Lapped Transforms in Perceptual Coding of Wideband Audio

Sien Ruan

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

December 2004

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering.

To my beloved parents

Abstract

Audio coding paradigms depend on time-frequency transformations to remove statistical redundancy in audio signals and reduce data bit rate, while maintaining high fidelity of the reconstructed signal. Sophisticated perceptual audio coding further exploits perceptual redundancy in audio signals by incorporating perceptual masking phenomena. This thesis focuses on the investigation of different coding transformations that can be used to compute perceptual distortion measures effectively; among them the lapped transform, which is most widely used in nowadays audio coders. Moreover, an innovative lapped transform is developed that can vary overlap percentage at arbitrary degrees. The new lapped transform is applicable on the transient audio by capturing the time-varying characteristics of the signal.

Sommaire

Les paradigmes de codage audio dépendent des transformations de temps-fréquence pour enlever la redondance statistique dans les signaux audio et pour réduire le taux de transmission de données, tout en maintenant la fidélité élevée du signal reconstruit. Le codage sophistiqué perceptuel de l'audio exploite davantage la redondance perceptuelle dans les signaux audio en incorporant des phénomènes de masquage perceptuels. Cette thèse se concentre sur la recherche sur les différentes transformations de codage qui peuvent être employées pour calculer des mesures de déformation perceptuelles efficacement, parmi elles, la transformation enroulé, qui est la plus largement répandue dans les codeurs audio de nos jours. D'ailleurs, on développe une transformation enroulée innovatrice qui peut changer le pourcentage de chevauchement à des degrés arbitraires. La nouvelle transformation enroulée est applicable avec l'acoustique passagère en capturant les caractéristiques variantes avec le temps du signal.

Acknowledgments

I would like to acknowledge my supervisor, Prof. Peter Kabal, for his support and guidance throughout my graduate studies at McGill University. Prof. Kabal's kind treatment to his students is highly appreciated. I would also like to thank Ricky Der for working with me and advising me through the work.

My thanks go to my fellow TSP graduate students for their close friendship; especially Alexander M. Wyglinski for the various technical assistances.

I am sincerely indebted to my parents for all the encouragement they have given to me. They are the reason for who I am today. To my mother, Mrs. Dejun Zhao and my father, Mr. Liwu Ruan, thank you.

Contents

List of Figures

List of Tables

List of Terms

AAC	MPEG-2 Advanced Audio Coding
ADPCM	Adaptive Differential Pulse Code Modulation
CELP	Code Excited Linear Prediction
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DPCM	Differential Pulse Code Modulation
DST	Discrete Sine Transform
EBU-SQAM	European Broadcasting Union — Sound Quality Assessment Material
ERB	Equivalent Rectangular Bandwidth
FIR	Finite Impulse Response
IMDCT	Inverse Modified Discrete Cosine Transform
ITU	International Telecommunication Union
MDCT	Modified Discrete Cosine Transform
MDST	Modified Discrete Sine Transform
MLT	Modulated Lapped Transform
MOS	Mean Opinion Score
MPEG	Moving Picture Experts Group
MP3	MPEG-1 Layer III
PCM	Pulse Code Modulation
NMN	Noise-Masking-Noise
NMR	Noise-to-Masking Ratio
NMT	Noise-Masking-Tone
LOT	Lapped Orthogonal Transform

LT	Lapped Transform
QMF	Quadrature Mirror Filter
PE	Perceptual Entropy
PEAQ	Perceptual Evaluation of Audio Quality
PESQ	Perceptual Evaluation of Speech Quality
PR	Perfect Reconstruction
Pre-DST	Pre-filtered Discrete Sine Transform
SFM	Spectral Flatness Measure
SMR	Signal-to-Masking Ratio
SNR	Signal-to-Noise Ratio
SPL	Sound Pressure Level
TDAC	Time-Domain Aliasing Cancellation
TMN	Tone-Masking-Noise
TNS	Temporal Noise Shaping
VQ	Vector Quantization

Chapter 1

Introduction

1.1 Audio Coding Techniques

Audio coding algorithms are concerned with the digital representation of sound using information bits. A number of paradigms have been proposed for the digital compression of audio signals. Roughly, audio coders can be grouped as either *parametric coders* or *waveform coders*. The concept of perceptual audio coding is relevant in the latter case, where auditory perception characteristics are applicable [1].

1.1.1 Parametric Coders

Parametric coders represent the source of the signal with a few parameters. Such coders are suitable for speech signals since a good source model of speech production is available. More specifically, the vocal tract is modelled as a time-varying filter that is excited by a train of periodic impulses (voiced speech) or a noise source (unvoiced speech) [2]. The parameters that characterize the filter are estimated, encoded and transmitted. In the decoder, the signal is synthesized from the decoded model parameters. More advanced parametric coders, such as the Code-Excited Linear Predictive (CELP) coders, may include the error signal resulting from the parametric reconstruction to represent the excitation to the vocal tract filter.

1.1.2 Waveform Coders

Waveform coders try to accurately replicate the waveform of the original signal. Such coders have been the best choice for audio encoding, since no appropriate source models are available to general audio signals. Efficient waveform coders remove redundancy within the coded signal by exploiting the correlation between signal components, either in time or frequency domain. Perceptual coders additionally remove information that is irrelevant to the perception of the signal.

Time domain waveform coders

Time domain coders perform the coding process on the time representations of the audio data. The well-known coding methods in the time domain are [2] Pulse Code Modulation (PCM), Differential Pulse Code Modulation (DPCM) and Adaptive Differential Pulse Code Modulation (ADPCM). For audio, the PCM scheme typically spends 16 bits to quantize each time sample. Although PCM provides high quality audio, the required bit rate is quite high. In DPCM, instead of the time samples, the difference between the original and predicted signal is quantized, which has a lower variance than the original signal and thus requires fewer bits to quantize. ADPCM, an enhanced version of DPCM, adapts the predictor and quantizer to local characteristics of the input signal and lowers the computation complexity.

Frequency domain waveform coders

Frequency domain coders carry out the compression on a frequency representation of the input signal. Main advantages of frequency domain coders include the ability to independently encode different parts of the frequency spectrum, adaptive bit allocation to shape the quantization noise, and the reconstruction of better sound quality [1]. Frequency domain coders are commonly categorized into two groups: *subband coders and transform coders*. Subband coders employ a small number of bandpass filters to split the input signal into subband signals which are coded independently. At the receiver the subband signals are decoded and summed up to reconstruct the output signal. Transform coders use a transformation to convert blocks of the input signal to frequency coefficients. Several advantages result from encoding the input signal in the transform domain [3]. Firstly, effective transforms compact the information of the signal into fewer coefficients which allows many

transform coefficients to be set to zero without affecting the quality. Secondly, transform coefficients are less correlated than temporal samples of the input signal, ensuring in a more efficient usage of quantizers. Furthermore, good frequency resolution is achievable by judiciously selecting the transformation. As such, frequency transform coders are the method of choice for the application of auditory masking characteristics.

Perceptual waveform coders

Perceptual audio coders work in frequency domain by employing a transform to decompose the input signal into spectral coefficients [1]. The auditory masking threshold is calculated from the signal spectrum. The transform coefficients are quantized and coded using the masking threshold. For example, if the coefficients have an energy less than the masking threshold, they are not quantized and not transmitted. Thus, the perceptual redundancy (these uncoded coefficients) is removed from the signal.

1.2 Time-to-Frequency Transformations

Time-frequency transformation maps the time-domain input to a set of coefficients which cover the entire spectrum and represent the frequency-localized signal energy. By confining significant values to subset of coefficients, the transformation plays an essential role in the reduction of statistical redundancies. Additionally, by providing explicit information about the distribution of signal and hence masking power over the time-frequency plane, the transformation also assists in the identification of perceptual redundancies when used in conjunction with a perceptual model. As a result, both statistical and perceptual redundancies in the signal are removed.

Coders typically segment input signals into quasi-stationary frames ranging from 2 to 50 ms in duration. Then the time-frequency mapping estimates the spectral components on each frame, attempting to match the analysis properties of the human auditory system. The time-frequency mapping section might contain [1]:

- Unitary transform;
- Time-invariant bank of critically sampled, uniform, or nonuniform bandpass filters;

- Time-varying (signal-adaptive) bank of critically sampled, uniform, or nonuniform bandpass filters.

The choice of time-frequency analysis methodology always depends on the overall system objectives and design philosophy.

1.3 Thesis Contributions

Extensive research has been performed by audio coding specialists to incorporate transformations within medium to high rate coders. At low coding rates (for instance, 1 bit per sample), some distortion is inevitable, which entails the need for a more effective representation of spectral components. Recent research work is primarily concerned with 50% overlapped and critically sampled transformations and their application to low-rate audio coding, with the aim of reducing audible artefacts and improving the audio quality.

In the thesis, two state-of-the-art time-frequency transformations are first presented and an assembly of transformation experiment results is analyzed (Chapter 4). They are both based on 50% overlapped frames. It is concluded that a pure transformation achieves better coding performance than a hybrid one (filter bank followed by a transformation). It is also suggested that the power spectrum generated from the transform coefficients should be used in the psychoacoustic analysis.

Moreover, a novel partially overlapped (less than 50%) transformation is proposed (Chapter 5). It is developed to reduce the noise-to-mask ratio mismatch associated with the 50% overlap transformations. At a smaller overlap, the novel transformation mitigates the pre-echo artefact (one generated from the noise-to-mask ratio mismatch) when coding transient audio events and delivers an overall better sound quality.

1.4 Thesis Synopsis

The thesis is organized into 6 chapters. Chapter 2 is concerned with the perceptual audio coding. Starting with a brief overview of the human auditory masking, Chapter 2 discusses the compression of audio in the perceptual domain with an emphasis on the psychoacoustic modelling of the input audio, followed by the description of the structure of a generic perceptual coder.

In Chapter 3, we discuss lapped transforms and their importance to audio coding. A thorough analysis of lapped transforms is given and the conditions for perfect reconstruction of the output signal are obtained in a matrix form. The role of the prototype window is investigated and the Modulated Lapped Transform (MLT) which is a special case of lapped transforms is analyzed. Finally window (length) switching is described as a traditional method to capture transient characteristics of audio.

Chapter 4 is dedicated to the evaluation of two widely used time-frequency transformations in the MPEG audio coding standards: hybrid filter bank (used in MP3) and pure MDCT (Modified Discrete Cosine Transform) filter bank (used in AAC). Their performance is compared based on informal subjective listening experiments. The comparison incorporated transforms for the masking threshold calculation in the psychoacoustic analysis.

In Chapter 5, we introduce the proposed partially overlapped lapped-transform, the Pre-DST (Pre-filtered Discrete Sine Transform). The matrix representation of the transform is obtained and the properties of perfect reconstruction and critical sampling are given. The functionality of each module is described. A comparison is made between the performance of Pre-DST and pure MDCT, based on the pre-echo mitigation.

Finally, a complete summary of our work is provided in Chapter 6, along with directions for future related research.

Chapter 2

Perceptual Audio Coding: Psychoacoustic Audio Compression

Perceptual audio coding has become an important key technology for many types of multimedia services these days. This chapter provides a brief tutorial introduction on a number of issues in today's low rate audio coders. After the discussion of psychoacoustic principles in the first part of this chapter, the second part will focus on the perceptual model along with the structure of generic perceptual audio coders using psychoacoustic approaches.

2.1 Human Auditory Masking

Audio coding algorithms must rely upon hearing models to optimize coding efficiency. In the case of audio, the receiver is ultimately the human ear and sound perception is affected by its psychoacoustic properties. For example, a speaker will be inaudible when the background noise is loud. This is one of various masking instances.

Most current audio coders incorporate several psychoacoustic principles, including absolute hearing thresholds, critical band analysis, and masking phenomena, to identify the "irrelevant" signal information during signal analysis. Further, combination of these psychoacoustic notions with properties of signal quantization leads to compressed audio with high fidelity.

2.1.1 Hearing System

The hearing system converts sound waves into mechanical movement and finally into electrical impulses perceived by the brain. This neuro-mechanical interaction in the ear is processed by three main parts: the outer ear, the middle ear and the inner ear.

The outer ear is composed of the pinna (auricle), the ear canal (external auditory meatus) and the eardrum (tympanic membrane) [4]. The pinna collects sounds (air pressure waves) in the air and directs them towards the ear canal. The canal acts as a quarter-wavelength resonator, amplifying sound pressures within the range of 3–5 kHz by as much as 15 dB. The sound pressure makes the eardrum to vibrate and this way it is converted into the mechanical energy.

The middle ear acts as an acoustical impedance-matching device that reduces the amount of reflected wave and improves sound transmission. Additionally, when the sound level exceeds a certain level, some of the tiny muscles in the middle ear contract to attenuate the vibrations passing through the middle ear, and others contract to keep the stirrup away from the oval window in order to weaken the vibrations passed to the inner ear [5].

The inner ear plays the most important role in perception within the auditory system. It includes the cochlea [4], from which mechanical vibrations emanating from the oval window are transformed into electrical impulses. The region of cochlea close to the oval window is recognized as the base, whereas the inner tip is known as the apex. The basilar membrane extends along the cochlea from the base to the apex. Each point along the basilar membrane is associated with a *Characteristic Frequency* for which the amplitude of its vibrations is maximal. The basilar membrane performs a frequency-to-place transformation and behaves like a spectrum analyzer. The motion of the basilar membrane causes the bending of sensory hair cells, leading to neural firings in the auditory nerve. Neural information propagates to the brain where it undergoes cognitive processing.

2.1.2 Perception of Loudness

The absolute threshold of hearing indicates the minimum *Sound Pressure Level* (SPL) that a sound must have for detection in the absence of other sounds. A mean threshold value is obtained by averaging the individual thresholds of numerous listeners. The audibility threshold exhibits a strong dependency on frequencies and is approximated by the function

proposed in [6],

$$T_q(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4, \tag{2.1}$$

where f is expressed in Hz and threshold in dB SPL. The threshold $T_q(f)$ is illustrated in Fig. 2.1.

Perceived loudness is a function of both frequency and level. Since coding algorithm designers have no a priori knowledge regarding the actual playback levels (SPL), it is typically assumed that the volume control (playback level) on a decoder will be set such that the smallest possible output signal will be presented close to 0 dB SPL. Hence, a scaling of loudness (SPL normalization) is needed and this procedure will be discussed in details in Section 2.2.1.

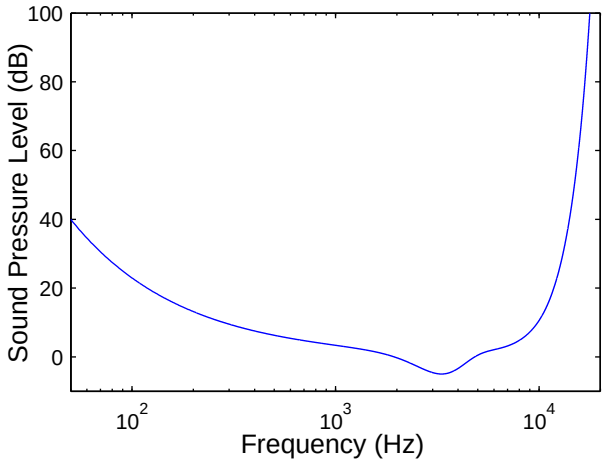


Fig. 2.1 Absolute threshold of hearing for normal listeners [6].

2.1.3 Critical Bands

As previously mentioned, a frequency-to-place conversion occurs within the cochlea that affects the frequency selectivity of the hearing system. As a result, the cochlea can be viewed from a signal-processing perspective as a bank of highly overlapping bandpass filters. The critical band refers to the frequency distance that quantifies the cochlea filter passbands.

The importance of the critical bands comes from two facts. First, the hearing system discriminates between energy in and out of a critical band. Within a critical band changes

in stimuli greatly affect perception and beyond a critical band subjective responses decrease abruptly. Additionally, the simultaneous masking property of the hearing system is related to the critical bands. When two sounds have energy in the same critical band, the sound having the higher level dominates the perception [2].

Experiments by Scharf have shown that the bandwidth of critical bands is a function of their center frequencies [7]. While attempting to represent the inner ear as a discrete set of non-overlapping auditory filters, Scharf determined that 25 critical bands were sufficient to represent the audible frequency range of the ear. The bandwidth of the resulting critical bands is listed in Table 2.1, with center frequencies spanning from 50 to 19.5 kHz.

Table 2.1 Critical bands measured by Scharf [7].

Band No.	Center Freq. (Hz)	Bandwidth (Hz)	Band No.	Center Freq. (Hz)	Bandwidth (Hz)
1	50	0–100	14	2150	2000-2320
2	150	100–200	15	2500	2320–2700
3	250	200–300	16	2900	2700–3150
4	350	300–400	17	3400	3150–3700
5	450	400–510	18	4000	3700–4400
6	570	510–630	19	4800	4400–5300
7	700	630–770	20	5800	5300–6400
8	840	770–920	21	7000	6400–7700
9	1000	920–1080	22	8500	7700–9500
10	1175	1080–1270	23	10500	9500–12000
11	1370	1270–1480	24	13500	12000–15500
12	1600	1480–1720	25	19500	15500–
13	1850	1720–2000			

It is evident that the critical bandwidths are wider at lower frequencies than those at higher frequencies. This nonlinear scale, on which the signal is processed in the inner ear, is called the Bark scale (where an increment of one Bark corresponds to one critical band). Zwicker suggested an analytical expression that converts from frequency in Hertz to the Bark scale [4],

$$Z(f) = 13 \arctan(0.00076 f) + 3.5 \arctan[(\frac{f}{7500})^2] \quad \text{Bark.} \tag{2.2}$$

The bandwidth of each critical band as a function of its center frequency can be ap-

proximated by [4]

$$BW(f) = 25 + 75[1 + 1.4(\frac{f}{1000})^2]^{0.69} \quad \text{Hz.} \tag{2.3}$$

An alternative measure, employing the concept of the *Equivalent Rectangular Bandwidth* (ERB) [8], was proposed by Moore and Glasberg. The discussion throughout the whole thesis is based on Bark scale measure.

2.1.4 Masking Phenomena

Auditory masking refers to the process where one sound is rendered inaudible by the presence of another sound. Varieties of masking occur in daily life. For example, a speaker must raise his/her voice in a very noisy environment in order to be understood. For applications on audio compression of discarding irrelevant spectral components, the simultaneous masking is most useful.

Simultaneous masking occurs when the masker and the maskee (masked signal) are presented to the hearing system at the same time. The nature of the masker as being noise-like or tone-like has impacts on the masking effects. For the purpose of coding noise shaping it is convenient to distinguish between only three types of simultaneous masking [1]: *noise-masking-tone* (NMT), *tone-masking-noise* (TMN), and *noise-masking-noise* (NMN). Different masking produces different masking power. For example, the masking threshold associated with NMT is significantly greater than with TMN.

The effect of simultaneous masking is not only felt in the current critical band, but also in the adjacent bands. This effect, also known as *the Spread of Masking*, is often modelled in coding applications by an approximately triangular spreading function [9]. When interband masking occurs, a masker centered within one critical band has some predictable effect on detection thresholds in the other critical bands.

The masking threshold is an estimate of the maximum quantization noise that can be injected into the signal and remains inaudible to human ear. The standard practice in perceptual coding involves first classifying masking signals as either noise or tone, next computing appropriate thresholds, then using this information to shape the quantization noise spectrum beneath the thresholds. The following section describes Johnston's model on perceptual entropy.

Masking can also take place even when the masking tone begins after and ceases before the masked sound. This is referred to as *forward* and *backward masking* respectively: they

fall under the category of *Temporal Masking*.

2.2 Example Perceptual Model: Johnston's Model

At the heart of any audio coder lies the auditory model. The goal of the digital model is to quantify the "irrelevant" information so that perceptual redundancies can be extracted. Various masking models have been proposed with different levels of accuracy and complexity: Johnston's Model [10], MPEG-1 Psychoacoustic Model 1 [11], AAC Audio Masking Model [12], and PEAQ Model [13]. All of these models are based on the masking patterns introduced in Section 2.1.4.

For our research, we use the auditory masking model proposed in [10] by Johnston. Johnston's model determines the energy threshold of the maximum allowable quantization noise in each critical band such that quantization noise remains inaudible. We introduce the functional mechanisms of the model and later the notion of perceptual entropy.

2.2.1 Loudness Normalization

As previously mentioned, some of the perceptual quality factors depend on the actual sound pressure level (SPL) of the test signal. A normalization step is needed to fix the mapping from input signal levels to loudness. The loudness normalization procedure in PEAQ Model works as follows [13].

First, spectral coefficients (e.g., DFT coefficients) are obtained by taking a sine wave of 1019.5 Hz and 0 dB full-scale as the input signal. Then the maximum absolute value of the spectral coefficients is compared to a 90 dB SPL reference level. The normalization factor is calculated such that the full-scale sinusoid will be associated with an SPL near 90 dB.

A more appropriate normalization would involve the total energy preserved in the frequency domain since sound pressure level is an energy phenomenon. Such a normalization factor is independent of the frequency of the test sinusoid [14].

2.2.2 Masking Threshold Calculation

The first step in Johnston's Model to calculate threshold corresponds to the critical band analysis. The complex Fourier spectrum of the input signal is converted to the power

spectrum as follows,

$$P(k) = \mathrm{Re}^2(X(k)) + \mathrm{Im}^2(X(k)), \tag{2.4}$$

where $X(k)$ represent the Discrete Fourier Transform (DFT) coefficients. The energy in each critical band is calculated by partitioning the power spectrum into critical bands (see Table 2.1) and then summed,

$$B_i = \sum_{k=b_{li}}^{b_{hi}} P(k), \tag{2.5}$$

where b_{li} and b_{hi} are the lower and upper boundaries of critical band i and B_i is the signal energy in critical band i (here, one critical band corresponds to one Bark).

The Bark power spectrum (critical band spectrum) is spread to estimate the effects of masking across critical bands. The spreading function S is described analytically by,

$$S_{ij} = 15.81 + 7.5((j-i) + 0.474) - 17.5(1 + ((j-i) + 0.474)^2)^{1/2} \quad \mathrm{dB}, \tag{2.6}$$

where i and j represent the Bark indices of the masked and masking signal respectively. The spread Bark spectrum is obtained by convolving the Bark spectrum B_i with the spreading function. The convolution is implemented as a matrix multiplication,

$$C_i = S_{ij} * B_i, \tag{2.7}$$

where C_i denotes the spread critical band spectrum. A conversion of S_{ij} from its decibel representation is required before carrying out the multiplication in the power spectrum domain.

As tonal maskers and noise maskers generate different masking patterns, Johnston uses the Spectral Flatness Measure (SFM) to determine the noise-like or tone-like nature of the signal. The SFM is defined as the ratio of the Geometric Mean (GM) to the Arithmetic Mean (AM) of the power spectrum

$$\mathrm{SFM_{dB}} = 10 \log_{10} \frac{\mathrm{GM}}{\mathrm{AM}}, \tag{2.8}$$

and is further converted to a coefficient of tonality α, according to

$$\alpha = \min(\frac{\mathrm{SFM_{dB}}}{\mathrm{SFM_{dBmax}}}, 1), \tag{2.9}$$

where $\mathrm{SFM}_{\mathrm{dBmax}} = -60$ dB. A signal that is completely tonal would result in $\alpha = 1$, whereas a purely noise-like signal would yield $\alpha = 0$.

The two threshold offsets is geometrically weighted by the tonality coefficient α, 14.5 $+i$ dB for tone-masking-noise and 5.5 dB for noise-masking-tone. The resulting offset O_i is set as,

$$O_i = \alpha(14.5 + i) + 5.5(1 - \alpha) \quad \text{dB}. \tag{2.10}$$

The spread threshold estimate T_i is then obtained by subtracting O_i from the spread Bark spectrum C_i

$$T_i = 10^{\log_{10}(C_i) - (O_i/10)}. \tag{2.11}$$

The next step involves renormalization of the noise energy threshold. Johnston argued that the spreading function increases the energy estimates in each band because of its shape. The renormalization multiplies each T_i by the inverse of the energy gain, assuming each band has unit energy. This renormalized T_i is designated as $\tilde{T}_i$.

Finally, the threshold $\tilde{T}_i$ is compared to the absolute threshold of hearing T_{qi} and replaced by $\max[\tilde{T}_i, T_{qi}]$, ensuring that masking thresholds do not demand a level of noise below the absolute limits of hearing. In a manner identical to the SPL normalization procedure, the final thresholds must be converted out of dB SPL by dividing back the normalization factor.

2.2.3 Perceptual Entropy

For transparent coding (perceptually lossless), the quantization noise injected at each frequency component must be set corresponding to the masking threshold. Then the total number of bits required to quantize all components represents an estimate of the minimum number of bits necessary to transmit that frame of the signal. The total bit rate divided by the number of samples coded, represents the per-sample rate, namely the "Perceptual Entropy".

By applying uniform quantization principles to the signal and associated set of masking thresholds, Johnston shows a lower bound on the number of bits required to achieve

transparent coding [15],

$$\mathrm{PE} = \frac{1}{N}\sum_{i=1}^{25}\sum_{k=b_{li}}^{b_{hi}} \log_2\{2[\mathrm{round}(\frac{\mathrm{Re}(X(k))}{\sqrt{6T_i/(b_{hi}-b_{li})}})]+1\} + \log_2\{2[\mathrm{round}(\frac{\mathrm{Im}(X(k))}{\sqrt{6T_i/(b_{hi}-b_{li})}})]+1\} \quad (2.12)$$

where N is the number of spectral coefficients, T_i is masking threshold in critical band i and round(.) denotes the nearest integer operation.

The measurement is applied on a frame-by-frame basis and the PE estimate is obtained by choosing a worst case value. Using a 2048-point FFT with a 1/16 overlapped Hann window, Johnston reported the PE of 2.1 bits/sample for transparent audio compression.

2.3 Perceptual Audio Coder Structure

Perceptual audio coders take into account mathematical models of human perception for purposes of quantization and noise shaping and the coding algorithm is essentially a psychoacoustic algorithm. Fig. 2.2 shows the structure of a generic perceptual audio encoder, including five primary parts: the filter bank, the psychoacoustic model, bit allocation, quantization, and bitstream formatting.

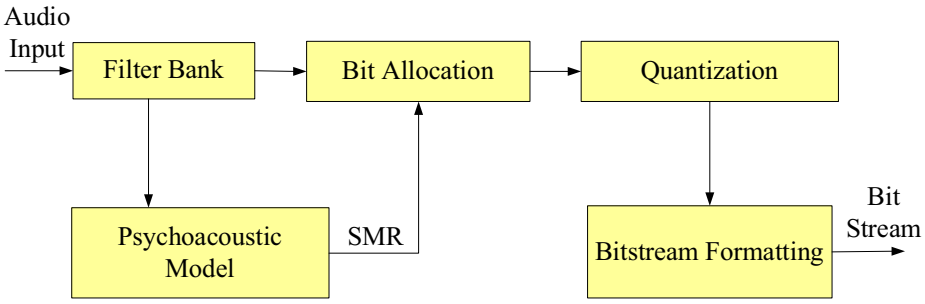


Fig. 2.2 Generic perceptual audio encoder [1].

2.3.1 Time-to-Frequency Transformation

All audio coders rely upon some type of time-frequency analysis to extract from the time domain input a set of frequency coefficients that is amenable to encoding in conjunction with a perceptual model. Encoding in frequency domain can take advantage of frequency characteristics of the input signal. For example, a spike (one coefficient) in the frequency domain can represent a sine wave, whereas a whole period of samples has to be encoded in the time domain.

The tool most commonly employed for the decomposition is the filter bank. The decomposing filter bank analyzes the frequency properties of the input signal and identifies the perceptual redundancies. For digital signals, the traditional decomposition is the Discrete Fourier Transform (DFT),

$$X_k = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi}{N} kn} \tag{2.13}$$

where n is the sample index and N is the number of samples in the transform. The filter bank widely used as a dominant tool in nowadays audio coders is the Modified Discrete Cosine Transform (MDCT) [1],

$$X_k = \sum_{n=0}^{2M-1} x(n) w(n) \cos\Big[\frac{(n + \frac{M+1}{2})(k + \frac{1}{2})\pi}{M}\Big] \tag{2.14}$$

where M is the number of transformed coefficients and $w(n)$ denotes the window function. In addition to an energy compaction capability similar to Discrete Cosine Transform (DCT), MDCT simultaneously achieves reduction of the blocking edge effects, critical sampling property and perfect signal reconstruction (Chapter 3).

Windowing

Windowing is multiplication of the audio signal directly by a window *w(n)*. The main consideration with designing a window is the shape of the window. For example, it is well known in digital signal processing theory that the rectangular window suffers from energy leakage. Most practical windows have a shape that emphasizes the mid-frame samples while de-emphasizes the edge samples such as Hann window and Hamming window.

- Example Window:

An example window is the "sine" window associated with MDCT, defined as

$$w(n) = \sin\left[(n + \frac{1}{2})\frac{\pi}{2M}\right] \tag{2.15}$$

for $0 \leq n \leq M - 1$. It offers good stopband attenuation (24 dB) [1], provides good attenuation of the blocking edge effects, and allows perfect reconstruction. This particular window is perhaps the most popular window in audio coding and is depicted in Fig. 2.3.

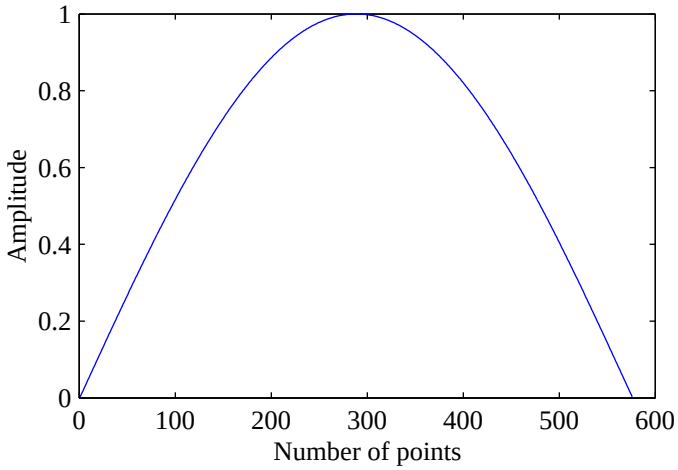


Fig. 2.3 Sine MDCT-window (576 points).

- Window Switch:

  If a sharp attack occurs at the end of a long frame, the psychoacoustic model would be misled to derive a higher masking threshold for that entire frame. As a result, the quantization noise would be spread over the entire frame and higher than the signal level at the beginning, manifesting itself as a perceptible *pre-echo* just before the attack of the signal. This situation can arise when coding recordings of percussive instruments such as the triangle, for example.

  To suppress the pre-echo, filter banks work by changing the analysis window length from "long" duration (e.g., 25 ms) during stationary segments to "short" duration (e.g., 4 ms) when transients are detected. For relatively stationary segments, long windows provide better compression with finer frequency resolution. On the other

hand, the characteristics of transients are better captured with short time windows. The switching decision is generally based on a measure of information content in the signal, like perceptual entropy.

2.3.2 Psychoacoustic Analysis

Psychoacoustic analysis, based on psychoacoustic models (Section 2.2), represents the core part of one perceptual audio coder. The purpose of psychoacoustic analysis is to estimate a just noticeable noise-level (masking threshold) in each band, represented as Signal-to-Noise Ratio (SMR), where S is the signal energy in the frequency band. This SMR is used in the bit allocation procedure to calculate Noise-to-Masking Ratio (NMR) [16],

$$\text{NMR} = \text{SMR} - \text{SNR} \quad \text{(dB)} \tag{2.16}$$

which determines the actual quantizer levels.

Psychoacoustic models are in frequency domain. It is possible to use output from the filter bank as input for the psychoacoustic model, or to perform a separate transform for the purpose of psychoacoustic analysis. For example, MDCT has been used as the decomposing filter bank in MPEG-1 Layer III (MP3) and MPEG-2 AAC (Advanced Audio Coding), but both coders still use the DFT for psychoacoustic analysis to more accurately apply their perceptual model.

2.3.3 Adaptive Bit Allocation

Information bits are allocated to frequency bands such that a distortion criterion is optimized. A adaptive bit assignment is used so that the spectrum of quantization noise is shaped to be less audible than a noise spectrum evenly distributed without shaping. The process is known as *Spectral Noise Shaping*, under the constraint that the total number of bits is fixed (though the number of bits assigned to each band can vary from frame to frame).

Two categories of distortion measures, perceptual and non-perceptual, are used to shape the audible noise [17]. In the perceptual approach, the quantization noise spectrum is shaped in parallel with the masking threshold curve. Noise-to-Masking Ratio, among others, is an example distortion measure. The non-perceptual approach employs criteria such

as the noise power above the masking threshold, for example.

(a) Noise-to-Mask Ratio (NMR)-based bit allocation

In this approach bit allocation is performed based on the Noise-to-Mask Ratio (NMR). As a result, the noise spectrum will be parallel to the masking threshold curve and be inaudible if it is below the masking threshold. This method attempts to distribute the noise power equally in all frequency bands.

(b) Noise energy-based bit allocation

In the energy-based approach, bit assignment is performed based on the audible part of the quantization noise, i.e., the noise above the masking threshold. Since it is not evenly distributed over the frequency range, the noise is audible to various degrees at different frequencies.

2.3.4 Quantization

In the earlier stage, a given number of bits are assigned to represent the spectral components of the audio signal. Now the spectral coefficients are quantized to integer levels according to the bits assignment. This quantization process is a lossy compression, meaning that the quantized signal is not mathematically equal to the original signal. However, this lossy coding scheme can be perceptually lossless (transparent) in the sense that the human ear cannot distinguish between the original and compressed signals. We introduce two major quantization schemes used in audio coding: *Scalar Quantization* and *Vector Quantization* (VQ).

(a) Scalar Quantization

A scalar quantizer operates on individual values. It divides the range of input values into L intervals (cells). Each cell is represented by a single decision level. It takes a single input value and selects the best match (the nearest scalar level, normally) to that value from a predetermined set of scalar levels. These scalar levels can be arranged in either a uniform or a non-uniform pattern.

- Uniform Quantization:

In this method, all the levels are equally spaced. *Step size* δ, the distance between two successive decision levels, is defined as

$$\delta = \frac{x_{\max} - x_{\min}}{L}, \tag{2.17}$$

where $x_{\max}$ and $x_{\min}$ are the maximum and minimum values of the input and L is the number of quantization levels. Based on the assumption that quantization noise is white and uniformly distributed in the interval $(-\delta/2, \delta/2)$, the variance of such uniform distribution noise is $\delta^2/12$ [18].

The uniform scalar quantizer can be implemented in a closed form. Let x be a scalar component which is quantized by a uniform scalar quantizer with a step size of δ. Then, the quantized value, $\hat{x}$ is given as (mid-riser case) [19]

$$\hat{x} = \delta \times \text{round}(x/\delta). \tag{2.18}$$

- Non-uniform Quantization:

  With non-uniformly spaced decision levels, the quantizer can be tailored to the specific input statistics such that considerably SNR is attained for a given input *pdf* (probability density function). In general, for arbitrary input signal, the decision levels are determined by minimizing the average distortion given by [18],

$$D = \sum_{i=1}^{L} \int_{R_i} (x - y_i)^2 p_{\mathbf{x}}(x) dx \tag{2.19}$$

  where y_i is the ith quantization level, R_i denotes the ith partition (cell) and $p_{\mathbf{x}}$ is the probability density function of the input. Iterative algorithms such as the Lloyd algorithm [20] can be used to design the quantizer.

(b) Vector Quantization

A vector quantizer is a mapping from a vector to a finite set of points, called *codewords*. By exploiting the correlation among the vector components, vector quantization achieves a bit-rate performance advantage over scalar quantization, at the expense of complexity and computation power when searching for the matched codeword in a large codebook. For

this reason, the uniform scalar quantizer was considered most appropriate and has been selected for the remainder of this thesis.

2.3.5 Bitstream Formatting

A bitstream formatter is used after quantization to achieve better data compression, which takes the quantized filter bank outputs, the bit allocation and other required side information, and assembles them in an efficient fashion. This process is known as *Entropy Coding*. In the case of MP3, variable-length Huffman codes are used to encode the quantized spectral coefficients. These Huffman codes are mathematically lossless and allow for more efficient bitstream representation of the quantized samples at the cost of additional complexity.

Chapter 3

Signal Decomposition with Lapped Transforms

In this chapter, we introduce an important family of time-frequency transformations, i.e., Lapped Transforms. They work to decompose the time-domain signal to transform-domain coefficients. Several properties are in consideration with designing a transform.

(a) Perfect Reconstruction

Perfect reconstruction of a transform refers to the signal decomposition from which the original signal can exactly be recovered from the reconstructed signal[1], in the absence of quantization [21]. In other words, the original and reconstructed signals are mathematically the same. This brings the advantage that reconstruction errors are due only to the quantization noise and thus it can be controlled and masked by the signal.

(b) Critical Sampling

The analysis/synthesis system should be *critically sampled* [21], i.e., the overall number of transformed domain samples is equal to the number of time-domain samples. Critical sampling ensures that all stages of the audio coder operate at the same sampling rate (input sampling rate) and the encoder does not carry an increase in the total number of samples to be processed.

[1]Here, $\hat{x}(n) = x(n - D)$, where $\hat{x}(n)$ and $x(n)$ are the reconstructed and original signal respectively, and D is a time delay.

(c) Frequency and Temporal Resolution

The bandwidths of the filter bank should emulate the analysis properties of the human auditory system. Spacings of the filter bank should match the large width-variation of the critical bands in frequency. At the same time, the analysis time window of the filter bank should be short enough to accurately estimate the masking thresholds for highly transient signals. Ideally, the analysis filter bank would have time-varying resolutions in both the time and frequency domains and this motivates many designs with switched and hybrid filter bank structures.

3.1 Block Transforms

Given a signal $x(n)$, we must group its samples into blocks before computing the transform. A signal block is defined as, $\mathbf{x} = [x(mM), x(mM-1), ...x(mM-M+1)]^T$, where m is the block index and M is the block length[2]. For a orthonormal matrix $\mathbf{A}$, $\mathbf{A}^{-1} = \mathbf{A}^T$, the *forward* and *backward transform* for the mth block $\mathbf{x}$ are defined as

$$\mathbf{X} = \mathbf{A}^T\mathbf{x} \tag{3.1}$$

and

$$\mathbf{x} = \mathbf{A}\mathbf{X}. \tag{3.2}$$

An orthogonal $\mathbf{A}$ brings advantages such as convenience to implement inverse transform by simply transposing the flowgraph of forward transform. Different choices of $\mathbf{A}$ lead to different transforms. DFT and DCT are some familiar cases.

We have used $\mathbf{A}^T$ in the forward transform and $\mathbf{A}$ in the backward so that the basis vectors (also called the basis functions) of the transform are the columns of $\mathbf{A}$. The coefficients of basis vectors represent the linear weights on block $\mathbf{x}$.

3.2 Lapped Transforms

The lapped transform [21] was originally developed in order to eliminate the blocking edge effects. The idea is to extend the basis functions beyond the block boundaries, creating an

[2]For simplicity of notation, we suppress the dependence of $\mathbf{x}(m)$ on m.

overlap between signal blocks. In a lapped transform (LT), L-sample input block is mapped into M transform coefficients, with $L > M$. Although $L - M$ samples are overlapped between blocks, the number of transform coefficients is the same as if there was no overlap. This critical sampling property is kept by computing M new transform coefficients for every new M input samples (i.e., frame update rate is M samples). Thus, there will be an overlap of $L - M$ samples in consecutive LT blocks. The LT of a signal block $\mathbf{x}$ is obtained by,

$$\mathbf{X} = \mathbf{H}_{\mathbf{M}\times\mathbf{L}}\mathbf{x} \tag{3.3}$$

where $\mathbf{x}$ is an extended signal block $\mathbf{x} = [x(mM), x(mM-1), ...x(mM-2M+1)]^T$ and $\mathbf{H}$ is the forward transform matrix. A diagram of signal processing with lapped transforms is shown in Fig. 3.1, where the block generation operates as set of serial to parallel converters for the input block and parallel to serial converters for the output block [21].

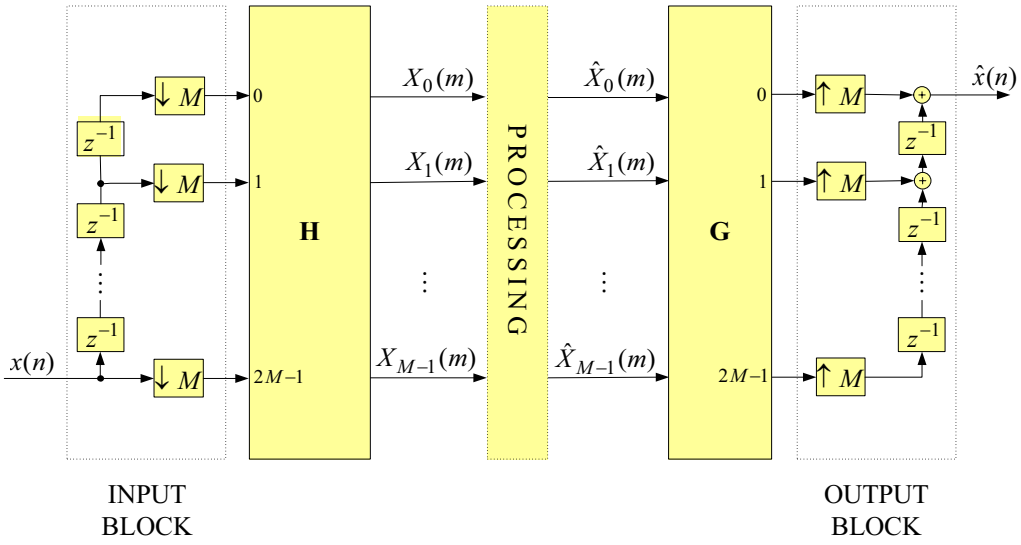


Fig. 3.1 General signal processing system using the lapped transform [21].

3.2.1 LT Orthogonal Constraints

Applying an inverse LT to $\mathbf{X}$,

$$\mathbf{y} = \mathbf{G}_{\mathbf{L}\times\mathbf{M}}\mathbf{X}, \tag{3.4}$$

the resulting L-sample $\mathbf{y}$ are not equal to the L-sample $\mathbf{x}$ used to compute the forward LT. The original signal $\mathbf{x}$ must be recovered in an overlap-add fashion. The whole procedure is illustrated in Fig. 3.2. As we can see, the total system is causal. For example, the $x(0)$ is the most recent input sample and so the first output sample occurs at $\hat{x}(0)$, with the algorithmic delay of $2M - 1$ samples.

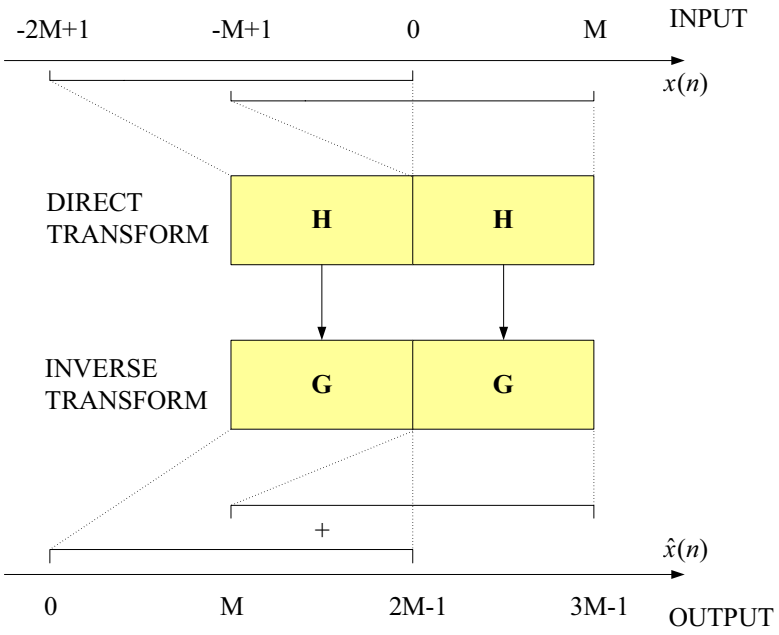


Fig. 3.2 Signal processing with a lapped transform with $L = 2M$ [21].

Assuming the overlap is 50% ($L = 2M$), we divide $\mathbf{H}$ into two $M \times M$ matrices and the first data block $\mathbf{x}^{(1)}$ into two $M \times 1$ vectors, we can rewrite Eq. (3.3) as follows,

$$\mathbf{X}^{(1)} = \begin{bmatrix} \mathbf{H}_a & \mathbf{H}_b \end{bmatrix} \begin{bmatrix} \mathbf{x}_a^{(1)} \\ \mathbf{x}_b^{(1)} \end{bmatrix} = \mathbf{H}_a \mathbf{x}_a^{(1)} + \mathbf{H}_b \mathbf{x}_b^{(1)} \tag{3.5}$$

where $\mathbf{H}_a$ and $\mathbf{H}_b$ are matrices containing the first M and last M columns of the analysis matrix $\mathbf{H}$; $\mathbf{x}_a^{(1)}$ and $\mathbf{x}_b^{(1)}$ contain the first and second M elements of $\mathbf{x}^{(1)}$. Similarly, the next transform block $\mathbf{X}^{(2)}$ can be denoted as

$$\mathbf{X}^{(2)} = \mathbf{H}_a \mathbf{x}_a^{(2)} + \mathbf{H}_b \mathbf{x}_b^{(2)}. \tag{3.6}$$

Also on the synthesis side, splitting the output vector $\mathbf{y}$ into two sub-vectors, the $2M \times 1$

reconstructed signal $\mathbf{y}$ can be represented as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_a \\ \mathbf{y}_b \end{bmatrix} = \begin{bmatrix} \mathbf{G}_a \\ \mathbf{G}_b \end{bmatrix} \mathbf{X}, \tag{3.7}$$

where $\mathbf{y}_a$ and $\mathbf{y}_b$ are the first and second half of $\mathbf{y}$; $\mathbf{G}_a$ and $\mathbf{G}_b$ are two $M \times M$ square matrices containing the first and second M rows of the synthesis matrix $\mathbf{G}$. This results in

$$\mathbf{y}^{(1)} = \begin{bmatrix} \mathbf{y}_a^{(1)} \\ \mathbf{y}_b^{(1)} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_a\mathbf{X}^{(1)} \\ \mathbf{G}_b\mathbf{X}^{(1)} \end{bmatrix} \tag{3.8}$$

$$\mathbf{y}^{(2)} = \begin{bmatrix} \mathbf{y}_a^{(2)} \\ \mathbf{y}_b^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_a\mathbf{X}^{(2)} \\ \mathbf{G}_b\mathbf{X}^{(2)} \end{bmatrix}. \tag{3.9}$$

Therefore, combining Eq. (3.8) and Eq. (3.9), the reconstructed signal in the overlapping parts of $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$ can be expressed as

$$\begin{aligned} \mathbf{y}_{overlap} &= \mathbf{y}_b^{(1)} + \mathbf{y}_a^{(2)} \\ &= \mathbf{G}_b\mathbf{H}_a\mathbf{x}_a^{(1)} + \mathbf{G}_b\mathbf{H}_b\mathbf{x}_b^{(1)} + \mathbf{G}_a\mathbf{H}_a\mathbf{x}_a^{(2)} + \mathbf{G}_a\mathbf{H}_b\mathbf{x}_b^{(2)}. \end{aligned} \tag{3.10}$$

This equation shows that an LT operates as 4 block transforms and then sums the overlapping parts.

For perfect reconstruction (PR), we have

$$\mathbf{y}_{overlap} = \mathbf{x}_b^{(1)} = \mathbf{x}_a^{(2)}. \tag{3.11}$$

This results in the following constraints

$$\mathbf{G}_b\mathbf{H}_a = \mathbf{G}_a\mathbf{H}_b = \mathbf{0_M} \tag{3.12}$$

and

$$\mathbf{G}_a\mathbf{H}_a = \mathbf{G}_b\mathbf{H}_b = \mathbf{I_M} \tag{3.13}$$

where $\mathbf{0_M}$ is an $M \times M$ zero matrix and $\mathbf{I_M}$ is an $M \times M$ identity matrix. In a special case

when $\mathbf{G} = \mathbf{H}^t$, the orthogonal constraints (PR conditions) become

$$\mathbf{H}_b^t \mathbf{H}_a = \mathbf{H}_a^t \mathbf{H}_b = \mathbf{0}_\mathbf{M}, \tag{3.14}$$

$$\mathbf{H}_a^t \mathbf{H}_a + \mathbf{H}_b^t \mathbf{H}_b = \mathbf{I}_\mathbf{M}. \tag{3.15}$$

This special case is referred to as a *Lapped Orthogonal Transform (LOT)* [22], meaning that $\mathbf{H_a}$ and $\mathbf{H_b}$ are orthogonal and the overlapping parts of the basis functions are also orthogonal.

3.3 Filter Banks: Subband Signal Processing

In many applications it is desirable to separate the incoming signal into several subband components, by means of bandpass filters, and then process each subband separately. The fundamental parts of subband signal processing systems are the analysis and synthesis filter banks. The analysis filter bank should do a good job of separating the incoming signal into its constituent subband signals, and the synthesis filter bank should be able to recover a good approximation to the input signal from the subband signals. The basic block diagram of a filter bank system is shown in Fig. 3.3, where $H(z)$ and $G(z)$ are the analysis and

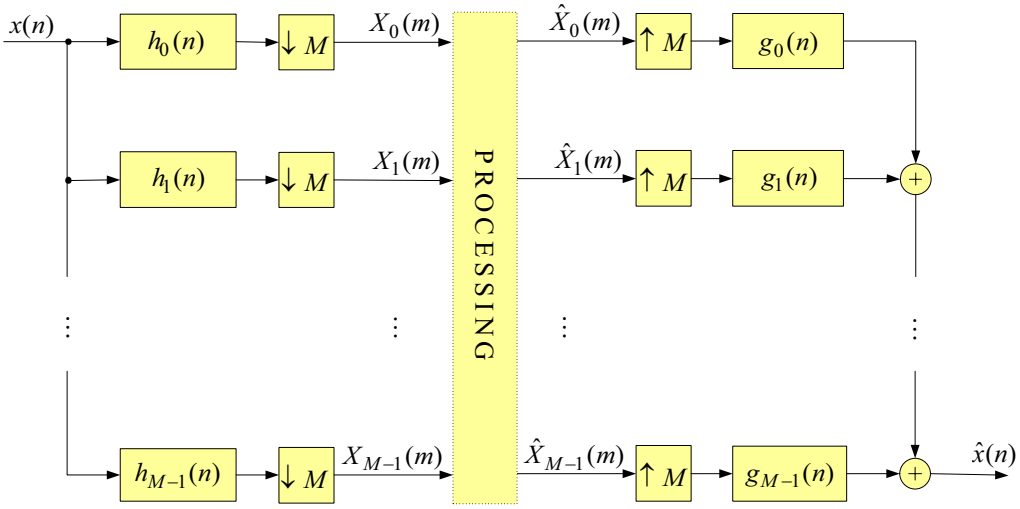


Fig. 3.3 Typical subband processing system, using the filter bank [21].

synthesis filters [21], respectively. The decimator after the analysis filter bank is to keep

the total sampling rate in the output of all M subbands identical to the input signal rate (critically sampled).

3.3.1 Perfect Reconstruction Conditions

In order to obtain the condition for signal reconstruction, we analyze the kth channel of the filter bank, since we have a similar structure for all channels. The output of the kth channel (after filtering and subsampling) is,

$$X_k(m) = \sum_{n=-\infty}^{\infty} x(n) h_k(mM - n) \tag{3.16}$$

where $h_k(n)$ is the impulse response of the kth analysis filter. If the subband signal $X_k(m)$ is not modified (e.g., no quantization) and thus $X_k(m) = \hat{X}_k(m)$, we have the reconstructed signal as,

$$y(n) = \sum_{k=0}^{M-1} \sum_{m=-\infty}^{\infty} X_k(m) g_k(n - mM) \tag{3.17}$$

where $g_k(n)$ is the impulse response of the kth synthesis filter. By substituting Eq. (3.16) into Eq. (3.17), we have the result,

$$\begin{aligned} y(n) &= \sum_{l=-\infty}^{\infty} x(l) h_T(n, l) \\ &= \sum_{l=-\infty}^{\infty} x(l) \Big[\sum_{k=0}^{M-1} \sum_{m=-\infty}^{\infty} g_k(n - mM) h_k(mM - l) \Big] \end{aligned} \tag{3.18}$$

where $h_T(n, l)$ denotes the time-varying impulse response of the total system. Perfect reconstruction is obtained if and only if $h^T(n, l) = \delta(n - l - N)$, that is,

$$\sum_{k=0}^{M-1} \sum_{m=-\infty}^{\infty} g_k(n - mM) h_k(mM - l) = \delta(n - l - D) \tag{3.19}$$

where D is a time delay and leads to $y(n) = x(n - N)$.

3.3.2 Filter Bank Representation of the LT

Comparing Fig. 3.1 and Fig. 3.3, it is clear that the LT is a special case of the multirate filter bank. The impulse responses of the analysis filters are the time-reversed rows of the analysis matrix **H**, and the impulse responses of the synthesis filters are the columns of the synthesis matrix **G**. For a block transform, it is possible to perfectly reconstruct $x(n)$ if PR conditions are satisfied. For lapped transforms, the above PR conditions cannot be satisfied because of the overlap-add of the inverse blocks.

3.4 Modulated Lapped Transforms

If we design each analysis filter $h_i(n)$ and synthesis filter $g_i(n)$ in Fig. 3.3 independently, then the computational complexity will be proportional to the number of bands. A more efficient implementation of the filter bank is to pass each subband through a cosine modulator to shift its center frequency to the origin. Then a lowpass filter $h(n)$ is used followed by the decimator. Modulated Lapped Transform (MLT) is a family of lapped transforms generated by modulating a lowpass prototype filter. The MLT basis functions are defined by [23],

$$h_i(n) = \sqrt{\frac{2}{M}} h(n) \cos\left[(n + \frac{M+1}{2})(k + \frac{1}{2})\frac{\pi}{M}\right] \tag{3.20}$$

where $k = 0, 1, ..., M-1, n = 0, 1, ..., 2M-1$ and $h(n)$ is the lowpass prototype filter[3]. The magnitude frequency response of MLT with a sine window (Section 2.3.1) is shown in Fig. 3.4.

3.4.1 Perfect Reconstruction Conditions

We start by analyzing the MLT with two different windows for analysis and synthesis stages. The output of the analysis filter bank is given by,

$$X(k) = \sqrt{\frac{2}{M}} \sum_{n=0}^{2M-1} x(n)h(n) \cos\left[(n + \frac{M+1}{2})(k + \frac{1}{2})\frac{\pi}{M}\right] \tag{3.21}$$

[3] $h(n)$ is the window function in time domain or the lowpass prototype filter in frequency domain.

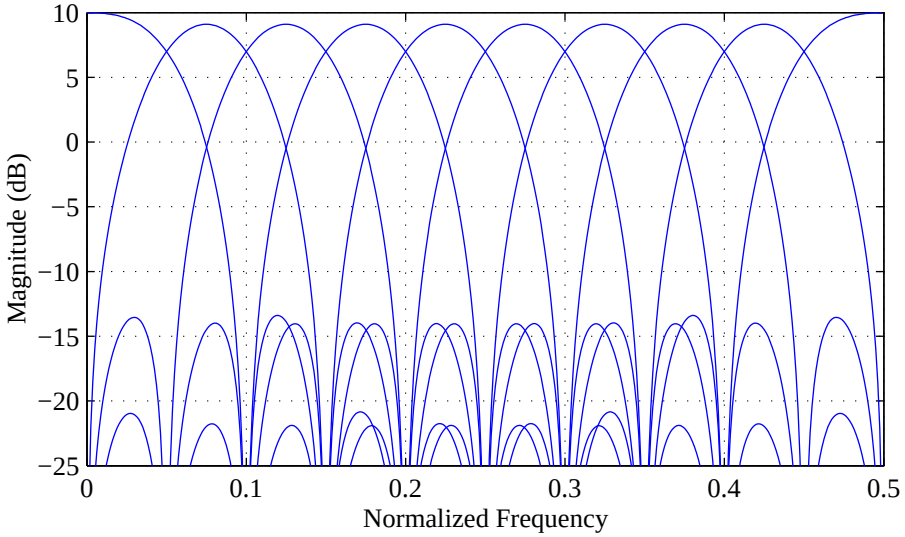


Fig. 3.4 Magnitude frequency response of a MLT ($M = 10$).

where $h(n)$ is the analysis window, M is the number of subbands and $k = 0, 1, ..., M-1$. The output of the synthesis filter bank is given by,

$$y(n) = \sqrt{\frac{2}{M}} \sum_{k=0}^{M-1} X(k) g(n) \cos\left[(n + \frac{M+1}{2})(k + \frac{1}{2})\frac{\pi}{M}\right] \tag{3.22}$$

where $g(n)$ is the synthesis window. Substituting Eq. (3.21) into Eq. (3.22) and simplifying, we obtain

$$\begin{aligned} y(n) &= \frac{1}{M} g(n) \sum_{m=0}^{2M-1} x(m)h(m) \sum_{k=0}^{M-1} \cos\left[(m+n+M+1)(k+\frac{1}{2})\frac{\pi}{M}\right] \\ &+ \frac{1}{M} g(n) \sum_{m=0}^{2M-1} x(m)h(m) \sum_{k=0}^{M-1} \cos\left[(m-n)(k+\frac{1}{2})\frac{\pi}{M}\right]. \end{aligned} \tag{3.23}$$

Observing that the first half of $y(n)$ is zero except for $m = n$ and $m = M-1-n$ and the second half of $y(n)$ is zero except for $m = n$ and $m = 3M-1-n$, we get

$$y(n) = g(n)h(n)x(n) - g(n)h(M-1-n)x(M-1-n), \quad n = 0, ..., M-1, \tag{3.24}$$

and

$$y(n) = g(n)h(n)x(n) - g(n)h(3M-1-n)x(3M-1-n), \quad n = M, ..., 2M-1. \tag{3.25}$$

Now, the desirable segment in the overlapping parts is given by

$$\mathbf{y}_{overlap} = \mathbf{y}_b^{(1)} + \mathbf{y}_a^{(2)}. \tag{3.26}$$

Since $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$ have different time references, we take the beginning of $\mathbf{y}^{(2)}$ as the time reference. Hence,

$$\begin{aligned} \mathbf{y}_{overlap} &= \mathbf{y}^{(1)}(n+M) + \mathbf{y}^{(2)}(n), \quad n = 0, ..., M-1 \\ &= g(n+M)h(n+M)\mathbf{x}^{(1)}(n+M) + g(n+M)h(2M-1-n)\mathbf{x}^{(1)}(2M-1-n) \\ &+ g(n)h(n)\mathbf{x}^{(2)}(n) - g(n)h(M-1-n)\mathbf{x}^{(2)}(M-1-n). \end{aligned} \tag{3.27}$$

Using a common time reference for the input blocks, we have,

$$\mathbf{x}_{overlap} = \mathbf{x}_b^{(1)} = \mathbf{x}_a^{(2)} = \mathbf{x}^{(1)}(n+M) = \mathbf{x}^{(2)}(n), \quad n = 0, ..., M-1, \tag{3.28}$$

and also,

$$\mathbf{x}^{(1)}(2M-1-n) = \mathbf{x}^{(2)}(M-1-n), \quad n = 0, ..., M-1. \tag{3.29}$$

Therefore, we need following conditions to achieve perfect reconstruction,

$$\begin{aligned} &h(n)g(n) + h(n+M)g(n+M) = 1, \\ &g(n)h(M-1-n) - g(n+M)h(2M-1-n) = 0. \end{aligned} \tag{3.30}$$

When the same window $h(n)$ is used for both analysis and synthesis, the perfect reconstruction conditions reduce to,

$$\begin{aligned} &h(n) = h(2M-1-n), \\ &h^2(n) + h^2(n+M) = 1. \end{aligned} \tag{3.31}$$

This special transform case is called *Modulated Lapped (Orthogonal) Transform*.

Comment: LOT vs. MLT

We have to distinguish two concepts here, namely, Lapped Orthogonal Transform (LOT) and Modulated Lapped Transform (MLT). They are both lapped transforms because they are realizations of the general filter bank in Fig. 3.3 with identical analysis and synthesis filters and satisfy the orthogonal conditions in Eq. (3.14) and Eq. (3.15).

The LOT, defined in Section 3.2, was developed to reduce the blocking effect in image coding. Eq. (3.15) forces orthogonality of the basis functions within the same block, whereas Eq. (3.14) forces orthogonality of the overlapping portions of the basis functions of adjacent blocks.

Fast LOTs can be constructed from components with well-known fast-computable algorithms such as the DCT and the DST, by matrix factorization of transform matrices. There are many fast solutions and one of the most elegant factorization is the *type-II fast LOT* proposed by Malvar [21]. The orthogonality conditions are satisfied by the construction of the inverse transform matrix $\mathbf{G}$ ($\mathbf{H}^t$) as

$$\mathbf{G} = \frac{1}{2} \begin{pmatrix} \mathbf{D}_e - \mathbf{D}_o & \mathbf{D}_e - \mathbf{D}_o \\ \mathbf{J}(\mathbf{D}_e - \mathbf{D}_o) & -\mathbf{J}(\mathbf{D}_e - \mathbf{D}_o) \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^{II}_{M/2}\mathbf{S}^{IV}_{M/2} \end{pmatrix} \mathbf{R}, \tag{3.32}$$

where $\mathbf{D}_e$ and $\mathbf{D}_o$ are the $M \times M/2$ matrices containing the even and odd DCT basis functions as

$$\mathbf{D}_e = c(k)\sqrt{\frac{2}{M}} \cos\left[2k(n+\frac{1}{2})\frac{\pi}{M}\right] \tag{3.33}$$

$$\mathbf{D}_o = \sqrt{\frac{2}{M}} \cos\left[(2k+1)(n+\frac{1}{2})\frac{\pi}{M}\right], \tag{3.34}$$

and $\mathbf{C}^{II}_{M/2}$ and $\mathbf{S}^{IV}_{M/2}$ are the DCT-II and DST-IV matrices, defined as

$$\mathbf{C}^{II}_{K} = c(k)\sqrt{\frac{2}{K}} \cos\left[k(r+\frac{1}{2})\frac{\pi}{K}\right] \tag{3.35}$$

$$\mathbf{S}^{IV}_{K} = \sqrt{\frac{2}{K}} \sin\left[(k+\frac{1}{2})(r+\frac{1}{2})\frac{\pi}{K}\right], \tag{3.36}$$

where

$$c(k) = \begin{cases} 1/\sqrt{2}, & k = 0 \\ 1, & \text{otherwise.} \end{cases}$$

The factor **J** is a antidiagonal matrix and the factor **R** is a permutation matrix. The LOT corresponds to a perfect reconstruction filter bank. It has been shown in Section 3.3 that any uniform PR FIR filter bank is a lapped orthogonal transform and the PR conditions in Eq. (3.19) are identical to the orthogonality conditions in Eq. (3.14) and Eq. (3.15).

The MLT, defined in Section 3.4, is developed independently in terms of filter bank theory. Calling $g_k(n)$ the impulse response of the kth synthesis filter, the modulated filter bank is constructed as [24]

$$g_k(n) = h(n)\sqrt{\frac{2}{M}}\cos\left[(k+\frac{1}{2})(n-\frac{L-1}{2})\frac{\pi}{M}+\frac{\pi}{4}\right] \tag{3.37}$$

for k even, and

$$g_k(n) = h(n)\sqrt{\frac{2}{M}}\sin\left[(k+\frac{1}{2})(n-\frac{L-1}{2})\frac{\pi}{M}+\frac{\pi}{4}\right] \tag{3.38}$$

for k odd, where L is the length of the lowpass prototype filter $h(n)$. The above construction is called *Quadrature Mirror Filter* (QMF) and it cancels frequency-domain aliasing terms between neighboring subbands. However, QMF does not cancel time-domain aliasing and thus perfect reconstruction is not necessarily achieved. The possibility of PR was first demonstrated by Princen and Bradley [23, 25] using the arguments of the *Time-Domain Aliasing Cancellation* (TDAC) filter bank. They have shown that if the lowpass prototype $h(n)$ satisfies the constraints in Eq. (3.31), both aliasing in time-domain and frequency-domain will be cancelled. Later, Malvar developed the concept of *Modulated Lapped Transform* (MLT) by restricting to a particular prototype filter and formulating the filter bank as a lapped orthogonal transform. Until recently, the consensus name in the audio coding for the lapped transform interpretation of this special-case filter bank has evolved into the *Modified Discrete Cosine Transform* (MDCT). In short, the reader should be aware that the different acronyms TDAC, MLT[4], and MDCT all refer essentially to the same PR cosine modulated filter bank. Only Malvar's MLT implies a particular choice for

[4]It is important to note that the MLT is not a particular case of the fast LOT in Eq. (3.32), since no matrix factorization can be generated from the MLT basis functions.

$h(n)$ as described in Eq. (2.15).

3.5 Adaptive Filter Banks

As previously mentioned in Chapter 2, some audio coders switch between a set of available windows to match the time-varying characteristics of the input signal [26, 27]. For stationary parts of the signal, a high coding gain can be achieved with a high frequency resolution (using long windows). On the other hand, for energy transient parts of the input signal, it is preferable to have a high temporal resolution (using short windows) to localize a burst of quantization noise and prevent it from spreading over different frames. The switching criterion is based on a measure of the signal energy [28] or perceptual entropy [15]. As an alternative to the window switch, Herre and Johnston [29] proposed *Temporal Noise Shaping* (TNS) to continuously adapt to the time-varying characteristics of the input signal.

To preserve the perfect reconstruction property of the overall system, the transition between windows has to be carefully chosen. Therefore, a start window is used in between to switch from a long window to a short window and a stop window is used to switch back. A start window is defined as

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

3.5.1 Window Switching with Perfect Reconstruction

Perfect reconstruction is maintained during the transition. Assuming that $h_{start}(n)$ is used for both the analysis and synthesis filter banks, the output of the synthesis filter bank is

given by

$$\begin{aligned} y(n) &= \frac{1}{M} h_{start}(n) \sum_{m=0}^{N-1} x(m) h_{start}(m) \sum_{k=0}^{M-1} \cos\left[((m+n+M+1)(k+\frac{1}{2})\frac{\pi}{M}\right] \\ &+ \frac{1}{M} h_{start}(n) \sum_{m=0}^{N-1} x(m) h_{start}(m) \sum_{k=0}^{M-1} \cos\left[((m-n)(k+\frac{1}{2})\frac{\pi}{M}\right]. \end{aligned} \tag{3.39}$$

Here, we analyze different segments of $y(n)$.

For $0 \leq n \leq M-1$, $h_{start}(n) = h_{long}(n)$, the output becomes,

$$y(n) = h_{long}^2(n)x(n) - h_{long}(n)h_{long}(M-1-n)x(M-1-n). \tag{3.40}$$

In a lapped transform, the first half of the current output of the synthesis filter bank will contain the same terms as the second half of the previous output block, differing in that the time reversed terms have opposite signs. Therefore by overlap-add operation those terms cancel each other resulting in perfect reconstruction of the original signal. In overlapping and adding, the second term in Eq. (3.40) will be cancelled by the time-reversed term from previous block.

For $M \leq n \leq M + \frac{M}{3} - 1$, $h_{start}(n) = 1$ and $y(n)$ equals to zero except when $n = m$. Therefore the output is

$$y(n) = x(n). \tag{3.41}$$

For $M + \frac{M}{3} \leq n \leq M + \frac{2M}{3} - 1$, $h_{start}(n) = h_{short}(n - M)$ (second half of the short window), we obtain

$$\begin{aligned} y(n) &= h_{start}^2(n)x(n) - h_{start}(n)h_{start}(3M-1-n)x(3M-1-n) \\ &= h_{short}^2(n)x(n) - h_{short}(n)h_{short}(2M-1-n)x(3M-1-n). \end{aligned} \tag{3.42}$$

Similarly, the second term will be cancelled by the time-reversed term in the next short block and perfect reconstruction is achieved.

For $M + \frac{2M}{3} \leq n \leq 2M-1$, $h_{start}(n) = 0$ and so is the output of the synthesis filter bank. The output signal is perfectly reconstructed by overlapping outputs from two successive short frames. Perfection reconstruction conditions can also be shown for transition from a short window back to a long window.

Chapter 4

MP3 and AAC Filter Banks

In this chapter, we first look at the time-to-frequency transformations used in standards of MP3 (MPEG-1 Layer III) and MPEG-2 AAC (Advanced Audio Coding). The performance of both filter banks is reported later on, along with different transforms for psychoacoustic analysis.

4.1 Time-to-Frequency Transformations of MP3 and AAC

4.1.1 MP3 Transformation: Hybrid Filter Bank

The transformation used in MPEG-1 Layer-III belongs to the class of hybrid filter bank. It is build by cascading two different kinds of filter banks: first the polyphase filter bank (as used in Layer-I and Layer-II) and then an additional Modified Discrete Cosine Transform (MDCT) filter bank. The polyphase filter bank has the purpose of making Layer-III more similar to Layer-I and Layer-II. The MDCT filter bank subdivides each polyphase frequency band into 18 finer subbands to increase the potential for redundancy removal. A complete MP3 decomposition structure is shown in Fig. 4.1. First we examine the prototype filter to understand the polyphase filter bank.

Polyphase filter bank

The polyphase filter bank [16] is common to all three layers of the MPEG/audio algorithm. This filter bank uses a set of bandpass filters to divide the input audio block into 32

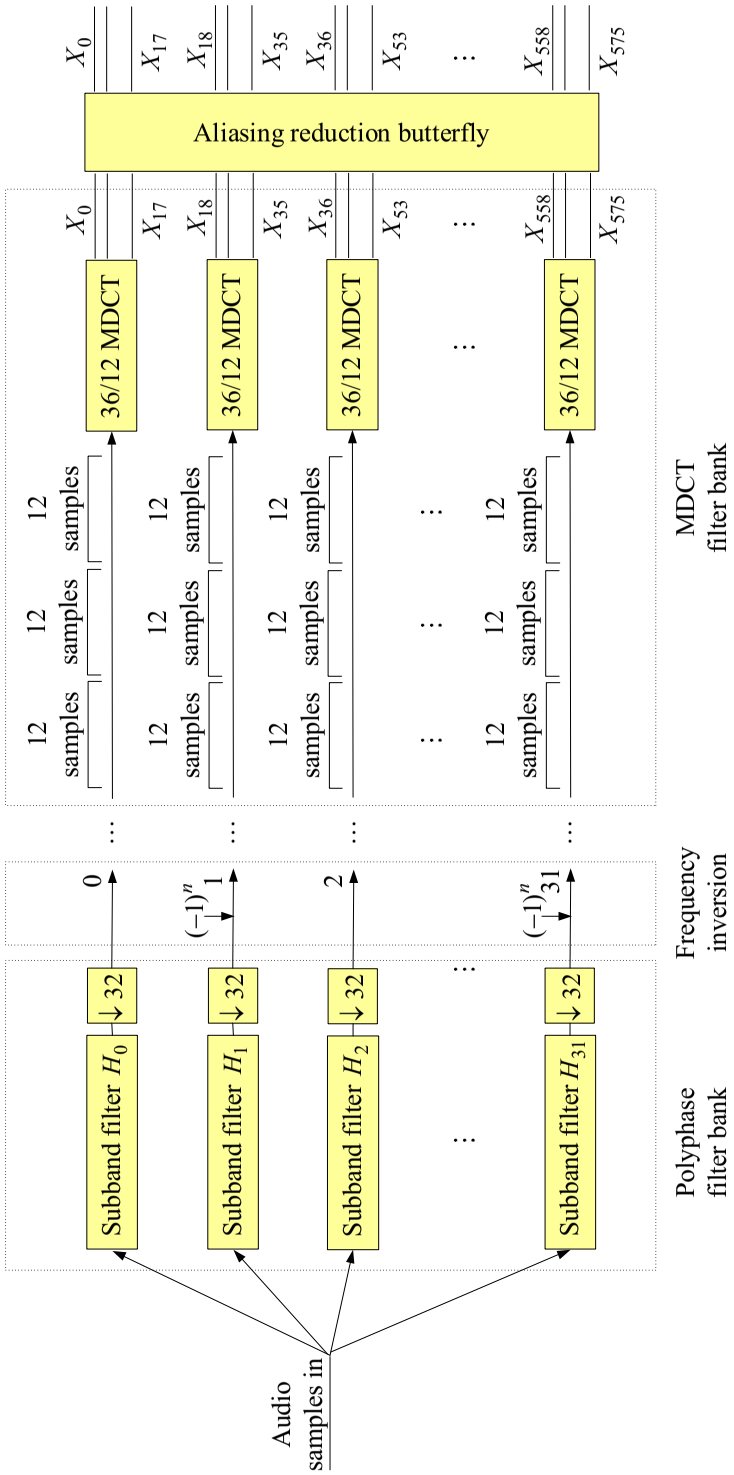
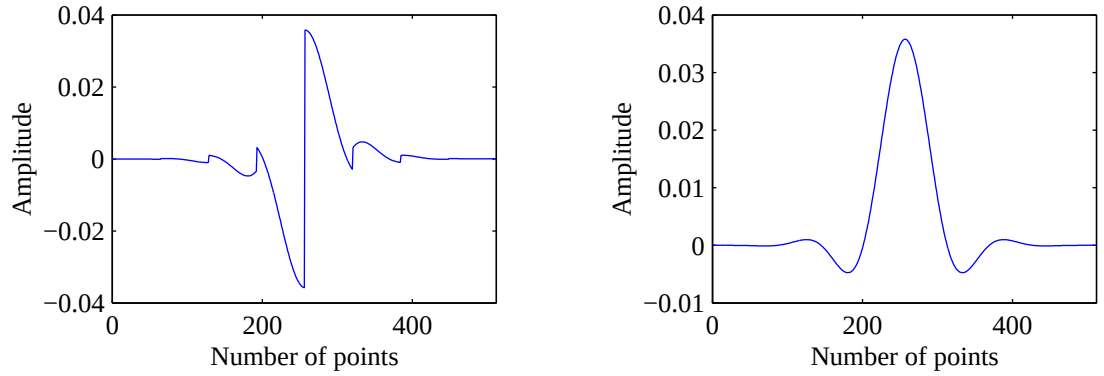


Fig. 4.1 MPEG-1 Layer III decomposition structure.

subbands, each of a nominal bandwidth $\pi/32$. The MPEG standard defines a 512-coefficient analysis window $C(n)$ to derive the lowpass prototype filter $h(n)$ of the filter bank, as

$$h(n) = \begin{cases} -C(n), & \text{floor}(n/64) \text{ is odd} \\ C(n), & \text{otherwise.} \end{cases}$$

A comparison of $h(n)$ and $C(n)$ is plotted in Fig. 4.2 and the magnitude frequency response of $h(n)$ is plotted in Fig. 4.3. The bandpass filter $H_i[n]$ of the ith subband of the filter bank



(a) Plot of $C(n)$

(b) Plot of $h(n)$

Fig. 4.2 Layer III prototype filter (b) and the original window (a).

is a modulation of the prototype filter with a cosine term to shift the lowpass response to the appropriate frequency band. Hence, they are called the *polyphase filter bank* and given by

$$H_i[n] = h[n] \cos\left[\frac{(2i+1)(n-16)\pi}{64}\right]. \tag{4.1}$$

As Fig. 4.4 shows, these filters have approximate "brick wall" magnitude responses with center frequencies at odd multiples of $\pi/64T$. The outputs of the filter bank are given by the filter convolution equation,

$$s_t[i] = \sum_{n=0}^{511} x[t-n]H_i[n]. \tag{4.2}$$

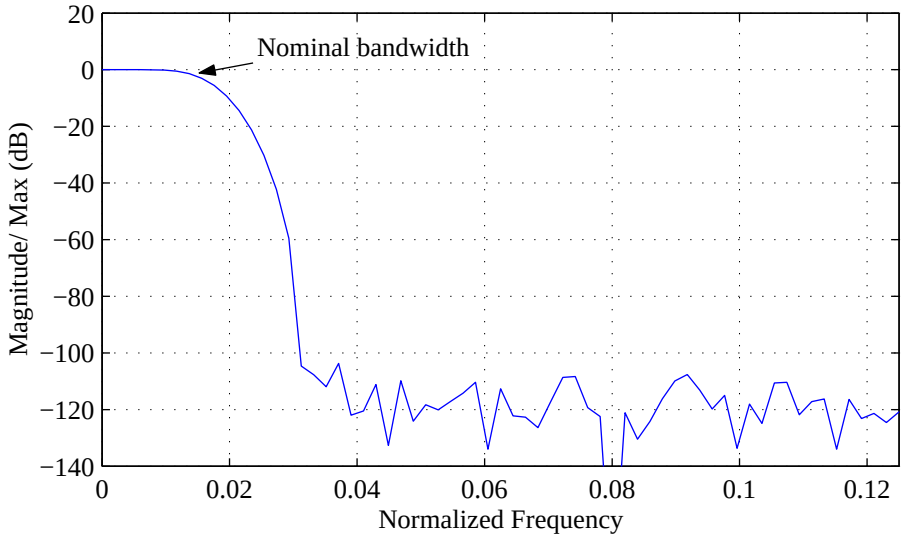


Fig. 4.3 Magnitude response of the lowpass filter.

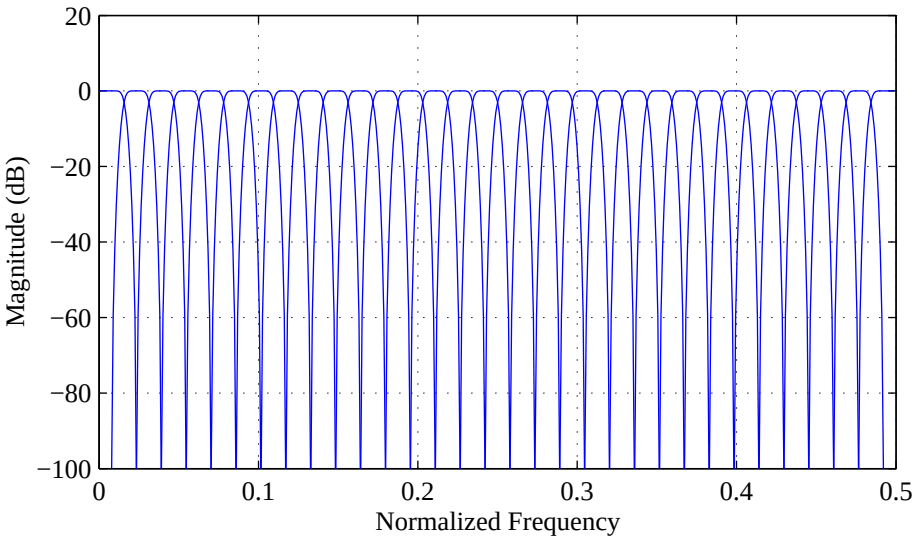


Fig. 4.4 Magnitude response of the polyphase filter bank ($M = 32$).

For a time instant t, which is an integral multiple of 32 audio sample intervals, the filter bank output for the subband i is given by,

$$s_t[i] = \sum_{k=0}^{63} \sum_{j=0}^{7} M[i][k] \times (C[k + 64j] \times x[k + 64j]) \tag{4.3}$$

where i is the subband index and ranges from 0 to 31, $x[n]$ is a 512-sample buffer of input audio, and $M[i][k]$ is a 32×64 analysis coefficient matrix for cosine modulation, defined as,

$$M[i][k] = \cos\left[\frac{(2i + 1)(k - 16)}{64}\right]. \tag{4.4}$$

We should note that the polyphase filter bank is critically sampled in that it produces 32 output samples for every 32 input samples. In effect, each of the 32 subband filters is followed by a decimator of factor 32, and thus only one sample out of 32 new samples is kept.

Computational requirement of the polyphase filter bank, as in Eq. 4.3, is moderate. 32 filter outputs need $512 + 32 \times 64 = 2560$ multiplies and $64 \times 7 + 32 \times 63 = 2464$ additions. Considerable further reductions in multiplies and adds are possible with, for example, FFT-like implementations. The flow diagram for computing the polyphase filter outputs is described in "Audio Content", part 3 of the MPEG/audio standard [11].

Although the response of the prototype is favorable, the polyphase filter bank has three notable drawbacks.

First, the lack of a sharp cut-off at the nominal bandwidth (see Fig. 4.3) results in an overlap in the frequency coverage of adjacent polyphase filters and this can cause signal energy near nominal subband edges to simultaneously appear in two adjacent subbands. To complicate matters further, the subsampler introduces a considerable amount of aliasing. To mitigate the problem, the analysis filter bank includes a stage of butterfly aliasing reduction, as discussed later.

Second, the division of the frequency content into subbands of equal width do not accurately reflect the response of the basilar membrane, of which the width of critical bands is a good indicator. As a result, at low frequencies, a single filter bank subband extends over several critical bands. In this circumstance the noise masking thresholds cannot be specifically computed for individual critical bands and thus they are inaccurate.

Third, the polyphase filter bank and its inverse are not lossless transformations. The filter bank and its inverse in tandem, without a quantization in between, cannot perfectly reconstruct the signal. However, by design the error introduced is imperceptible (less than 0.07 dB ripple [11]).

Frequency inversion

Prior to cascading the subband outputs with the MDCT filter bank, each of the odd subbands must undergo a frequency inversion correction so that the spectral lines will appear in proper monotonic ascending order [30]. The frequency inversion consists of multiplying each odd sample in each odd subband by -1, as illustrated in Fig. 4.1.

MDCT filter bank

To compensate for some polyphase filter deficiencies, the frequency inversed samples are processed using the Modified Discrete Cosine Transform (MDCT), of which the block diagram is shown in Fig. 4.1. Unlike the polyphase filter bank, the MDCT filter bank is a lossless transform. It further subdivides the subband outputs to provide finer frequency resolution. Layer III specifies two different block sizes for the MDCT: a short block of 6 samples and a long block of 18 samples. Since there is 50% overlap between adjacent time windows, the MDCT transforms 12 time samples to 6 spectral coefficients in the short block mode, and 36 time samples to 18 spectral coefficients in the long block mode. Although the short block length is one third that of the long block, the number of MDCT coefficients for a frame remains constant irrespective of the block size. This is achieved by replacing one long block by three short blocks in the short block mode.

When the conditions of pre-echo are detected, MDCT is triggered to switch to short windows. For the purpose of perfect reconstruction, the switching between short and long windows has to be smooth. A start and stop time window is employed in the transition between short and long block modes (Section 3.5). Fig. 4.5 displays the process of the long-start-short window switching. As we can see, the short sequence is composed of three short blocks, which overlap 50% with each other and with start (and stop) window at window boundaries. Thus, time is synchronized for different subband channels.

The polyphase filter bank and the MDCT filter bank are together called as the *Hybrid Filter Bank* for their adaptation to signal characteristics.

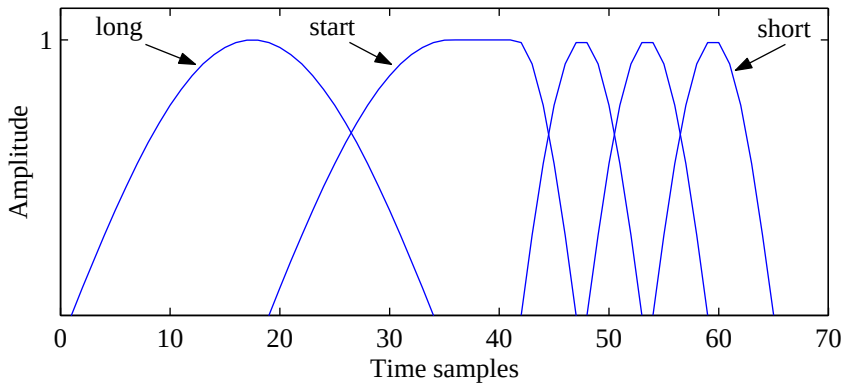
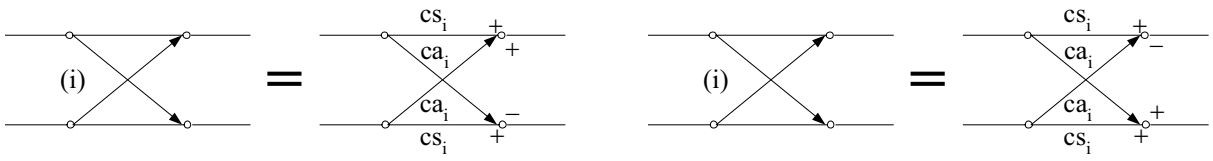


Fig. 4.5 Switching from a long sine window to a short one via a start window.

Aliasing reduction butterfly

Now the subband components are subdivided in frequency, some aliasing introduced by the polyphase filter bank can be partially cancelled. Layer III specifies a method of processing the MDCT coefficients to remove some aliasing caused by the overlapping bands of the polyphase filter bank.

This anti-alias operation is a number of butterflies applied to the 576 frequency lines (X_0 to X_{575})[1]. The butterfly is calculated between adjacent subbands by reading two values, multiplying and adding the values according to the butterfly in Fig. 4.6, and then put the new values back. The butterfly rotation coefficients are defined in "Audio Content", part 3 of the MPEG/audio standard [11].



(a) Encoder butterfly

(b) Decoder butterfly

Fig. 4.6 Layer III aliasing-butterfly, encoder/decoder [11].

Fig. 4.7 shows the alias reduction operation. As we can see, the group of 576 coefficients are rearranged for the anti-alias purpose: 18 coefficients of each subband are grouped

[1] $576 = 32 \times 18$, as 32 subbands contain 18 subsamples each.

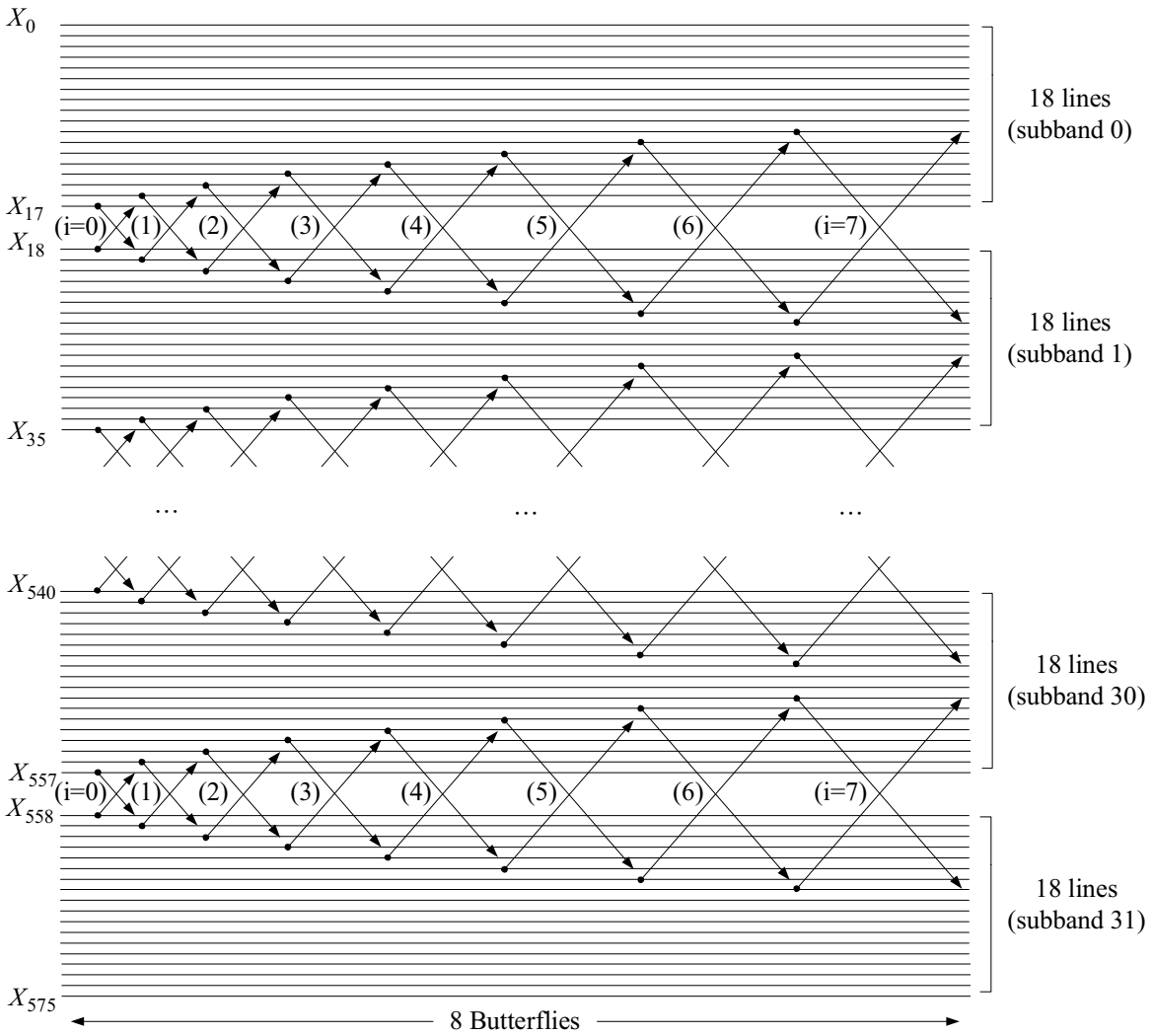


Fig. 4.7 Layer III aliasing reduction encoder/decoder diagram.

together. So, the butterfly is calculated on one of the eight designated pairs of spectral lines in every alternate subband.

4.1.2 AAC Transformation: Pure MDCT Filter Bank

Unlike the MP3 coder, AAC eliminates the polyphase filter bank and relies on the MDCT exclusively. Each block of input samples is overlapped by 50% with the immediately preceding block and following block.

The AAC filter bank resolution is adaptive to the characteristics of the input signal. This is done by switching between MDCT transforms whose block lengths are either 2048 or 256. Stationary signals are analyzed with a 2048-point window, while transients are analyzed with a 256-point window [31]. Therefore, the maximum frequency resolution is 23 Hz for a 48 kHz sample rate, and the maximum time resolution is 2.6 ms[2]. Block switching potentially generates a problem of time synchronization. If one channel uses a 2048-point transform and another channel uses a 256-point transform, the time interval will not be aligned. To maintain block synchronization for different block size channels, AAC combines eight 256-point short windows to a block and uses a start and stop window to bridge between long and short windows. The start window is defined as,

$$h_{start}(n) = \begin{cases} h_{long}(n), & 0 \le n \le M-1 \\ 1, & M \le n \le M + \frac{7M}{16} - 1 \\ h_{short}(n - \frac{4M}{16}), & M + \frac{7M}{16} \le n \le M + \frac{9M}{16} - 1 \\ 0, & M + \frac{9M}{16} \le n \le 2M - 1. \end{cases}$$

It preserves the time-domain aliasing cancellation property of MDCT and maintains block synchronization. The whole switching procedure is similar to the MDCT in MP3 hybrid filter bank, though it has different window sizes and thus different number of short windows to align with.

[2] 24 kHz/1024 MDCT coefficients = 23 Hz; 128 new samples/48000 samples per second = 2.6 ms.

4.2 Performance Evaluation

4.2.1 Full Coder Description

Full audio coders are implemented to explore the effectiveness of two decomposition structures, the hybrid filter bank (MP3 decomposition) and the pure MDCT filter bank (AAC decomposition). Our coders work in a wide-band regime with the sample frequency of 44.1 kHz, consisting of the decomposition filter banks, the psychoacoustic models, the scalar quantizers, and the decoders. The block diagram of the encoder is shown in Fig. 4.8.

Our main goal is not to design a complete coder but merely a prototype of one sufficiently sophisticated and general to produce NMR-coded files, thereby permitting experimental comparisons between the two decomposition structures. Consequently, we are not concerned with the design of an entropy coder mapping the quantized values to binary sequences, nor the coding of the side information.

Decomposition Filter bank

In our two encoder-decoder structures, the first time-frequency decomposition is the hybrid filter bank as used in the MP3, shown in Fig. 4.1, and the second a pure MDCT filter bank, as used in the AAC. The block sizes we are testing are 1152-sample blocks (26 ms time frames), using 50% overlap. In both decomposing structures, the window switching controls are not implemented. In effect, we are comparing a polyphase filter bank followed by 36-point MDCT with aliasing reduction butterfly to a pure 1152-point MDCT filter bank.

We notice that both the hybrid and the pure MDCT systems are critically sampled, because of the 50% overlap. The pure MDCT structure is quite straightforward, as the MDCT is itself critically sampled. For the hybrid structure (Fig. 4.1), a 1152-sample block is first subband-filtered and decimated by 32, thus producing 36 outputs in each channel, which are then transformed to 18 spectral lines by the 36-point MDCT. The butterfly stage does not affect the sampling rate. Therefore, there are totally 18×32 subbands $= 576$ spectral lines. As the 1152-sample blocks are 50% overlapped with each other and thus contain only 576 new samples, the whole system functions to convert 576 new input samples to 576 spectral lines and maintains the critical sampling property.

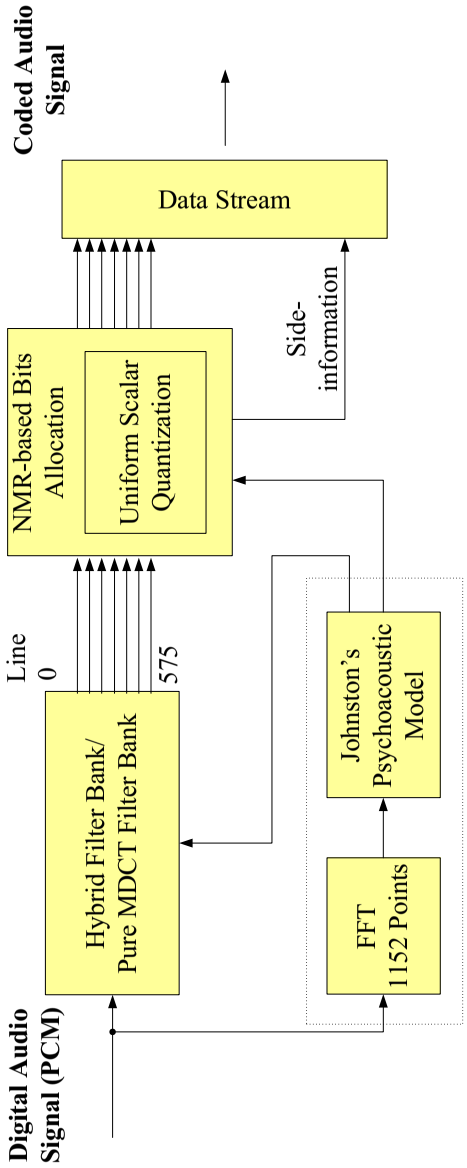


Fig. 4.8 Block diagram of the encoder of the full audio coder.

Psychoacoustic model

There is one psychoacoustic evaluation per frame. The audio data is first mapped to frequency domain. Both our coders use a Fourier transform for this mapping. The frequency values are then processed based on the steps in Johnston's psychoacoustic model (Section 2.2). Johnston's model is not identical to the psychoacoustic models used in Layer II and Layer III, but it is a reasonable choice here, since our purpose is to test the influence of different time-frequency decompositions as long as the distortion function is fixed. The output of the psychoacoustic model is a group of 25 masking thresholds, corresponding to 25 critical bands in the perceptual domain.

The audio data sent to the psychoacoustic model must correspond with the audio data to be coded, meaning that the psychoacoustic analysis should be applied to the exact 1152 samples to be transformed. It is desirable to have the number of mapped values equal to the number of coefficients decomposed from the filter bank. For this purpose, we choose to eliminate the DC term (first value) of the DFT because human hearing only goes down to 20 Hz, so it is irrelevant what the frequency content is at 0 Hz.

A standard Hann weighting, conditioning the data to reduce the edge effects, is applied to the audio data before Fourier transformation. We use a 1152-point Hann window to provide complete coverage for the samples to be coded, while Layer III uses a smaller window size of 1024 to reduce the computational load[3].

Quantization and bit allocation

Spectral components decomposed by the filter bank are firstly grouped into subbands (generally critical bands). A group of spectral coefficients is normalized by a common factor and is then quantized using the same quantizer resolution (same step size δ). Accordingly, the common normalization factor is called *scalefactor* and different groups of spectral coefficients are called *scalefactor bands* [32, 33].

In our coders, we use the spectral peak value (maximum absolute value) of coefficients within critical band i as the scalefactor R_i to give a good indication about the signal amplitude in that band. Scalar uniform quantization (Section 2.3.4) is then done for each normalized X_k ($X_k = X_k/R_i$) in each of the ith critical band. For each critical band i, the

[3]This is a compromise though, in that samples falling outside the analysis window generally have no major impact on the psychoacoustic evaluation [16].

scalar quantizer (mid-riser) is tested and operates as follows:

- If $R_i = 0$, then $B_i = 0$, $L_i = 1$ and $\hat{X}_k = 0$,
- If $R_i \neq 0$, then $L_i = 2^{B_i}$ and $\hat{X}_k = (2/L_i) \times \text{round}(X_k \times L_i/2)$,

where R_i is the quantizer range (scalarfactor), B_i is the allocated bits, L_i is the number of quantizer levels, round(.) denotes the nearest integer operation and $\hat{X}_k$ represents the quantized levels (between $[-1, 1]$) of each normalized spectral coefficient X_k. This operation assigns a bit rate of zero bits to any signal with an amplitude that does not need to be quantized, and assigns a bit rate of B_i to those that must be quantized. For example, if the bit assignment is 1, two levels $(-1, +1)$ are generated to quantize the particular component. As the signs of the various spectral coefficients are random, the sign information must be included. When no levels are necessary, a 0 is assigned and transmitted to the decoder.

The resolution of the quantizer (step size δ) is controlled carefully in the bit allocation loop according to the time-frequency dependent masking thresholds which are supplied by the perceptual model. If the quantization noise in a given band is found to exceed the masking threshold estimated by the perceptual model, the step size for this band is adjusted to reduce the quantization noise. Since achieving a smaller quantization noise requires a smaller value of quantization step size and thus a higher bit rate, the noise control loop (bit allocation loop) has to be repeated until the actual noise power (computed from the difference between the original spectral values and the quantized spectral values) is below the masking threshold for every scalefactor band and the total number of allocated bits satisfies the bit requirement[4]. The allocation of bits is performed with the *Greedy Algorithm* [18], which assigns one bit at each iteration to the band with the largest update NMR. A step-by-step implementation of the NMR-based greedy algorithm is described in Appendix A.

Bit rate

The data stream sent to the decoder consists of the quantized spectral values and the side information. In our particular coders, the side information includes a vector of 25 scalefactor values and a vector of 25 bit assignments. The information allows the receiver

[4]This means that bit allocation procedure continues even when transparent quality is achieved, provided that extra bits are available.

to recover the quantization scheme in the same way as the encoder, thus making explicit information on masking threshold unnecessary. Since the quantization and bit allocation scheme is fixed, the amount of side information is identical in both of our coders and thus is not quantized and used in the bit rate calculation.

We use the per sample bit rate to represent the decomposing capacities of different transformations. The reason is that all the decompositions used in our experiments are critically sampled, meaning that the number of spectral coefficients is a constant, though they are decomposed by different filter banks. Assuming a per sample bit rate of b, the bit rate of the audio signal is calculated as $(b \times M) \times (f_s/M) = b \times f_s$ bits/sample, where M is the number of quantized samples (also the number of spectral coefficients and the frame update rate), and f_s is the sample frequency of the input audio. For example, the MP3 codec operates on 48 kHz sampling rate and the bit rate is 64 kb/s for one channel, which leads to the per sample bit rate of 1.3 bits/sample.

The bits per sample value is computed based on the information content (entropy) of the quantized spectral coefficients from frame to frame. A step-by-step entropy calculation procedure is described in Appendix A. After the entropy of each coefficient is obtained, it is averaged across all quantized coefficients and the result value is the empirical entropy (bits/sample) of the encoded audio, which is exactly the lower bound of bits information if a entropy coder is used.

Decoder

The decoding operation is straight forward. The inversely quantized values are directly generated from the received scalefactors and bit assignment information. These values are further decoded by the synthesis filter bank. There are two synthesis filter banks corresponding to the two decomposition structures separately. The first one is precisely the transpose of the hybrid filter bank in Fig. 4.1, i.e., first butterfly decoded, and then the inverse 36-point MDCT and polyphase filter bank with frequency inversion in between. The decoding flow chart is described in "Audio Content", part 3 of the MPEG/audio standard [11]. The second one is simply an inverse 1152-point MDCT. Synthesized outputs shall be the reconstructed PCM audio samples.

4.2.2 Audio Quality Measurements

To compare different audio coders, we can refer to a number of factors, such as signal bandwidth, bit rate, quality of reconstructed audio and computational complexity. Among them, bit rate and quality of the reconstructed audio signal are two fundamental attributes of an audio coder, and they are intimately related: the lower the bit rate, the lower the quality.

There are basically three quality measurement methods: objective measurement, subjective listening tests, and perceptual measurement techniques.

The objective measurement is the traditional signal-to-noise ratio (SNR),

$$SNR = \frac{\sigma_x^2}{\sigma_e^2} = \frac{E\{x^2[n]\}}{E\{e^2[n]\}} \tag{4.5}$$

defined from the concept of mean square error $e(n)$ between the original signal and the decoded signal. However, relying on SNR of the decoded signal does not show much understanding of the paradigm of perceptual coders, which is: separate the inaudible artefacts from audible distortions and improve the subjective quality by shaping the quantization noise over frequency. Thus, objective measures do not satisfy the evaluation requirements of the perceptual audio coders. The reliable method to assess audio quality of perceptually coded signals has been subjective listening tests.

In the subjective listening test, the listeners can switch between the original signal (reference), R, and two other signals, A and B. Both these two signals are reconstructed signals, though they are processed by different audio codecs. The test has to be *double blind*, meaning that neither the listeners nor the supervisor knows which of the signals A and B is decomposed by its corresponding coding structure. The listeners have to judge the overall quality of the signals and decide that, which signal sounds better or whether two signals sound no difference.

Because listening tests are very time consuming and expensive, there has been new measurement methods which are capable of yielding a reliable estimate of the perceived sound quality. For years' work, ITU-R Task Group standardized the perceptual measurement techniques and recommended a system called PEAQ (Perceptual Evaluation of Audio Quality). Similarly, in the field of speech coding, perceptual measurement methods have been introduced known as PESQ (Perceptual Evaluation of Speech Quality) [34]. PESQ

simulates the ear model and predicts the subjective *Mean Opinion Score* (MOS) [35]. Although MOS operates on a scale from 1.0 (unacceptable quality) to 5.0 (excellent quality), as shown in Table 4.1, the PESQ values lie between 1.0 (bad) and 4.5 (no distortion).

Table 4.1 MOS is a number mapping to the above subjective quality.

Excellent	Good	Fair	Poor	Bad
5.0	4.0	3.0	2.0	1.0

4.2.3 Experiment Results

Experiment set-ups

We conduct subjective listening tests on the coded audio files. While our listening tests are carried out on a small scale, we apply the PESQ tests to coded speech files as a supplement to the subjective results.

A wide range of source files must be tested to ascertain which decomposition has a better frequency interpretation and is more robust to quantization. In our case, we choose representation set of material including single instrumental music, single speaker speech, and music with mixed types. Six test audio files are from EBU-SQAM (European Broadcasting Union — Sound Quality Assessment Material) and the other two are difficult-to-code material.

In the subjective tests, all sound files were first randomly ordered to eliminate the order-preference to the testing sequence. Then the quality of the coded signals were evaluated through informal listening tests. Eight coded files including speech and music were presented over loud speakers to five untrained listeners in a quiet room. The test is double blind, in our case, none of the listeners know which of the signals is decomposed by the hybrid filter bank or by the pure MDCT filter bank. The listeners had to judge the overall sound quality and give their preference.

Results: subjective and PESQ tests

The results of subjective listening tests are shown in Table 4.2. The per sample bit rates of the coded files, decomposed by the hybrid and pure MDCT filter banks, have been adjusted as close to each other as possible. Comparing between them, the listeners preferred the

quality of 5 files using the pure MDCT decomposition of all 8 coded files. For the other 3 cases, the quality of sound files decomposed by the pure MDCT filter bank is not worse than that decomposed by the hybrid filter bank, with the only exception for the glockenspiel. On average, the pure MDCT filter bank outperforms the hybrid filter bank.

Table 4.2 Subjective listening tests: Hybrid filter bank (*Hybrid*) vs. Pure MDCT filter bank (*Pure*)

Sound Files	*Hybrid* (bits/sample)	*Pure* (bits/sample)	*Hybrid* - No Preference - *Pure*
Violin	0.572	0.569	1 - 1 - 3
Flute	0.614	0.594	1 - 0 - 4
Glockenspiel	0.405	0.404	3 - 2 - 0
Piano	0.371	0.372	0 - 2 - 3
Vega	0.981	0.979	2 - 0 - 3
Seal	1.292	1.289	2 - 1 - 2
Female Speech	0.853	0.856	2 - 1 - 2
Male Speech	0.980	0.975	0 - 1 - 4

PESQ tests are applied on the two speech files from EBU-SQAM. Since the PESQ software runs on a narrowband basis, both the reference speech files and the coded ones are first subsampled to 8 kHz and then PESQ tested. The results are shown in Table 4.3. The pure MDCT filter bank slightly outperforms the hybrid filter bank in both files. The PESQ experiment results are in accordance with the subjective judgements. Therefore, we conclude that a pure filter bank provides better performance than a hybrid one.

Table 4.3 PESQ MOS values: Hybrid filter bank (*Hybrid*) vs. Pure MDCT filter bank (*Pure*)

Speech Files	*Hybrid* (bits/sample)	*Pure* (bits/sample)	*Hybrid* - *Pure* MOS Values
Female Speech	0.853	0.856	3.206 - **3.211**
Male Speech	0.980	0.974	3.546 - **3.549**

The per sample bit rate is low because none of the test passages was coded at a transparent quality. All files were purposely coded at a close-to-transparent level and thus slight distortions were introduced for the goal of subjective comparison. For transparent coding,

we used the notoriously hard-to-code material "vega" to test our coders since the coder will be transparent for all audio inputs if it is transparent in the crucial test involving difficult-to-code material. Our subjective tests report a bit rate of 2.305 bits/sample for transparent coding of "vega". In addition, our PESQ tests report a bit rate around 2.074 bits/sample for transparent coding of the two speech files from EBU-SQAM. The reconstructed files can give a PESQ value of 4.

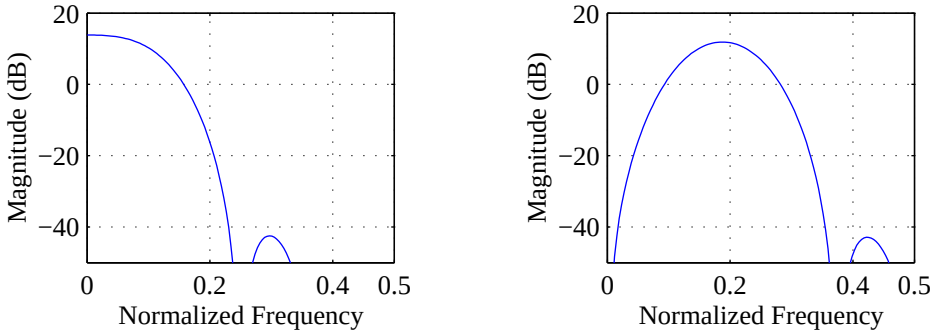
4.3 Psychoacoustic Transforms of DFT and MDCT

4.3.1 Inherent Mismatch Problem

The purpose of psychoacoustic model is to estimate the maximum allowable distortion, represented as masking thresholds. Since the psychoacoustic model runs in frequency domain, it is possible to use the output from time-frequency mapping filter bank as the input for psychoacoustic model, or to perform a separate transform for the purpose of psychoacoustic analysis. For example, the AAC codec uses the MDCT filter bank to decompose the audio while using a separate DFT filter bank for psychoacoustic processing.

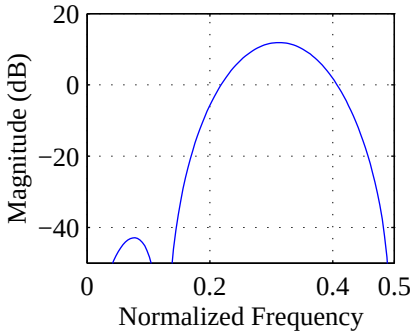
This could be a problem. Simply put it, the DFT filter bank used in the model cannot always simulate energy values from the codec MDCT filter bank. Because of this, energy estimation might be incorrect and therefore psychoacoustic output would be inaccurate. In the quantization stage, the audio coder decouples the psychoacoustic part from the quantization part: a DFT computes masking thresholds but one quantizes in MDCT domain. But for the analysis-synthesis loop involving excitation distortion minimization, the two cannot be decoupled since there would be an inherent mismatch. For instance if the quantizer is removed, the distortion should be zero, but it would not be since each excitation pattern would be generated by a different time-frequency processing. This mismatch is illustrated in Fig. 4.9, represented by the frequency responses of the MDCT basis functions.

In the figure, the MDCT seems to perform a projection of the time samples onto one set of basis functions $h_k[n]$ for the forward transform (MDCT), and then use this set and the set $h_k[n + M]$ for signal reconstruction (IMDCT). The document looks at the DFT of functions $h_k[n]$. For the set of basis of functions, except for the DC term at $k = 0$, none of the frequency responses for $h_k[n]$ achieve their maximum at the "expected" value of $k \times \pi/M$. The main lobe for each of the responses has very large bandwidth, the center

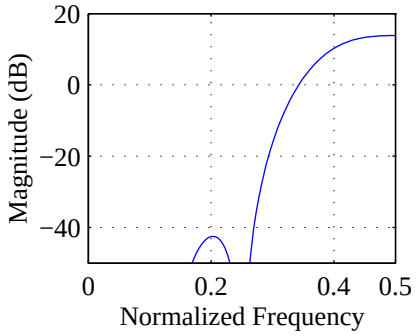


(a) $H_0(\omega)$

(b) $H_1(\omega)$



(c) $H_2(\omega)$



(d) $H_3(\omega)$

Fig. 4.9 Frequency response of the MDCT basis function $h_k(n)$, $M = 4$.

of which is moving to higher frequencies as k increases.

It is important to note that, with the MDCT, while past samples can be used to get rid of the time aliasing in the first 50% of the frame, the time aliasing in the latter 50% of a frame would still be very severe, and so the computation of the DFT given the MDCT coefficients of such a frame would be misleading. This raises the question of developing a transform which has less time aliasing, that is, less amount of overlap. We will elaborate this motivation in Section 5.1.

4.3.2 Experiment Results

In this part, we investigate the use of MDCT spectrum as input to the perceptual model. Performances of coded files, using the DFT spectrum or the MDCT spectrum for psychoacoustic analysis, are tested under the condition of a pure MDCT filter bank decomposition. We experimented both the subjective listening tests and PESQ measurements.

In the subjective tests, compared to the MDCT spectrum, the listeners unanimously believed that the DFT spectrum delivered better quality for most music passages and never performed worse. The PESQ values are shown in the Tables 4.4. As we can see, comparing to the MDCT spectrum, the DFT spectrum generated better speech quality.

Experiment results are different from what we had expected. One possible reason is that DFT spectrum is a complex spectrum and the imaginary values are instrumental to tonality estimation. In our tests, the DFT spectrum produced better psychoacoustic analysis because the advantage of complex spectrum outweighted the disadvantage of energy mismatch.

Table 4.4 PESQ MOS values: DFT spectrum (*DFT*) vs. MDCT spectrum (*MDCT*)

Speech Files	*DFT* (bits/sample)	*MDCT* (bits/sample)	*DFT - MDCT* MOS Values
Female Speech	0.856	0.867	**3.211** - 3.197
Male Speech	0.975	0.961	**3.549** - 3.460

Chapter 5

Partially Overlapped Lapped Transforms

In this chapter, we present a new partially overlapped yet critically sampled transform, the Pre-DST lapped transform. The transform can vary the amount of overlap between neighboring blocks (let $M < L \leq 2M$) and, hence, have fine control over the coding performance.

5.1 Motivation of Partially Overlapped LT: NMR Distortion

We explored the current decomposition structures in MPEG standards: the hybrid filter bank in MP3 and the pure MDCT filter bank in AAC (Chapter 4). They are all based on 50% overlapped frames and use overlap-add for signal reconstruction. There is a problem with these coders: quantized NMR is not the same as reconstructed NMR. Der showed in [36] that, by overlapping frames, there exists two versions of the "reconstructed" NMR patterns. Version 1 is derived directly from the quantized spectral coefficients. Version 2 is derived from spectral analysis of the final time-domain coded signal, *after* overlap-add. These two versions of NMR patterns will not be the same if there is overlap in frames. It is obvious that the "correct" reconstructed NMR pattern is the one obtained only after time-domain addition, because this is the signal upon which the listeners perform the perceptual processing. Thus, *NMR distortion*, generated from the difference between the intermediary signal (version 1) and reconstruction signal (version 2), will apply at any transforms with overlap greater than zero.

There exist two solutions to the posed dilemma. The first is to forego overlapped representations. There are a few problems with this approach. First, it prohibits any lapped-transform decomposition; among them the Modified Discrete Cosine Transform (MDCT), which is perhaps the most popular and widely-used transform in audio coding. Second, the quantized coefficients must come from a non-overlapped analysis; by the Nyquist constraints for perfect reconstruction, the only time-window that may be used is a rectangular one, which has relatively poor sidelobe suppression properties. Finally, it is well-known that non-overlapped reconstruction in transform coders result in discontinuities at frame boundaries due to quantization: the result is a highly audible low-frequency clicking, known as blocking edge effects.

The other possibility, and the one which we pursue in this chapter, is to develop a transform which has less time aliasing in the overlap-add procedure, that is, less overlap. The transform we explore would be a partially overlapped yet critically sampled transform.

5.2 Construction of Partially Overlapped LT

5.2.1 MLT as DST via Pre- and Post-Filtering

The 50% overlapped modulated lapped transform (MLT) can be implemented efficiently by means of a fast transform of length M, as explained in [37] by Malvar. Assuming we have the MLT in a common form as

$$X_k(m) = \sqrt{\frac{2}{M}} \sum_{n=0}^{2M-1} x_m(n) h(n) \cos\Big[\frac{(n+\frac{M+1}{2})(k+\frac{1}{2})\pi}{M}\Big], \tag{5.1}$$

where m is the block index and $h(n)$ is the lowpass prototype filter.

Defining a new sequence $y_m(n)$ as

$$y_m(n) = \begin{cases} x_m(n+M/2)h(n+M/2) - x_m(M/2-n-1)h(M/2-n-1), \\ \qquad n = 0, \ldots, M/2-1, \\ x_m(n+M/2)h(n+M/2) + x_m(5M/2-n-1)h(5M/2-n-1), \\ \qquad n = M/2, \ldots, M-1, \end{cases} \tag{5.2}$$

we can rewrite the Eq. (5.1) as,

$$X_k(m) = \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} y_m(n) \sin\Big[\frac{(n+\frac{1}{2})(k+\frac{1}{2})\pi}{M}\Big]. \tag{5.3}$$

The above equation shows that the MLT outputs can be obtained by applying on the sequence $y_m(n)$ the Type-IV Discrete Sine Transform (DST-IV), as defined in Eq. (3.36). Therefore, a MLT filter bank can be implemented in two steps: first, we compute the butterflies in Eq. (5.2); and second, we calculate the DST-IV of the sequence $y_m(n)$ as in Eq. (5.3).

As we have shown in Chapter 3, MDCT is a special case of MLT with a particular choice of the prototype filter (a sine window), so that it can be implemented in the same fashion of the two steps. The flowgraph of the forward MDCT is illustrated in Fig. 5.1, where the butterfly coefficients are $C_i = \cos\big[(2i-1)\pi/4M\big]$ and $S_i = \sin\big[(2i-1)\pi/4M\big]$. The flowgraph of the inverse MDCT is just the transpose of that of Fig. 5.1. Now the transform consists of a butterfly pre-processor and a DST-IV decomposition stage.

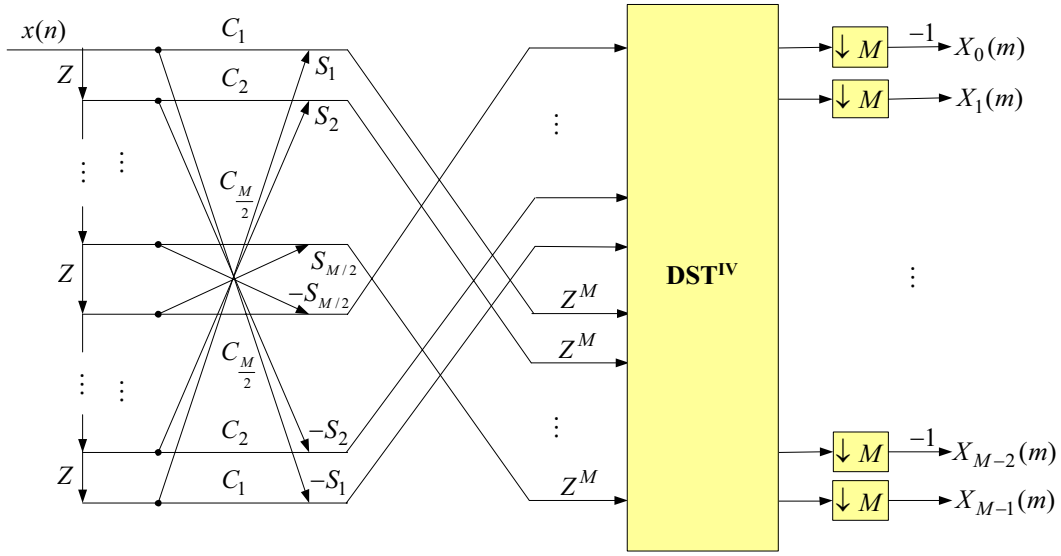


Fig. 5.1 Flowgraph of the Modified Discrete Cosine Transform [37].

In Fig. 5.1, we note that the input signal passes through a M-decimator after the DST-IV, that $M/2$ channels with a advance of M samples now appear before the DST-IV, and

that the outputs are only computed for every M samples that are shifted in. We put the M-decimator before the butterfly so that the z^M advance becomes z^1 advance. The rearranged system is illustrated in Fig. 5.2, where the matrix $\mathbf{B}$ is the butterfly-coefficient

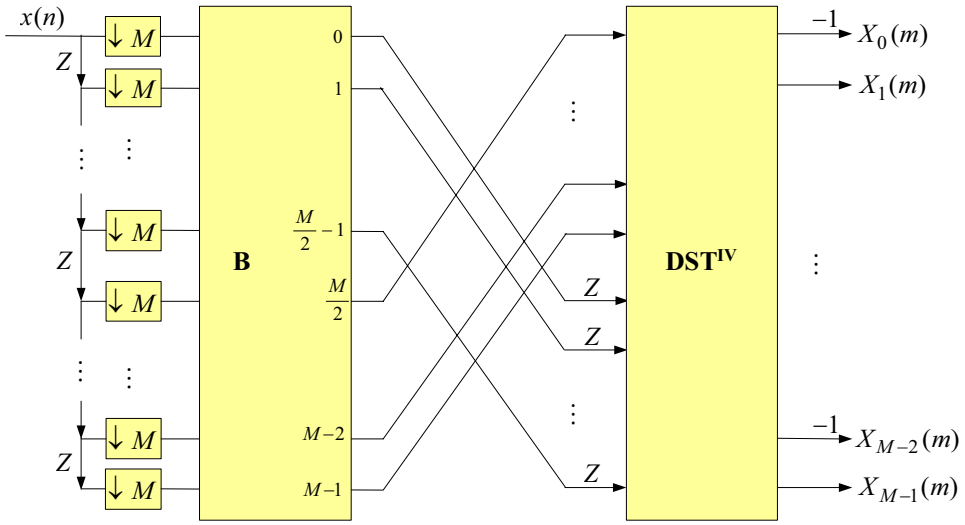


Fig. 5.2 Flowgraph of MDCT as block DST via butterfly pre-filtering.

matrix, as given in Eq. (5.4).

$$\mathbf{B_N} = \begin{bmatrix} C_1 & \cdots & & & & \cdots & S_1 \\ \vdots & C_2 & & & & S_2 & \vdots \\ & & \ddots & & \iddots & & \\ & & & C_{\frac{N}{2}} & S_{\frac{N}{2}} & & \\ & & & -S_{\frac{N}{2}} & C_{\frac{N}{2}} & & \\ & & \iddots & & \ddots & & \\ \vdots & -S_2 & & & & C_2 & \vdots \\ -S_1 & \cdots & & & & \cdots & C_1 \end{bmatrix}_{\mathbf{N}\times\mathbf{N}} \tag{5.4}$$

Now the MDCT can be viewed as a combination of the common block-based DST with simple time-domain pre-filtering. Since the inputs to the $M/2$ branches with z^1 advances are from next data frame, the input block to DST-IV is essentially the second half of the current butterfly outputs plus the first half of the next butterfly outputs. Thus, the whole

system can be viewed globally as in Fig. 5.3.

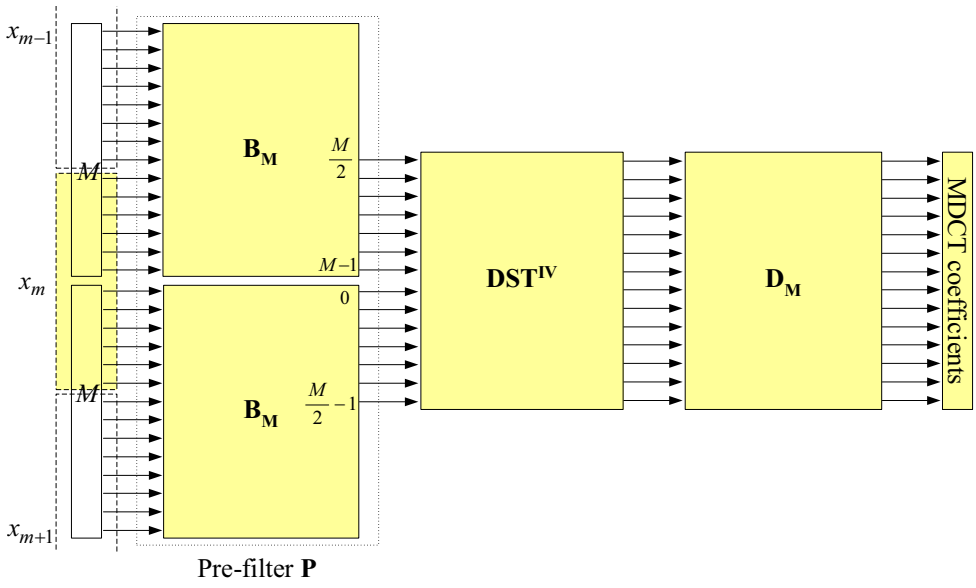


Fig. 5.3 Global viewpoint of MDCT as pre-filtering at DST block boundaries.

It is important to note that the current frame contains the M samples in the dotted-line block, instead of the M samples of MDCT frame in the solid-line block. The 50% overlapping is achieved by borrowing $M/2$ samples in the dotted-line blocks from neighboring frames. In the decomposition stage, $\mathbf{B}$ acts as the pre-filter working across the block boundaries, taking away interblock correlation; the pre-filtered time samples are then fed to the DST to be encoded as usual.

In Fig. 5.3, the forward transform matrix $\mathbf{H}$ can be expressed in the matrix form as

$$\mathbf{H} = \mathbf{D}_M \mathbf{S}_M^{IV} \mathbf{H}_{\text{pre}} \mathbf{P}_{2M}, \tag{5.5}$$

where

$$\mathbf{P} = \begin{bmatrix} \mathbf{B}_M & \mathbf{0}_{M\times M} \\ \mathbf{0}_{M\times M} & \mathbf{B}_M \end{bmatrix}_{\mathbf{2M\times 2M}} \tag{5.6}$$

$$\mathbf{H}_{\text{pre}} = \begin{bmatrix} \mathbf{0}_{\frac{M}{2}\times\frac{M}{2}} & \mathbf{I}_{\frac{M}{2}\times\frac{M}{2}} & \mathbf{0}_{\frac{M}{2}\times\frac{M}{2}} & \mathbf{0}_{\frac{M}{2}\times\frac{M}{2}} \\ \mathbf{0}_{\frac{M}{2}\times\frac{M}{2}} & \mathbf{0}_{\frac{M}{2}\times\frac{M}{2}} & \mathbf{I}_{\frac{M}{2}\times\frac{M}{2}} & \mathbf{0}_{\frac{M}{2}\times\frac{M}{2}} \end{bmatrix}_{\mathbf{M\times 2M}} \tag{5.7}$$

$$\mathbf{D_M} = \begin{bmatrix} -1 & & & & \\ & 1 & & & \\ & & -1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}_{\mathbf{M\times M}} . \tag{5.8}$$

The pre-filter matrix $\mathbf{P}$ is composed of the butterfly matrix $\mathbf{B}$ and applied to $2M$ time samples, $\mathbf{H}_{\text{pre}}$ is the $2M \rightarrow M$ mapping operator, $\mathbf{S}_M^{IV}$ is the DST-IV transform of length M, and $\mathbf{D}$ is the diagonal matrix inverting the polarity of the transform coefficients. We label this kind of Pre-filtered Discrete Sine Transform as the *Pre-DST* lapped transform.

5.2.2 Smaller Overlap Solution

Similar to the smaller overlap approach on the type-II fast Lapped Orthogonal Transform (LOT) in [38], our partially overlapped MLT is derived from an observation of the structure in Fig. 5.3. The amount of overlap can be lowered by reducing the size of the pre-processing matrix $\mathbf{B}$. An $M \times L$ LT, where $L \leq 2M$ and $M \geq 2$, can be easily constructed with a $(L-M) \times (L-M)$ pre-filter $\mathbf{B}$, which has the same form as Eq. (5.4), except that the matrix $\mathbf{B}$ is now of the size $L-M$. In an extreme situation where $L = M$ (0% overlap), the pre-filtering is turned off and the system turns into a disjoint DST-IV transform. The diagram of the partially overlapped LT at arbitrary overlaps is shown in Fig. 5.4.

The matrix representation of the Pre-DST LT in Fig. 5.4 is

$$\mathbf{H} = \mathbf{D}_M \mathbf{S}_M^{IV} \mathbf{H}_{\text{pre}} \mathbf{P}, \tag{5.9}$$

where $\mathbf{P}$ is the pre-filtering matrix applied to L time samples and $\mathbf{H}_{\text{pre}}$ is a $L \rightarrow M$ mapping operator. The full system functions to convert L time samples to M transform coefficients.

$$\mathbf{P} = \begin{bmatrix} \mathbf{B}_{L-M} & \mathbf{0}_{(L-M)\times(2M-L)} & \mathbf{0}_{(L-M)\times(L-M)} \\ \mathbf{0}_{(2M-L)\times(L-M)} & \mathbf{I}_{(2M-L)\times(2M-L)} & \mathbf{0}_{(2M-L)\times(L-M)} \\ \mathbf{0}_{(L-M)\times(L-M)} & \mathbf{0}_{(L-M)\times(2M-L)} & \mathbf{B}_{L-M} \end{bmatrix}_{\mathbf{L\times L}} \tag{5.10}$$

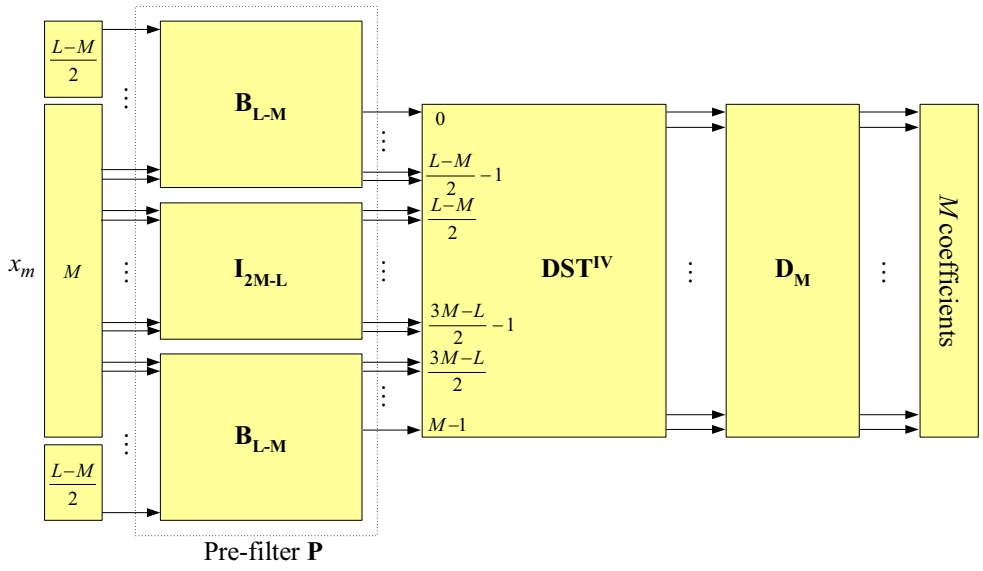


Fig. 5.4 Pre-DST lapped transforms at arbitrary overlaps ($L < 2M$).

$$\mathbf{H}_{\text{pre}} = \begin{bmatrix} \mathbf{0}_{\frac{L-M}{2}\times\frac{L-M}{2}} & \mathbf{I}_{\frac{L-M}{2}\times\frac{L-M}{2}} & \mathbf{0}_{\frac{L-M}{2}\times(2M-L)} & \mathbf{0}_{\frac{L-M}{2}\times\frac{L-M}{2}} & \mathbf{0}_{\frac{L-M}{2}\times\frac{L-M}{2}} \\ \mathbf{0}_{(2M-L)\times\frac{L-M}{2}} & \mathbf{0}_{(2M-L)\times\frac{L-M}{2}} & \mathbf{I}_{(2M-L)\times(2M-L)} & \mathbf{0}_{(2M-L)\times\frac{L-M}{2}} & \mathbf{0}_{(2M-L)\times\frac{L-M}{2}} \\ \mathbf{0}_{\frac{L-M}{2}\times\frac{L-M}{2}} & \mathbf{0}_{\frac{L-M}{2}\times\frac{L-M}{2}} & \mathbf{0}_{\frac{L-M}{2}\times(2M-L)} & \mathbf{I}_{\frac{L-M}{2}\times\frac{L-M}{2}} & \mathbf{0}_{\frac{L-M}{2}\times\frac{L-M}{2}} \end{bmatrix}_{\mathbf{M}\times\mathbf{L}} \tag{5.11}$$

The flowgraph of the inverse transform is described in Fig. 5.5. The received signal is first inverse-DST transformed, then post-filtered along with the $(L-M)/2$ coefficients from the previous and next frame. Finally the reconstructed signal is obtained as the combination of the post-filtered samples with the outputs directly from the inverse DST.

The Pre-DST system has several advantages. First, it is a perfect reconstruction system, which is structurally guaranteed by the orthogonality of the butterfly-coefficient matrix $\mathbf{B}$, matrix $\mathbf{S}^{IV}$ and polarity conversion matrix $\mathbf{D}$. In addition, the whole system is fast computable because the DST-IV module can be implemented by one of the many fast algorithms.

Second, though the block length of the transform is L, the partially-overlap Pre-DST is critically sampled in that one data block contains only M new time samples. The various overlap percentage is achieved by borrowing different amount of samples from neighboring

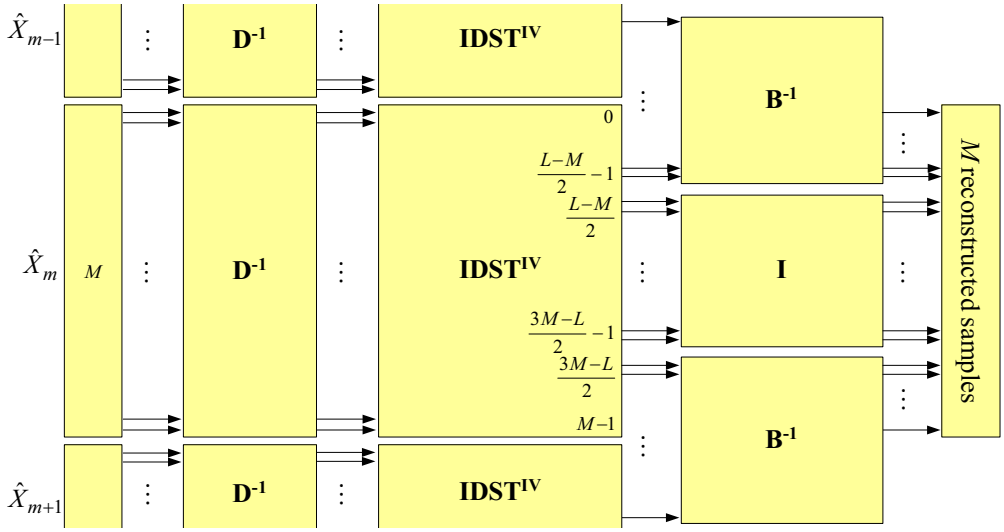


Fig. 5.5 Post-DST lapped transforms at arbitrary overlaps ($L < 2M$).

frames.

Third, the Pre-DST system has the property of cascade linear phase. It is true that the impulse responses of the Pre-DST do not have even/odd symmetry, therefore their frequency responses do not have linear phase. Nevertheless, linear phase alone is not generally a required property, since in coding applications if a signal is processed by the kth analysis filter, it will also be processed by the kth synthesis filter. If the analysis filters are equal to the time-reversed synthesis filters, the overall impulse response of any channel has even symmetry and so the cascade connection has linear phase. Through some elementary matrix manipulations, it is easy to verify that the Pre-DST has the property of the cascade linear phase.

5.3 Performance Evaluation

5.3.1 Pre-echo Mitigation

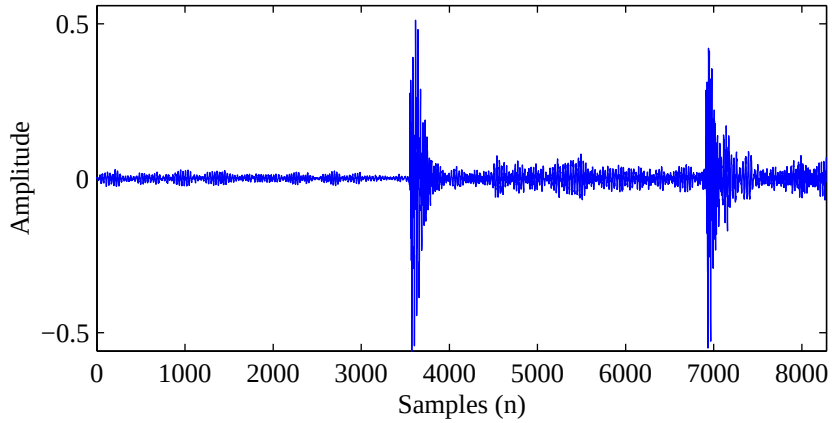
As mentioned, the motivation behind developing a partially overlapped LT is to reduce the NMR distortion in the overlap-add procedure. In this section, we show that the partially overlapped Pre-DST can effectively mitigate the frame-to-frame pre-echo artefact, one NMR distortion.

The pre-echo artefact arises in perceptual coding systems. It occurs when a signal with a sharp attack begins near the end of a transform block immediately following a region of low energy. The quantization error will then be spread out over some time before the music attack. For a block-based algorithm, when quantization and encoding are performed in order to satisfy the masking thresholds associated with the average spectral estimate within the analysis window, the quantization error in the coder is added to the spectral components as a signal of the window length. Thus, the inverse transform will spread the quantization error evenly in time over the full window length. This results in audible distortions throughout the low energy region preceding in time the signal attack. Pre-echoes can arise when coding recordings of percussive instruments such as the castanets. There are a number of techniques to avoid audible pre-echoes, including window switching and temporal noise shaping, as mentioned in Section 2.3.1.

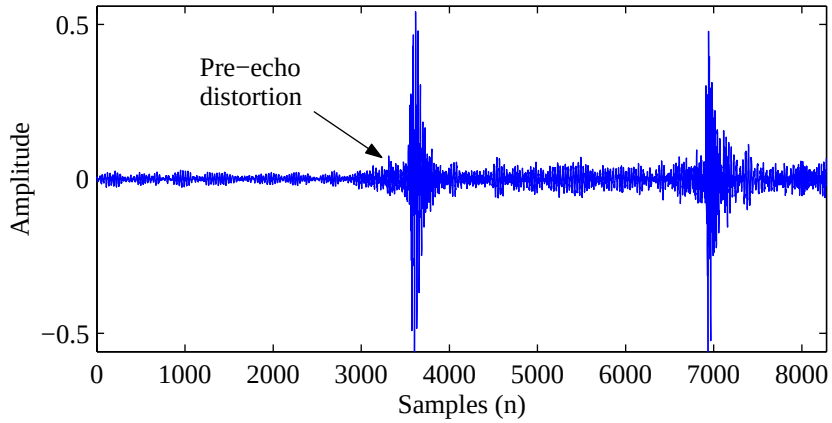
Our Pre-DST framework can be designed adaptive to transient signals and thus compensate for pre-echoes. Based on the energy of the transform coefficients generated, we can vary the number of borrowing samples dynamically while the frame size is fixed to M. For example, the pre-filtering operator can be chosen amongst: no filtering (0% overlap), borrowing $M/4$ samples (20% overlap), borrowing $2M/3$ samples (40% overlap), or borrowing M samples (50% overlap). Thus, we are switching from a $M \times M$ to a $5M/4 \times M$ to a $5M/3 \times M$ and to a $2M \times M$ Pre-DST. The price to be paid is a small increase of the side information used to specify the overlap. For instance, if the number of borrowing samples can be chosen from the set $\{0, M/4, 2M/3, M\}$, the side information increase for each frame is then 2 bits.

We use the standard test file "castanets" which has sharp transients to examine our Pre-DST decomposition in a full audio coder. Other parts of the audio coder, such as the psychoacoustic model and quantization, are identical to the audio coder in Section 4.2.1. The experiment set-ups are the same as those in Section 4.2.3. So, we are testing the Pre-DST decomposition at different overlaps and note that, at 50% overlap, the Pre-DST structure becomes the MDCT and our coder is precisely the *pure* MDCT audio coder in Section 4.2.1. The difference of experiment results are not only audible but also visible. Coded waveforms are shown in Fig. 5.6. Comparing to the MDCT (Pre-DST at 50% overlap), the 20% overlapped Pre-DST significantly reduces the pre-echoes of the waveform.

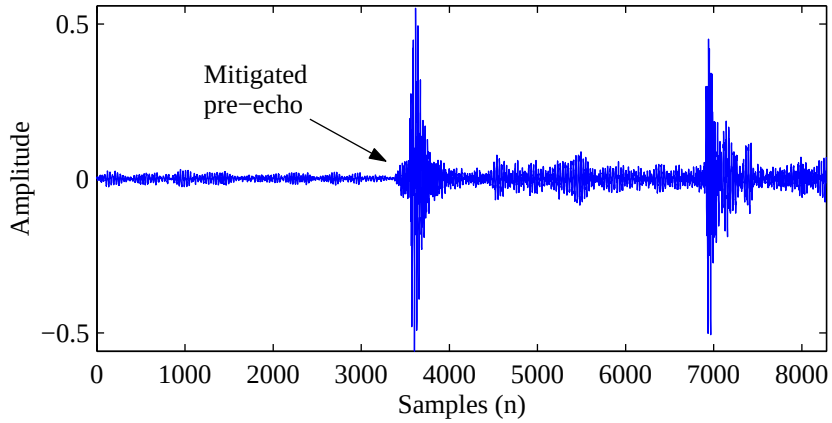
Bit rate performance is measured by computing the empirical entropy, which is a realistic measure in our critically sampled Pre-DST system. Results are shown in Table 5.1:



(a) Uncoded files.



(b) MDCT coded files, 576 new samples per block at 50% overlap.



(c) Pre-DST coded files, 576 new samples per block at 20% overlap.

Fig. 5.6 Partially overlapped Pre-DST example showing pre-echo mitigation for sound files of castanets.

as overlap reduces, bits per sample (empirical entropy) monotonically decreases. This is understandable because fewer coefficients are coded and thus less information content is involved. Again, all coding is performed at a non-transparent level and the per sample bit rate has been adjusted as close as possible.

5.3.2 Optimal Overlapping Point for Transient Audio

Obviously, fewer overlapping samples at each block boundary to be borrowed, more can we capture the time-varying characteristics of the transient audio and keep the pre-echoes under control. However, when reducing the overlap percentage, we increase another artefact of the blocking edge effects. Thus, from 0% overlap to 50% overlap, the pre-echoes increase while the blocking edge effects decrease. The question is how much overlapping balances both artefacts and sounds the reconstructed audio best as a whole. We refer to this overlap percentage as the *Optimal Overlapping Point*. Preliminary experiments are carried out on castanets to find the solution and the results are shown in Table 5.1.

Table 5.1 Subjective listening tests of Pre-DST coded test files of castanets.

Overlap Percentage	Sound Quality	Bits/Sample
10%	most distorted	0.950
20%	**least distorted**	0.969
30%	distorted	0.982
40%	more distorted	1.008
50%	more distorted	1.008

Our experiments show that the optimal overlapping point for sound files of castanets would be around 20%. This implies that an overlap of 20% is sufficient for the blocking effect reduction and simultaneously conceals the pre-echoes under the backward temporal masking thresholds of the sharp attack (Section 2.1.4). To reach a conclusive optimal overlapping point, more transient audio files have to be tested.

Chapter 6

Conclusion

The purpose of our research has been to explore the decomposition structures which could be used to compute perceptual distortion measures effectively, and to develop a transform operating at smaller overlaps. To accomplish our goal, two widely-used decomposition filter banks are implemented and their performance compared. In addition, we have proposed a partially overlapped lapped transform, the *Pre-DST*. This structure can be designed adaptive to the time-varying characteristics of input audio.

6.1 Thesis Summary

In Chapter 1 the major classes of coding paradigms, i.e., parametric coding and waveform coding were presented. Frequency-domain waveform coders employing the perceptual principle have been rendered as the best alternative for the coding of general audio signals. Additionally, the basic concept of time-frequency transformation was introduced and its application to audio coding was described.

Chapter 2 started with a description of the physiology of the human auditory system. The basic theory of loudness perception was introduced since it maps the input signal energy to sound pressure levels. The important concept of critical bands, which approximate the bandwidth of the auditory bandpass filters, was presented to explain the frequency resolution of the ear. Subsequently, auditory masking effects were discussed, highlighting on the simultaneous masking phenomenon.

Section 2.2 presented the well-known auditory models that predict the amount of masking produced by a complex audio signal. The model under study was the Johnston's model.

Following the discussion, the basic structure of a perceptual audio coder was presented in Section 2.3. Among other components, the decomposition filter bank, the psychoacoustic model, bit allocation and quantization were examined.

Chapter 3 provided a detailed analysis of lapped transforms. Lapped transforms are a proper choice for transform coders because they perform on overlapping blocks of data which reduces blocking edge effects. Modulated Lapped Transforms (MLT), which is produced through modulating cosine functions by a prototype lowpass time window, was analyzed; among them, the Modified Discrete Cosine Transform (MDCT) was noted for its wide popularity in audio coding. Modulated Lapped Transforms were compared to a equivalent filter bank representation. The effect of the prototype window on the frequency response of the resulting filter bank was discussed. Perfect reconstruction conditions in which an identical window is used in the analysis and synthesis stages were compared to the Lapped Orthogonal Transforms (LOT) in which two symmetric matrices are used. Finally, the issue of adaptive filter banks was addressed and a window switching method was analyzed as a form of adaptive filter bank to reduce pre-echo artefacts in audio coding.

In Chapter 4, we thoroughly discussed two main classes of decomposition schemes in audio coding of MP3 (MPEG-1 Layer III) and AAC (MPEG-2 Advanced Audio Coding). MP3 decomposition is a hybrid filter bank, which consists of a subband filter bank and a transform filter bank. Some information is lost during its signal decomposition. AAC decomposition is a pure MDCT filter bank which can perfectly reconstruct the original signal in the absence of quantization. However, in terms of perceptually transparent coding, no difference between the original and reconstructed signal can be perceived by the human ear in both methods.

In the following sections we described different blocks of our audio coders along with the related algorithms. An hybrid or pure MDCT filter bank was used to decompose the input signal into its spectral components. The spectral coefficients were grouped into 25 subbands to emulate the frequency analysis in the ear. To quantize the transform coefficients, a scalar quantization approach was taken. The bit allocation algorithm based on the Noise-to-Masking (NMR) ratio was introduced. In the process of quantization, the simultaneous masking thresholds were used to determine the acceptable noise level. Subsequently, following a description of performance evaluation measures, the performance assessment of MP3 and AAC decomposition was presented. It was argued that pure transform filter bank performs better than hybrid structure with subband and transform filter banks.

In Section 4.3, DFT-based and MDCT-based psychoacoustic analysis approaches were compared and the DFT-based approach performed better than the MDCT-based one.

Chapter 5 introduced a novel coding structure called Pre-filtering DST (Pre-DST). The novel structure first included the analysis of the Modified Discrete Cosine Transform (MDCT) that was presented in Chapter 3. Based on the analysis, the framework is proposed which can vary the overlap percentage at arbitrary degrees between blocks. The performance evaluation of the proposed coding structure was presented in Section 5.3. Performance improvements of Pre-DST were observed when coding transient audio signals, compared to the MDCT decomposition. In addition, the optimal overlap percentage to model transient signals was investigated and we reported an amount around 20%.

6.2 Future Research Directions

In this section, we make some suggestions for future research on more general aspects of lapped transforms in audio coding.

- Use a complex MDCT representation and compare to the DFT spectrum. The experiment results in Section 4.3 showed that DFT spectrum produced better reconstructed sound quality than MDCT spectrum. One possible reason is DFT generates a complex excitation power spectrum, and the incorporated imaginary parts can be instrumental. A complex MDCT spectrum could be constructed of MDCT and MDST.

- Further designs on the Pre-DST structure.

  - Instead of using a unitary matrix, experiment other diagonal matrices between the butterfly matrices to represent the pre-filter.
  - Use time windows other than the sine window to construct the butterfly matrix and test the performance.

- We have argued that applying the NMR criterion on the signal before overlap-add is inappropriate. This can be avoided by adopting a partially-overlap transform. However, several problems arise if a partially-overlap representation is used. As the overlap is reduced, the time window will have poor sidelobe suppression properties (consider the extreme condition, a rectangular window comes from 0%), resulting in poor frequency separation. As the overlap is reduced, the blocking edge effects at

frame boundaries due to quantization become more audible. Therefore, the task will be to find an overlap percentage which could balance all the beneficial and deleterious effects

Appendix A

Greedy Algorithm and Entropy Computation

A.1 Greedy Algorithm

The greedy algorithm is a simple and intuitive method for achieving integer-constrained bit allocation [18]. The algorithm is performed iteratively, ensuring an integer assignment of bits to each quantizer. At each iteration, one bit is allocated to the quantizer for which the decrease in a distortion measure is largest. The algorithm is *greedy* since bit allocations are optimized per iteration rather than considering the final distortion. The algorithm is summarized below.

Assume that B bits are available for N quantizers. Let $W_i(b)$ represent the distortion function associated with the ith quantizer having b bits. Additionally, let $b_i(m)$ represent the number of bits allocated to the ith quantizer after m iterations.

0 - Initialize the number of bits assigned to each quantizer to zero such that $b_i(0) = 0$ for $i = 1 \dots N$.

1 - Find the index j such that: $j = \text{argmax}_i\{W_i(b_i(m-1)) - W_i(b_i(m)\}$.

2 - Set $b_j(m+1) = b_j(m) + 1$ and $b_i(m+1) = b_i(m)$ for all $i \neq j$.

3 - Set $m = m + 1$. If $m \leq B$, return to step 1.

A.2 Entropy Computation

The entropy of a discrete random variable X is a function of its PMF (probability mass function) and is defined by [39]

$$H(X) = -\sum_{i=1}^{K} p_i \log(\frac{1}{p_i}), \tag{A.1}$$

where p_i is the probability of random event $X = a_i$, for all $i = 1, 2, \ldots, K$.

Assume we have the encoded signal $\hat{S}$, composed of N frames with M quantized coefficients each frame, in a matrix as

$$\hat{S} = \begin{pmatrix} C_{f_1}(1) & C_{f_2}(1) & \ldots & C_{f_N}(1) \\ C_{f_1}(2) & C_{f_2}(2) & \ldots & C_{f_N}(2) \\ \vdots & \vdots & \vdots & \vdots \\ C_{f_1}(M) & C_{f_2}(M) & \ldots & C_{f_N}(M) \end{pmatrix}, \tag{A.2}$$

where the ith column corresponds to the ith frame. All coefficients $C_f(i)$ (row-wise) constitute a sample space of real numbers and represent a random variable, denoted as X_i. It consists of values generated from the jth subband scalar quantizer of L levels. The set $L = \{l_1, l_2, \ldots, l_L\}$ denotes the set in which the random variable X_i takes its values. Similar to Eq. (A.1), the entropy of the discrete random variable X_i is given by

$$H(X_i) = -\sum_{i=1}^{L} p_i \log(\frac{1}{p_i}), \tag{A.3}$$

where p_i is the probability of random event $X = l_i$, for all $i = 1, 2, \ldots, L$.

Now that we have the entropy of the ith component X_i, the entropy of the whole encoded signal is simply the average entropy of M components, as

$$E_s = \sum_{i=1}^{M} H(X_i). \tag{A.4}$$

This is the per sample bit rate of the encoded audio signal, also known as the empirical entropy of the quantized coefficients.

References

[1] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proc. IEEE*, vol. 88, pp. 451–513, Apr. 2000.

[2] D. O'Shaughnessy, *Speech Communications: Human and Machine*. IEEE Press, second ed., 2000.

[3] H. Najafzadeh-Azghandi, *Perceptual Coding of Narrowband Audio Signals*. PhD thesis, McGill University, Montreal, Canada, Apr. 2000, (http://tsp.ECE.McGill.CA).

[4] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*. Springer-Verlag, 1990.

[5] B. C. J. Moore, *An Introduction to the Psychology of Hearing*. Academic Press, fourth ed., 1997.

[6] E. Terhardt, "Calculating virtual pitch," *Hear. Res.*, vol. 1, pp. 155–182, May 1979.

[7] B. Scharf, "Critical bands," in *Foundations of Modern Auditory Theory* (J. V. Tobias, ed.), New York: Academic Press, 1970.

[8] B. C. J. Moore and B. R. Glasberg, "Derivation of auditory filter shapes from notched-noise data," *Hear. Res.*, vol. 47, pp. 103–138, Aug. 1990.

[9] G. S. E. Terhardt and M. Seewann, "Algorithm for extraction of pitch and pitch salience from complex tonal signals," *J. Acoust. Soc. Am.*, vol. 71, pp. 679–688, Mar. 2000.

[10] J. D. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE J. Selected Areas in Comm.*, vol. 6, pp. 314–323, Feb. 1988.

[11] International Standards Organization, *Coding of Moving Pictures and Associated Audio*, Apr. 1993. ISO/IEC JTC/SC29/WG 11.

[12] International Standards Organization, *Generic Coding of Moving Pictures and Associated Audio Information (Part-7)-Advanced Audio Coding (AAC)*, 1996. ISO/IEC DIS 13818-7.

[13] International Telecommunication Union, *Method for Objective Measurements of Perceived Audio Quality*, July 1999. ITU-R Recommendation BS.1387.

[14] P. Kabal, "An examination and interpretation of ITU-R BS. 1387: Perceptual Evaluation of Audio Quality," tech. rep., McGill University, Montreal, Canada, Dec. 2003, (http://tsp.ECE.McGill.CA).

[15] J. D. Johnston, "Estimation of perceptual entropy using noise masking criteria," in *Proc. IEEE Int. Conf. on Acoustic, Speech, Signal Processing*, (New York, USA), pp. 2524–2527, Apr. 1988.

[16] D. Pan, "A tutorial on MPEG/audio compression," *IEEE Multimedia*, vol. 2, pp. 60–74, Summer 1995.

[17] W. B. Kleijn and e. K. K. Paliwal, *Speech Coding and Synthesis*. Elsevier, 1995.

[18] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer, 1991.

[19] A. D. Subramaniam and B. D. Rao, "PDF optimized parametric vector quantization of speech line spectral frequencies," *IEEE Trans. Speech and Audio Processing*, vol. 11, pp. 130–142, Mar. 2003.

[20] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. 28, pp. 129–137, Mar. 1982.

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

[22] H. S. Malvar, "The LOT: Transform coding without blocking effects," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 37, pp. 553–559, Apr. 1989.

[23] J. P. Princen and A. B. Bradley, "Analysis/synthesis filter bank design based on-time domain aliasing cancellation," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 34, pp. 1153–1161, Oct. 1986.

[24] J. H. Rothweiler, "Polyphase quadrature filters - A new subband coding technique," in *Proc. IEEE Int. Conf. on Acoustic, Speech, Signal Processing*, (Somerville, USA), pp. 1280–1283, Apr. 1983.

[25] J. P. Princen, A. W. Johnson, and A. B. Bradley, "Subband/transform coding using filter bank designs based on time domain aliasing cancellation," in *Proc. IEEE Int. Conf. on Acoustic, Speech, Signal Processing*, (Guildford, UK), pp. 2161–2164, Apr. 1987.

[26] A. Sugiyama, F. Hazu, M. Iwadare, and T. Nishitani, "Adaptive transform coding with adaptive block size (ATC-ABS)," in *Proc. IEEE Int. Conf. on Acoustic, Speech, Signal Processing*, (Albuquerque, USA), pp. 1093–1096, Apr. 1990.

[27] D. Sinha and J. Johnston, "Audio compression at low bit rates using a signal adaptive switched filter bank," in *Proc. IEEE Int. Conf. on Acoustic, Speech, Signal Processing*, (Minneapolis, USA), pp. 1053–1056, May 1996.

[28] D. Sinha and A. Tewfik, "Low bit rate transparent audio compression using adapted wavelets," *IEEE Trans. Signal Processing*, vol. 41, pp. 3463–3479, Dec. 1993.

[29] J. Herre and J. D. Johnston, "Enhancing the performance of perceputal audio coders by using temporal noise shaping TNS," in *Proc. 101st Conv. Aud. Eng. Soc.*, (Los Angeles, USA), Aug. 1996. Preprint 4384.

[30] S. Shlien, "The modulated lapped transform, its time-varying forms, and its applications to audio coding standards," *IEEE Trans. Speech and Audio Processing*, vol. 5, pp. 359–366, July 1997.

[31] M. Bosi, Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, M. Dietz, J. Herre, G. Davidson, and Y. Oikawa, "ISO/IEC MPEG-2 advanced audio coding," *J. Aud. Eng. Soc.*, vol. 45, pp. 789–812, Oct. 1997.

[32] J. Herre, "Temporal noise shaping, quantization and coding methods in perceptual audio coding: a tutorial introduction," in *Aud. Eng. Soc., 17th International Conference on High Quality Audio Coding*, (Florence, Italy), pp. 17–31, Aug. 1999.

[33] K. Brandenburg, "MP3 and AAC explained," in *Aud. Eng. Soc., 17th International Conference on High Quality Audio Coding*, (Florence, Italy), pp. 9–17, Aug. 1999.

[34] British Telecommunications and Royal KPN NV, "Software: Perceptual Evaluation of Speech Quality (PESQ)," November 2000.

[35] X. Huang, A. Acero, and H. wuen Hon, *Spoken Language Processing.* Prentice Hall, 2001.

[36] R. Der, P. Kabal, and W.-Y. Chan, "Bit allocation algorithms for frequency and time spread perceptual coding," in *Proc. IEEE Int. Conf. on Acoustic, Speech, Signal Processing*, (Montreal, Canada), pp. 201–204, May 2004.

[37] H. S. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 38, pp. 969–978, Jun. 1990.

[38] T. D. Tran, J. Liang, and C. Tu, "Lapped transform via time-domain pre- and post-filtering," *IEEE Trans. Signal Processing*, vol. 51, pp. 1557–1571, Jun. 2003.

[39] J. G. Proakis and M. Salehi, *Communication Systems Engineering*. Prentice Hall, 2002.