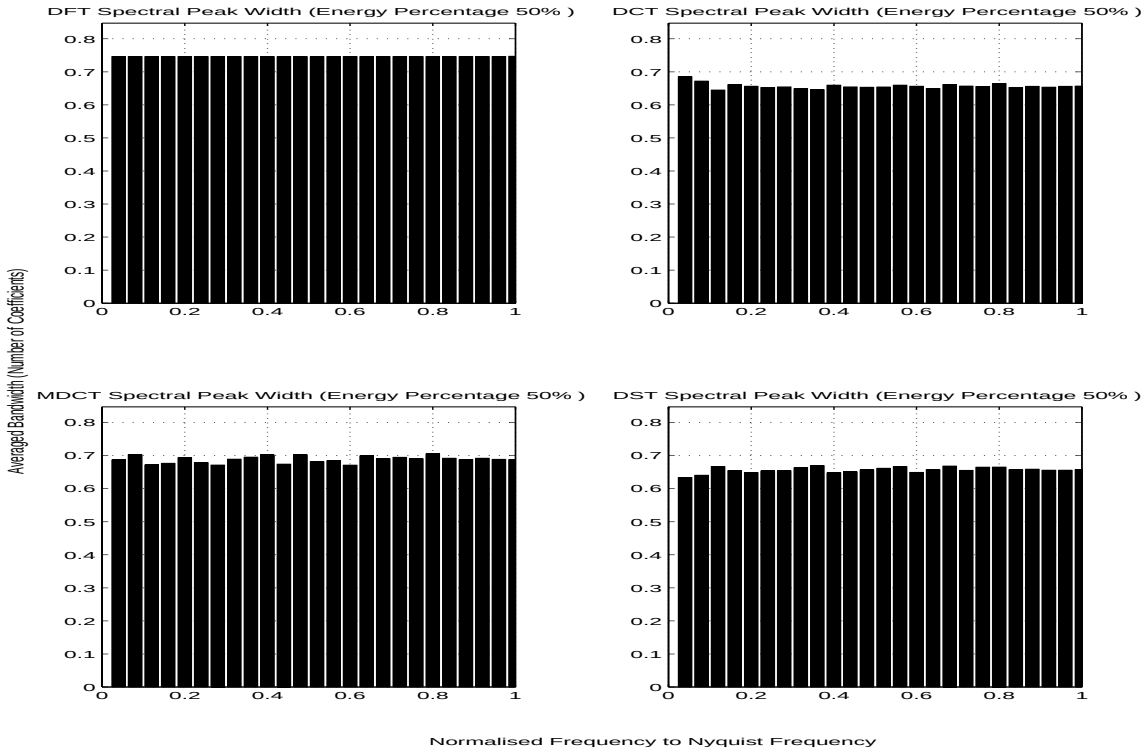

Fig. 16. Comparison of spectral resolutions (spectral peak width) of DFT, DCT, MDCT and DST using cosine signals with random phase shifts (sine window length = 512 samples) as a function of signal frequency.

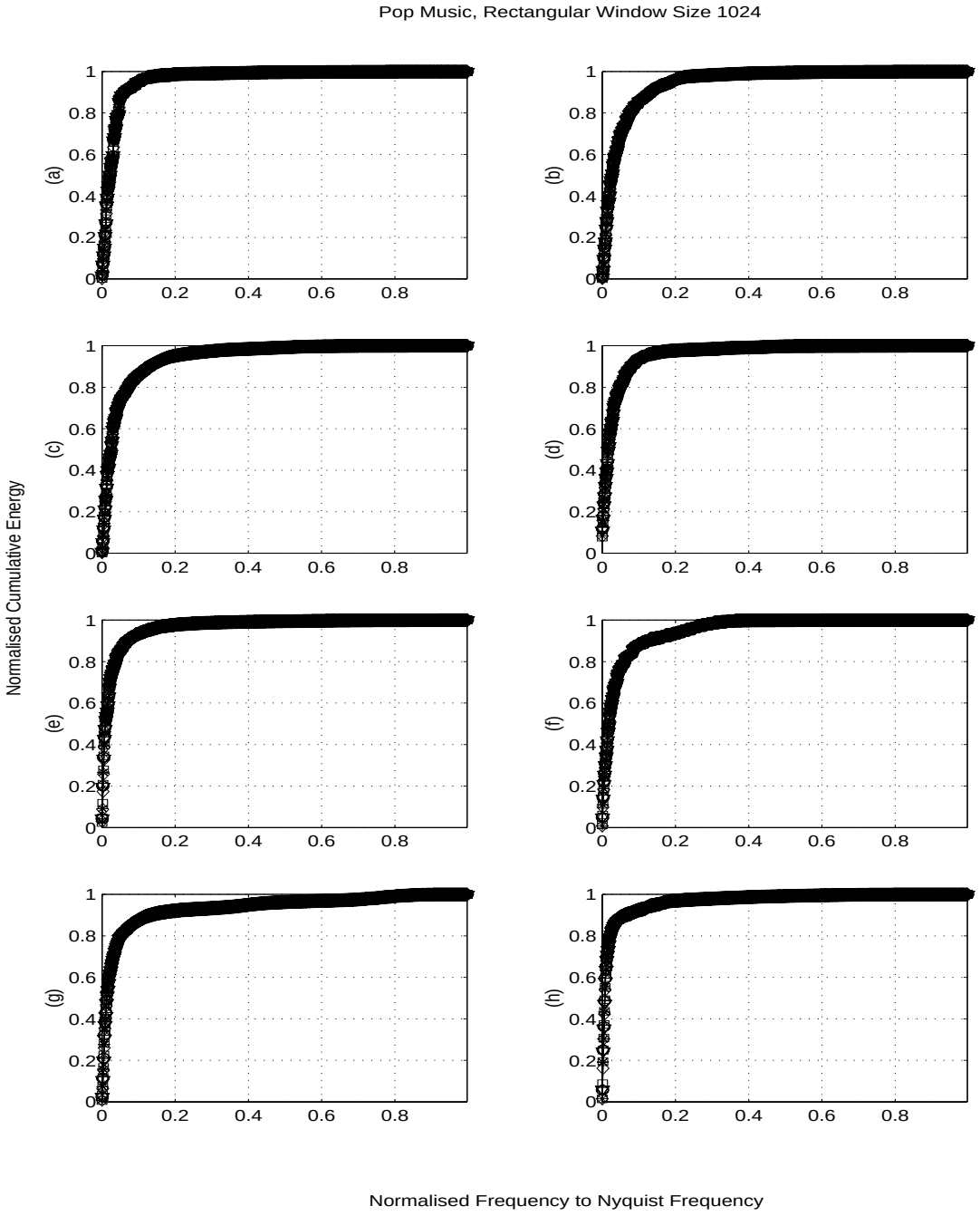

Fig. 17. Comparison of energy compaction property of DFT (star), $SDFT_{\frac{N+1}{2},\frac{1}{2}}$ (circle), DCT (diamond), MDCT (triangle) and DST (square) using 8 pieces of pop music with a rectangular window size = 1024 samples. In this scale, it is impossible to distinguish the difference between different transforms.

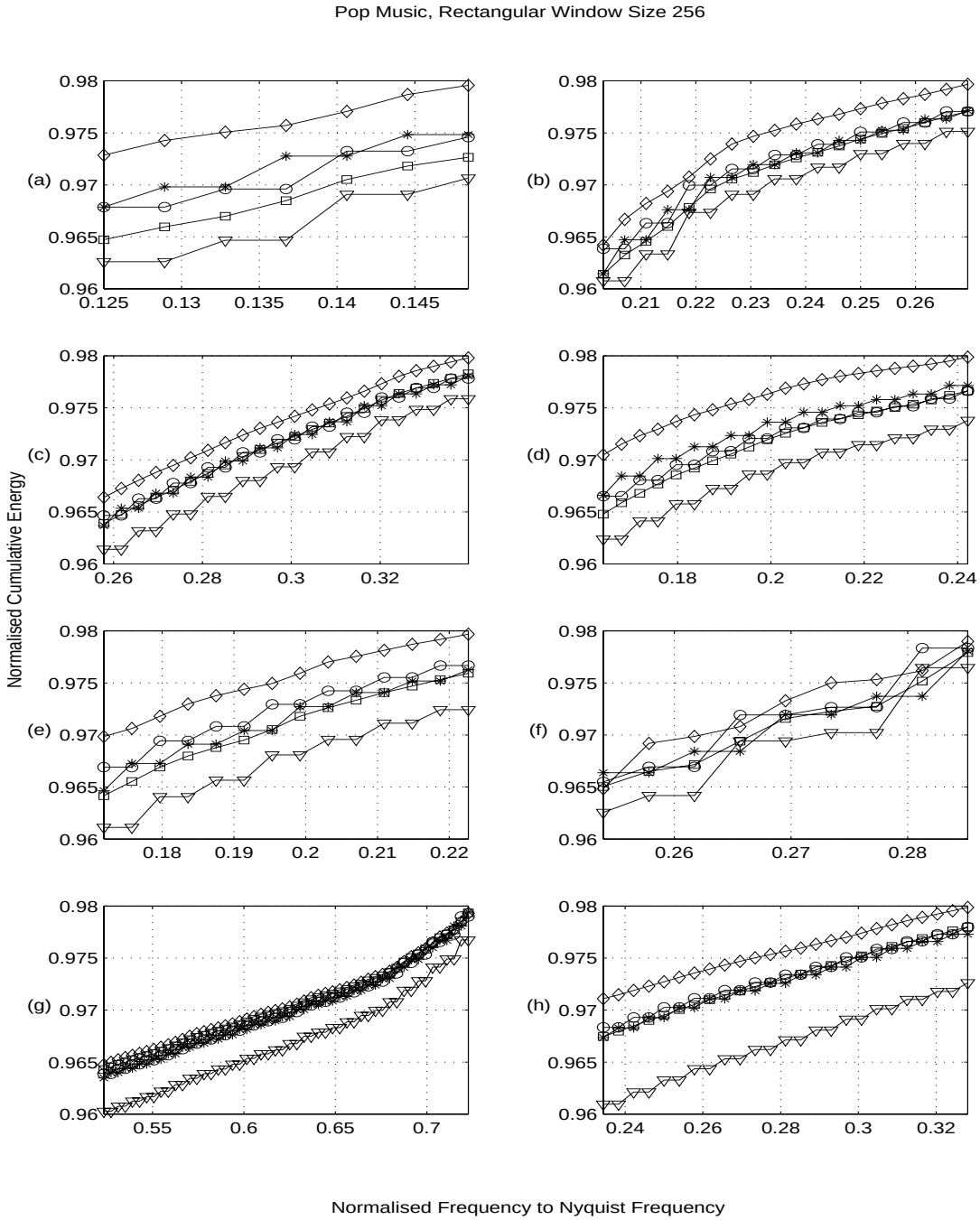

Fig. 18. Comparison of energy compaction property of DFT (star), $SDFT_{\frac{N+1}{2},\frac{1}{2}}$ (circle), DCT (diamond), MDCT (triangle) and DST (square) using 8 pieces of pop music with a rectangular window size = 256 samples. This is a zoomed version for better illustration.

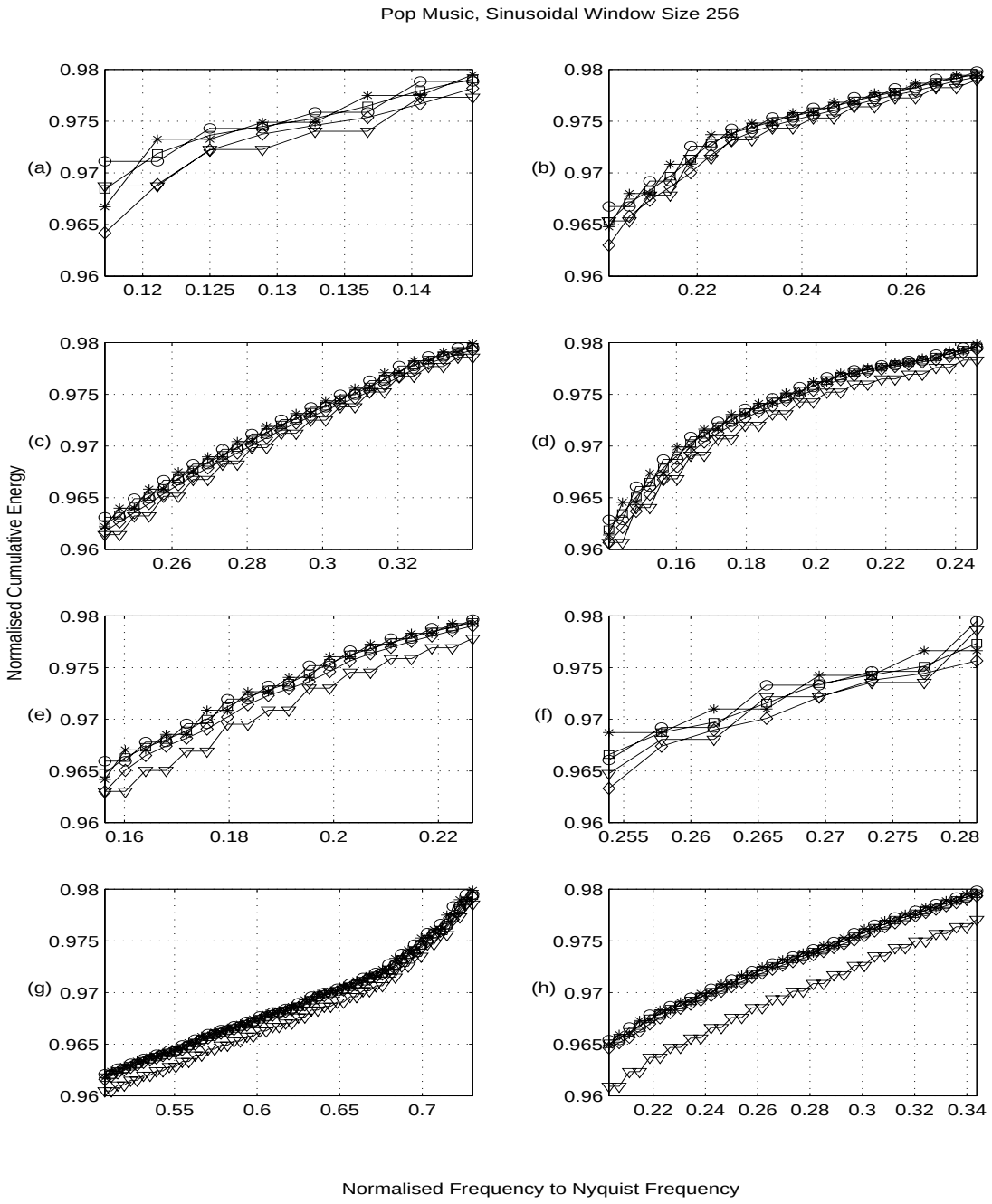

Fig. 19. Comparison of energy compaction property of DFT (star), $SDFT_{\frac{N+1}{2},\frac{1}{2}}$ (circle), DCT (diamond), MDCT (triangle) and DST (square) using 8 pieces of pop music with a sine window size = 256 samples. The sine window clearly reduces the difference between different transforms.

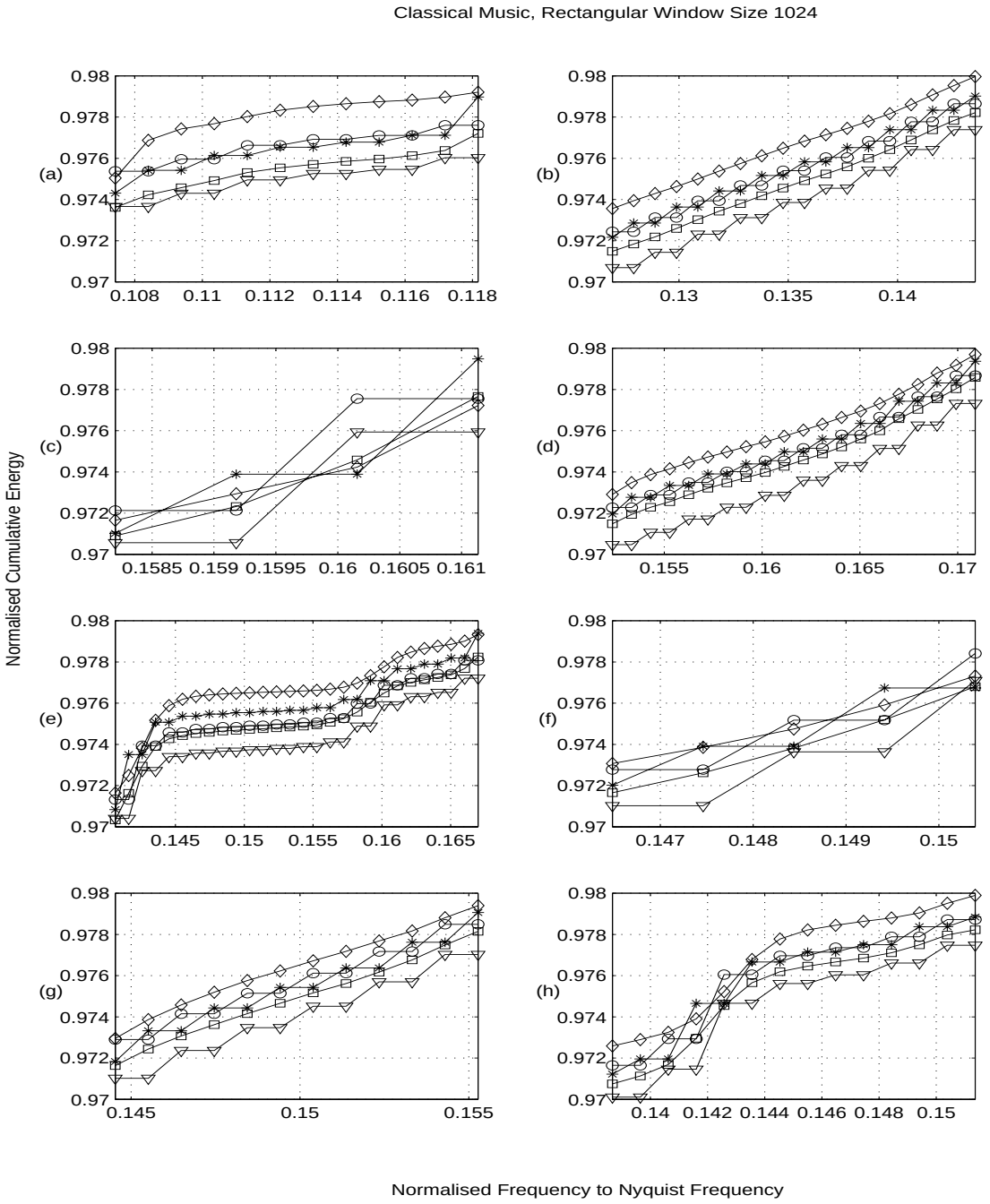

Fig. 20. Comparison of energy compaction property of DFT (star), $SDFT_{\frac{N+1}{2},\frac{1}{2}}$ (circle), DCT (diamond), MDCT (triangle) and DST (square) using 8 pieces of classic music with a rectangular window size = 1024 samples

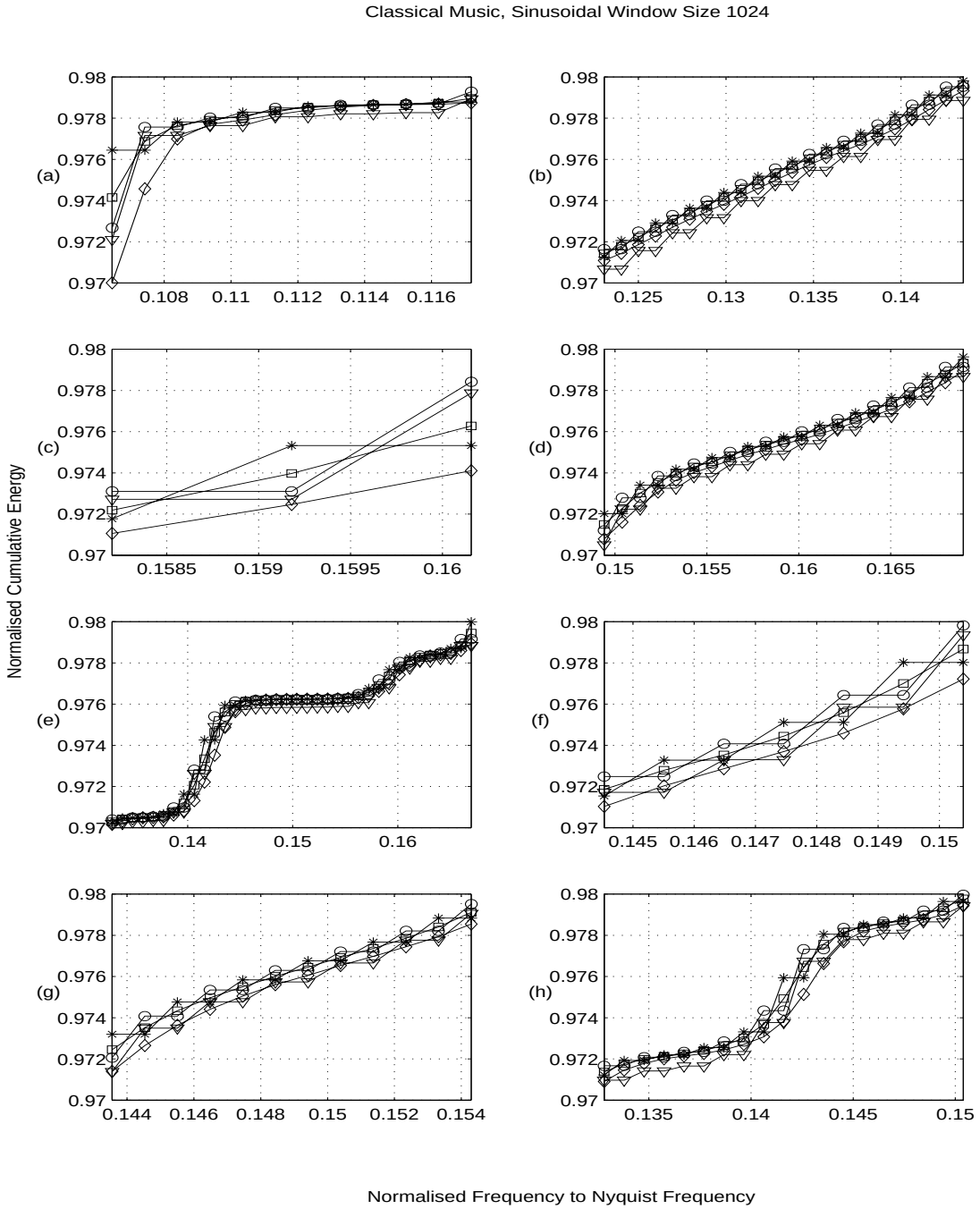

Fig. 21. Comparison of energy compaction property of DFT (star), $SDFT_{\frac{N+1}{2},\frac{1}{2}}$ (circle), DCT (diamond), MDCT (triangle) and DST (square) using 8 pieces of classic music with a sine window size = 1024 samples

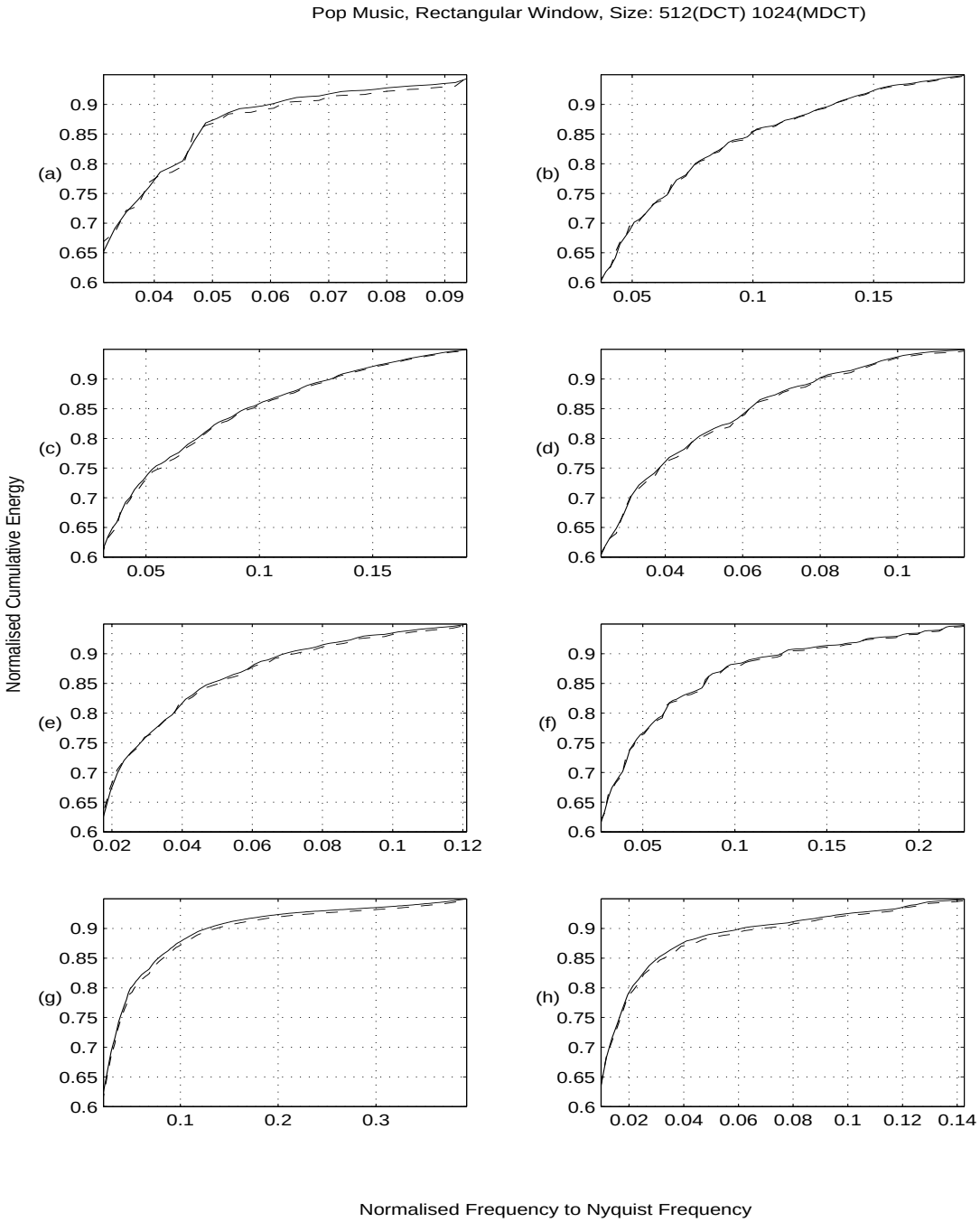

Fig. 22. Comparison of energy compaction property using 8 pieces of pop music with a rectangular window size of 512 samples for DCT (solid lines) and a rectangular window size of 1024 samples for MDCT (dashed lines).

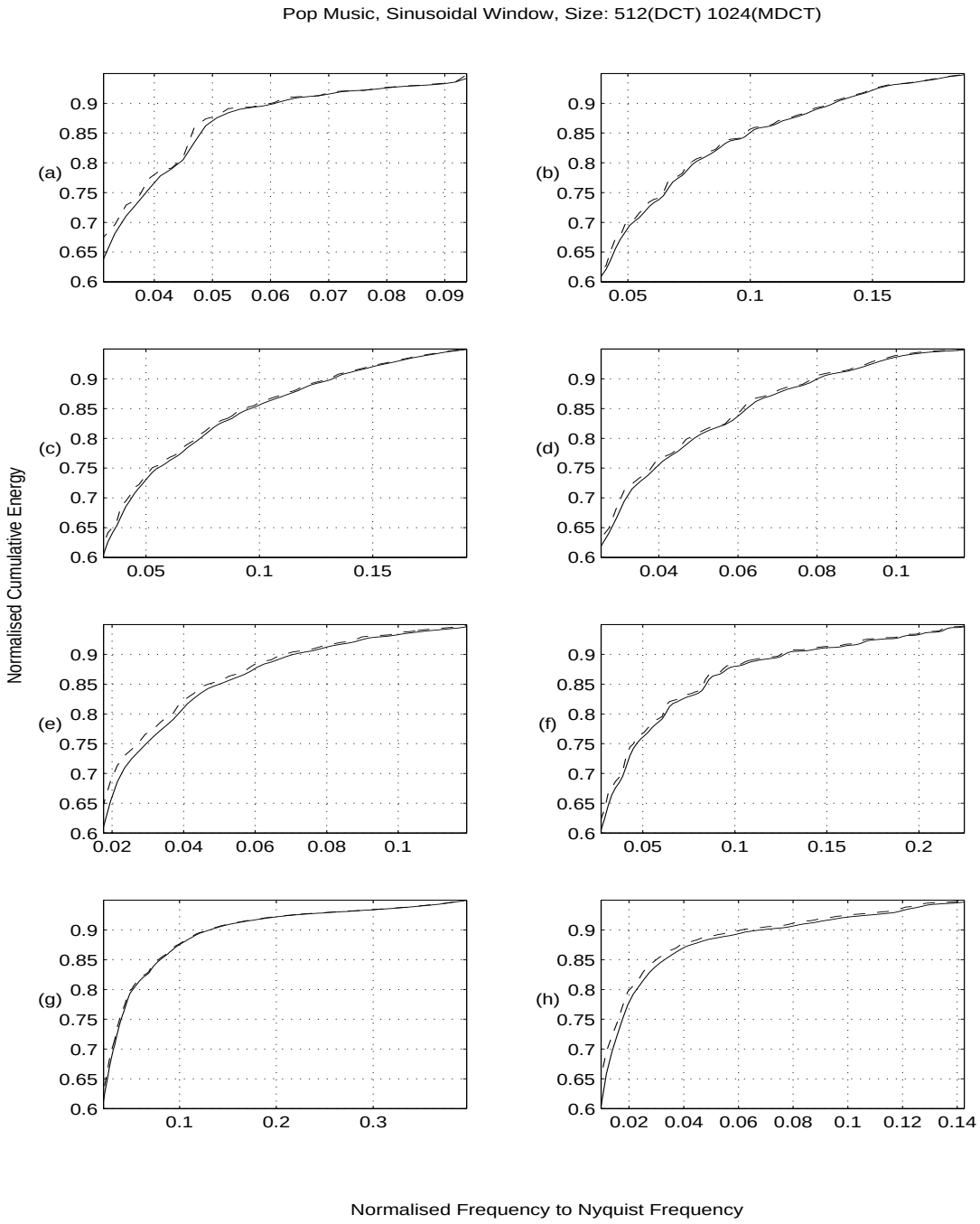

Fig. 23. Comparison of energy compaction property using 8 pieces of pop music with a sine window size of 512 samples for DCT (solid lines) and a sine window size of 1024 samples for MDCT (dashed lines).

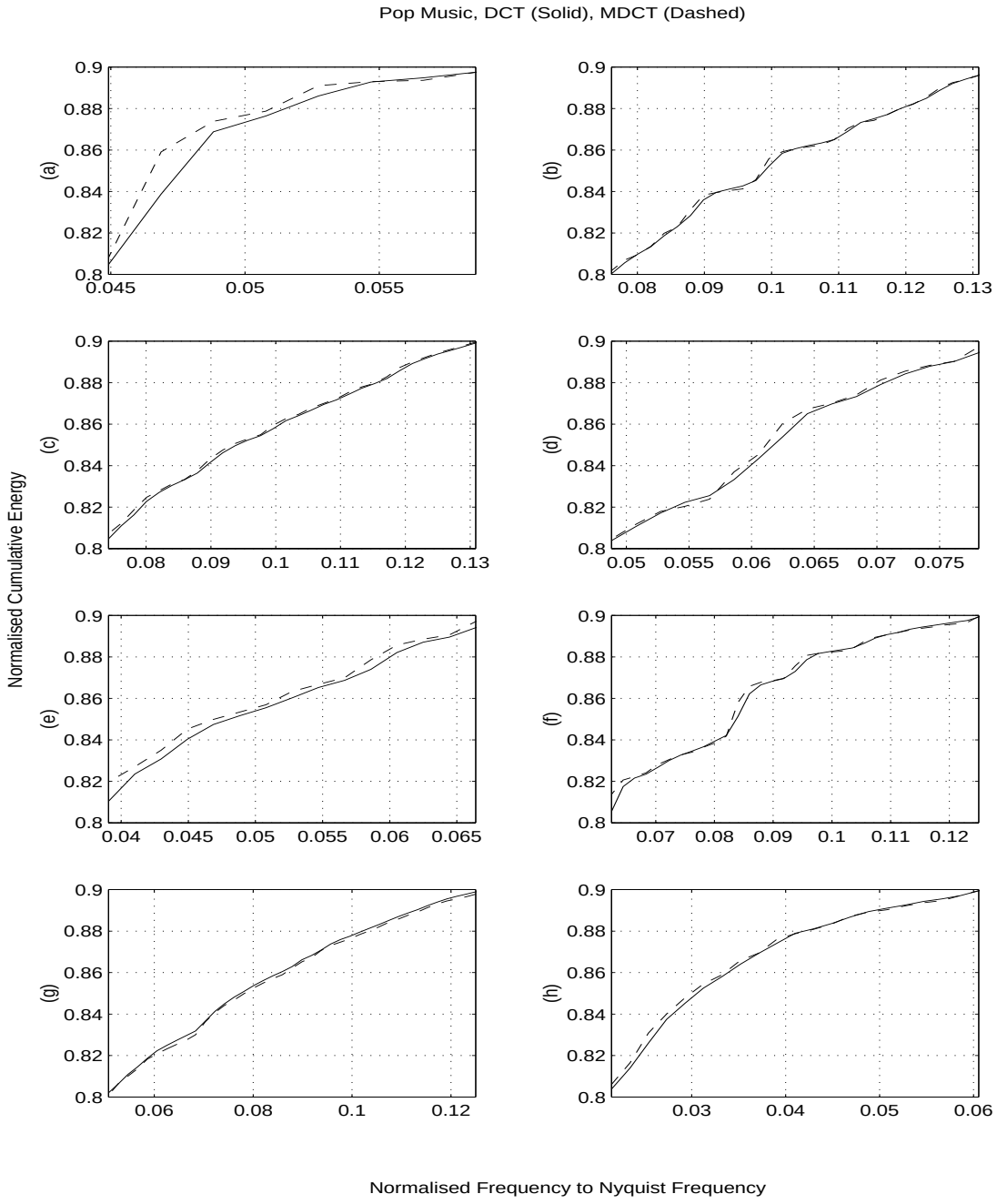

Fig. 24. Comparison of energy compaction property using 8 pieces of pop music with a rectangular window size of 512 samples for DCT (solid lines) and a sine window size of 1024 samples for MDCT (dashed lines).

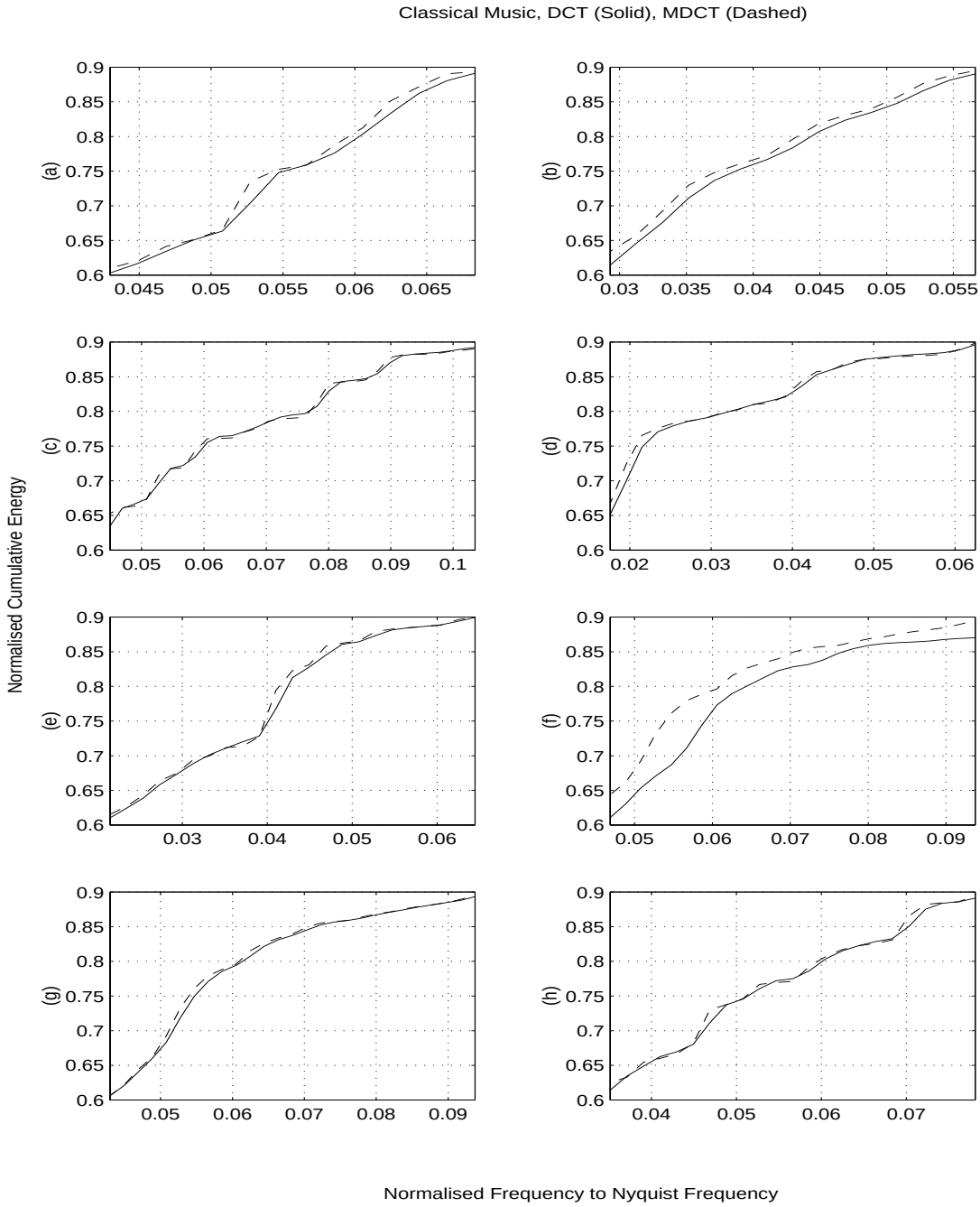

Fig. 25. Comparison of energy compaction property using 8 pieces of classic music with a rectangular window size of 512 samples for DCT (solid lines) and a sine window size of 1024 samples for MDCT (dashed lines).

[P4] Wang, Y., Vilermo, M., Isherwood, D. “The Impact of the Relationship Between MDCT and DFT on Audio Compression: A Step Towards Solving the Mismatch”, The First IEEE Pacific-Rim Conference on Multimedia (IEEE-PCM2000), December 13-15, 2000, Sydney, Australia, pp. 130-138

The Impact of the Relationship Between MDCT and DFT on Audio Compression: A Step Towards Solving the Mismatch

Ye Wang
Nokia Research Center
P.O. Box 100
FIN-33721 Tampere, Finland
ye.wang@nokia.com

Miikka Vilermo
Nokia Research Center
P.O. Box 100
FIN-33721 Tampere, Finland
miikka.vilermo@nokia.com

David Isherwood
Nokia Research Center
P.O. Box 100
FIN-33721 Tampere, Finland
david.isherwood@nokia.com

ABSTRACT

Most state-of-the-art audio encoders have two fundamental coding tools: a MDCT and DFT based psychoacoustic model. The MDCT coefficients are quantized according to the masking threshold calculated by the psychoacoustic model. However, this kind of encoder structure can fail for some test signals. Research has been undertaken to find the reasons behind this failure, during which it has been found that the failure may be caused by the peculiar properties of MDCT and the mismatch between MDCT and a DFT based psychoacoustic model. We have established a direct and compact formulation of the MDCT with the help of a *Shifted* Discrete Fourier Transform (SDFT). This formulation has a clear physical interpretation. It enables us to i) clarify the symmetric properties of MDCT, ii) to illustrate the Time Domain Alias Cancellation (TDAC) concept in a very intuitive and illustrative way, and iii) to show some peculiar properties of MDCT, which may affect the coding performance of a MDCT based audio codec. Based on these new interconnections we propose a new encoder structure as a first step towards solving the mismatch. A small, formal listening test was initiated to verify the relative performance of our optimized codec. The mismatch between the two fundamental coding tools is relevant for multimedia codec design, watermark embedding, etc. The improvement of the computational efficiency discussed in this paper is essential for hand-held devices such as mobile phones.

Keywords

Modified Discrete Cosine Transform (MDCT), Time Domain Alias Cancellation (TDAC), Shifted Discrete Fourier Transform (SDFT), Discrete Fourier Transform (DFT), mismatch and audio coding.

1. INTRODUCTION

Audio signal representation and the human auditory system perceptual model are two fundamental tools of audio coding. Signal representation in the Modified Discrete Cosine Transform (MDCT) domain has emerged as a dominant tool in high quality audio coding because of its special properties: in addition to the energy compaction capability similar to DCT, MDCT combines critical sampling, reduction of block effect and flexible window switching. However, auditory system perceptual models are often based on the Fourier transform domain implemented by means of DFT [1]. Using the masking curve of a DFT based psychoacoustic model to quantise MDCT coefficients is problematic in some special cases. This may be one reason behind the failure of MDCT based audio codecs with certain test signals. In order to gain improved understanding of the mismatch between DFT and MDCT, we have recently studied the interconnections between DFT and MDCT via SDFT [2][3], and discovered some peculiar properties of MDCT [4]. Illustrative examples presented in this paper will help the readers to understand the Time Domain Alias Cancellation (TDAC) concept of the MDCT. Our new formulation of MDCT via SDFT also clarifies the symmetric property as well as some other peculiar properties of the MDCT. The mismatch between the two fundamental coding tools exists in most perceptual encoders (including image and video coding), and has not been adequately addressed in literature. A good understanding of the interaction between these two fundamental tools in audio coding may lead us to further advances in coding performance. It may help us to design an improved watermarking algorithm in combination with compression.

The complex version of the MDCT has been investigated in [5][6][7] in terms of filterbank theory. Our research has approached the problem from a different perspective: Fourier spectrum analysis. We believe that the analysis presented in this paper is an elegant and hopefully clearer clarification of the often-confusing concept of MDCT and TDAC.

In this paper we present a bridge between the MDCT and DFT via SDFT((N+1)/2,1/2) in chapter 2. We then present some peculiar properties of MDCT and the TDAC concept in a very intuitive and illustrative way in chapter 3. A new SDFT((N+1)/2, 1/2) based audio encoder structure is proposed as a further step towards solving the mismatch and to improve computational

efficiency in chapter 4. Experimental results are discussed in chapter 5. Concluding remarks are in chapter 6.

2. THE INTERCONNECTION BETWEEN MDCT, SDFT AND DFT

The direct and inverse MDCT are defined as [8][9]:

$$\alpha_r = \sum_{k=0}^{2N-1} \tilde{a}_k \cos\left[\pi \frac{(k+(N+1)/2)(r+1/2)}{N}\right], \quad (1)$$

$$r = 0,..., N-1$$

$$\hat{a}_k = \frac{1}{N}\sum_{r=0}^{N-1} \alpha_r \cos\left[\pi \frac{(k+(N+1)/2)(r+1/2)}{N}\right], \quad (2)$$

$$k = 0,...,2N-1$$

where $\tilde{a}_k = h_k a_k$ is the windowed input signal, a_k is the input signal of $2N$ samples. h_k is a window function. We assume an identical analysis-synthesis time window. The constraints of perfect reconstruction are [5][7]:

$$h_k = h_{2N-1-k} \quad (3)$$

$$h_k^2 + h_{k+N}^2 = 1 \quad (4)$$

A sine window is widely used in audio coding because it offers good stop-band attenuation, provides good attenuation of the block edge effect and allows perfect reconstruction. Other optimized windows can be applied as well [5]. The sine window is defined as:

$$h_k = \sin\left[\pi (k+1/2)/2N\right], \quad (5)$$

$$k = 0,...,2N-1$$

$\hat{a}_k$ in (2) are the IMDCT coefficients of α_r, which contains time domain aliasing:

$$\hat{a}_k = \begin{cases} \frac{1}{2}\tilde{a}_k - \frac{1}{2}\tilde{a}_{N-1-k}, & k = 0,...,N-1 \\ \frac{1}{2}\tilde{a}_k + \frac{1}{2}\tilde{a}_{3N-1-k}, & k = N,...,2N-1 \end{cases} \quad (6)$$

The relationship between MDCT and DFT can be established via Shifted Discrete Fourier Transforms (SDFT). The direct and inverse SDFTs are defined as [10]:

$$\alpha_r^{u,v} = \sum_{k=0}^{2N-1} a_k \exp\left[i2\pi(k+u)(r+v)/2N\right], \quad (7)$$

$$a_k^{u,v} = \frac{1}{2N}\sum_{r=0}^{2N-1} \alpha_r^{u,v} \exp\left[-i2\pi(k+u)(r+v)/2N\right], \quad (8)$$

where u and v represent arbitrary time and frequency domain shifts respectively. SDFT is a generalization of DFT that allows a possible arbitrary shift in position of the samples in the time and frequency domain with respect to the signal and its spectrum coordinate system.

We have proven that the MDCT is equivalent to the SDFT of a modified input signal [2][3].

$$\alpha_r = \sum_{k=0}^{2N-1} \hat{a}_k \exp\left[i2\pi \frac{(k+(N+1)/2)(r+1/2)}{2N}\right] \quad (9)$$

The right side of (9) is $SDFT_{(N+1)/2,1/2} = \left(\alpha_r^{(N+1)/2,1/2}\right)$ of the signal $\hat{a}_k$ formed from the initial windowed signal $\tilde{a}_k$ according to (6). Physical interpretation of (6) is straightforward. MDCT coefficients can be obtained by adding the $SDFT_{(N+1)/2,1/2}$ coefficients of the initial windowed signal and the alias. In other words, we can rewrite (9) as:

$$MDCT(signal) = \frac{1}{2}SDFT_{(N+1)/2,1/2}(signal) + \frac{1}{2}SDFT_{(N+1)/2,1/2}(alias) \quad (10)$$

α_r in (1) is expressed as $MDCT(signal)$ in (10) for the sake of explicitness. With reference to (6)(9) and Figure 1(f), the alias is added to the original signal in such a way that the first half of the window (the signal portion between points A and B) is mirrored in the time domain and then inverted, before being subsequently added to the original signal. The second half of the window (signal portion between points B and C) is also mirrored in the time domain and added to the original signal.

From (1)(2)(6)(9) and Figure 1(f) we can see that, in comparison with conventional orthogonal transforms, MDCT has a special property: the input signal cannot be perfectly reconstructed from the MDCT coefficients, even without quantization. MDCT itself is a lossy process (therefore not an orthogonal transform). That is, the imaginary coefficients of the $SDFT_{(N+1)/2,1/2}$ are lost in the MDCT transform. However, the lost information can be recovered using the redundancy of the 50% overlap of neighboring frames to gain perfect reconstruction. Applying a MDCT and then an IMDCT converts the input signal into one that contains a certain twofold symmetric alias (see (6) and Figure 1(f)). The introduced alias will be cancelled in the overlap-add process (see Figure 5).

In comparison with the Odd-DFT concept discussed in [5], the formulation in (9) is clearly different. The Odd-DFT is $SDFT_{0,1/2}$ of the initial windowed signal $\tilde{a}$.

The $SDFT_{(N+1)/2,1/2}$ can be expressed by means of the conventional DFT as:

$$\sum_{k=0}^{2N-1} \hat{a}_k \exp\left[i2\pi \frac{(k+(N+1)/2)(r+1/2)}{2N}\right] = \left\{\sum_{k=0}^{2N-1}\left[\hat{a}_k \exp\left(i2\pi\frac{k}{4N}\right)\right]\exp(i2\pi\frac{kr}{2N})\right\}\exp\left(i2\pi\frac{(N+1)r}{4N}\right)\exp\left(i\pi\frac{N+1}{4N}\right) \quad (11)$$

To the right side of (11), the first exponential function corresponds to a modulation of $\hat{a}$ that results in a signal spectrum shift in the frequency domain by ½ of the frequency-sampling interval. The second exponential function corresponds to the conventional DFT. The third exponential function modulates the signal spectrum that is equivalent to a signal shift by (N+1)/2 of the sampling interval in the time domain. The fourth term is a constant phase shift. Therefore, $SDFT_{(N+1)/2,1/2}$ is the conventional DFT of this signal shifted in the time domain by (N+1)/2 of the sampling interval and evaluated with the shift of ½ the frequency-sampling interval.

3. THE CONCEPT OF TIME DOMAIN ALIAS CANCELLATION AND SOME PERCULIAR PROPERTIES OF THE MDCT

3.1 Symmetric Properties of MDCT

$SDFT_{(N+1)/2,1/2}$ coefficients exhibit symmetric properties:

$$\alpha_{2N-r-1}^{(N+1)/2,1/2} = (-1)^{N+1}\left(\alpha_r^{(N+1)/2,1/2}\right)^* \quad (12)$$

where $*$ is the complex conjugate of the coefficients.

Similarly, MDCT coefficients exhibit symmetric properties:

$$\alpha_{2N-r-1} = (-1)^{N+1}\alpha_r \quad (13)$$

whereby, the MDCT coefficients are odd symmetric, only if N is even, which is often true in audio coding applications. However, they are even symmetric, if N is odd. This new conclusion is more general in comparison with [11].

We have proved that:

$$IMDCT(\alpha_r) = ISDFT_{(N+1)/2,1/2}(\alpha_r) \quad (14)$$

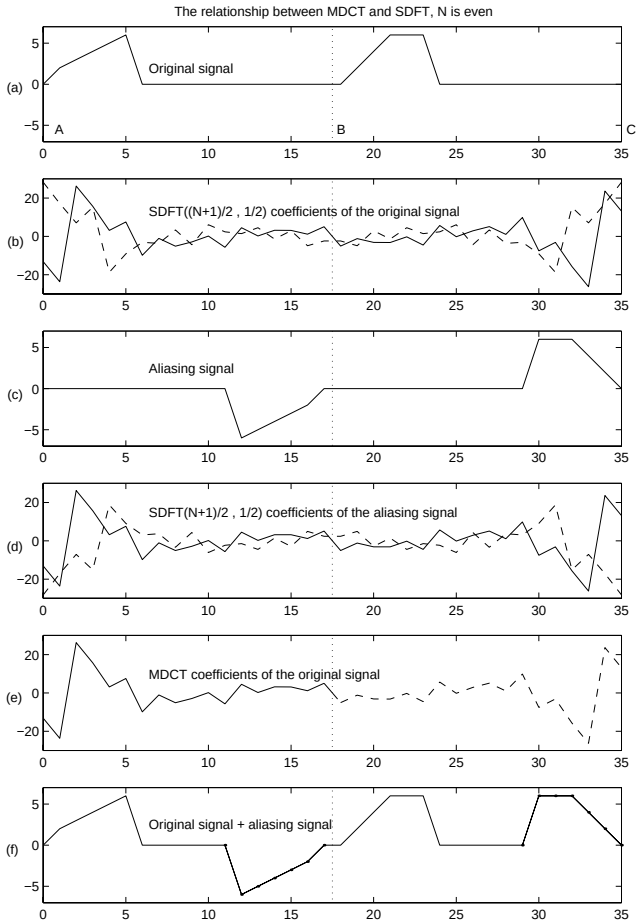

Figure 1. Illustration of the interconnection between MDCT and $SDFT_{(N+1)/2,1/2}$. (a) an artificial time domain signal of 36 samples; (b) $SDFT_{(N+1)/2,1/2}$ coefficients of the signal in (a); (c) The time domain alias; (d) $SDFT_{(N+1)/2,1/2}$ coefficients of the alias, where the solid lines are the real parts, the dotted lines are the imaginary parts in both (b) and (d); (e) MDCT coefficients of the time signal in (a), where the dotted line is odd symmetric to the solid line, and therefore it is redundant; (f) the alias embedded time signal.

In order to illustrate the symmetric properties of MDCT and the interconnection between MDCT and $SDFT_{(N+1)/2,1/2}$ in an intuitive way, we have employed two artificial time domain signals (N=18, 17 respectively) as shown in Figure 1(a) and Figure 2(a). The $SDFT_{(N+1)/2,1/2}$ coefficients of the original signals are shown in Figure 1(b) and Figure 2(b). The time domain alias is illustrated in Figure 1(c) and Figure 2(c). Its $SDFT_{(N+1)/2,1/2}$ coefficients are presented in Figure 1(d), Figure 2(d). In both Figure 1(b, d) and Figure 2(b, d) the solid lines are the real parts, the dashed lines are the imaginary parts. The MDCT coefficients are shown in Figure 1(e) and Figure 2(e). They are equivalent to the real parts of the $SDFT_{(N+1)/2,1/2}$ coefficients of the original signals in Figure 1(a) and Figure 2(a). The dashed lines in Figure 1(e) and Figure 2(e) are odd/even symmetric to the solid lines, and these dashed lines represent the redundant coefficients, which are left out in the MDCT definition. The alias embedded time signal is presented in Figure 1(f) and Figure 2(f). It equals the IMDCT of the MDCT coefficients scaled by factor two.

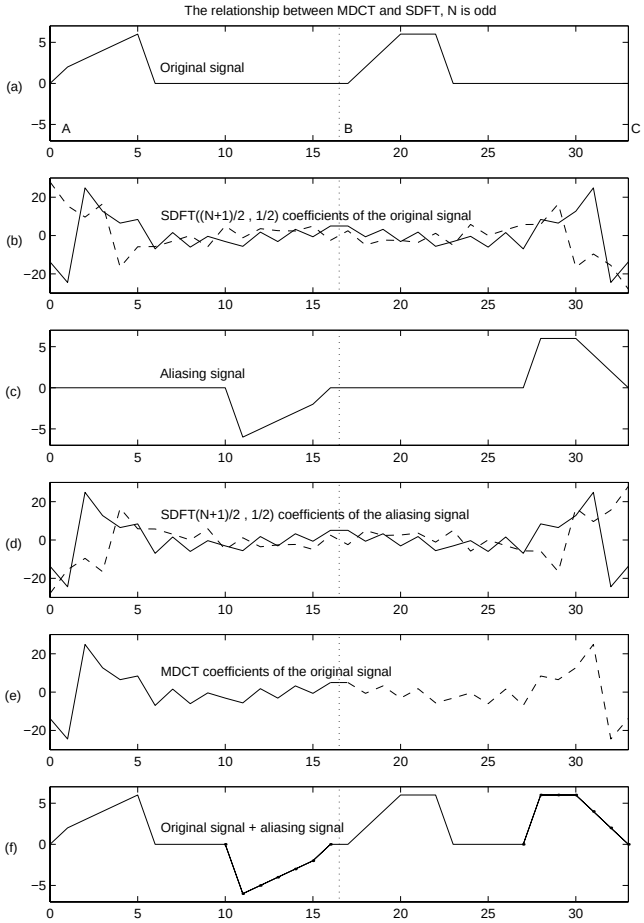

Figure 2. Illustration of the interconnection between MDCT and $SDFT_{(N+1)/2,1/2}$. (a) An artificial time domain signal of 34 samples; (b) $SDFT_{(N+1)/2,1/2}$ coefficients of the signal in (a); (c) The time domain alias; (d) $SDFT_{(N+1)/2,1/2}$ coefficients of the alias, where the solid lines are the real parts, the dotted lines are the imaginary parts in both (b) and (d); (e) MDCT coefficients of the time signal in (a), where the dashed line is even symmetric to the solid line, and therefore it is redundant; (f) the alias embedded time signal, which equals the IMDCT of the MDCT coefficients scaled by factor two.

3.2 Non-Orthogonal Properties of MDCT

The MDCT differs somewhat from orthogonal transforms used for signal coding. The main peculiar properties of MDCT are:

- MDCT is not an orthogonal transform. Perfect signal reconstruction can be achieved in the overlap-add (OA) process. For the overlap-add window of 2*N* samples, the first *N* and last *N* samples of the signal will remain modified according to (6). One can easily see this from the fact that performing MDCT and IMDCT of an arbitrary signal $\tilde{a}$ reconstructs the signal $\hat{a}$ defined in (6).
- If a signal exhibits local symmetry such that
$$\begin{cases} \tilde{a} = \tilde{a}_{N-1-k}, & k = 0,...,N-1 \\ \tilde{a} = -\tilde{a}_{3N-1-k}, & k = N,...,2N-1 \end{cases}, \quad (15)$$
its MDCT degenerates to zero: $\alpha_r = 0$ for $r = 0,...,N-1$. This property follows from (6). This is a good example that MDCT does not fulfill Parseval's theorem, i.e. the time domain energy is not equal to the frequency domain energy (see Figure 3).
- If a signal exhibits local symmetry such that
$$\begin{cases} \tilde{a}_k = -\tilde{a}_{N-k-1}, & k = 0,...,N-1 \\ \tilde{a}_k = \tilde{a}_{3N-k-1}, & k = N,...,2N-1 \end{cases}, \quad (16)$$
MDCT and IMDCT will perfectly reconstruct the original time domain samples. This property also follows from (6).
- Nevertheless, on average, MDCT, similar to such orthogonal transforms as DFT, DCT, DST, etc, possesses energy compaction capability and acceptable Fourier spectrum analysis.

In order to illustrate the special characteristics of the MDCT and their impact on audio coding in an intuitive way, we have designed a phase/frequency-modulated time signal in Figure 3(a), which has two different frequency elements with the duration of half of the frame size (frame size = 512 samples). Dashed lines in Figure 3 (a) illustrate the 50% window overlap. However, MDCT spectra of different time slots in Figures 3(b)(d)(f) are calculated with rectangular windows for simplicity. The IMDCT time domain samples of frame 1, 2, 3 are shown in Figures 3(c)(e)(g) respectively. The reconstructed time domain samples after overlap-add (OA) procedure is shown in Figure 3(h). With frame 2 the condition in (15) holds, and the MDCT coefficients are all zero! Nevertheless, the time domain samples in frame 2 can still be perfectly reconstructed after the overlap-add procedure. With frame 3 the condition (16) holds, and the original time samples are perfectly reconstructed even without overlap-add procedure. These are, of course, very special occurrences, which are rare in real life audio signals. If the signal is close to the condition in (15) however, MDCT spectrum will be very unstable in comparison with DFT spectrum. In this case, using the output of the DFT based psychoacoustic model to quantise MDCT coefficients will not be logical. This is an important limitation of MDCT.

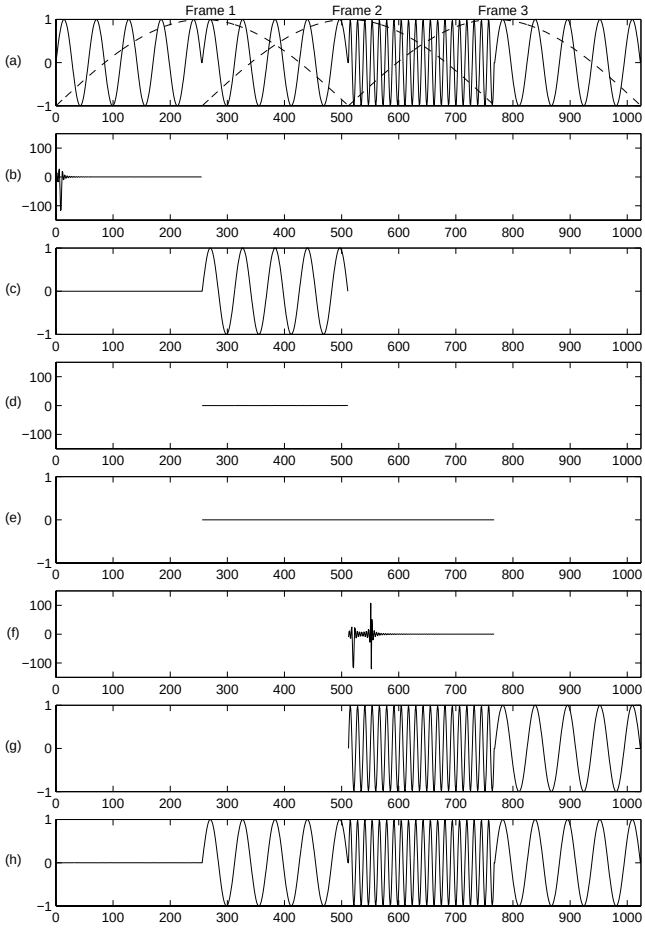

Figure 3. Illustration of signal analysis/synthesis with MDCT, overlap-add procedure and perfect reconstruction of time domain samples. (a) a phase/frequency-modulated time signal; (b)(d)(f) MDCT spectra in different time slots, indicated as frames 1, 2, 3 in (a); (c)(e)(g) reconstructed time domain samples (with IMDCT) of frames 1, 2, 3 respectively; (h) the reconstructed time samples after the overlap-add procedure.

Figure 4 shows the fluctuation of MDCT spectrum in comparison with DFT and $SDFT_{(N+1)/2,1/2}$ spectra. With a frequency-modulated time signal in Figure 4(a), the DFT power spectrum is very stable despite a moving window. Conversely, the MDCT spectrum is very unstable. The $SDFT_{(N+1)/2,1/2}$ spectrum is in between. This is at least one evidence that the $SDFT_{(N+1)/2,1/2}$ can be used as a bridge to connect MDCT and DFT in audio coding applications.

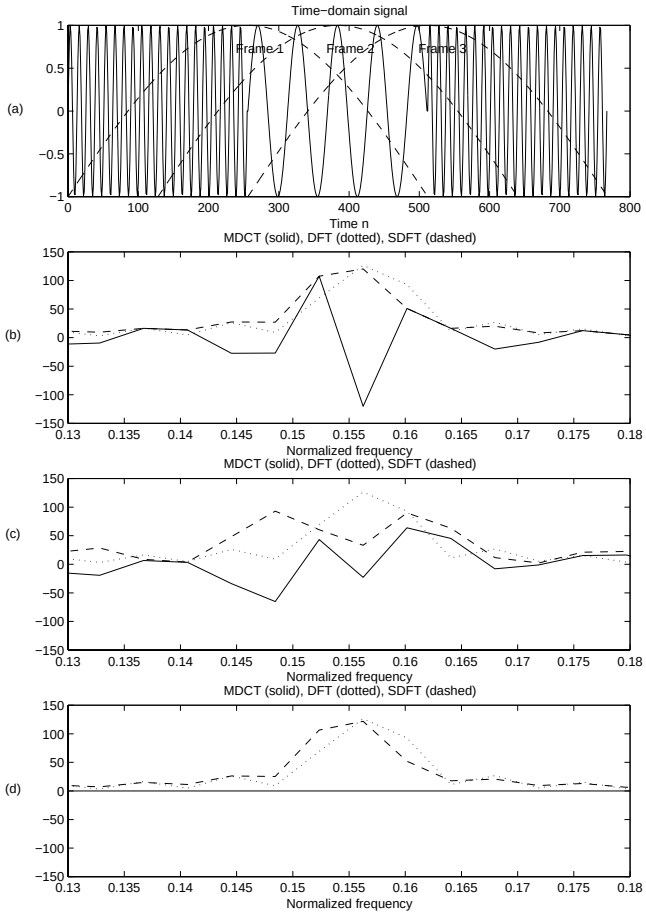

Figure 4 Comparison of DFT, $SDFT_{(N+1)/2,1/2}$ and MDCT spectra in different time slots. (a) a frequency-modulated time signal (solid line) with a moving window, (b)(c)(d) DFT (dotted lines), $SDFT_{(N+1)/2,1/2}$ (dashed lines) and MDCT (solid lines) spectra of Frames 1, 2, 3.

3.3 Intuitive illustration of the Concept of Time Domain Alias Cancellation (TDAC)

Based on (1) (2) (6) (9), we have used a similar artificial time domain signal as in Figure 1(a) to illustrate the Time Domain Aliasing Cancellation (TDAC) concept in an intuitive way. The artificial signal of 54 samples is shown in Figure 5(a). The MDCT coefficients of the signal in Window 1 are shown in Figure 5(b). For simplicity we have used rectangular windows. Obviously the coefficients are subsampled by 50% in MDCT (from 2N time domain samples to N independent frequency domain coefficients), and the alias is introduced as well. The IMDCT coefficients of the signal in Figure 5(b) are illustrated in Figure 5(c). This step introduces redundancy (from N frequency domain coefficients to 2N time domain samples). The MDCT coefficients of the signal in Window 2 are presented in Figure 5(d). The corresponding IMDCT time domain signal is shown in Figure 5(e). If the overlap-add procedure is performed with Figure 5(c) and (e), perfect reconstruction (PR) of the original signal in the overlapped part (between points B and C) can be achieved.

It is clear that one cannot achieve perfect reconstruction (PR) for the first half of the first window and the second half of the last window as indicated in Figure 5.

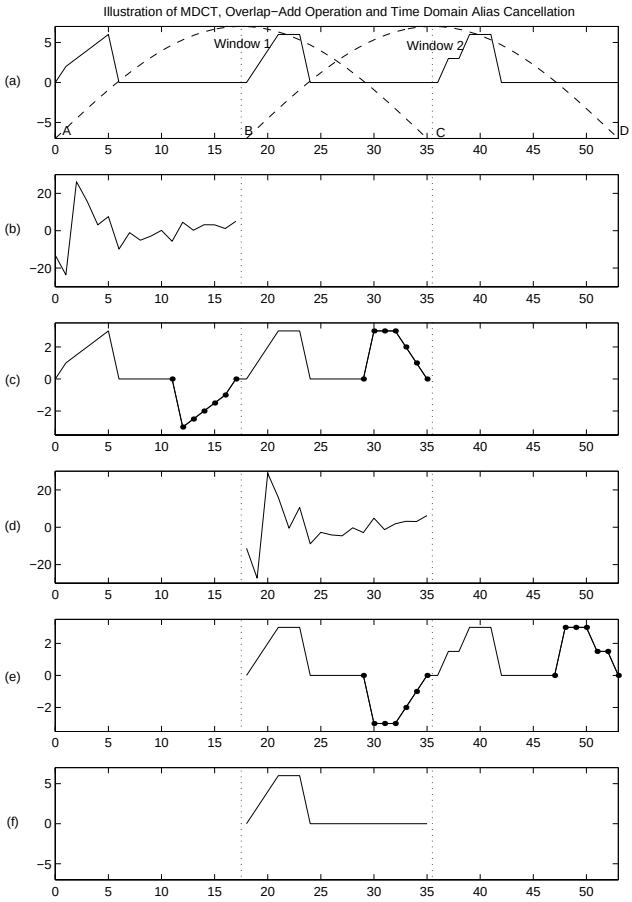

Figure 5. Illustration of the MDCT, overlap-add (OA) procedure and the concept of the Time Domain alias cancellation (TDAC). (a) An artificial time signal, dashed lines indicating the 50% overlapped windows; (b) MDCT coefficients of the signal in Window 1; (c) IMDCT coefficients of the signal in (b), the alias is shown by markers on the line; (d) The MDCT coefficients of the signal in Window 2; (e) IMDCT coefficients of the signal in (d), the alias shown by markers on the line; (f) The reconstructed time domain signal after the overlap-add (OA) procedure. The original signal in the overlapped part (between points B and C) is perfectly reconstructed.

In order to illustrate the TDAC concept during the window switching specified in the MPEG-2 AAC ISO/IEC standard [1], we define two overlapping windows with window functions h_k and g_k. The conditions for perfect reconstruction are [12]:

$$h_{N+k} \cdot h_{2N-1-k} = g_k \cdot g_{N-1-k} \tag{17}$$

$$h_{N+k}^2 + g_k^2 = 1 \tag{18}$$

Using (6) one can easily see one of the important properties of MDCT: the time domain alias in each half of the window is independent, which allows adaptive window switching [12]. The

TDAC concept during window switching in MPEG-2 AAC is illustrated in Figure 6.

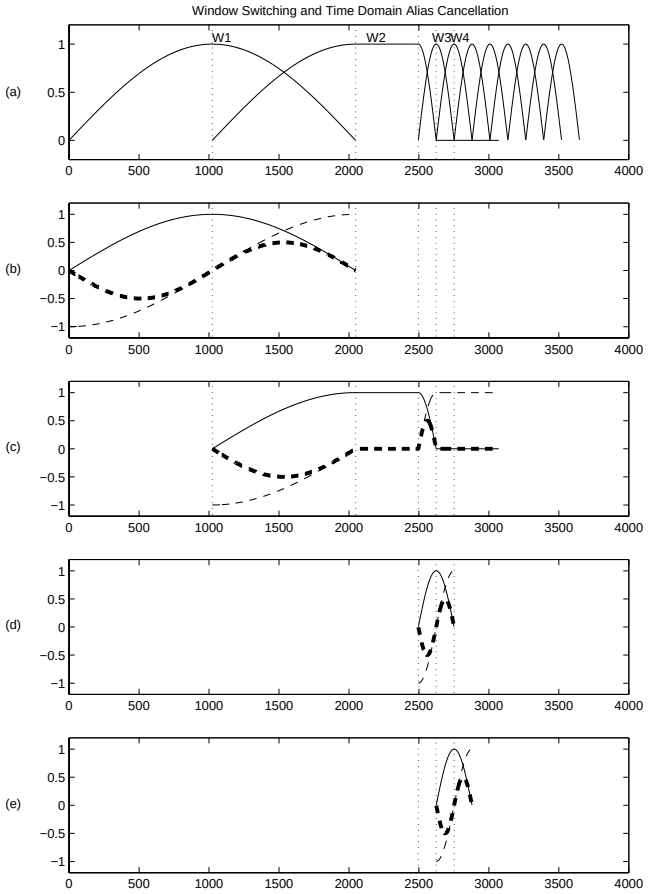

Figure 6. TDAC in the case of window switching. (a) three types of window shape in MPEG-2 AAC indicated with W1, W2, W3; (b) window function in the long window (solid line), time domain alias (thin dashed line), time domain alias after weighting with the window function (thick dashed line); (c) window function in the transition window (solid line), time domain alias (thin dashed line), time domain alias after weighting with the window function (thick dashed line); (d)(e) window function in the short window (solid line), time domain alias (thin dashed line), time domain alias after weighting with the window function (thick dashed line).

4. A SDFT BASED AUDIO ENCODER STRUCTURE

Figure 1 shows the interconnection between MDCT and $SDFT_{(N+1)/2,1/2}$ when N is even, which is often the case in audio coding applications. For practical reasons, we have considered here only real-valued signals. In this case, one can prove that MDCT coefficients are equivalent to the real part of the $SDFT_{(N+1)/2,1/2}$ of the input signal. That is

$$MDCT(signal) = real\{SDFT_{(N+1)/2,1/2}(signal)\} \qquad (19)$$

This result can be used for the optimization of our audio encoder published previously [2]. The new encoder structure is illustrated in Figure 7. $SDFT_{(N+1)/2,1/2}$ can be implemented via existing FFT routines with some minor modifications as described in (11). The structure in Figure 7 has obvious advantages in the sense of system optimization. We can achieve the same coding performance without a parallel FFT routine in the psychoacoustic model [2]. In comparison with [2], we have used $SDFT_{(N+1)/2,1/2}$ instead of MDCT coefficients as the input of the psychoacoustic model, because $SDFT_{(N+1)/2,1/2}$ provides the necessary phase information for the psychoacoustic model, which has significantly improved the coding performance. Another advantage is that FFT routines are widely available for implementation. Conceptually, our approach is similar to that in [5][6]. However, we have tackled the problem from a different perspective – solving the mismatch. This is only a first step and a partial solution.

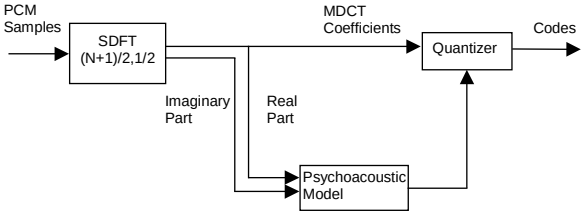

Figure 7. A simplified structure of the proposed audio encoder

5. EXPERIMENTAL RESULTS

This section describes the codec we have used for a listening test, including details of the samples, test method and analysis of the results.

5.1 Codec description

A codec similar to MPEG-2 AAC using a psychoacoustic model [13] is shown in Figure 8 and Figure 9. This codec was used to encode the test samples for inclusion in a listening test. The subjective comparison was done by comparing samples encoded with the original codec in Figure 8 and Figure 9 to the modified structure described in Figure 7. Essentially the only modification is the DFT or SDFT((N+1)/2,1/2) as the input to the psychoacoustic model. The prediction blocks were disabled during the test.

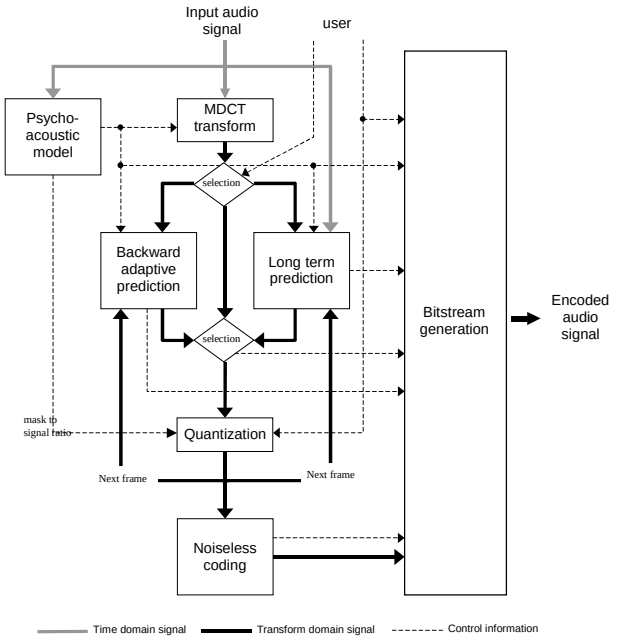

Figure 8. Block diagram of the encoder

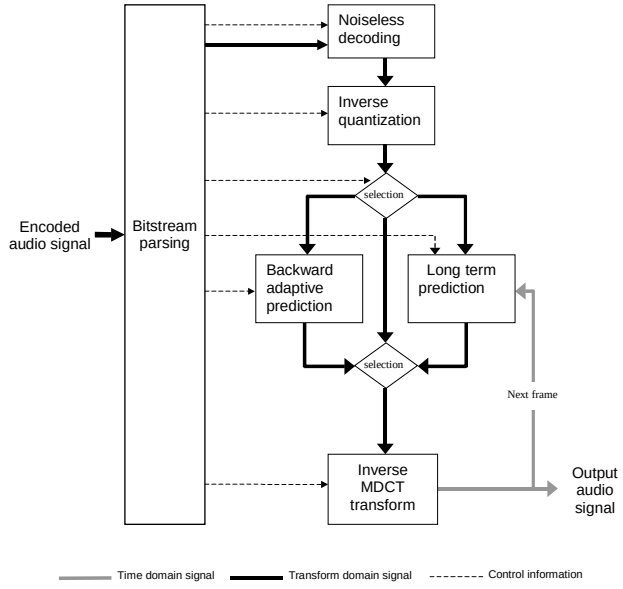

Figure 9. Block diagram of the decoder

5.2 Listening test description

A small formal listening test was initiated to investigate whether the SDFT optimized encoder presents any subjectively appreciable artefacts not inherent in the DFT-based encoder. An A/B/X test [14] designed to find whether listeners could distinguish between the two encoding methods was created and implemented using the GuineaPig [15] subjective testing software, developed jointly between NRC and Helsinki University of Technology. The A/B/X test is a paired comparison paradigm that utilizes a forced-choice grading method to tests whether a listener can correctly differentiate between two sources. Two audio samples (a reference and comparator) are assigned to the three sample items, A, B, and X, by test rules such that;

- X is randomly assigned with the reference or comparator
- A is also randomly assigned with the reference or comparator
- B is assigned the alternative sample to A

A listener is required to identify and grade which sample item, A or B, is identical to X; or described another way, which of the two differential intervals (A-X, B-X) is imperceptible. This results in a binomial score associated with a correct or incorrect grade.

The experimental design for this test presented each sample pair (reference and comparator) eight times, split equally between two test blocks for each listener. The GuineaPig software GUI used for presentation of sample items and grading is shown in Figure 10. Each sample is played in parallel synchronously with only the sample associated with the chosen sample item being audible. This allows subjects to listen to each sample item monadicly or to crossfade between each sample simply by clicking alternate sample items during playback. For this experiment a linear crossfade lasting 20ms (equivalent to 960 samples for f_S=48kHz) was used. Grading was achieved by checking one of the boxes, A or B, after which clicking 'Done' progresses the subject onto the next test item.

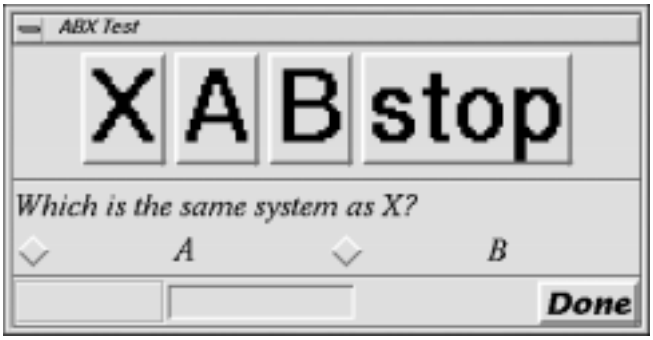

Figure 10. Software GUI for replay and grading of each test item

All test data is saved automatically disk during the test.

For the test, a panel of seven listeners were chosen from the staff of the Audio Coding group at NRC. Each had experience in listening to coded audio material and its associated artifacts and experience in subjective tests involving small impairments in coded audio material. Results from previous tests had classified the individuals as "expert" listeners in accordance with ITU-R Rec. BS 1116 [16]. Verifying listener expertise in tests involving forced-choice paradigms is difficult. Where no subjectively detectable differences are present, which is a likely feature of small impairment tests, a measure of a subjects grading reliability, purely from the material under examination, will not be achievable. A test to check intra-listener reliability based on the grading error of a low-anchor sample having intentionally appreciable coding artifacts was thus included as a sub-set of the experiment. This was presented eight times randomly in the second test block.

The audio material used for the test was comprised of six stereo 16bit, 48kHz standard MPEG-4 audio test samples. Each was summed to mono before encoding. The six samples were classified as:

1. es01 - Solo female singing (Susanne Vega)
2. es02 - german male speech
3. sc03 - contemporary pop
4. si01 - harpsichord
5. si02 - castanets
6. si03 - pitchpipe

The codec applied to the test material was similar to the MPEG-2 AAC, with only the basic coding tools. The maximum average bitrate for the coded samples was 64 kbps. Due to the difficulty in estimating the errors inherent in the SDFT version, it was decided to use sample 'es01' encoded at 48kbps utilizing the DFT version of the psychoacoustic model as the lower-anchor.

5.3 Results analysis

The statistical measures involved in analyzing forced-choice paired comparison listening test data are extensively described elsewhere [17-18]. The statistical hypothesis for the A/B/X test states that

$$H_0: \quad p = 0.5$$
$$H_1: \quad p > 0.5$$

where p is the observed proportion of correct gradings. The null hypothesis H_0 states that the listener is unable to distinguish between the two samples a significant proportion of the time.

The low-anchor test items were analyzed separately and found to have $p = 1.0$ for all listeners, indicating reliability at detection of the intended artefact. The results of the test material are shown in Table 1.

Table 1(a,b). Test results

Subject	1	2	3	4	5	6	7
p	.71	.67	.54	.50	.58	.69	.63
Sig. level	.002	.008	.097	.115	.059	.004	.026

(a)

Sample	es01	es02	sc03	si01	si02	si03
p	.71	.64	.45	.54	.71	.64
Sig. level	.000	.011	.077	.092	.000	.011

(b)

The results show that for $\alpha = .050$ more than half the subjects show a probable discriminatory ability, with the pooled p for all grades being .62 with a .000 significance level. The results for each subject have pooled samples and for each sample the subject grades are pooled. The number of trials and subjects are not so large as to have confidence in any conclusions drawn from these results. However, within the confines of this small test there appears to exists appreciable differences between the encoding methods that are dependent on the source material. Further tests should be initiated to clarify the existence and nature of any artefacts caused by the SDFT method.

6. CONCLUSION AND FUTURE WORK

This paper has addressed the mismatch issue between the two fundamental tools used in advanced audio coding, and has examined the interconnection between MDCT, SDFT and DFT. A direct and compact formulation of the MDCT has been established with the help of a SDFT enabling us to clarify the symmetric properties of MDCT, as well as illustrate the Time Domain Alias Cancellation (TDAC) concept in a very intuitive and illustrative way. We have also shown some of the peculiar properties of MDCT affecting the coding performance of a MDCT based audio codec. Based on this analysis we have suggested a modified structure of audio encoder implemented via existing FFT routines. The suggested encoder has demonstrated improved coding performance. The optimization presented here is especially important for applications with limited computational and storage capacities, such as hand-held devices.

Our subjective examination of the SDFT implementation has outlined the need for further study of the potential artifacts inherent in this new process, and the tradeoff between QoS and greater efficiency.

MDCT is an efficient and elegant concept in terms of signal analysis and synthesis, especially with its Time Domain Alias Cancellation (TDAC) characteristics. However, its mismatch with the DFT domain based psychoacoustic model has limited its coding performance. We believe that the encoder structure proposed in this paper represents one step further to solve this mismatch by introducing $SDFT_{(N+1)/2,1/2}$ as a bridge between DFT and MDCT.

In recent years, more attention was paid to wavelet filterbank based audio coding algorithms [19]. However, their rather disappointing performance may be also caused by the mismatch between the two fundamental tools of audio coding - audio signal representation and the human auditory system perceptual model. Therefore, it is hoped that further study on the interconnection between discrete wavelet and Fourier transform may lead to a breakthrough in wavelet domain based audio coding algorithms.

7. ACKNOWLEDGMENTS

Ye Wang wishes to thank Mr. James Johnston (AT&T Research Laboratories) and Prof. Anibal Ferreira (University of Porto) for very inspiring discussions during the 17th AES International Conference on High Quality Audio Coding in Florence, Italy, 2nd – 5th September, 1999. Those discussions inspired us to conduct this research. Ye Wang also wishes to thank Prof. Deepa Kundur (University of Toronto) and Dr. Jie Yang (Carnegie Mellon University) for reading the initial draft of this paper during ACM Multimedia99 international conference in Orlando, Florida, USA, 30th October – 5th November, 1999, and for constructive suggestions.

8. REFERENCES

[1] ISO/IEC JTC1/SC29/WG11, "Coding of moving pictures and audio - MPEG-2 Advanced Audio Coding AAC," ISO/IEC 13818-7 International Standard, 1997.

[2] Wang, Y., Yaroslavsky, L., Vilermo, M., Väänänen, M. "Restructured Audio Encoder for Improved Computational

Efficiency," AES 108th International Convention, February 19-22, 2000, Paris, France.

[3] Wang, Y., Yaroslavsky, L., Vilermo, M., "On the Relationship between MDCT, SDFT and DFT," 16th IFIP World Computer Congress (WCC2000)/5th International Conference on Signal Processing (ICSP2000), August 21-25, 2000, Beijing, China.

[4] Wang, Y., Yaroslavsky, L., Vilermo, M., Väänänen, M. "Some Peculiar Properties of the MDCT," 16th IFIP World Computer Congress (WCC2000)/5th International Conference on Signal Processing (ICSP2000), August 21-25, 2000, Beijing, China.

[5] Ferreira, A., "Spectral Coding and Post-Processing of High Quality Audio," Ph.D. thesis http://telecom.inescn.pt/doc/phd_en.html

[6] Malvar, H., "A Modulated Complex Lapped Transform and Its Apllications to Audio Processing," IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999, Phoenix, USA.

[7] Malvar, H., "Signal Processing with Lapped Transforms," Artech House, Inc., 1992.

[8] Princen, J. P., Bradley, A. B., "Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No. 5, October 1986.

[9] Princen, J. P., Johnson, A. W., Bradley, A. B., "Subband/Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation," IEEE International Conference on Acoustics, Speech, and Signal Processing, 1987, Dallas, USA, pp. 2161-2164.

[10] Yaroslavsky, L., Eden, M., "Fundamentals of Digital Optics," Birkhauser, Boston, 1996.

[11] Bosi, M., Brandenburg, K., Quackenbush, S., Fielder, L., Akagiri, K., Fuchs, H., Dietz, M., Herre, J., Davidson, G., Oikawa, Y., "ISO/IEC MPEG-2 Advanced Audio Coding," Journal of Audio Engineering Society, vol. 45, no. 10, 1997.

[12] Edler, B., "Coding of Audio Signals with Overlapping Block Transform and Adaptive Window Functions," (in German), *Frequenz*, vol.43, pp.252-256, 1989

[13] Wang, Y., Vilermo, M., "An Excitation Level Based Psychoacoustic Model for Audio Compression," The 7th ACM International Multimedia Conference, October 30 to November 4, 1999, Orlando, Florida, USA.

[14] Clarke, D., "High-Resolution Subjective Testing Using a Double-Blind Comparator," *J. Audio Eng. Soc.,* vol 30, no. 5, pp. 330-338 (1982 May).

[15] Hynninen, J., and Zacharov, N., "GuinePig – A generic subjective test system for multichannel audio," *AES preprint 4871, AES 106th convention,* May 8-11, 1999, Munich

[16] ITU-R Rec. BS-1116, "Methods for the Subjective Assessment of Small Impairments in Audio Systems Including Multichannel Sound Systems," International Telecommunications Union (1994)

[17] Leventhal, L., "Type 1 and Type 2 Errors in the Statistical Analysis of Listening Tests," Journal of Audio Engineering Society, vol. 34, no. 6, 1986.

[18] Burstein, H., "Transformed Binomial Confidence Limits for Listening Tests," Journal of Audio Engineering Society, vol. 37, no. 5, 1989.

[19] Erne, M., Moschytz, G., "Audio Coding based on Rate Distortion and Perceptual Optimization Techniques," 17th AES International Conference on High Quality Audio Coding, September 1999, Florence, Italy, pp. 220-225.

[P5] Wang, Y., Vilermo, M., Väänänen, M., Yaroslavsky, L. “A Multichannel Audio Coding Algorithm for Inter-Channel Redundancy Removal”, AES110th International Convention, May 12-15, 2001, Amsterdam, The Netherlands, preprint 5295

Audio Engineering Society

Convention Paper

Presented at the 110th Convention
2001 May 12–15 Amsterdam, The Netherlands

A Multichannel Audio Coding Algorithm for Inter-Channel Redundancy Removal

Ye Wang[1], Miikka Vilermo[1], Mauri Väänänen[1] and Leonid Yaroslavsky[2]
[1]Speech and Audio Systems Laboratory
Nokia Research Center
P.O.Box 100 (Visiokatu 1)
FIN-33721 Tampere, Finland
{ye.wang, miikka.vilermo, mauri.vaananen}@nokia.com
[2]Department of Interdisciplinary Studies
Tel Aviv University
Ramat Aviv 69978, Israel
yaro@eng.tau.ac.il

ABSTRACT

This paper presents a novel lossless multichannel audio coding algorithm to remove inter-channel redundancy. We employ an Integer-to-Integer Discrete Cosine Transform (INT-DCT) to perform inter-channel decorrelation after quantization of Modified Discrete Cosine Transform (MDCT) coefficients of individual channels. When compared with a Karhunen-Loeve Transform (KLT) based approach our new method has three major advantages: 1) avoids quantization noise spreading to other channels; 2) computational simplicity; 3) uses less overhead information (a quantized covariance matrix or eigenvector is avoided in our algorithm), while having a similar decorrelation capability.

INTRODUCTION

With rapid deployment of DVD, high-quality multichannel audio compression has finally found its way from research labs to widespread applications. In spite of a steady increase in storage capacity and transmission bandwidth, multichannel audio could still poise a problem to traditional and new media delivery systems.

Among several existing multichannel audio compression algorithms, MPEG Advanced Audio Coding (AAC) is currently the most powerful one in the MPEG family, which supports up to 48 audio channels and provides perceptually lossless audio at 64 kbits/s per channel [1]. The most widely adopted multichannel configuration is the 5.1 channel configuration, which refers to left (L), center (C), right (R), left surround (LS), right surround (RS) and an optional low-frequency-enhancement (LFE) channel.

Some effort has been made on reducing the inter-channel redundancy inherent in multichannel audio. In the established technologies such as AAC, only "Intensity Stereo Coding/Coupling" and "MS Stereo Coding" have been employed. Coupling is adopted based on psychoacoustic evidence that at high frequencies (above approximately 2 kHz) the human auditory system localizes sound based primarily on the "envelopes" of critical-band-filtered versions of the signals reaching the ears, rather than on the signals themselves. MS stereo coding encodes the sum and difference of the signal in two symmetric channels instead of the original signals in left and right channels [2]. Both MS stereo and intensity stereo coding operate on Channel-Pair-Elements (CPEs). In the case of the most widely adopted 5.1 surround sound constellation, the diagram of the transform part is as in Figure 1 (LFE channel has not been considered in this paper).

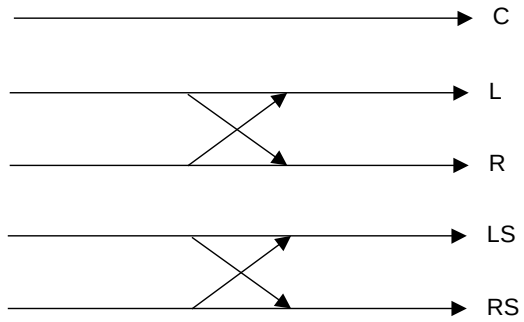

Figure 1. The pair structure of AAC surround sound coding

In order to further reduce inter-channel redundancy, an interesting algorithm was proposed in [3], which utilizes a Karhunen-Loeve Transform (KLT) for inter-channel decorrelation. A simplified block diagram is illustrated in Figure 2. However, that algorithm has a few unsolved challenges:

1) How to map the masking threshold requirements for each of the original channels in the MDCT domain into the inter-channel transformed (MDCTxKLT) domain?
2) How to quantize the inter-channel transformed MDCTxKLT coefficients optimally so as to satisfy the masking threshold requirements in the original channels in the MDCT domain?

This paper presents a new attempt to solve the problem. This algorithm is especially efficient to the class II and III multichannel audio material defined in [3], that is, when the audio material has more than 2 correlated channels.

INTER-CHANNEL DECORRELATION USING INT-DCT

Generally, a N channel surround sound system, running with a bit rate of M bps/ch does not necessarily have a total bit rate of MxN bps, but rather an overall bit rate significantly less than MxN due to inter-channel redundancy. The effect of adding more channels will further increase the efficiency of the coding algorithm described in this paper, in the case of multichannel correlated material.

The only significant difference between our approach and the one in [3] is that we put an INT-DCT unit between the quantizer and Huffman coder instead of putting it directly after MDCT, so as to yield a lossless approach. The block diagrams of the prior arts and new method are illustrated in Figure 2 and 3 respectively.

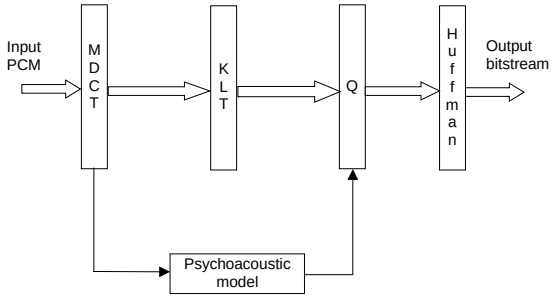

Figure 2. A simplified block diagram of prior arts. A KLT is employed for inter-channel decorrelation.

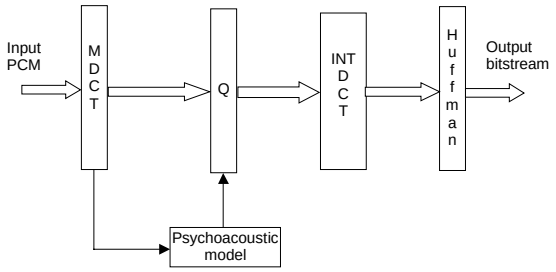

Figure 3. A simplified block diagram of the proposed method. A INT-DCT is employed for inter-channel decorrelation.

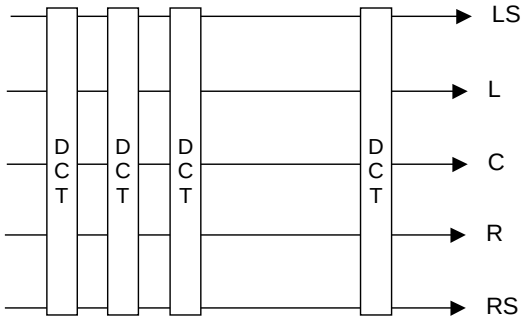

Figure 4. The new transform structure of surround sound. The horizontal lines represent the quantized MDCT coefficients of each individual channel.

DCT is a well-known decorrelation transform, which usually has similar energy compaction property as that of KLT. The diagram of the new method is illustrated in Figure 4. The horizontal lines

represent the quantized MDCT coefficients of different channels and the DCT is approximated by an Integer-to-Integer DCT (INT - DCT) discussed in the next section.

INTEGER-TO-INTEGER DISCRETE COSINE TRANSFORM (INT-DCT)

We use essentially the same approach as in [4]. Creating an integer-to-integer transform starts by first factorizing the transform matrix into matrices that have ones on the diagonal and nonzero off-diagonal elements only in one row or column. If such a factorization exists rounding the result after each intermediate step results in an integer-to-integer transform that not only approximates the original transform but that is also precisely reversible.

The factorization is not unique. One straightforward method is to use elementary matrices to reduce the transform matrix into unit matrix (if possible) and then use the inverses of the elementary matrices as the factorization. For orthogonal matrices, such as the DCT matrix, one can also first factorize the transform matrix into Givens matrices and then further factorize each of the Givens matrices into three matrices that can be used as building blocks for an integer-to-integer transform [4][5].

Lifting Scheme

A matrix that has ones on the diagonal and nonzero off-diagonal elements only in one row or column can be used as a building block when constructing an integer-to-integer transform. This is called 'the lifting scheme'. Such a matrix has an inverse also when the end result is rounded in order to map integers to integers.

Let us consider the case of a 3 x 3 matrix ($a, b \in R$, $x_i \in Z$)

$$\begin{bmatrix} 1 & 0 & 0 \\ a & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_\Delta = \begin{bmatrix} x_1 \\ ax_1 + x_2 + bx_3 \\ x_3 \end{bmatrix}_\Delta = \begin{bmatrix} x_1 \\ x_2 + \lfloor ax_1 + bx_3 \rceil_\Delta \\ x_3 \end{bmatrix} \quad (1)$$

where $\lfloor \ \rceil_\Delta$ denotes rounding for the nearest integer. The inverse of (1) is

$$\begin{bmatrix} 1 & 0 & 0 \\ -a & 1 & -b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 + \lfloor ax_1 + bx_3 \rceil_\Delta \\ x_3 \end{bmatrix}_\Delta = \begin{bmatrix} x_1 \\ -ax_1 + x_2 + \lfloor ax_1 + bx_3 \rceil_\Delta - bx_3 \\ x_3 \end{bmatrix}_\Delta = \begin{bmatrix} x_1 \\ x_2 + \lfloor -ax_1 + \lfloor ax_1 + bx_3 \rceil_\Delta - bx_3 \rceil_\Delta \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (2)$$

Factorizing the DCT Kernel into Givens Rotations

A Givens rotation is a matrix of the form [6]:

$$G(i,k,\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c & s & 0 \\ 0 & -s & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} \\ i \\ \\ k \end{matrix}, \quad (3)$$

where $c = \cos(\theta)$, $s = \sin(\theta)$

A Givens matrix is clearly orthogonal and the inverse is

$$G(i,k,\theta)^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c & -s & 0 \\ 0 & s & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} \\ i \\ \\ k \end{matrix} \quad (4)$$

Any m x m orthogonal matrix can be factorized into m(m-1)/2 Givens rotations and m sign parameters [5]. As an example:

Let A be an orthogonal matrix.

Firstly we choose θ_1 such that $\tan(\theta_1) = \frac{a_{2,3}}{a_{3,3}}$, then it follows that

$$G(2,3,\theta_1)^{-1} \cdot A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_1) & -\sin\theta_1 \\ 0 & \sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ b_{2,1} & b_{2,2} & 0 \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix} = B \quad (5)$$

If $a_{3,3} = 0$ then we simply choose $\theta_1 = \pi/2$ i.e. $\cos(\theta_1) = 0$, $\sin(\theta_1) = 1$. This matrix still has an inverse even when used to create an integer-to-integer transform.

Secondly we choose θ_2 such that $\tan(\theta_2) = \frac{a_{1,3}}{b_{3,3}}$,

$$G(1,3,\theta_2)^{-1} \cdot B = \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) \\ 0 & 1 & \\ \sin(\theta_2) & 0 & \cos(\theta_2) \end{bmatrix} \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ b_{2,1} & b_{2,2} & 0 \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & 0 \\ b_{2,1} & b_{2,2} & 0 \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} = C \quad (6)$$

Now, since both $G(2,3,\theta_1)^{-1}$ and $G(1,3,\theta_2)^{-1}$ and also A are orthogonal therefore C has to be orthogonal and thus every row and column in C has unit norm. Therefore $c_{3,3} = \pm 1$ and $c_{3,1}, c_{3,2} = 0$

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & 0 \\ b_{2,1} & b_{2,2} & 0 \\ 0 & 0 & \pm 1 \end{bmatrix} \tag{7}$$

Lastly we choose θ_3 such that $\tan(\theta_3) = \frac{c_{1,2}}{b_{2,2}}$,

$$\begin{aligned} & G(1,2,\theta_3)^{-1} \cdot C \\ &= \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{1,1} & c_{1,2} & 0 \\ b_{2,1} & b_{2,2} & 0 \\ 0 & 0 & \pm 1 \end{bmatrix} \\ &= \begin{bmatrix} d_{1,1} & 0 & 0 \\ d_{2,1} & d_{2,2} & 0 \\ 0 & 0 & \pm 1 \end{bmatrix} = D \end{aligned} \tag{8}$$

Since $G(1,2,\theta_3)^{-1}$ and C are orthogonal, D has to be orthogonal and similarly to (7)

$$D = \begin{bmatrix} \pm 1 & 0 & 0 \\ 0 & \pm 1 & 0 \\ 0 & 0 & \pm 1 \end{bmatrix}$$

Finally we have:

$$G(1,2,\theta_3)^{-1} \cdot G(1,3,\theta_2)^{-1} \cdot G(2,3,\theta_1)^{-1} \cdot A = D \tag{9}$$

Taking D as the sign matrix we have:

$$D \cdot G(1,2,\theta_3)^{-1} \cdot G(1,3,\theta_2)^{-1} \cdot G(2,3,\theta_1)^{-1} \cdot A = I \tag{10}$$

Therefore A can be factorized as:

$$A = G(2,3,\theta_1) \cdot G(1,3,\theta_2) \cdot G(1,2,\theta_3) \cdot D \tag{11}$$

For m x m matrices the operation is similar [5].

Factorizing a Givens Rotation

Givens rotations can in turn be factorized as follows [7]:

$$\begin{aligned} G(i,k,\theta) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c & s & 0 \\ 0 & -s & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & (1-c)/s & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -s & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & (1-c)/s & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{12}$$

when θ is not an integer multiple of 2π. If it is then the Givens rotation matrix equals the unity matrix and no factorization is necessary. Let us denote these factors as $G(i,k,\theta)_1$, $G(i,k,\theta)_2$ and $G(i,k,\theta)_3$. A transform that behaves similarly to matrix A, maps integers to integers and is reversible is then

$$\begin{aligned} & \left\lfloor G(2,3,\theta_1)_1 \cdot \left\lfloor G(2,3,\theta_1)_2 \cdot \left\lfloor G(2,3,\theta_1)_3 \cdot \right\rfloor \right\rfloor \right\rfloor \cdots \\ & \left\lfloor G(1,2,\theta_3)_1 \cdot \left\lfloor G(1,2,\theta_3)_2 \cdot \left\lfloor G(1,2,\theta_3)_3 \cdot D \cdot \underline{x} \right\rfloor_\Delta \right\rfloor_\Delta \right\rfloor_\Delta \cdots \left\rfloor_\Delta \right\rfloor_\Delta \right\rfloor_\Delta \right\rfloor_\Delta \end{aligned} \tag{13}$$

where $\underline{x}$ is the integer 3 x 1 input vector.

Integer-to-Integer DCT (INT-DCT)

Now we have all the necessary building blocks to construct an integer-to-integer DCT. DCT matrix is orthogonal, therefore we can factorize it into Givens rotations. Givens rotations can in turn be factorized into matrices that are the basic building block matrices of the integer-to-integer DCT. The original integer input is multiplied by each of these matrices and every intermediate result is rounded to the nearest integer.

In our case the 5 x 5 DCT matrix is factorized into 10 Givens rotations. Givens rotations are factorized into 3 matrices each, resulting the total of 15 matrix multiplications. However the internal structure of these matrices guarantees that only 15 multiplications and 15 rounding operations are needed in total.

THE EXPERIMENT

In order to evaluate the performance of the new algorithm, we have used a mono MPEG-2 AAC encoder [8] to compare the bitrates with and without INT-DCT inter-channel decorrelation. The block diagram of the modified AAC encoder and decoder in our experiment are illustrated in Figures 5 and 6, where both the Intensity Coupling and MS Stereo Coding are disabled. The bitrate of individual channel as well as total bitrate were compared with and without inter-channel decorrelation. A 5-tap INT-DCT was used on the quantized MDCT coefficients across the five channels resulting in new coefficients to be Huffman coded and written to bitstream.

On the decoder side the original quantized MDCT coefficients can be perfectly reconstructed from the INT-DCT coefficients, due to the invertibility of the INT-DCT. Scale factors and other original AAC side information were unaffected by this process. The INT-DCT was used on a scalefactor band basis. As a result, a flag bit was needed for each scalefactor band based on the inter-channel prediction gain. The flag bit indicated whether to use the INT-DCT in a scalefactor band. This flag bit is the only extra side information to be added to the bitstream. For the sake of simplicity only long windows were used. The usage of the INT-DCT was restricted for the first 40-scalefactor bands. The maximum bitrate for the test was set for 64 kbps while the sampling rate of all the five channel samples was 48 kHz. Since 40 extra bits were needed

for each frame, the total amount of new side information was 40x48000/1024 = 1875 bps, which was ca. 0.5% of the total bitrate of the 5 channels. Because the algorithm presented in this paper was a lossless scheme, we were able to list the bitrate reduction in table 1 using some 5-channel surround sound samples.

In order to show the energy compaction property of the INT-DCT using an example, the original quantized MDCT coefficients and inter-channel decorrelated coefficients are compared in Figure 7. It is clear that the energy is concentrated into fewer channels. A comparison of the original bitrate and the resulting bitrate after an INT-DCT is illustrated in Figure 8. It can be seen the total bitrate is reduced after the INT-DCT inter-channel decorrelation.

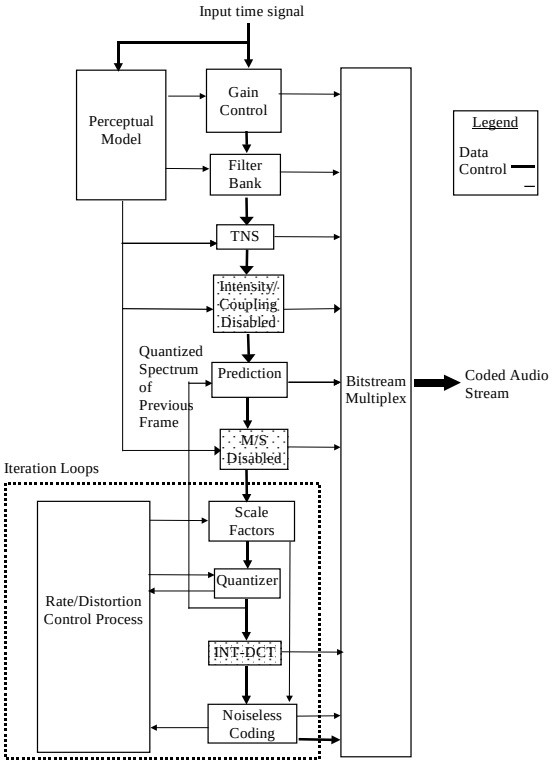

Figure 5. Modified AAC encoder block diagram.

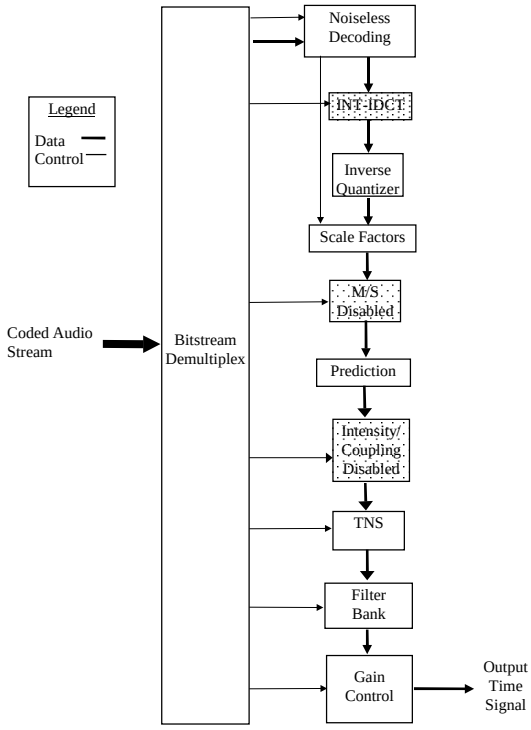

Figure 6. Modified AAC decoder block diagram.

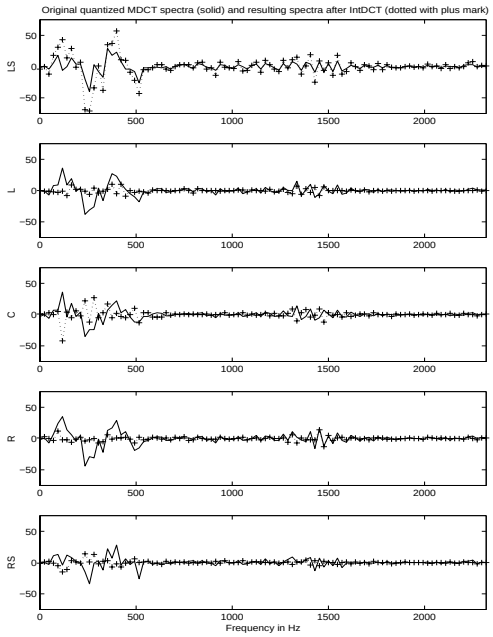

Figure 7. Comparison of the original quantized MDCT coefficients and the inter-channel decorrelated coefficients

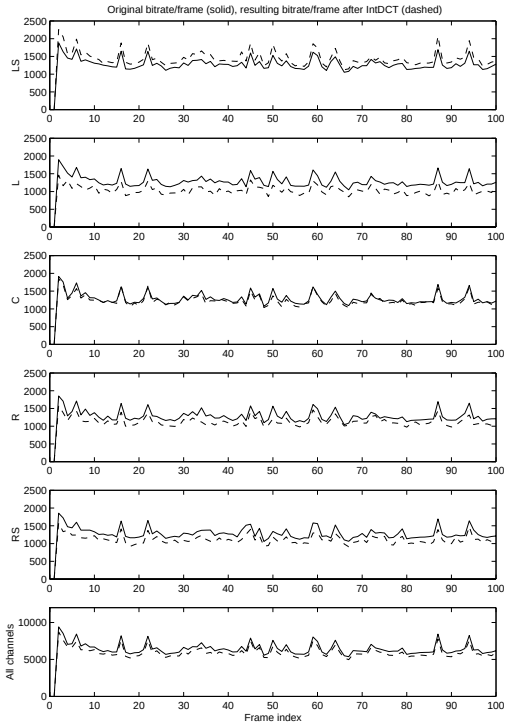

Figure 8. Comparison of the original bitrate and the resulting bitrate after an INT-DCT inter-channel decorrelation.

Sample index	Total bitrate reduction (%)	Sample description
1	7.0	Male speech
2	1.7	Guitar
3	2.1	Coffee table
4	4.0	Concert hall
5	3.3	Passing train
6	3.4	Bus stop
7	2.3	Cafeteria

Table 1. Bitrate reduction performance after an INT-DCT. 7 five-channel surround sound samples were tested.

CONCLUSION

We have presented a novel lossless multichannel audio-coding algorithm based on INT-DCT. The computational complexity of the algorithm is negligible in comparison with the original AAC codec.

Acknowledgement

Ye Wang wishes to thank Mr. James Johnston (AT&T research Labs) for helpful discussions.

References

1. Brandenburg, K., Bosi, M. “ISO/IEC MPEG-2 Advanced Audio Coding: Review and Applications”, AES 103rd Convention, September 1997, New York, USA, preprint 4641
2. Bosi, M., Brandenburg, K., Quackenbush, S., Fielder, L., Akagiri, K., Fuchs, H., Dietz, M., Herre, J., Davidson, G., Oikawa, Y. “ISO/IEC MPEG-2 Advanced Audio Coding”, J. Audio Eng. Soc., Vol. 45, No. 10, October 1997
3. Yang, D., Ai, H., Kyriakakis, C., Kuo, C.-C. “An Inter-channel Redundancy Removal Approach for High-Quality Multichannel Audio Compression”, AES 109th Convention, September 2000, Los Angeles, USA, preprint 5238
4. Goyal, V.K., “Transform Coding with Integer-to-Integer Transforms”, IEEE Transactions on Information Theory, Volume 46, Issue 2 , March 2000, pp: 465 –473
5. Vaidyanathan, P.P., “Multirate Systems and Filter Banks”, Prentice Hall, 1993
6. Golub, G.H., Van Loan, C.F., “Matrix Computations”, The Johns Hopkins University Press, 1996
7. Daubechies, I., Sweldens, W., “Factoring Wavelet Transforms into Lifting Steps”, J. Fourier Anal. Appl., **4** (no. 3), pp. 247-269, 1998
8. ISO/IEC JTC1/SC29/WG11, “Coding of moving pictures and audio - MPEG-2 Advanced Audio Coding AAC,” ISO/IEC 13818-7 International Standard, 1997.

[P6] Wang, Y. “A Beat-Pattern based Error Concealment Scheme for Music Delivery with Burst Packet Loss”, IEEE International Conference on Multimedia and Expo (ICME2001, CD-ROM proceeding), August 22-25, 2001, Tokyo, Japan

A Beat-Pattern based Error Concealment Scheme for Music Delivery with Burst Packet Loss

Ye Wang
Speech and Audio Systems Laboratory
Nokia Research Center
P.O.Box 100, FIN-33721 Tampere, Finland

ABSTRACT

Error concealment is an important method to mitigate the degradation of the audio quality when compressed audio packets are lost in error prone channels, such as mobile Internet and digital audio broadcasting. This paper presents a novel error concealment scheme, which exploits the beat and rhythmic pattern of music signals. Preliminary simulations show significantly improved subjective sound quality in comparison with conventional methods in the case of burst packet losses. The new scheme is proposed as a complement to prior arts. It can be adopted to essentially all existing perceptual audio decoders such as an MP3 decoder for streaming music.

1. INTRODUCTION

The transmission of audio signals in compressed digital packet formats, such as MP3, has revolutionized the process of music distribution. Recent developments in this field have made possible the reception of streaming digital audio with handheld network communication devices, for example. However, with the increase in network traffic, there is often a loss of audio packets because of either congestion or excessive delay in the packet network, such as may occur in a best-effort based Internet.

Under severe conditions, for example, errors resulting from burst packet loss may occur which are beyond the capability of a conventional channel-coding correction method, particularly in wireless systems such as GSM, WCDMA or Bluetooth. Under such conditions, sound quality may be improved by the application of an error-concealment algorithm. Error concealment is an important process used to improve the quality of service (QoS) when a compressed audio bitstream is transmitted over an error-prone channel, such as found in mobile network communications and in digital audio broadcasts.

The focus of the new scheme is given to bitstream errors in the compressed domain, because a compressed domain bitstream, after removing most of the signal redundancy and irrelevance, is more sensitive to channel errors in comparison with an uncompressed domain bitstream.

With sufficient overhead and cost of the codec, it is theoretically possible to devise a perfect error detection and correction method. However, such a scheme would be impractical and undesirable. A practical error-correction method balances those limitations against the probability of uncorrected errors, and allows severe errors to remain uncorrected [1]. Then, error concealment methods are used as the last resort to mitigate the degradation of the audio quality in case of uncorrected errors.

In principle, all error concealment methods exploit the correlation of the signal and characteristics of human hearing to reduce the effects of uncorrected errors or packet losses. Since all perceptual audio codecs use frame-wise compression of audio signals, the new scheme is designed as segment-oriented error concealment in connection with an audio decoder.

Though error protection (detection/correction) and error concealment methods are closely related, they are, however, different concepts for tackling errors. A good system design should include [2][3]:

- Detailed analysis of the channel status and error pattern. This is the basis for choosing an appropriate error concealment strategy. The error detection is a prerequisite for error concealment.
- Careful consideration of the interdependency among channel coding, source coding and error concealment, in order to find the optimal trade-off between error resilience and bandwidth efficiency.

This paper presents a new error concealment scheme to exploit the beat and rhythmic pattern of music signals. This long-term time domain correlation has not been exploited in any existing perceptual audio-coding algorithm. Our preliminary simulation shows promising results in comparison with conventional methods in the case of long burst packet loss, which does happen now and then in the Internet [4] and Wireless LANs [5]. Even with one or two packet loss, the proposed method may produce better results than prior arts.

This paper is organized as follows. A brief review of the prior arts is given in section 2. Then our new concept and method are described in section 3. Some preliminary evaluations of the new scheme are presented in section 4. Finally, section 5 concludes the paper with some discussions and indicates some future work.

2. PREVIOUS METHODS

A lot of investigations into error concealment have been conducted during the development of a digital audio broadcasting (DAB) system within the EUREKA Project 147 [2][3][6]. A good summary of previous methods can be found in [7]. A more recent method can be found in [8].

The most relevant prior arts for error concealment employ small segments (typically around 20 ms) oriented concealment methods including: 1) muting, 2) repeating prior packet, 3) interpolation, and 4) time-scale modification. However, a fundamental limitation of conventional error concealment systems is that they all operate with the assumption that the audio signals are

stationary. Thus, if the lost or distorted portion of the audio signal includes a short transient signal, such as a 'beat,' the conventional system will not be able to recover the signal. This paper presents a first attempt to solve this problem by exploiting the beat and rhythmic pattern of music signals.

3. PROPOSED METHOD

The new error concealment scheme results from the observations that a music signal typically exhibits rhythm and beat characteristics, which do remain fairly constant. This is one of the most important features that makes the music flow unique and differentiates it from other audio signals.

A segment of audio data lost from one defined interval can be replaced by a segment of audio data from a corresponding preceding interval. By exploiting the beat pattern of music signals, error concealment performance can be significantly improved, especially in the case of long burst packet loss.

In western music, especially pop music, it is well known that beat patterns are composed of regularly spaced strong and weak beats. For the sake of brevity, we have considered only pop music with clear time signature of 4/4 in this paper. The block diagram of the proposed system is shown in Figure 1. An MP3 decoder is used to perform all simulations.

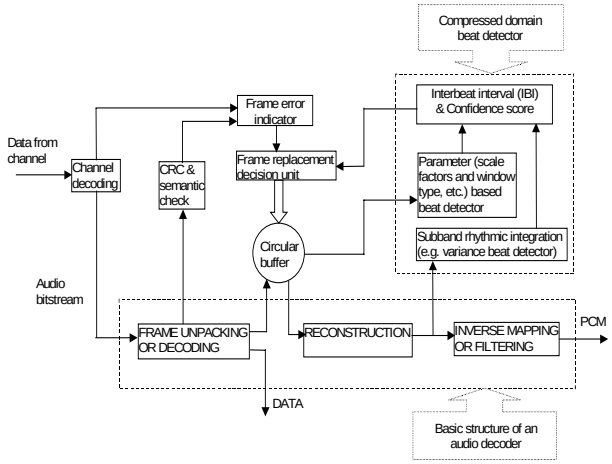

Figure 1. Block diagram of an extended audio decoder system including an error detection section, a compressed domain beat detector and a circular FIFO buffer in accordance with the proposed error concealment algorithm.

3.1 Frame Error Detection

The channel decoder is able to derive some information concerning the reliability of the received frames (status-bit). The Frame-CRC and semantic check in MPEG audio can also provide frame error information. In the case of packet-based network, the time stamp of the packet is a reliable cue for missing packets. The frame error indicator in Figure 1 analyzes the type, structure and duration of the errors. The determination of the suitable error concealment technique is based on these results. Optionally, the encoder should provide the determination and transmission of some concealment control information. It could directly point to the best error concealment strategy for a defined error situation [2].

3.2 Compressed Domain Beat Detector

Beat refers to a perceived pulse marking off equal duration units [9]. Beats are usually created by certain instruments such as drums and bass guitars.

The beat detector tries to determine the beat location, beat width and inter-beat interval. A detailed description of the compressed domain beat detector will be published elsewhere. The key ideas are summarized in this section.

In this paper, beat detection is accomplished by two methods. The more reliable method uses the energy of the music signal, which is derived from decoded Modified Discrete Cosine Transform (MDCT) coefficients available in an MP3 decoder. This method detects primarily strong beats. An adaptive statistical model is employed for improved detection accuracy. The second method uses a window-switching pattern to identify the beats present. The window-switching method detects both strong and weak beats. However, the window-switching method alone is not reliable, thus must be applied together with other more reliable methods.

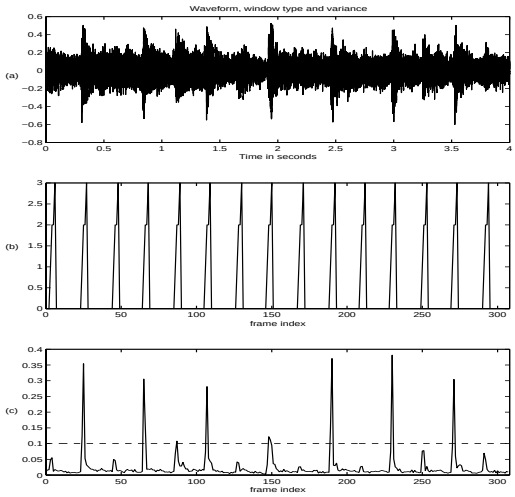

Figure 2. A sample of a pop music recording from ABBA. (a) time domain waveform, (b) window switching pattern, the vertical axis values indicate the window types: 0 – long window, 1 – long to short window, 2 – short window, 3 – short to long window, (c) energy of the signal and a threshold for beat detection.

In accordance with the energy method, the energy $EN(\tau)$ of the music signal at time τ is calculated directly by summing the squares of the decoded MDCT coefficients to give:

$$EN(\tau) = \sum_{j=0}^{575} [X_j(\tau)]^2 \quad (1)$$

where $X_j(\tau)$ is the j[th] MDCT coefficient decoded at time τ. The location of the beats are determined to be those places where $EN(\tau)$ exceeds a pre-determined threshold value (see dashed line in Figure 2 (c)).

A confidence score on beat detection is included to the audio decoder system in Figure 1 to prevent erroneous beat replacement. The confidence score measures how reliably beats can be detected within an observation window. Accordingly, a threshold value is specified. If the confidence score is above the threshold value, the beat replacement is enabled. Otherwise, the beat replacement is disabled.

3.3 Error Concealment

After the error type and duration has been determined, and the beat pattern has been detected, the error concealment is fairly straightforward.

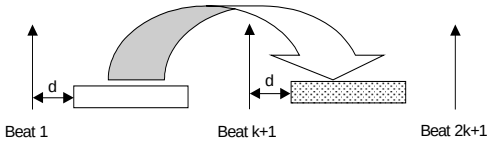

Figure 3. The replacement of an erroneous audio segment in an inter-beat interval using the system of Figure 1. k is a positive integer.

Figure 3 illustrates the replacement procedure. In this case, the audio frames making up the first inter-beat interval have been found error-free. If errors are detected between beat (k+1) and (2k+1) by the frame error indicator, the erroneous segment will be replaced by a corresponding segment from the first inter-beat interval as indicated by the arrow in Figure 3.

For music signals with time signature of 4/4, the error concealment can be performed in consecutive bars as indicated in Figure 4.

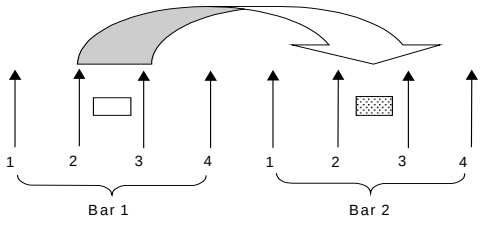

Figure 4. The replacement of an erroneous audio segment in a bar of music using the system of Figure 1.

The above error concealment configuration would require considerable memory consumption and delay in the decoder. In order to save memory and to restrict delay, an alternative configuration stores only selected audio frames around beats rather than every audio frame in the coming bitstream as illustrated in Figure 5. When the reduced memory capacity is used, only the beat structure is preserved. In this case, it is desirable to combine the new method and the conventional method to achieve better error concealment.

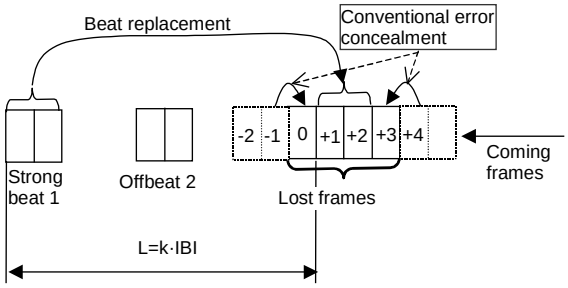

Figure 5. A scenario of burst error concealment with both the new and conventional methods. IBI indicates the inter-beat interval and k is a positive integer.

4. PRELIMINARY EVALUATIONS

As the aim of the concealment process itself is to avoid the degradations in signal quality perceived by the listener, the performance criterion can be formulated as the best restoration of the distorted signals in terms of subjective signal quality [7].

The proposed method belongs to the non-estimating algorithm, which does not attempt to give an optimum estimate but uses a replacement signal which is close enough to the original data in its structure [7].

The non-estimating technique substitutes a whole period of audio data by some other, more or less similar segment, which is available to the algorithm. Thus the processed signal is not intended to approximate the original one and a measurement of the output signal in respect to the reference signal makes no sense.

Nevertheless, we have performed some informal listening tests to evaluate the new algorithm in comparison to conventional ones.

Figure. 6 presents a comparison of the new method with some conventional methods. An error-free audio segment is represented in the top graph by two consecutive inter-beat intervals.

Consider an audio data loss between the two dotted lines, it corresponds to an interval approximately 520 ms in duration (i.e., approximately 20 MP3 audio data frames). Because most conventional error-concealment methods are not intended to deal with errors longer than one audio frame length used in the applied transfer protocol in duration, the conventional error concealment method will not produce satisfactory results. One

conventional approach, for example, is to mute the entire segment, as shown in the next graph. Unfortunately, this waveform will be objectionable to a listener as there is an abrupt transition, and the second strong beat is missing.

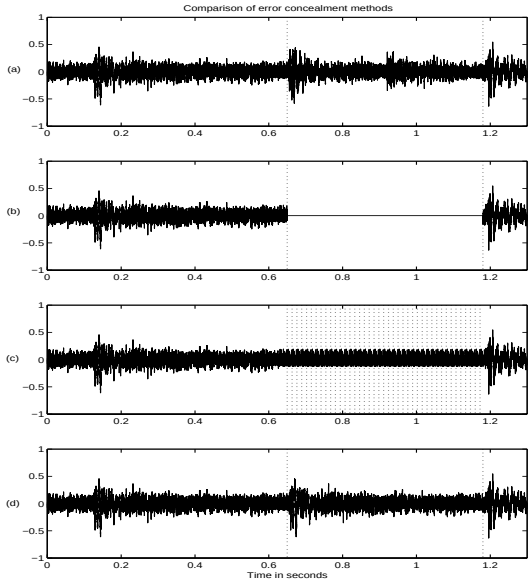

Figure 6. Comparison of the new error concealment method with some conventional methods. (a) waveform of original music signal; (b) muting the long burst errors between the two dotted lines; (c) repeating the previous MP3 frame in case 20 consecutive MP3 frames are lost; (d) beat-pattern based error concealment method.

In another conventional approach, shown in the underlying graph, an audio data frame occurring just before the lost segment is repeatedly copied and added to fill the entire interval, resulting in a monotonic waveform in Figure 6(c). This configuration will also be objectionable to a listener, as there is little if any musical content in the monotonic waveform, and the second strong beat is also missing.

The proposed method exploits the beat pattern knowledge and substitutes the missing audio segment from the previous inter-beat interval as shown in the bottom graph. By casual evaluation by some researchers at Nokia Research Center, the new method provides very promising results.

5. DISCUSSION

In this paper we have described a new error concealment technique for streaming music via error prone channels. The new method has demonstrated its capability to recover burst packet loss, which may include transient parts such as beats in music signals.

The experiments with different audio material have revealed that the current algorithm is quite effective for pure music signals with an obvious and constant beat pattern. If the signal does not have a clear beat structure or if speech and singing are considered, the current system cannot guarantee satisfactory results, because the beat-pattern does not give sufficient information about the similarity between different speech and singing segments. However, it can serve as a basis for future work in this direction.

Future research may include:

- The compressed domain beat detector employed in the proposed system may be generalized into a multi-band approach for improved beat detection.
- A segment similarity measure may be introduced in the encoder side in order to reduce the "blind" segment replacement. A good audio similarity measure should take not only music but also speech and singing sounds into account.
- An intelligent selection agent may be developed to choose the right error concealment method in a given error condition.
- Simulations with a realistic streaming channel such as mobile Internet to give some quantitative information about the subjective performance of the method.

6. ACKNOWLEDGEMENT

The author wishes to thank P. Haavisto, N. Courtis, M. Väänänen, M. Vaalgamaa, M. Vilermo, J. Tian, N. Zacharov, D. Isherwood, J. Ojanperä (Nokia Research Labs) for helpful discussions. Academy of Finland and Nokia Foundation are acknowledged for funding support.

7. REFERENCES

[1] K.C. Pohlmann, "Principles of Digital Audio," 3rd Ed., NY, McGraw-Hill, Inc. 1995

[2] D. Wiese, "Error Concealment Strategies for Digital Audio Broadcasting," 92nd AES Convention, Vienna, 1992, preprint 3264

[3] D. Wiese, "Error Concealment for DSB: Impact on Channel and Source Coding," 94th AES Convention, Berlin, 1993, preprint 3467

[4] J.C. Bolot, H. Crepin, A.V. Garcia, "Analysis of Audio Packet Loss in the Internet," Proc. of 5th Int. Workshop on Network and Operating System Support for Digital Audio and Video, pp. 163-174, Durham, April 1995

[5] http://www1.acm.org/sigs/sigmm/MM2000/ep/mckinley/

[6] M. Barberis, E. F. Schroeder, "Burst Error Concealment for Digital Audio Tape and Broadcast Application," 90th AES Convention, Paris, 1991, preprint 3012

[7] J. Herre, E. Eberlein, "Evaluation of Concealment Techniques for Compressed Digital Audio," 94th AES Convention, Berlin, 1993, preprint 3460

[8] A. Stenger, K.B. Younes, R. Reng, B. Girod, "A New Error Concealment Technique for Audio Transmission with Packet Loss," Proc. EUSIPCO-96, pp.1965-1968, Trieste, Italy, 1996

[9] W.J. Dowling, D.L. Harwood, "Music Cognition," Academic Press, 1986

[P7] Wang, Y., Vilermo, M. “A Compressed Domain Beat Detector using MP3 Audio Bitstream”, The 9th ACM International Multimedia Conference (ACM Multimedia 2001), September 30 – October 5, 2001, Ottawa, Ontario, Canada, pp. 194-202

A COMPRESSED DOMAIN BEAT DETECTOR USING MP3 AUDIO BITSTREAMS

Ye Wang, Miikka Vilermo
Speech and Audio Systems Laboratory
Nokia Research Center
P.O.Box 100
FIN-33721 Tampere, Finland
{ye.wang, miikka.vilermo}@nokia.com

ABSTRACT
This paper presents a novel beat detector that processes MPEG-1 Layer III (known as MP3) encoded audio bitstreams directly in the compressed domain. Most previous beat detection or tracking systems dealing with MIDI or PCM signals are not directly applicable to compressed audio bitstreams, such as MP3 bitstreams. We have developed the beat detector as a part of a beat-pattern based error concealment scheme for streaming music over error prone channels. Special effort was used to obtain a tailored trade-off between performance, complexity and memory consumption for this specific application. A comparison between the machine-detected results to the human annotation has shown that the proposed method correctly tracked beats in 4 out of 6 popular music test signals. The results were analyzed.

Keywords
Error concealment, Beat detection, Beat tracking, Compressed domain processing, Bitstream processing, MP3, MPEG audio.

1 INTRODUCTION

With rapid deployment of audio compression technologies, more and more audio content is stored and transmitted in compressed formats. The transmission of audio signals in compressed digital packet formats, such as MP3, has revolutionized the process of music distribution. Consequently, compressed bitstream processing is becoming a subject of study [1][2][3][4]. However, compressed domain bitstream processing is still in its infancy and many aspects such as beat detection remain unaddressed.

Beat detection or tracking is an important initial step in computer processing of music and is useful in various multimedia applications, such as automatic classification of music, content-based retrieval, audio track analysis in video, etc.

Beat detection or tracking systems can be classified according to the input data type. Most existing beat-tracking systems deal with musical score information (typically MIDI signals) [9][10][11], or PCM samples [12][13][14][15][16][17]. Some are designed for real-time applications.

None of the above-mentioned systems is directly applicable to a compressed domain bitstream such as MP3 bitstream, which has gained popularity not only in the Internet world, but also in consumer products. In addition, existing algorithms usually have such a high computational complexity that it is beyond the capability of a normal laptop computer (not to mention handheld devices) to perform a real-time application task – beat-pattern based error concealment for streaming music over error prone channels having burst packet losses [5]. Our objective here was not to develop a general-purpose beat detector, but to develop a beat tracking method as a building block of the error concealment system proposed in [5] with strict constraints on complexity and memory requirement. The proposed beat detector serves to segment music signals according to beats. The ultimate goal is to define a segment-similarity measure that relates closely to the subjective similarity, which will enable us to perform beat-pattern based error concealment and coding tasks better [5][7].

This paper is organized as follows. The concept of beat-pattern based error concealment is first outlined in section 2. It serves to clarify the necessity and requirements of such a beat tracking method. A window-type based beat detector is presented separately in section 3 due to its importance on error concealment. A Modified Discrete Cosine Transform (MDCT) domain beat detector is then detailed in section 4. Some preliminary evaluations of the new scheme are presented in section 5. Finally, section 6 concludes the

paper with some discussions and indicates some future work.

2 CONCEPT OF BEAT-PATTERN BASED ERROR CONCEALMENT

Error concealment usually serves as the last resort to mitigate the degradation of the audio quality when compressed audio packets are lost in error prone channels, such as mobile Internet and digital audio broadcasts.

Conventional error concealment methods include muting, interpolation or simply repeating a short segment immediately preceding the lost segment. They are useful if the lost segment is short (an usual assumption in the literature is around 20 ms) and the signal is fairly stationary. However, if these conditions do not hold, conventional methods will not produce satisfactory results.

To solve this problem, a new scheme was proposed to exploit the beat-pattern similarity of music signals to recover a possible burst packet loss in a best-effort based network such as the Internet. [5].

The beat-pattern based error concealment scheme results from the observations that a music signal typically exhibits rhythm and beat characteristics. And the beat-patterns of most music, particularly pop, march and dance music are fairly stable and repetitive.

The time signature of pop music is typically 4/4. The average inter-beat interval (IBI) is about 500 ms, thus the duration of a bar is about 2 s. Such long-term similarity of music has not been exploited in any existing audio coding technology. The concept is quite simple and straightforward. If the lost or distorted segment of the audio signal includes a beat, it would be better to replace it with a segment from a previous beat.

A conventional error concealment method and our new approach are illustrated in Figure 1 and 2 respectively. The small segments represent MP3 granules. An MP3 frame consists of two granules where each granule consists of 576 frequency components.

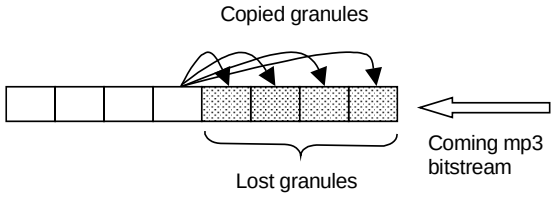

Figure 1. Illustration of a conventional error concealment method. Rectangles filled with dots represent corrupted MP3 granules. Blank rectangles represent error-free ones. The thin arrows indicate the repetitive copy of the immediately preceding granule to fill the erroneous audio segment.

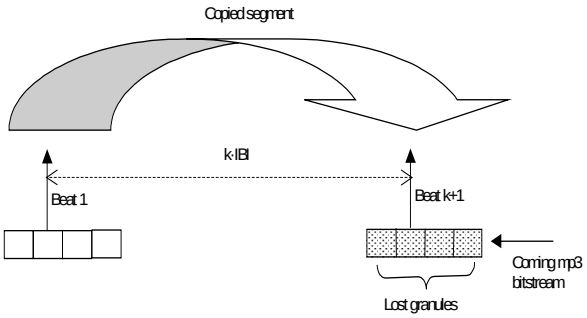

Figure 2. Concept of the beat-pattern based error concealment method. It replaces an erroneous audio segment around beat (k+1) with a corresponding segment from a previous beat as indicated by the thick arrow. k is a positive integer which is determined by the employed level of beat information (e.g. quarter-note or half-note level). IBI stands for inter-beat interval. Rectangles filled with dots indicate corrupted MP3 granules. Blank rectangles indicate error-free ones.

The assumption for this approach is that a segment around a beat, which often corresponds to a transient produced by a rhythmic instrument such as a drum, is subjectively more similar to a segment around a previous beat than its immediate neighboring segment. A possible psychological verification of this assumption is explained by the following example. If we observe typical pop music with a drum sound marking the beat in a 3-D time-frequency representation (see Figure 6), the drum sound usually appears as a ridge, short in the time domain and broad in the frequency domain, which masks all other sounds such as singing and other instruments well. It is usually so dominant in pop music that one perceives only the drum sound during the event. In spite of some variations in consecutive drum sounds, it is logical to propose that it would be subjectively more pleasant to replace a missing drum sound with a previous drum sound segment rather than with any other sound, such as singing. It becomes evident from this that a beat detector is a crucial element of the scheme. And it is reasonable to perform the beat detection directly in the compressed domains to avoid redundant operations.

The requirement of such a beat detector depends on the constraint on computational complexity and memory consumption. In our current implementation, the beat detector employs only the window types and the MDCT coefficients decoded from the MP3 bitstream to perform beat tracking. It outputs 3 parameters: beat position, IBI and confidence score. However, if the constraint on complexity and memory were relaxed, higher level structure (e.g. bar-level structure) would improve error

concealment performance.

3 WINDOW TYPE BASED BEAT DETECTION

MP3 uses 4 different window types: long, long-to-short, short and short-to-long which are indexed with 0, 1, 2, 3 respectively (see Figure 3(b)). The short window is introduced to tackle transient signals better. From our experiments with pop music, short windows often coincide with beats and offbeats since they are the most frequent events to trigger window-switching. We have observed that 99% of the window-switching patterns in all of our test signals appear in the following order: long => long-to-short => short => short => short-to-long => long. This pattern can be indexed as a sequence of 012230 (see Figure 3(b)).

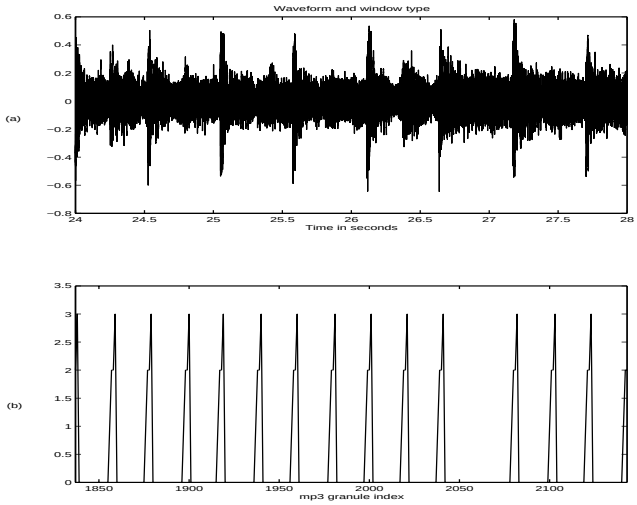

Figure 3. Comparison of music waveform and its corresponding mp3 window-switching pattern. (a) music waveform versus time in seconds, (b) window types (vertical axis) versus mp3 granule index (horizontal axis). The four window types (long, long-to-short, short and short-to-long) are indexed with 0, 1, 2, 3 respectively.

It should be noted that the window-switching pattern depends not only on the encoder implementation, but also on the applied bitrate. Therefore, window-switching alone is not a reliable cue for beat detection. For general purpose beat detection, we could even completely discard the window type information. A MDCT based method alone would be sufficient.

However, for error concealment purposes window type information plays an important role. Therefore, we take the following strategy to handle the beat information from the two separate sources. The MDCT based method serves as the baseline beat detector due to its reliability. Then the beat information (position and IBI) is checked with the window-switching pattern. If the window-switching also indicates a beat and its position departs from the MDCT based one less than 4 MP3 granules (ca. 13x4 = 52 ms), we take the beat information from the window-switching and adjust the beat information accordingly. That is, the window-switching method always has priority. The beat information from MDCT based method is used only in the absence of window-switching (see Figure 3, 5 and 6).

The rational of this strategy is that the window shapes in all MDCT based audio codecs including MPEG-2/4 advance audio coding (AAC) must satisfy certain conditions to achieve time domain alias cancellation (TDAC) [6]. If these conditions are violated due to the error concealment operation, the time domain alias will not be able to cancel each other during the overlap-add (OA) operation [6]. This will result in clearly audible distortion as a consequence.

For example, if the two consecutive short window granules indexed as 22 in a window-switching sequence of 012230 are lost in a transmission channel, it is easy to deduce their window types from their neighboring granules. And a previous short window granule pair should replace them to mitigate the subjective degradation. However, if we disregard the window-switching information available from the audio bitstream and replace the short window with any other neighboring window types, resulting in window-switching patterns such as 011130, the TDAC conditions will be violated. This will create annoying artifacts. We define the phenomenon as *window type mismatch phenomenon*.

To our best knowledge, this important issue has not been addressed in any publications to date.

Let's consider the same example of a MP3 granule sequence of 012230 as discussed above. In case a segment of four consecutive granules indexed as 1223 is *partially* corrupted in a communication channel, it is still possible to detect the transient, if we can correctly decode only the window type information (2 bits) of one *single* granule in the segment of four consecutive granules, even if their main data is totally corrupted.

The above analysis clearly suggests why *partially* damaged audio packets due to channel error should not be simply discarded because they can still be utilized to improve quality of service (QoS) in applications such as streaming music. This clarifies the significance of the window type information and the rational of our strategy to combine beat information from the two separate detection methods.

4 MDCT DOMAIN BEAT DETECTION

The MDCT coefficients based method has the following building blocks (see Figure 4 and 5):

- Feature Vector (FV) calculation: calculates the multi-band energy within each granule (ca. 13 ms) as a feature, and then forms a FV of each band within a search window. FV serves to separate beats and non-beats as much as possible. An element to mean ratio (EMR) can be used to improve the feature quality.

- Beat candidate selection: This process is performed in two stages. Beat candidates are first selected in individual bands based on a threshold method in a given search window. Within each search window the number of candidates in each band is either one or zero. If there are one or more valid candidates selected from individual bands, they are then clustered and converged to a single candidate according to certain criteria.
- Confidence score: A confidence score is calculated for each beat candidate from individual bands to score their reliability. Based on them, a final confidence score is calculated, which is used to determine whether a converged candidate is a beat.
- Statistical model: An inter-onset interval (IOI) histogram is usually employed to select the correct inter-beat interval (IBI) [13]. The idea is to use the IBI derived from the IOI histogram to predict the next beat. In our system a valid candidate in each individual band is defined as an onset. A set of previous IOIs in each band is stored in a FIFO for computing the candidate's confidence score of that band. Instead of a usual histogram approach, our statistical model employs a median in the FIFO buffer to predict the position of the next beat, which works quite well.
- Mark and output beat information: Before a beat candidate is finally marked and stored as a beat, it has to pass a confidence test. Only a candidate with sufficient confidence is selected as a beat (see Figure 9). Its position, IBI and confidence score are stored and also fed back to calculate the confidence score of future beat candidates. This beat information then is checked with the window-switching information, adjusted accordingly.

A high-level block diagram of the MDCT domain beat detector is illustrated in Figure 4. More detailed information about each block is given in the subsequent sub-sections.

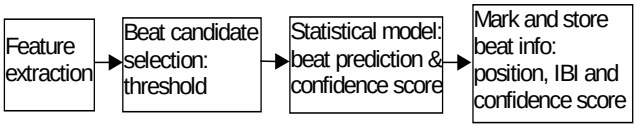

Figure 4. Block diagram of a MDCT based beat detector

Feature Extraction

We use subband energy or EMR of the subband energy in a search window as a feature vector (FV). The FV is directly calculated from decoded MDCT coefficients as illustrated in Figure 5. We chose an approach, which extracts FV from the full-band and individual subbands separately to avoid possible loss of information. The frequency boundaries of the new subbands are defined in table 1 and 2 for long and short windows respectively for a sampling frequency of 44.1 kHz. For other sampling frequencies the subbands can be defined in a similar manner.

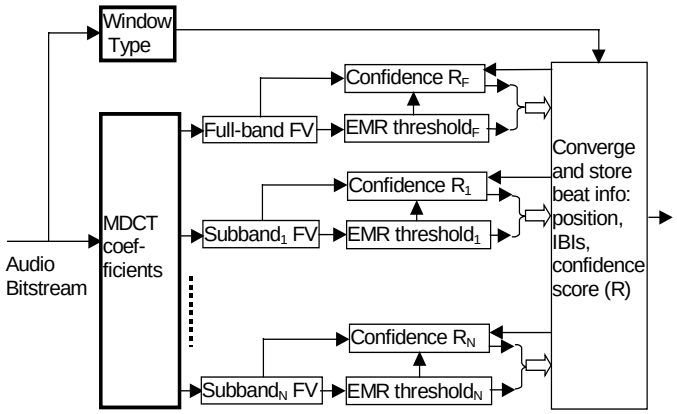

Figure 5. Block diagram of a compressed domain beat detector using MP3 bitstream. FV stands for feature vector.

Sub-band	Frequency interval (Hz)	Index of MDCT coefficients	Scale factor band index
1	0-459	0-11	0-2
2	460-918	12-23	3-5
3	919-1337	24-35	6-7
4	1338-3404	36-89	8-12
5	3405-7462	90-195	13-16
6	7463-22050	196-575	17-21

Table 1. Subband division for long windows

Sub-band	Frequency interval (Hz)	Index of MDCT coefficients	Scale factor band index
1	0-459	0-3	0
2	460-918	4-7	1
3	919-1337	8-11	2
4	1338-3404	12-29	3-5
5	3405-7465	30-65	6-8
6	7463-22050	66-191	9-12

Table 2. Subband division for short windows

MP3 employs a hybrid filterbank. In principle, the feature extraction can also be performed after an Inverse Modified Discrete Cosine Transform (IMDCT) step [8]. We chose the decoded MDCT coefficients for feature extraction, in order to make the algorithm more general and applicable to

other codecs such as MPEG2/4 AAC, which uses only a MDCT.

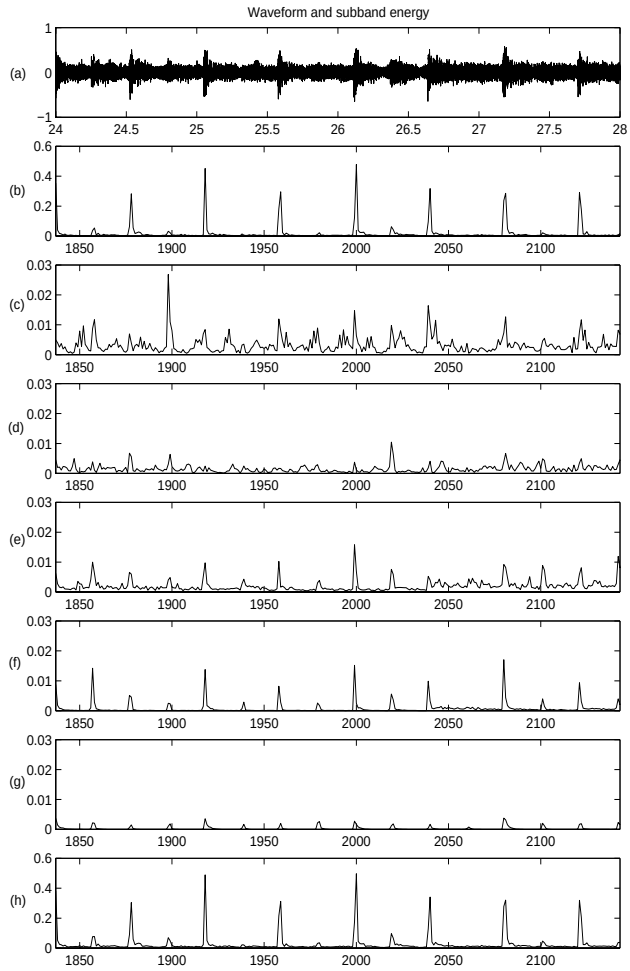

Figure 6. Music waveform of a 4 seconds segment and its corresponding subband energy employing the same pop music sample as in Figure 3. (a) music waveform versus time in seconds, (b)-(h) energy in subbands 1-6 and full-band versus mp3 granule index.

MP3 has an option to use long or short windows. The window length is 36 subband samples in the case of long windows and 12 subband samples in the case of short window. 50% window overlap is used in the MDCT. In order to have a consistent frequency resolution for both long and short windows we grouped the MDCT coefficients of each granule into 6 newly defined subbands (see tables 1 and 2) for feature extraction. For other codecs or configurations, similar frequency divisions can be performed. This frequency division is different in comparison to most previous beat detectors due to the constraint of the MPEG standard and system complexity.

In Figure 5, each band gives only one value by summation of the energy within a granule [8]. Thus the time resolution of our beat detector is one MP3 granule (ca. 13 ms) as opposed to a theoretical beat event, which has no duration.

The energy $E_b(n)$ of band b in granule n is calculated directly by summing the squares of the decoded MDCT coefficients to give:

$$E_b(n) = \sum_{j=N1}^{N2} [X_j(n)]^2 \qquad (1)$$

where $X_j(n)$ is the j[th] normalized MDCT coefficient decoded at granule n, $N1$ is the lower bound index and $N2$ is the higher bound index of MDCT coefficients defined in Table 1 and 2. Since the feature extraction is performed in granule level, the energy in three short windows (equal to one long window in duration) is combined into one so that we have comparable energy for both long and short windows.

Based on the observations of the extracted features from different pop music, we have concluded that subbands 1, 5, 6 and the full-band features are generally reliable for pop music beat tracking. The features extracted from a pop music extract are illustrated in Figure 6.

For simplicity our current system only uses these 4 bands to extract the feature vector. The reason that subbands 2, 3 and 4 usually give rather poor features is that singing and instruments other than drums are mostly concentrated in these bands. Consequently, the beat and non-beat separation is usually rather difficult in these bands. As illustrated in Figure 6, feature vectors are extracted in multiple bands and then processed separately.

Search window

The search window size determines the FV size, which is used for selecting beat candidates in individual bands. The search window size can be fixed or adaptive. Based on our experiments both methods are feasible. In the case of the fixed window size, the minimal possible IBI (~325 ms) is chosen as the search window size so that the maximal number of possible beats within the search window is one. The current system uses an adaptive window size because of its slightly better performance. It is calculated as the closest odd integer to the median of the stored IOIs, so that we have a symmetric window around a valid sample:

$$window\_size\_new = 2\lfloor median(\overline{IOI})/2 \rfloor + 1 \qquad (2)$$

The hop size is selected to be half of the new search window size.

$$hop\_size\_new = round(window\_size\_new/2) \qquad (3)$$

Beat candidate selection

The basic principle of beat candidate selection is setting a proper threshold for the extracted FV. The local maxima within a search window, which fulfils certain conditions, are selected to be beat candidates. This process is performed in each band separately. There are two threshold-based approaches for selecting beat candidates. The first approach uses the primitive FV (multi-band energy) directly and the second approach uses an improved FV (EMR).

The first method is based on the absolute value of the multi-band energy of beats and non-beats. A threshold is set based on the distribution of beat and non-beat for selecting beat candidates within the search window. This approach is computationally simple but needs some knowledge of the feature in order to set a proper threshold. It has three possible outputs in the search window: no candidate, one candidate or multiple candidates. In the case of one or multiple candidates, it is desirable to have a subsequent statistical model to determine the reliability of each candidate as a beat. The beat detector in [5] was based on this method.

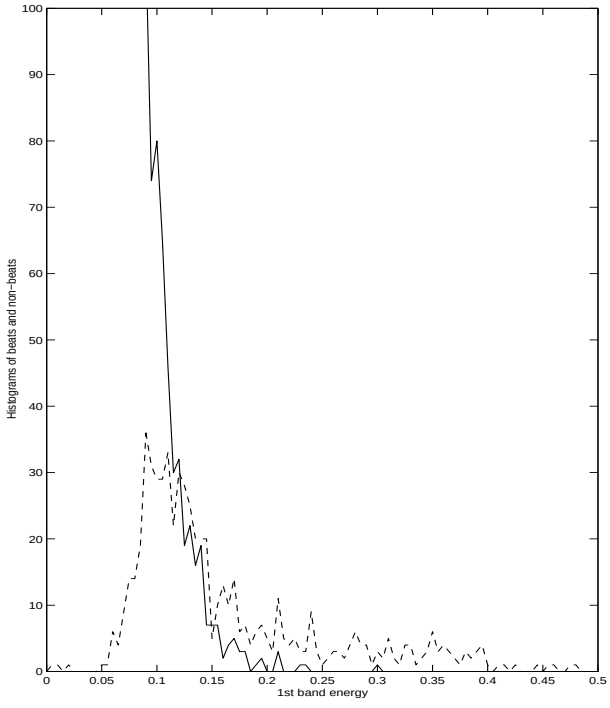

Figure 7. Histogram of beats (dashed line) and non-beats (solid line) versus their first-band energy (feature vector extracted with the first method) employing a pop music sample (6 minutes in duration).

The second method uses the primitive FV to calculate an EMR within the search window to form a new FV. That is, we calculate the ratio of each element (energy in each granule) to the mean value (average energy in the search window). And then the maximum EMR is compared with a given threshold. If the EMR is greater than the threshold, this local maximum is selected as a beat candidate. The beat candidate is sent to the next stage for further processing as illustrated in Figure 5.

The second approach seems to be superior to the first approach in most cases since it measures the relative distance between the individual element and the mean, not their absolute values. Therefore, the EMR threshold can be set as a constant value, while the threshold in the first method should be adaptive to cope with the wide dynamic range in music signals. EMR is used in our current implementation. Comparison of the two methods with an identical sample (6 minutes in duration) is shown in Figures 7 and 8. The beats were picked up manually by a human subject. Although the EMR method has slightly better separation between beats and non-beats, none of the two FVs is good enough to separate beats and non-beats reliably without a subsequent statistical model. The wide signal dynamic range and relatively strong offbeats mainly cause the bad separation.

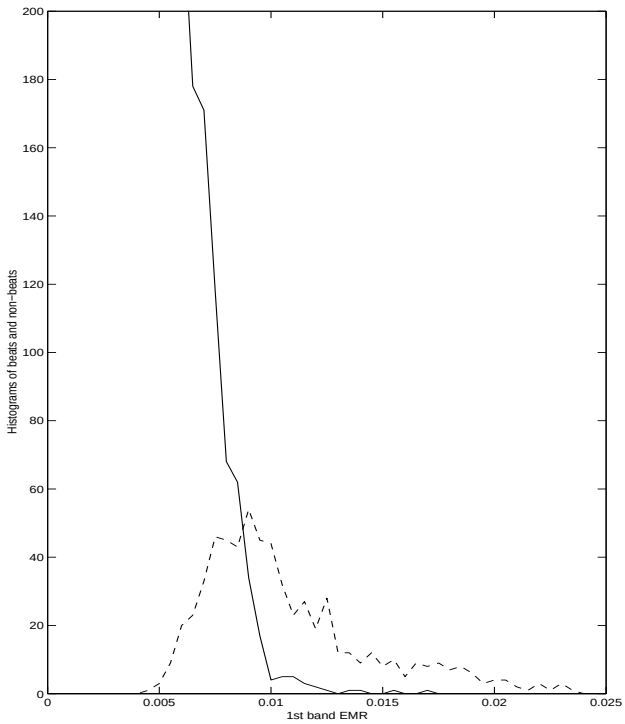

Figure 8. Histogram of beats (dashed line) and non-beats

(solid line) versus their EMR measure (feature vector extracted with the second method) employing the same pop music sample as in Figure 7.

In order not to miss a possible beat, we were forced to set a threshold towards the lower end of the beat population, which is about 0.005 in Figure 8. Thus the probability of selecting non-beats as beat candidates is rather high. Subsequent statistical models will eventually remove false selections.

Statistical model

We define a valid candidate in each band as an onset and store a number of previous IOI values in a FIFO buffer for beat prediction in each band. Then we use the median of the IOI vector to calculate the confidence scores of all beat candidates in individual bands. This simple statistical model has proven to be quite effective.

The IOI vector size is a tunable parameter for adjusting the responsiveness of the beat detector. If the IOI vector size is kept small, the beat detector is quick to adapt to a changed tempo at the cost of instability. If the IOI vector size is large, it becomes slow to adapt to a changed tempo, but it can tackle more difficult situations better. In the current implementation, the FIFO buffer size is 9. Since we store the IOI as opposed to the final IBI in the buffer, the tempo change is registered in the FIFO. However, the search window size is only updated to follow the new tempo after 4 IBIs, which is about 2~3 seconds in duration.

Confidence score

The confidence score for an individual beat candidate is calculated to measure its reliability:

$$R_i = \max_{k=1,2,3} \left\{ \frac{median\left(\overline{IOI}\right)}{median\left(\overline{IOI}\right) + \left| median\left(\overline{IOI}\right) - \frac{\left(I_i - I_{last\_beat}\right)}{k} \right|} \right\} \cdot f\left(E_i\right) \quad (4)$$

where k =1, 2, 3. k is introduced to cope with the situation that the current *IOI* is 2 or 3 times longer than the predicted value due to a decreased tempo or a missed candidate. $\overline{IOI}$ is a vector of previous inter-onset intervals. The size of $\overline{IOI}$ is an odd number. $median\left(\overline{IOI}\right)$ is used as a prediction of the current beat. i is the current beat candidate index. I_i is the MP3 granule index of the current beat candidate. I_{last_beat} is the MP3 granule index of the previous beat.

$$f\left(E_i\right) = \begin{cases} 0, & E_i < threshold_i \\ 1, & E_i \geq threshold_i \end{cases} \quad (5)$$

where E_i is energy of each candidate. $f\left(E_i\right)$ is introduced to discard candidates having too low energy.

The confidence score of the converged beat stream R is calculated by

$$R = \max\left\{R_F, R_1, \cdots, R_N\right\} \quad (6)$$

where N indicates the number of subbands and F indicates the full-band. The dashed line in Figure 9 shows the converged confidence score of a pop music, which is used to reject non-beats.

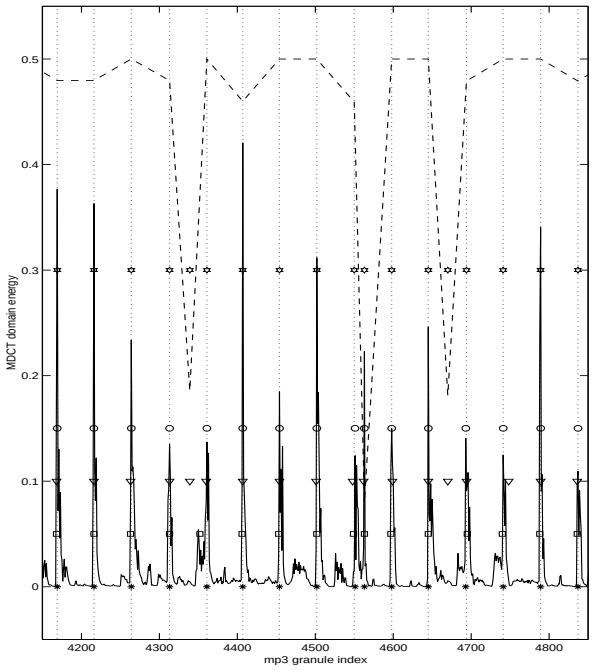

Figure 9. Multi-band features of a pop music sample: full-band energy (solid line), candidates from subband 1 (star), subband 5 (squares), subband 6 (triangles), and full-band (circles), converged beat candidates (hexagram), detected beats (dotted lines). The dashed line indicates the confidence score of the converged beat candidates, which is used to discard non-beats at this stage. For illustration purposes, the confidence score is shifted downwards by 0.5.

Converge and store beat information

Beat candidates together with their confidence scores from all the bands are converged. The candidate that has the greatest confidence score within a search window is selected as a center point. If candidates from other bands are close to the selected center point (less than 4 MP3 granules, for example), they are clustered. The confidence of a cluster is the maximum confidence of its members and the location of the cluster is the rounded mean of all locations of its members. All other candidates are ignored. As the final step, the candidate is accepted as a beat if its final confidence score is above a constant threshold. Beat position, IBI, and overall confidence score are sent to the application module after checking with the window

switching pattern.

5 PRELIMINARY EXPERIMENTAL RESULTS

We tested the proposed beat detector on 6 popular songs with durations from 1 to 6 minutes. The input was monaural audio signals sampled from a few commercial compact discs. The signals are then compressed using a MP3 encoder at bitrate of 64 ~ 96 kbps.

The system utilizes some basic musical knowledge to track beats at the quarter-note level. It assumes that beats generally have more energy than offbeats and the tempo is constrained to be between 50 and 180 M.M. (Mälzel's Metronome: the number of quarter notes per minute) and is roughly constant. Since the sampling frequency of all test sounds is 44.1 kHz, the MP3 granule length is 576, the time resolution of our system is ~ 13 ms (=576/44.1).

The beat annotation of the 6 test samples were performed by the second author, who is a M.Sc. student at Tampere University of Technology and a violinist at Tampere Conservatoire. The reason to choose a musician for beat annotation was that we wanted a more precise and consistent result.

The machine-detected results were then compared to human annotation. The criteria to count a failure were: (1) if the detected beat position departs from the annotation by or more than 4 MP3 granules that is ~ 52 ms; (2) if the algorithm simply fails to detect a beat; (3) if the algorithm picks up a non-beat as a beat.

The proposed method correctly tracked beats in 4 out of 6 popular music test signals.

We found that the algorithm worked almost without error if there was a simple strong bass drum pattern marking the beat such as songs from the band ABBA. However, the algorithm failed completely, if there was no clear drum beat or the beat pattern was rather complex. The algorithm often made some mistakes at the beginning of each music sample due to irregularity of the intro and at the end of each sample where the signal was fading away. If we disregard the beginning and the end for a few IBIs, the algorithm made only one mistake (missed one beat) during a 6-minutes test song, for example.

We discuss the reason why the beat tracker fails completely in two of the test samples. The algorithm relies on the assumption that the actual beats are in general stronger than the offbeats for example. If this assumption does not hold, the algorithm fails since it does not use any advanced musical knowledge. In particular, one failed sample has no drums and uses only a synthesized shaker sound marking beats. Another failed sample has rather complex beat-pattern. The bass drum beat varies a lot mixed with snare drum and hi-hats.

These preliminary results show that the proposed algorithm can deal with realistic music signals. However, some improvements are still necessary.

6 DISCUSSION

A beat tracking system is developed as a building block of an error concealment scheme. It should be noted that the error concealment and beat detection concept could be easily applied to cope with other audio bitstreams with minor modifications.

The beat detector was implemented partly in C and partly in Matlab. Its memory consumption and computational complexity are modest.

Essentially, the beat detection and error concealment are still two separate tasks. In order to reduce the complexity of the decoder, it might be a better alternative to implement the beat detector in the encoder side and to embed the current beat information in its preceding beat as ancillary data in the MP3 bitstream. The decoder could then directly use the beat information for error concealment. In this way we not only reduce the complexity of the decoder but also know with certainty whether the missing segment has a beat or not from the embedded beat information from its previous beat. Otherwise the decoder would have difficulties to guess a beat with damaged packets.

The algorithm does not work with signals such as speech and classic music. It is just intended for pop music with quite regular beat structure, which is an important class of music in streaming applications.

We believe that it may be a better option to use the 32-subband signal for beat detection instead of the 576-MDCT coefficients of a MP3 granule. This will not only improve the time resolution but also avoid the alias introduced by MDCT [6]. After all a beat is more a temporal phenomenon.

The current implementation is clearly an application oriented work in nature. We intend to port a good PCM domain beat tracker into the compressed domain to examine its performance shift.

The implemented system is still in its early version. There are many avenues open for further work. Because of its ad hoc nature, all major building blocks (e.g. the subband division, the statistical model and the confidence score) can be further optimized.

Another possible extension is to include more high-level musical knowledge into the system for better performance at the expense of complexity.

For applications other than error concealment, some modifications and optimizations may be necessary to satisfy the specific requirements.

ACKNOWLEDGEMENTS

The Academy of Finland and Nokia Foundation are

acknowledged for providing the first author scholarships which enabled him to conduct a major part of this research at Department of Experimental Psychology, University of Cambridge, UK under supervision of Prof. Brian C.J. Moore. We thank Mr. Juha Ojanpera for assistance with programming, and Dr. Jilei Tian, Mr. Jarno Seppänen, Prof. Anibal Ferreira, Prof. Brian C.J. Moore and three anonymous reviewers for helpful comments on an earlier version of this paper.

REFERENCES

1. Patel, N.V., Sethi, I.K. "Audio Characterization for Video Indexing", Proc. SPIE Vol. 2670, Storage and Retrieval for Image and Video Databases IV, Jan/Feb 1996, San Jose, CA, USA, pp. 373-384.
2. Nakajima, Y., Lu, Y., Sugano, M., Yoneyama, A., Yanagihara, H., Kurematsu, A. "A Fast Classification from MPEG Coded Data", Proc. of International Conference on Acoustic, Speech and Signal Processing (ICASSP), 1999, Phoenix, Arizona, USA, pp.3005-3008.
3. Boccignone, G., DeSanto, M., Percannella, G. "Joint Audio-Video Processing of MPEG Encoded Sequences", Proc. IEEE Intl Conf. On Multimedia Computing and Systems (ICMCS99), 1999, pp.225-229.
4. Pfeiffer, S., Robert-Ribes, J.; Kim, D. "Audio Content Extraction from MPEG-encoded sequences", First International Workshop on Intelligent Multimedia Computing and Networking (IMMCN2000), Feb./March 2000, Atlantic City, New Jersey, pp. 513-516.
5. Wang, Y., "A Beat-Pattern based Error Concealment Scheme for Music Delivery with Burst Packet Loss", accepted by IEEE International Conference on Multimedia and Expo (ICME2001), August, 2001, Tokyo, Japan.
6. Wang, Y., Vilermo, M., Isherwood, D. "The Impact of the Relationship Between MDCT and DFT on Audio Compression: A Step Towards Solving the Mismatch", The First IEEE Pacific-Rim Conference on Multimedia (IEEE-PCM2000), December 13-15, 2000, Sydney, Australia, pp. 130-138.
7. Wang, Y., Ojanpera, J., Vilermo, M., Vaananen, M. "Schemes for Re-compressing MP3 Audio Bitstreams", accepted by the Audio Engineering Society (AES) 111th International Convention, September 21-24, 2001, New York, USA.
8. ISO/IEC 11172-3, "Information Technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s", 1993.
9. Dannenberg, R.B., Mont-Reynaud, B., "Following an improvisation in real time," Proc. Int. Comp. Music Conf., 1987, pp.241-248.
10. Desain, P., Honing, H., "Advanced issues in beat induction modeling: syncopation, tempo and timing," Proc. Int. Comp. Music Conf., 1994, pp. 92-94.
11. Rosenthal, D., "Machine Rhythm: Computer Emulation of Human Rhythm Perception," PhD thesis, MIT, 1992.
12. Scheirer, E.D., "Tempo and beat analysis of acoustic musical signals," J. Acousti. Soc. Am., 1998, vol. 103, no.1, pp. 588-601.
13. Goto, M., Muraoka, Y. "Music understanding at the beat level: Real-time beat tracking for audio signals", in "Computational Auditory Scene Analysis", edited by Rosenthal D. and Okuno H., 1998, Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, pp. 157-176.
14. Todd, N.P.M., "The auditory 'primal sketch': A multi-scale model of rhythmic grouping," J. New Music Research, 1994, vol. 23, no. 1, pp.25-70.
15. Smith, L.M., "A multi-resolution time-frequency analysis and interpretation of musical rhythm," PhD thesis, University of Western Australia, 1999.
16. Dixon, S.E., "A beat tracking system for audio signal," Proc. Conf. Computat. And Mathemat. Methods in Music, Vienna, Austria, 1999, pp.101-110.
17. Klapuri, A., "Sound onset detection by applying psychoacoustic knowledge," Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc., 1999, vol. 6, pp. 3089-3092.

[P8] Wang, Y., Ojanperä, J., Vilermo, M., Väänänen, M. “Schemes for Re-Compressing MP3 Audio Bitstreams”, accepted by the AES111th International Convention, November 30 - December 3, 2001, New York, USA

Audio Engineering Society
Convention Paper
Presented at the 111th Convention
2001 September 21–24 New York, NY, USA

Schemes for Re-Compressing MP3 Audio Bitstreams

Ye Wang, Juha Ojanperä, Miikka Vilermo, Mauri Väänänen
Speech and Audio Systems Laboratory
Nokia Research Center
P.O.Box 100 (Visiokatu 1)
FIN-33721 Tampere, Finland
{ye.wang, juha.ojanpera, miikka.vilermo, mauri.vaananen}@nokia.com

ABSTRACT

This paper presents three schemes for re-compressing MP3 (MPEG-1 Layer III) audio bitstreams. The first two schemes are lossless ones, which exploit the inter-frame redundancies of the main data (the scale factors and the quantized MDCT coefficients) of the MP3 bitstream. The third scheme is a lossy approach, which exploit the redundancies between consecutive beat-patterns. The aim is to study the potential of the new coding schemes. Preliminary results are demonstrated in this paper.

INTRODUCTION

Since the acceptance of the ISO MPEG-1 international standard [1], MPEG audio coding has been used in a variety of applications for transmission and storage of high quality digital audio. The MPEG-1 Layer III, commonly known as MP3, is the most sophisticated coding method offered by MPEG-1 and it has shaken the traditional model of audio distribution. However, as an international standard, MP3 has been designed for various applications, thus the frame sizes are kept small and redundancies between samples in neighboring frames are not effectively exploited. An algorithm to exploit such redundancies in quantized MDCT domain is reported in [2].

The necessity to develop new algorithms on top of MP3 is based on the consideration of the following scenarios:

1) Downloading MP3 files using an analogue modem is a time-consuming process. If music delivery via wireless channels is considered, it is even more necessary to have a better coding scheme for bandwidth efficiency.

2) In addition to the use of MP3 files on PCs, portable MP3 players have been introduced to the market, which store MP3 files on a flash memory card. The limited size of flash memory cards (typically 32 MB or 64 MB) places a limit on the amount of audio, which can be stored on the device.

These are the motivations for us to investigate better coding algorithms, which can result in reduced file size and in more efficient use of channel or memory capacity.

Analysis of MP3

MP3 uses a bit reservoir technique that smoothes the bitstream fluctuation to meet the requirement of a certain channel capacity and thus makes a little inter-frame dependency. Bit reservoir does not compress the amount of data but only changes the distribution of bits among consecutive frames. What we have proposed in this paper is a compression algorithm exploiting the inter-frame redundancies by increasing the delay and the actual frame length.

The main reason why the inter-frame redundancy has not been exploited in MPEG-1 Layer III is the need to keep the frame size small and to lower the computational complexity of the decoder [2]. The advantages of using smaller frames that usually do not

match the natural characteristics of music signals, such as beat pattern and verse, are:
1) Less buffer memory (RAM) required in the decoder.
2) Lower computational complexity.
3) Increased robustness to channel errors.
4) Lower coding delay.
However, the falling price/capacity of memory (RAM) and CPU means that memory and computational complexity have become less of an issue. Additionally, robustness to channel errors and low coding delay are less critical in some applications. The coding can be performed offline and the files are reliably transmitted over Internet or saved in the flash memory.
Scheme 1 & 2 are lossless methods. Scheme 3 is a lossy method, which aims to achieve bits reduction with little additional degradation of the subjective audio quality.

PROPOSED SCHEME 1: LONG-TERM ZERO-ORDER PREDICTION + FIXED LENGTH CODING FOR THE SCALE FACTORS

In this section we propose a coding scheme for lossless encoding of MPEG-1 layer III encoded scale factors in the main data.
In MP3 bitstream, the main data consists of scale factors and quantized MDCT coefficients. The scheme results from the observation that patterns of the scale factors in consecutive granules are quite similar (see Fig. 1).

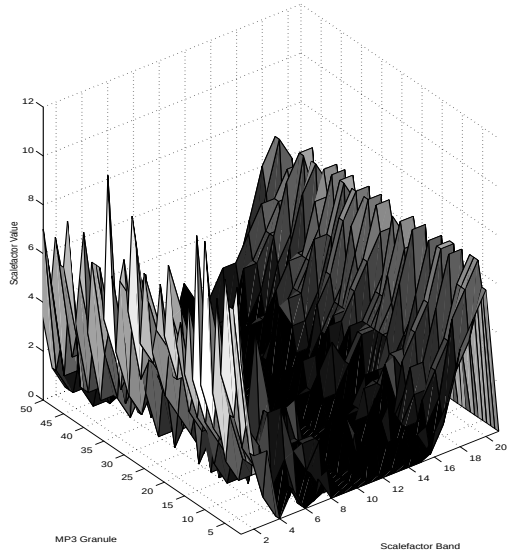

Fig. 1. Scale factors in consecutive granules. Test signal is a piece of pop music.

In order to exploit the inter-granule redundancy, we have developed a long-term zero-order prediction (LTZP) algorithm to re-compress the MP3 scale factors. The residual signal after the LTZP is shown in Fig. 2 using the same test signal as shown in Fig.1. Essentially, this algorithm has compressed the dynamic range of the MP3 scale factors.

The structure of the proposed scheme is depicted in Fig. 3. Since we aim to re-compress only the main data, the MP3 side information is kept intact and is sent directly to the output bitstream so that it can be used for MP3 decoding.

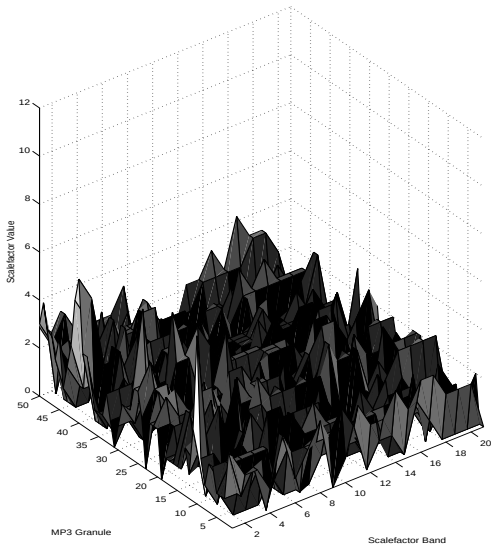

Fig. 2. The residual of the scalefactors after LTZP and lifting.

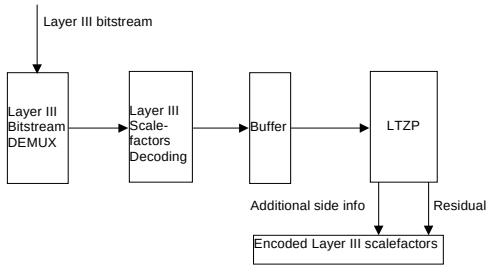

Fig. 3. Block diagram of the proposed scheme 1

After decoding the scale factors from each granule, we buffer scale factors of S_d consecutive granules in a FIFO working memory for re-compression. Search depth S_d of 32 seems to be a good compromise to balance the prediction gain and additional side information. S_d of 32 requires 5 bits additional side information to be represented. We perform a simple subtraction of the scale factors in the current granule from a previous granule within the search window S_d and find the residual that needs least bits. In case the residuals in the entire search window can not achieve any net bit-reduction, no subtraction is performed and a flag-bit is set to 0. Consequently no additional side information is needed in this case.

In order to limit the amount of side information, we divide the 21 scalefactor bands of each granule into two parts according to MPEG-1 standard. That is, the first part is scalefactor bands 0-10, the second part is scalefactor bands 11-20. In case any of the two parts has negative values, it is lifted to non-negative values with a lifting-offset. Each part has its own lifting-flag (one bit), that indicates whether a lifting is necessary for that particular part. If the lifting-flag is 0, no lifting is performed, if the lifting-flag is 1, a lifting is performed. In order to reduce the side information, the minimum residual is limited to be –2. Therefore only 1 bit is needed to represent the lifting-offset. Lifting-offset of 0 represents

lifting by 1 e.g. from –1 to 0 and lifting-offset of 1 represents lifting by 2.

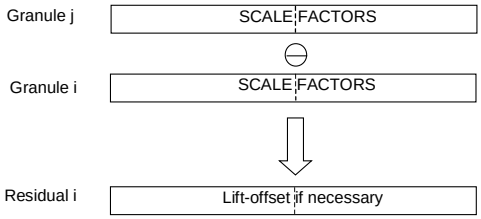

Fig. 4. Illustration of the differential coding of scale factors.

The residual after LTZP and lifting is then coded with the same fixed length coding method as in MP3.

With this simple coding scheme, we have achieved an average lossless bit reduction of 1.5 %. The price for the bit-reduction is a coding delay of roughly 32x13 ms and a buffer memory to save the scale factors of previous 32 granules. The bitrate in the test was 64 kbps and the sampling frequency was 44.1 kHz.

PROPOSED SCHEME 2: CODEBOOK-BASED DIFFERENTIAL CODING + HUFFMAN CODING FOR THE SCALE FACTORS

In certain applications such as handheld devices, the working memory is quite limited. To reduce the memory requirements of the decoder in scheme 1, we have developed a second scheme to reduce the dynamic range of the scale factors.

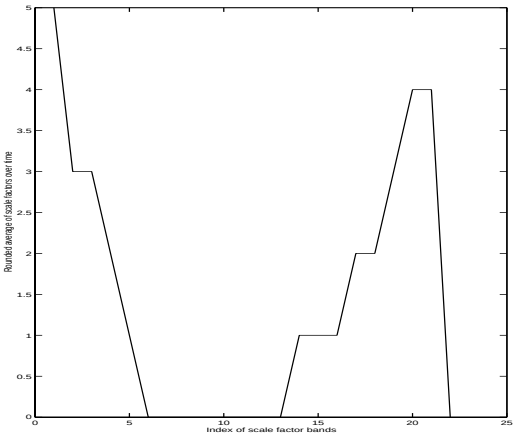

Fig. 5 Average MP3 scale factors of a test signal. Horizontal axis is the index of the scale factor bands.

The mean of the scale factors in all granules of the signal is calculated and transmitted as side information. An example of the mean is shown in Fig. 5. A subtraction is performed between the current granule and the mean (centroid). The residual is coded with the Huffman tables in MPEG-1 Layer III standard. Since the Huffman tables in MPEG-1 Layer III are always multidimensional (quadruples or pairs), zeros are appended in the end for Huffman coding. With scheme 2 we have achieved a similar bit-reduction as with scheme 1. However, the working memory and computational complexity in the decoder is reduced by a factor of 32.
Although it is possible to combine schemes 1 & 2 to achieve better results, the improvement will still be marginal at this bitrate, since the number of bits needed to encode the scale factors takes only ca. 8 % of the total MP3 bitstream. The potential for re-compressing the scale factors is rather limited. Nevertheless it can serve as a complement to the algorithm proposed in [2].
We have also tried a scheme similar to scheme 1&2 to re-compress quantized MDCT coefficients. However, the performance is far behind the reported bit-reduction in [2].

PROPOSED SCHEME 3: SIMILARITY MEASURE BASED CODING OF STRONG AND WEAK BEATS

This scheme results from the observation that a music signal typically exhibits characteristics of self-similarity. This is particularly true for certain types of music such as pop, march, dance music etc.
In western music, especially in pop music, it is well known that beat patterns are composed of regularly spaced strong and weak beats as shown in Fig. 6. The beat-patterns are highly repetitive for a large amount of music. This is the basis for scheme 3.

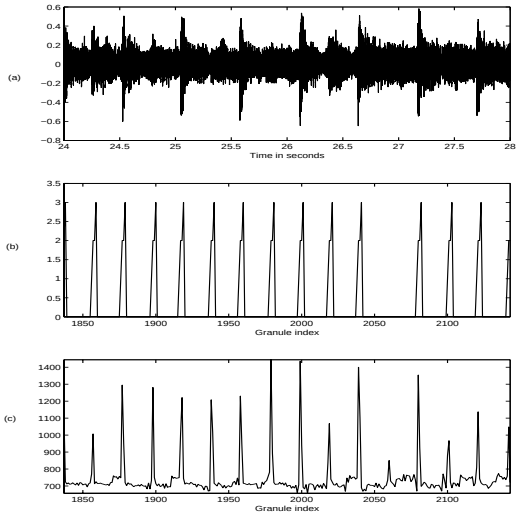

Fig. 6. (a) Waveform of a piece of a pop music signal, (b) window types in the corresponding MP3 bitstream, (c) Huffman bits fluctuation.

The short window segments are mostly associated with beats, which we classify to two classes: strong and weak beats. The total energy of the decoded MDCT coefficients in one granule is used as a simplified similarity measure. We calculate the average (mean) energy of all short window granules and take it as a threshold. Using this threshold short window granules are divided into strong and weak beats. Then the average of the strong beats and the average of the weak beats will be used as centroids.

We store the two centroids of decoded MDCT coefficients as a codebook to approximate all short window segments. Since there are only two elements in the codebook, we need only 1 bit for indexing (0 represents a strong beat and 1 represents a weak beat). The concept is illustrated in Fig. 7.

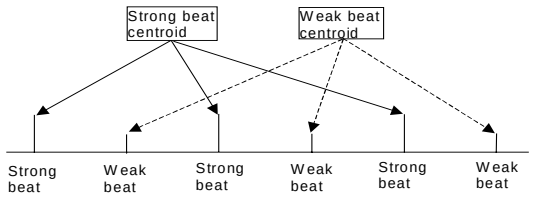

Fig. 7. Concept of self-similarity based coding method

In this way, we can achieve 4-8% bitrate reduction with little additional degradation in the subjective audio quality. Informal listening tests have confirmed that this concept is viable.

Our original intention was to design the codebook on inter-beat interval (IBI) based segment similarity measure, so that we could represent similar IBIs with a relatively small codebook. Unfortunately we have not yet found a satisfactory segment-similarity measure that relates closely to the subjective similarity.

As a starting point, we have implemented a similarity measure proposed in [3] in MP3 compressed domain. This similarity measure in MP3 compressed domain is good enough for beat recognition, but is not good enough to distinguish subjectively very similar and clearly different segments.

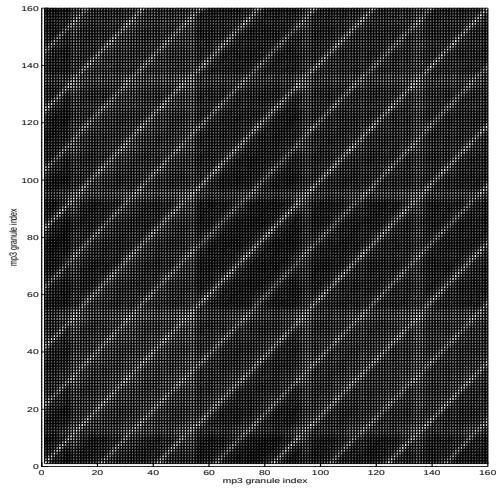

Fig. 8. Similarity matrices for a piece of ABBA song.

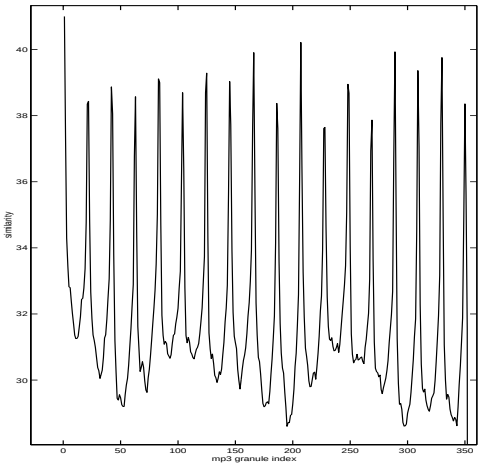

Fig. 9. Beat spectrum of a piece of ABBA song.

The similarity measure is based on the distance matrix, which is a two-dimensional embedding of the audio self-similarity [3]. Firstly the audio is parameterized on a granule-by-granule basis. These parameters are directly calculated from the MP3 bitstream. The scalefactor band energies of each granule form our feature vectors. Secondly a similarity measure D between feature vectors v_i and v_j is calculated from granules i and j.

$$D(i,j) \equiv \frac{v_i \bullet v_j}{\|v_i\| \|v_j\|} \quad (1)$$

$D(i,j)$ can be visualized in two dimensions as a square image as shown in Fig. 8, where each pixel i, j is given a gray scale value proportional to $D(i,j)$.

To compare the similarity between two IBIs, we calculate a measure similar to the "beat spectrum" [3]. Adding values of the similarity measure $D(i,j)$ diagonally over the granules in one IBI does this. This is calculated as

$$B(l) \approx \sum_{k=Start_b}^{End_b} D(k, k+l) \quad (2)$$

where $Start_b$ and End_b are the first and last granule in IBI b. $B(l)$ is then a comparison between IBI b and subsequent segments of the signal in a running window of equal duration. l is the relative distance in MP3 granules to the running window b. The approximation in the equation comes from the fact that the granule boundaries in mp3 don't usually match with the real IBI boundaries. An example of $B(l)$ is shown in Fig. 9.

The disadvantage of this similarity measure is its huge computational complexity and memory consumption. For the purpose of beat detection in MP3 compressed domain, a much simpler algorithm is reported in [4].

CONCLUSION

In this paper we have investigated a few coding schemes for re-compressing MP3 audio bitstreams. It was demonstrated that it is possible to achieve some additional coding gain by exploiting inter-frame redundancies.

The investigations we have performed with scheme 1 & 2 seem to suggest that the room for further bits reduction is rather limited. Further effort in this direction will only produce marginal improvements.

Scheme 3 has shown some promising results. However, it will eventually degrade the subjective audio quality. It may be more useful to utilize the beat-pattern redundancies for error concealment as suggested in [5].

ACKNOWLEDGEMENT

The Academy of Finland and Nokia Foundation are acknowledged for providing the first author scholarships to initiate and to conduct the research.

References

1. ISO/IEC 11172-3 International Standard, "Information Technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s", 1993.
2. Golchin, F., Paliwal, K., "Lossless Coding of MPEG-1 Layer III Encoded Audio Stream", ICASSP2000, Istanbul, Turkey, June 5-9, 2000.
3. Foote, J., Uchihashi, S., "The Beat Spectrum: A New Approach to Rhythm Analysis", IEEE International

Conference on Multimedia and Expo (ICME2001), Tokyo, Japan, August 22-25, 2001.

4. Wang, Y., Vilermo, M., "A Compressed Domain Beat Detector using MP3 Audio Bitstreams", The 9th ACM International Multimedia Conference (MM2001), Ottawa, Canada, September 30 – October 5, 2001.

5. Wang, Y., "A Beat-Pattern based Error Concealment Scheme for Music Delivery with Burst Packet Loss", IEEE International Conference on Multimedia and Expo (ICME2001), Tokyo, Japan, August 22-25, 2001.