Wavelet Filter Banks in Perceptual Audio Coding

by

Peter Lee

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2003

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.
I understand that my thesis may be made electronically available to the public.

Abstract

This thesis studies the application of the wavelet filter bank (WFB) in perceptual audio coding by providing brief overviews of perceptual coding, psychoacoustics, wavelet theory, and existing wavelet coding algorithms. Furthermore, it describes the poor frequency localization property of the WFB and explores one filter design method, in particular, for improving channel separation between the wavelet bands. A wavelet audio coder has also been developed by the author to test the new filters. Preliminary tests indicate that the new filters provide some improvement over other wavelet filters when coding audio signals that are stationary-like and contain only a few harmonic components, and similar results for other types of audio signals that contain many spectral and temporal components.

It has been found that the WFB provides a flexible decomposition scheme through the choice of the tree structure and basis filter, but at the cost of poor localization properties. This flexibility can be a benefit in the context of audio coding but the poor localization properties represent a drawback. Determining ways to fully utilize this flexibility, while minimizing the effects of poor time-frequency localization, is an area that is still very much open for research.

Acknowledgements

I would like to first thank Prof. G.H. Freeman for taking me on as his student and allowing me to pursue my graduate studies at the University of Waterloo. I would also like to thank Profs. J. Vanderkooy and K.T. Wong for their helpful review of my thesis.

I'd like to express my appreciation to friends that have stuck with me through my sometimes trying period at the University. I am grateful to my friend Pastor J.M. Park for always reminding me of what was important in life. I am indebted to my dear friend Jisoo Lee without whom I would not have been able to finish my thesis. I am most appreciative to my friend Joonghee Huh for being the best roomate a guy could ask for. And I am thankful to my fellow student Alex Lee for his friendship and encouragement.

Finally, I wish to thank my family for their continued love and support throughout all my studies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Audio coding, an application that falls under the general area of digital waveform coding [1], has seen significant progress in the past two decades with advances made by the inter-workings of coding theory, signal processing, and psychoacoustics. Developing tools for encoding audio signals has been instrumental in providing practical and cost-effective ways to store and transmit audio data in a variety of applications. For example, audio coders are widely used on the *Internet* for transmitting audio files, broadcasting radio, and sharing music. As applications of audio coding continue to grow with growing demand for multimedia, developing better and more efficient audio coding algorithms will continue to be important.

An audio coding algorithm essentially operates on bitrate-intensive audio data and reduces its required data-rate while providing transparent or near-transparent quality. A common audio source is the Compact Disc (CD) audio format, which provides a bit resolution of 16 bits and a sampling rate of 44.1 kHz. This results in a bitrate of 705.6 kilo bits per second (kbps) for a monaural channel and 1.41 Mbps for a stereo channel, both of which are far too large for transmission over common networks. But when compressed by, for example, the MPEG-2 Advanced Audio Coding (AAC) algorithm (which represents the state-of-the-art in audio coding) near-transparent coding of stereo signals can be achieved at a bitrate of 128 kbps [2]. This represents a compression ratio of about 11:1 and a much more practical bandwidth requirement.

1.1 Historical Overview of Audio Coding

Early work on signal compression dates back to the information-theoretic foundation that was laid out by Shannon [3]. Shannon introduced the idea of *entropy* as a quantity expressing the information content of a signal and showed that a source could be coded with zero error if encoding was done at a bitrate equal to or greater than the entropy of the signal (and with coding delay that approached infinity). An implication of this was that sources with infinite alphabets, such as analog audio, required infinite bitrates for error-free coding.

In practice, however, audio signals are first digitized before any meaningful processing is done. This digitization of a signal from analog to digital domain, typically done through the use of an analog-to-digital (A/D) convertor, can actually be thought of as a coding stage that reduces the entropy of a signal to a finite level while introducing some distortion or *coding noise*. The type of coding done at this stage is usually simple and results in a high bitrate so that complexity and coding noise can be minimized, e.g. pulse code modulation (PCM). In order to further reduce the bitrate and still maintain high signal quality, removal of statistical redundancy and perceptual irrelevancy is required [1].

A group of coding algorithms developed early on, commonly referred to as *entropy* or *lossless* coders, were designed to exploit the statistical redundancy of the source signal. Although the entropy provided a measure of the bitrate required to encode a signal, practical coders were only able to approach this theoretical limit. Examples of lossless coding schemes developed for both speech and audio have appeared in [1, 4, 5]. Since most of the early coding work was done in speech, wideband audio coding finds its root in speech coding. A number of differences can, however, be noted. Wideband audio generally has a wider sampling range, wider dynamic range, and higher expectation of quality by the listener. In terms of coding, the most notable difference could be the use of a production model in speech coding that leads to highly efficient ways to encode speech signals [6], whereas nothing similar exists for general audio signals.

The most significant advances in audio coding came with the introduction of perceptual coders. Perceptual coders are designed to take advantage of the masking phenomena that occurs in the ear so that coding noise can be introduced in a way that minimizes or eliminates perceived distortion. It has been noted that many of the innovations in perceptual coding came from people closely familiar with audio applications rather than those involved in research, and this has caused the technology and literature of audio coding to evolve somewhat independently [3]. A number of notable examples of perceptual coders

are mentioned next.

The earliest examples of perceptual coders were developed in the 1970's by Crochiere [7], Schroeder [8], and Zelinski and Noll [9]. These algorithms utilized a time to frequency transformation stage, e.g. Short-time Fourier Transform in [8] and 4-channel non-uniform filter bank [7], that allowed noise shaping in the frequency domain according to some well known psychoacoustic principles. They were followed by the work of several other people in the 1980's who tried to improve on the choice of the transformation stage, accuracy of the psychoacoustic model, and use of other coding techniques that further improved coding efficiency. Most notable of these were the works by Schroeder [10], Brandenburg [11], Johnston [12], and Mahieux [13]. One in particular, an algorithm called MUSICAM developed by Dehery et al. [14], was adopted in the application of digital audio broadcasting (DAB) in Europe and also became part of the well known MPEG-1 audio coding algorithm. Another algorithm called ASPEC [15] also became part of the MPEG-1 audio coding algorithm as the basis to Layer III. The MPEG-1 audio coding algorithm, perhaps the most well known audio coding algorithm, was developed in the early 1990's through a collaborative effort led by the International Standardization Organization (ISO) [16, 17] and was designed to provide three layers of complexity and performance. Layer I provided the lowest complexity and lowest performance, layer II provided medium complexity and medium performance, and layer III provided the highest complexity and highest performance. Layer III, also commonly referred to as "MP3", became popular and widely used on the Internet. Subsequent development of the MPEG audio coding standard appeared as the MPEG-2 and MPEG-4 standard where several improvements were made over the original algorithm in terms of performance, scalability, and functionality [18, 19]. Variants of the MPEG algorithm outside the standard also appeared from other groups, e.g. MPEGplus and MP3 Pro [20]. Other well known audio coders have appeared more commercially, including the AC-3 family of coders developed by Dolby [21], the ATRAC coder developed by Sony [22], and the PAC coder developed by Lucent (formerly AT&T) [23]. More recently, an open-source and patent-free audio codec called Ogg Vorbis [24, 25] appeared as an alternative to the popular but somewhat proprietary MP3 algorithm. Many of these algorithms are used in a variety of applications that include transmission and broadcasting on the Internet, portable audio players and recorders, and multichannel digital sound system in DVD and movie theatres.

1.2 Wavelets in Audio Coding

All perceptual coders share a similar structure in that they contain a filter bank, a psychoacoustic model, and an encoding and quantization stage, as will be described in chapter 2. The filter bank stage provides a decomposition of the input signal that makes the application of perceptual criteria possible and also provides some decorrelation of the input signal. Many types of filter banks and time-to-frequency transforms exist where each offers a different set of trade-offs in its design. Many have been considered and explored in the context of audio coding [2, 26] and one in particular called the *Wavelet Transform* (WT) has shown to be interesting and potentially very useful.

The wavelet transform, or more generally the wavelet filter bank (WFB), is an iterated filter bank that provides a flexible way of analyzing a signal at various resolutions and across various frequency regions. This flexibility is especially appealing in audio coding since the WFB can provide an analysis of the input signal according to the critical band (CB) resolution of the inner ear and, more generally, provide a scheme that can adapt to the time-varying nature of the audio signal [27]. However, the WFB has also been found to provide poor localization properties that can be a drawback in audio coding. The application of wavelets in perceptual audio coding, therefore, requires us to explore ways to maximize its benefits and minimize its drawbacks.

1.3 Thesis Overview

This thesis studies the application of the wavelet filter bank in the context of perceptual audio coding and explores one approach in minimizing the artifacts associated with the poor localization properties of the WFB. An overview of the thesis is given as follows.

- Background and overview of:
  - Perceptual Audio coding (chapter 2)
  - Psychoacoustics (chapter 3)
  - Wavelets and Filter Banks (chapter 4)
- Survey of existing wavelet audio coders (chapter 5)
- Study of the poor frequency localization behaviour of the WFB (chapter 5)
- Exploration of one method based on the modified Remez exchange algorithm for eliminating side-lobes in wavelet subbands (chapter 5)

- Implementation of a wavelet audio coder with some preliminary results (Appendix A and B)

Chapter 2

Overview of Perceptual Audio Coding

Perceptual coders are *lossy* coders that introduce coding error into the encoded signal while trying to minimize its perceived effects. In general, coding error is comprised of pre-filtering, aliasing, and quantization components that are the result of (re)sampling and (re)quantization [3]. Among the three, quantization error is typically the error that we seek to minimize. Traditional coders have relied on distortion metrics such as the mean-squared-error (MSE) in order to minimize coding error according to an objective criteria. Such minimization typically resulted in an optimized signal-to-noise ratio (SNR) that corresponded to a flat noise floor in the frequency domain [28], but not necessarily to an optimized quality in terms of how a listener perceived it. In perceptual coding, however, coding noise is shaped so that the perceived quality of the audio signal is optimized according to some subjective criteria.

The perceptual distortion criteria in a perceptual coder are usually computed by a psychoacoustic stage that tries to model the behaviour of the human auditory system (HAS). The perceptual criteria, commonly referred as the *just noticeable distortion* (JND) or the *masking threshold*, provides a threshold below which coding noise remains imperceptible. The masking threshold is usually represented by a frequency domain curve that covers the range of human hearing, and more generally by a time-frequency contour if temporal masking is also taken into account. A perceptual coder essentially tries to control quantization noise so that its shape remains below the masking threshold in the frequency domain.

This chapter gives a description of a generic perceptual audio coder and covers is-

sues that are pertinent to its design. Section 2.1 describes the overall structure of the perceptual coder and sections 2.2 to 2.4 describe the three main stages of the encoder. Section 2.2 describes the filter bank stage, section 2.3 describes the psychoacoustic model, and section 2.4 describes the quantization and coding stage. Lastly, section 2.5 gives a summary and some concluding remarks.

2.1 Structure of a Generic Perceptual Audio Coder

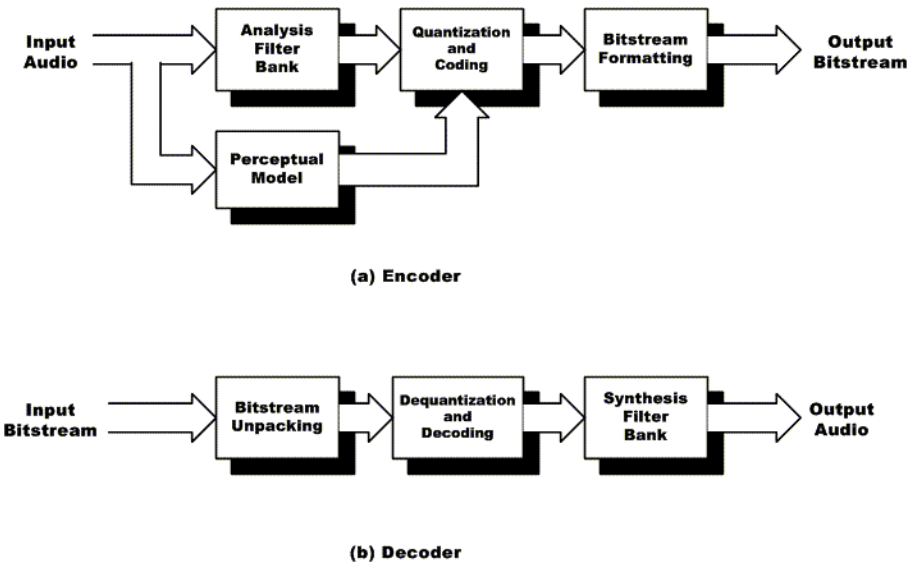


Figure 2.1: A generic perceptual coder (a) encoder (b) decoder

Figure 2.1 shows a diagram of the structure of a generic perceptual audio coder. Figure 2.1(a) shows the structure of the encoder, which has three main stages and a fourth bitstream formatting stage, and Figure 2.1(b) shows the decoder, which has three stages. The encoder operates on the input audio signal and outputs the encoded bitstream, and the decoder operates on the encoded bitstream and reconstructs the original signal. The three stages in the decoder, as a result, are reverse operations of three stages in the encoder. Namely, the signal analysis, quantization and encoding, and bitstream formatting stages of the encoder correspond to the signal synthesis, de-quantization and decoding, and bitstream extraction stages of the decoder, respectively. The extra stage in the encoder is the psychoacoustic model, which is not required in the decoder since the information is implicitly encoded as side-information. This means that perceptual

coders are asymmetrical in that the encoder has a greater computational requirement than the decoder, which actually can be desirable in certain applications where one "server" encodes the signal for many "clients".

The encoder works as follows. The input signal is typically segmented into contiguous *blocks* or *frames* so that each block can be processed individually. This is done for a number of reasons. First, processing a signal in terms of smaller segments reduces the required computational and memory load. Second, segmentation serves as a way to localize the signal in time so that a frequency domain masking can be applied to a time-localized signal. And third, the encoded bitstream can be sent as "packets" that can be transmitted, decoded, and played on a real-time basis. Inside the encoder, the input frame first enters the filter bank and the psychoacoustic stage. The filter bank stage transforms the signal into a frequency domain representation or into a joint time-frequency representation. The psychoacoustic model first applies a high frequency-resolution transform (sometimes the same one as the filter bank, in which case the output from the filter bank is used instead) and then applies rules from psychoacoustics to calculate the frequency domain masking threshold. The output from both the filter bank and psychoacoustic stage then goes to the quantization and encoding stage where the actual bitrate reduction occurs. The coding and quantization stage decides how bits are allocated among the filter bank coefficients and a quantizer is used to (re)quantize the filter bank coefficients. Sometimes, an additional lossless coding step is applied at this stage to further remove statistical redundancy. The quantized coefficients, along with some side information, are finally formatted into the output bitstream. The decoder, on the receiving end, simply performs the reversing operations.

Although most perceptual coders follow this basic structure, some are difficult to describe in terms of this simple and clear-cut model. Nevertheless, all perceptual coders do incorporate the given four stages in some shape or form and this basic framework can be useful in understand other existing approaches.

2.2 The Filter Bank

The choice of the optimal filter bank has, historically, been a subject of much research and discussion in the development of perceptual coders [29]. Inherent to every filter bank is a trade-off between time and frequency resolution. Filter banks that have high frequency resolution, e.g. Discrete Fourier Transform (DFT), have low temporal resolution and filter banks with high temporal resolution, e.g. 2-channel QMF filter bank, have

low frequency resolution. Effective coding depends to a great extent on how the time-frequency resolution of the filter bank is matched to the requirements of the input signal. It has been found that no single resolution satisfies the requirements of all audio signals [27, 18]. That is because audio signals vary greatly in their time-frequency characteristics over time and between signals. Figure 2.2 gives examples of two musical instruments whose masking thresholds are shown in the time-frequency plane. The two instruments, the castanets being an atonal percussive instrument and the piccolo being a pitched wind instrument, are in a sense diametric opposites. Note that the energy of the piccolo is distributed with fine frequency resolution but remains essentially invariant across time, while the castanets is localized in time but spread-out in frequency. Clearly, we require finer frequency resolution for the piccolo and better time resolution for the castanets.

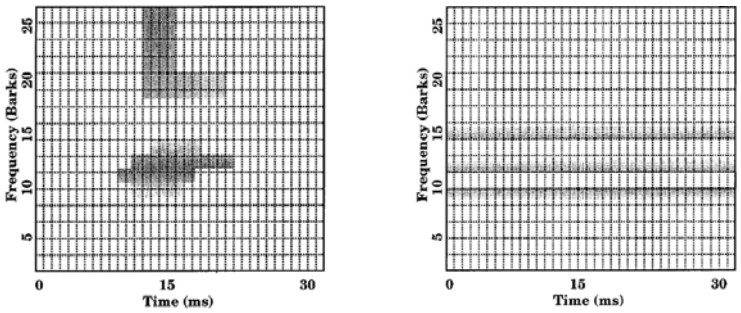


Figure 2.2: Examples of signal masking thresholds in the time-frequency plane (a) castanets (b) piccolo (after [30])

This section covers issues involved in the choice and design of the filter bank. In particular, the importance of providing a resolution similar to the masking resolution and a resolution that decorrelates the input signal is described. Other important and desirable filter bank properties are also described and several examples from the literature are given.

2.2.1 Masking Resolution

The resolution of the human auditory system (HAS) is characterized by the critical band (CB) scale that corresponds to approximately uniform bands at the lowest frequencies and approximately 1/3 octave bands at higher frequencies (see chapter 3). This means that frequency domain masking also occurs according to a resolution similar to the CB

resolution and that the ear experiences masking according to the CB scale. Although critical band measurements have been provided as fixed bands that cover the range of hearing [31], it is important to note that the CB scale is continuous and that the masking resolution also varies continuously (section 3.4.3). Nevertheless, common psychoacoustic models found in coders today only provide masking thresholds according to a fixed CB resolution, where the computed masking threshold level is constant within each masking band. As such, filter banks need to be designed with a similar fixed CB resolution so that coding noise can be controlled properly in the frequency domain. The cost associated with using an inappropriate resolution can be illustrated by the following scenario.

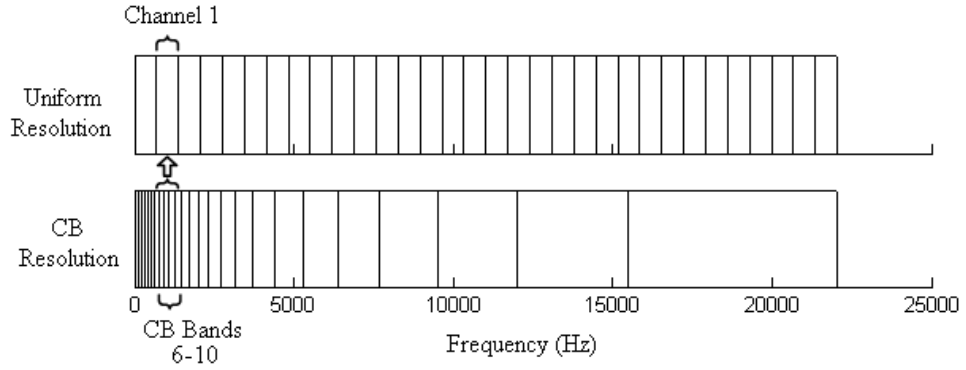


Figure 2.3: Example of a mis-match between subband and masking band resolution

Figure 2.3 shows an example of the frequency resolution provided by the 32-band uniform filter bank and the Psychoacoustic Model I as used in MPEG-1 Layer 1 algorithm. The filter bank provides a frequency resolution of 1378 Hz for each band (at 44.1 kHz sampling frequency) and the psychoacoustic model provides 24 masking bands where the resolution ranges from about 100 Hz at the lowest frequencies to about 4000 Hz at the highest frequencies, a variation of approximately 40:1. The discrepancy that exists between the two resolutions results in over-coding requirements, particularly for the lower bands where the filter bank provides insufficient resolution with respect to masking resolution. For example, channel 1 of the filter bank overlaps with 4 different masking bands (bands 5 to 8, inclusively) in the same frequency region and transparent coding requires that the noise level remains below the masking threshold for the entire channel, i.e. the masking level of the lowest masking band is used as the masking level for the entire channel. Although this satisfies the most stringent masking requirements, this also results in inefficient coding of the remaining bands that do not require such a high bit

resolution. For high frequency channels where multiple channels of the filter bank fall into one masking band, such problems do not exist, but there are redundancies in the subband representation in that threshold values are repeated across multiple channels. Ideally, a filter bank with matching resolution to the masking threshold provides the most efficient resolution for coding in the frequency domain.

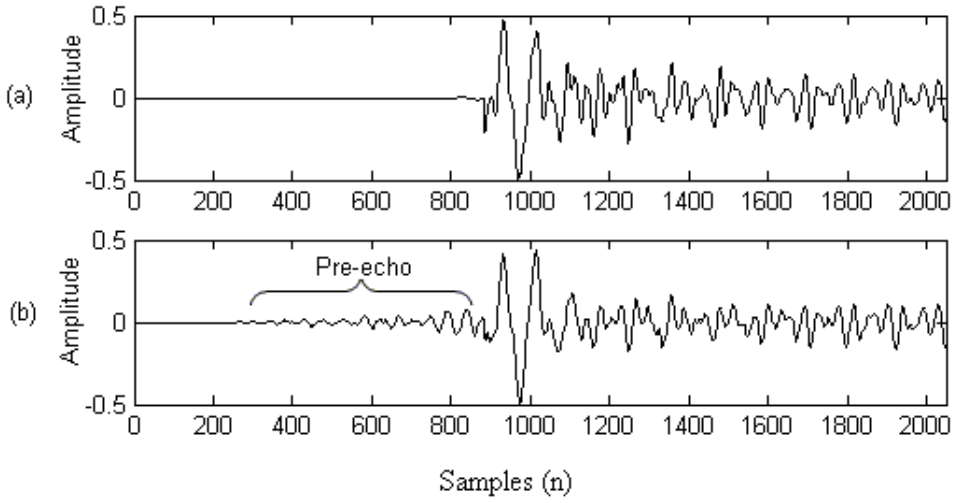


Figure 2.4: Example of a "Pre-echo" scenario (a) original castanets signal (b) signal encoded with MDCT

The duality of this also exists in the time domain. Although some time localization is provided when the input signal is processed in terms of blocks, additional time-varying masking characteristics may exist within a given block. Unfortunately, our understanding of temporal masking is rather limited and few attempts have been made include details of temporal masking. However, it has been found that better time resolution is still sometimes required in order to eliminate this so-called *pre-echo*. Pre-echoes occur when the onset of a transient signal appears towards the middle or latter half of the input block, and coding noise spreads across the frame without regard to where the onset appears in time as shown in Figure 2.4. The listener perceives this as an "echo" that precedes the actual onset of the signal. Note that coding noise spreads beyond the localized region where the signal occurs, where for a high frequency resolution filter bank the noise can spread throughout the entire frame. As a result, the lack of temporal resolution can sometimes force a coder to over-code in order to reduce the amount of pre-echo.

2.2.2 Redundancy Extraction

Audio signals, like many other signals, contain redundancies that can be extracted by the use of an appropriate filter bank. It has been found that redundancy extraction depends on the characteristics of the signal and the type of filter bank used [27]. Audio signals, in general, contain regions that are quasi-stationary in nature with a well-defined harmonic structure and regions that are highly transient and noise-like [32]. For example, a typical musical excerpt contains regions that are highly tonal or stationary, e.g., pitched instruments, and regions that contain lots of transients, e.g., percussive instruments. Audio signals have also been described in terms of the following three categories in [29]:

- Stationary or Pseudo-stationary:
  Stationary signals, such as piccolo or harpsichord, have many frequency components with varying degrees of harmonic structure and typically with envelops that contain a steady-state region. Filter banks with high frequency resolution are required to resolve the spectral characteristics and to provide a good decorrelation of the signal.

- Transient or Noise-like:
  Transient signals, such as percussive instruments, exhibit high non-stationarity that appears as time-dependent events that lack fine structure in the frequency domain. A filter bank with a CB resolution has been found to provide a good way of controlling these temporal details.

- Pitch-periodic:
  Pitch-periodic signals, such as speech or pulse trains, have high frequency contents that are clustered in time around some pitched event. Coding the high frequency details in a time-dependent manner is required if the temporal masking provided by the fundamental pitched-signal (corresponding to the pitch period) does not mask all of the time-domain artifacts. High-resolution filter banks provide inadequate time localization for such signals and a filter bank based on the CB resolution has been suggested as an alternate solution.

A common measure of redundancy in a signal representation is the so-called *spectral flatness measure* (SFM) that is defined as the ratio between the geometric mean (GM) and the arithmetic mean (AM) of the energy distribution of a signal given by [33]

$$\text{SFM} = \frac{(\prod_{k=0}^{N-1} x_k^2)^{\frac{1}{N}}}{\frac{1}{N}\sum_{k=0}^{N-1} x_k^2}, \tag{2.1}$$

where

$$\begin{aligned} N &= \text{number of spectral lines} \\ x_k &= \text{energy of } k^{\text{th}} \text{ spectral component.} \end{aligned}$$

The values of SFM vary between 0 and 1, where 1 represents a flat spectrum with no redundancy, i.e. x is white, and values near 0 indicate high redundancy. If the signal is distributed with equal energy throughout the whole spectrum, then there is nothing gained in using such a representation as each component requires the same number of bits. But if the signal is distributed such that the energy is concentrated into fewer components, then more efficient coding is possible through the redistribution of the bit pool according to the energy spectrum. Since GM $\leq$ AM, and only equal if all spectral components are equal, GM will decrease as more energy gets concentrated into fewer coefficients, and as a result also decrease the value of SFM. Therefore, a filter bank that provides the best energy compaction, i.e. best decorrelation, will provide the smallest SFM and ideally the best *coding gain*. *Coding gain*, consequently, is defined as the inverse of the *SFM*, whose values range between 1 (no coding gain, with respect to PCM) and ∞ (infinite coding gain).

Johnston in [27] used the SFM measure based on an N-point FFT to study the characteristics of various audio signals and found that redundancy varied considerably as a function of both audio signal and frame length. He found that redundancy extraction generally grew with increasing frame size, which meant that signals without substantial time-domain artifacts required long filter banks with high frequency resolution in order to provide efficient coding. For signals with more time-dependent structures, longer filter banks still provided better redundancy extraction but the cost associated with overcoding, e.g. for eliminating pre-echoes, outweighed the gain in redundancy extraction. As a result, shorter filter banks were considered more appropriate for such signals. Furthermore, Johnston found that signal models remained constant for long periods of time, i.e. remained pseudo-stationary, and then changed suddenly, i.e. a transient occurred. This meant that a filter bank with high frequency resolution was required for many parts of the signal, but also a filter bank that could switch adaptively to provide a better time resolution when it was required.

2.2.3 Summary of Design Issues

A summary of design issues for the filter bank is given as follows.

1) Time-Frequency Resolution
   As described above in sections 2.2.1 and 2.2.2, filter banks need to consider the time-frequency resolution of the HAS, the resolution that provides the greatest amount of coding gain, and the resolution that provides adequate control of temporal artifacts. In general, no single fixed filter bank can accomplish all the above requirements and an adaptive scheme is therefore required.

2) Channel Separation
   Channel separation, or frequency localization, refers to how well one channel of the filter bank is separated from the other channels. Inter-band leakages always occur in practical filter banks due to the non-ideal nature of the filtering operation. Channel separation is important in the context of perceptual coding since adjacent channels are assumed to be independent and non-overlapping when perceptual results are applied. This is particularly important for tonal components of the signal, which require high frequency resolution but also good frequency separation. The amount of channel separation, or lack thereof, has been found to have a direct impact on the coding performance in a perceptual coder [33, 18].

3) Boundary Handling and Blocking Artifacts
   The segmentation or windowing of the input signal into smaller blocks gives rise to blocking artifacts at the frame boundaries. This can be perceived as distortions in the reconstructed signal, particularly for portions of the signal that are stationary. To avoid blocking artifacts, coding noise must be made to be somewhat correlated at the frame boundaries. Minimizing boundary distortions is typically done by applying an overlap-add window [34], or by using a lapped transform [33].

4) Perfect Reconstruction
   Perfect reconstruction (PR) requires that the reconstructed signal be identical to the input signal (with a possible delay) in the absence of any coding error. Although the PR condition is not a strict requirement, as there are coders that use a non-PR filter bank, e.g. [18], it generally simplifies the design of a coding system.

5) Maximal Decimation
   Maximally decimated, or critically sampled, filter banks provide the same number of transform-domain coefficients as there are time-domain coefficients in the original signal, i.e. the number of input samples per second is equal to the number of frequency domain samples per second. Since the ultimate goal is to decrease the data

rate while maintaining high audio quality, critically sampled systems are desirable.

6) Computational Complexity
Although it is becoming less of as an issue, computational complexity and coding delay can still be important requirements for certain applications [33]. The analysis and synthesis filter banks should provide efficient algorithms and fast implementations, e.g. FFT or DCT [32].

2.2.4 Filter Bank Examples

Most time-to-frequency transforms and filter banks used in audio coding are now viewed under the framework of multirate signal processing [35, 36, 37]. As such, we have a way of comparing and designing various types of filter banks where each provide a different set of benefits and drawbacks. In general, we can not design a filter bank that provides all the desirable properties, e.g. PR, critical sampling, good time-frequency localization, flexible time-frequency resolution, "transparent" boundary handling, and low complexity, that were described above. A trade-off always exists between certain parameters in any given filter bank.

Examples of filter banks commonly found in the audio coding literature are given next. More in depth studies and descriptions can be found in [35, 2, 38, 33, 39].

1) Fourier Transform Based Filter Banks
Some of the earliest high-quality audio coders were based on the discrete Fourier Transform (DFT) and discrete Cosine Transform (DCT) [38]. First introduced for speech coding, Fourier based transforms provided a relatively simple way of obtaining a frequency domain representation and decorrelating the input signal. To reduce blocking artifacts, a window and overlap-add technique was commonly used. In general, high-frequency transforms provide low computational complexity, but lack the temporal resolution that is sometimes required.

2) Quadrature Mirror Filter (QMF) Based Filter Banks
QMF filter banks, first introduced by Croisier, Estaban, and Galand in 1976 [40], have also been proposed in a number of early speech and audio coders [38]. A QMF filter bank provides a two-way split that could be cascaded together and used to divide the frequency spectrum in a number of different ways. The original QMF filters were near-PR solutions with alias cancellation, e.g. Johnston QMF filters

[40], but PR solutions were later found as well, e.g. *conjugate quadrature filters* (CQF) and *generalized-QMF* [40].

Another more popular variant was the *pseudo-QMF* (PQMF) filter bank, or also called the *polyphase filter bank*, that was designed to provide near-PR and a uniform M-channel decomposition [41]. The PQMF was a cosine-modulated filter bank that required the design of only one prototype filter and which could be implemented efficiently through the use of a polyphase structure. As an example, the MPEG-1 audio coder uses a PQMF filter bank that is based on a 511 tap prototype filter and 32 uniform subbands [38].

3) Modified Discrete Cosine Transform (MDCT)
The MDCT is one of the most popular filter banks that combines a list of features that make it particularly attractive for audio coding. Also known as *modulated lapped transform* (MLT) and *time domain aliasing cancellation* (TDAC) transform [2], the MDCT is a cosine-modulated filter bank that provides PR, high-frequency resolution, high coding gain, elimination of blocking artifacts through its lapped structure, critical sampling, simple design procedure with only one prototype filter, and efficient implementation through an FFT-like algorithm [32]. The MDCT is used in a number of perceptual coders, including the MPEG-1 Layer III, MPEG AAC, AC-3, ATRAC, and PAC [33].

4) Wavelet Filter Bank
The wavelet filter bank is closely related to two-channel QMF filter banks and can be used to provide a flexible division of the frequency spectrum. An in-depth description and examination the WFB is given in chapters 4 and 5.

5) Hybrid Filter Banks
A Hybrid filter bank is created by cascading two or more filter banks together so that a more flexible time-frequency resolution can be obtained. Hybrid filter banks are commonly used for providing non-uniform divisions of the frequency spectrum where different regions require different resolutions. Examples of hybrid filter banks include the MPEG-1 Layer III algorithm that uses a PQMF filter bank followed by an MDCT, and the ATRAC algorithm that uses two-channel QMF filter banks followed by also an MDCT [2]. In addition to the design flexibility, hybrid filter banks can also be used in adaptive schemes, as described next. The cost associated with this increased flexibility is usually higher complexity and longer coding delays.

6) Adaptive Filter Banks
   Adaptive filter banks are time-varying filter banks that make switching decisions based on the characteristics of the input signal. As discussed earlier, a filter bank needs to accommodate to the requirements of the signal. Adaptive schemes can generally be implemented in one of three ways. The first is to use a hybrid filter bank that can adaptively decide how the signal gets analyzed by the subsequent stages of the filter bank, e.g. MPEG-1 Layer III and ATRAC. The second approach is to use a tree-structured decomposition scheme, e.g. wavelet filter bank, that can make switching decisions using different tree structures. A third approach involves using two or more entirely different filter banks and making switching decisions between them, e.g. the EPAC coder makes a switching decision between an MDCT and a wavelet filter bank. Adaptive filter banks generally provide the greatest amount of flexibility as well as complexity, and are usually employed in "high-end" audio coders.

2.2.5 Summary

The design requirements for a filter bank can be summarized as:

- Flexible time-frequency resolution
- Good time-frequency localization
- Minimized blocking artifacts
- Perfect or near-perfect reconstruction
- Maximal or near-maximal decimation
- Low computational complexity and coding delay

A flexible time-frequency resolution is a particularly important property that needs to be driven by the requirements of the input signal and the masking characteristics of the HAS. It has been found that filter banks with approximate CB resolution provided good control of time-dependent artifacts, but lacked the coding gain required for stationary or pseudo-stationary signals. On the other hand, high-frequency resolution filter banks have shown to provide high coding gain for pseudo-stationary signals, but lacked the time-resolution required to control time-domain artifacts and pre-echoes. As a result, adaptive filter banks are generally employed to provide the flexibility required by the time-varying input audio.

2.3 The Psychoacoustic Model

The psychoacoustic model is arguably the most critical and the most difficult component to design in a perceptual coder. The output of the psychoacoustic model directly controls the quantization and coding stage, and indirectly controls the filter bank stage. Moreover, its accuracy directly determines the performance of the overall algorithm. An in-depth overview of the psychoacoustic model is given in chapter 3.

2.4 The Quantization and Coding Stage

The goal of the quantization and coding stage is to essentially achieve a data representation that is as compact as possible while introducing as little perceptual distortion as possible. The quantization and coding stage, in many ways, is the "intelligent" part of the coding algorithm that determines the overall coding strategy.

The quantization and coding stage is usually designed in three substages [38, 42] as described next. First, a *control structure* determines how the bit pool gets distributed among the spectral coefficients using the masking results provided by the psychoacoustic model. Second, a quantization (or re-quantization) step maps the filter bank coefficients to a representation of lower resolution according to the bit distribution determined by the control structure. And third, the quantized data is sometimes encoded with an additional lossless coding step that further tries to remove statistical redundancy. In general, these three stages work together in an interactive way and can be designed to provide a simple scheme like *block companding*, or a more complex scheme like *analysis-by-synthesis* with noiseless coding. The three stages are described in more detail next.

2.4.1 The Control Structure

In order to determine the permissible coding noise level in each band, three terminologies are commonly used according to Figure 2.5. The three measures, namely, SNR, SMR, and NMR, are used to relate the masking threshold to the quantization noise level for a given spectral component [43]. The three measures are described as follows:

- The Signal-to-Noise-Ratio (SNR) is defined as the ratio between the signal energy and the quantization noise energy that results from, for example, an m-bit uniform quantizer. This is an objective measure of the noise level where a higher bit resolution results in a higher local SNR.

- The Signal-to-Mask-Ratio (SMR) is defined as the ratio between the signal energy level and the masking threshold level as computed by the psychoacoustic model. This is the parameter that indicates how much noise is permissible while still maintaining transparency. Note that if the signal level is below the masking level, i.e. SMR < 0 dB, then the signal does not require coding, i.e. zero bit resolution. On a related note, the idea of SMR has been extended to a concept called *Perceptual Entropy* (PE), which like entropy, gives a measure of the information content of a signal, but unlike entropy, quantifies only the perceptually relevant information [12].

- Noise-to-Mask-Ratio (NMR) is defined as the ratio between the quantization noise level and the masking threshold level. The NMR is the measure that indicates the perceptual quality of a signal component for a given bit resolution. In a simple scheme, the NMR can be calculated by NMR = SMR - SNR. For transparent coding, we need NMR ≤ 0 dB.

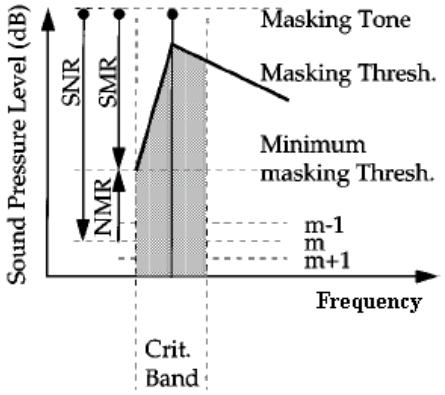


Figure 2.5: Relationship between SNR, SMR, and NMR (after [43])

The hypothetical scenario that is shown in Figure 2.5 represents the masking that occurs as a result of a single tonal masker. The tonal signal is centered at the given critical band and the masking threshold is approximated by the spreading function (section 3.3.2). If we are using an m-bit uniform scalar quantizer, then noise might be introduced at level m. Now, for coding noise to remain imperceptible, it needs to lie below the minimum masking threshold within the entire band, i.e. NMR ≤ 0 dB for entire band. If we apply this simple scenario to the encoding stage, then we need to quantize each coder band or channel so that quantization noise remains below the lowest masking level within the

given band. Determining the actual bit resolution in each coder band typically involves one of two approaches. In the first approach, called *constant quality coding*, a constant level of perceptual quality is maintained while the bitrate requirement for each frame is made to vary. In the second approach, called *constant rate coding*, the bitrate for each frame is maintained at a constant level while the perceptual quality is made to vary. The two approaches are described in more detail next.

2.4.1.1 Constant Quality Coding

Maintaining constant quality is equivalent to shaping the coding distortion so that coding noise remains parallel to the masking threshold across the whole frequency spectrum as shown in Figure 2.6. This is also equivalent to keeping the values of NMR constant across all the bands, where the constant value of NMR determines the level of perceptual quality, e.g. transparency when NMR ≤ 0 dB and decreasing quality for increasing NMR when NMR > 0 dB. In practice, however, noise levels are adjusted to give approximately constant NMR's as the quantizer resolution is only able vary in discrete steps.

The variability in bitrate associated with constant quality coding essentially arises from the variability of the input signal and the masking threshold that results from it. Due to this variability, constant quality coders are less popular and only used when the application allows variable bitrates, e.g. Digital Video Disks (DVDs) and some audio applications on the Internet [42].

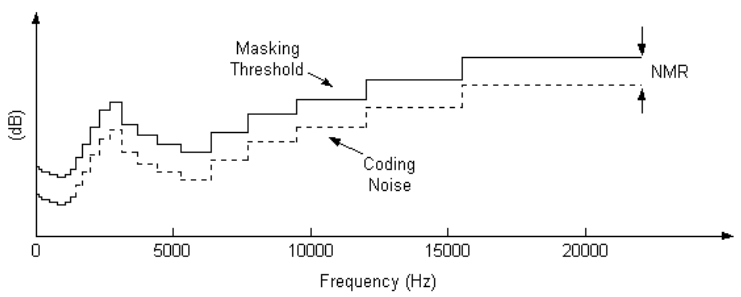


Figure 2.6: Coding at constant quality

2.4.1.2 Constant Rate Coding

In constant rate coding, the overall bitrate is kept constant by keeping the bitrate of each frame constant, even if this results in variable coding quality from frame to frame. In

terms of the distortion curve in Figure 2.6, constant rate coding also tries to shape the coding noise so that it remains parallel to the masking threshold, but the value of NMR is made to vary from frame to frame depending on the Perceptual Entropy of a given frame, e.g. NMR < 0 dB if PE is less than the given bitrate, NMR > 0 dB if PE is greater than the given bitrate, and NMR $= 0$ dB if the two are equal. Most audio coders used toady are based on constant rate coding since the fixed bitrate constraint is required in many applications.

2.4.2 Quantization

The Quantization stage is where the actual data reduction occurs during the coding process. In the context of perceptual coding, providing a flexible control of the quantizer resolution is an important criteria for the quantizer since the coding noise needs to be shaped carefully according to the calculated masking threshold. Usually, transparent coding of a signal component will require a local signal-to-noise-ratio from anywhere between 0 dB (no coding required) to 30 dB, and even higher levels for critical signal components [42]. Two types of quantizers are commonly used as described next.

2.4.2.1 Scalar Quantizer

Scalar quantizers, the simpler of the two, perform quantization on scalar or individual data [44]. They are also the more popular choice in perceptual coding due to their low computational complexity and scalability over a wide range of resolutions. Among scalar quantizers, uniform quantizers are the simplest type. They are designed with uniform step sizes and assume a basic uniform distribution of the input signal. Some perceptual coders that employ a uniform scalar quantizer include MPEG-1 Layer I/II, ATRAC, and AC-3. More sophisticated in structure are the non-uniform scalar quantizers, which are designed with a variable step size that can be adapted to the statistics of the signal or used to exploit additional properties of masking [1]. For example, the non-uniform quantizer used in MPEG-1 Layer III and MPEG-2 AAC audio coder is designed with a power law, e.g. $x^{3/4}$, that distributes coding noise more towards regions of higher magnitude where masking occurs more [42].

Perceptual coders that employ scalar quantizers are usually implemented in one of two ways. In the first, traditionally known as *block companding* or *block floating point*, all coefficients within a coder band are first normalized by a common multiplier, also called the *scalefactor*, so that the range of the coefficients can be matched to the input

of the quantizer, e.g. $|x| \leq 1$. The normalized input coefficients are then quantized with a given bit resolution and the resulting samples and side information, which include the *scalefactor* and the bit resolution, are encoded into the output bitstream. In the second method, all the coefficients within a coder band are first scaled by a "scalefactor" value (different from the scalefactor above) that is determined by the bit resolution assigned to the given band. The scaled coefficients are then quantized directly using a fixed non-uniform quantizer. The scalefactors, in this case, control the quantizer resolution, i.e. SNR, since larger input values, which result from larger scalefactors, become quantized with finer resolution relative to the same input coefficient that are scaled by smaller scalefactors. Note that the meanings of "scalefactor" are different in the two cases, where in the first case it gives a good indication of the amplitudes involved, while in the second case it does not give any such indication as the actual amplitudes also depend on the size of inputs to the quantizer.

2.4.2.2 Vector Quantizer

Vector quantization (VQ), the more general of the two, performs quantization on a vector, or group, of data rather than just on a single value. This provides a framework for performing joint coding where greater amounts of statistical redundancies can be extracted from the input signal. VQ-based perceptual coders, as a result, are generally more complex as they try to identify greater amounts of statistical redundancies as well as providing a way of applying perceptual criteria. This has, actually, been found to be rather difficult since adapting perceptual results to a simple table-based VQ scheme, for example, required fairly large VQ tables and a computationally heavy search process in order to accommodate the high variability in coding resolution. It has been found that VQ-based schemes are most useful in the context of low bitrate audio coding that provides coding qualities in the low to intermediate range, e.g. the TwinVQ algorithm in the MPEG-4 Audio Coding Standard [42].

2.4.3 Noiseless Coding

Noiseless coding is typically an additional step included as part of the coding stage that tries to obtain a further reduction in bitrate. Some well known noiseless coding techniques used in audio coding include the *grouping* method used in MPEG-1 Layer II coder and Huffman coding used in MPEG-1 Layer III and AAC audio coder [42]. Others that have been suggested for audio coding include run-length coding, bit-sliced arithmetic coding

(BSAC), and the LZW algorithm [2].

2.4.4 Discussion

In addition to the various techniques described above, the quantization and coding stage can be designed to incorporate any number of other coding techniques and tools that have appeared in various audio coders [42, 38]. In general, the overall coding strategy of a perceptual coder involves a great deal of flexibility in terms of its complexity and performance, and freedom in terms of its actual implementation. Furthermore, it has been noted in [42] that this is where specific implementation know-hows and "secrets of audio coding" contribute significantly to the overall performance. For example, the MPEG-1 Layer III audio coding standard provides enough guidelines for guaranteeing inter-operability between different implementations, but enough freedom so that different implementations can add their own improvements. This is the reason why different implementations of the MP3 algorithm, e.g. standard ISO distribution, LAME, and Fraunhofer, provide different coding performances.

2.5 Summary and Discussion

The overview in this chapter shows that there is a great deal of choice and flexibility in designing and implementing a perceptual audio coder, both in terms of the overall coding strategy and in terms of the individual stages. The type of filter bank, the accuracy of the psychoacoustic model, the coding strategy used in the quantization and coding stage, and other coding techniques that are additionally employed all play significant roles in determining the overall performance and complexity of the algorithm.

We can generally describe the design goals of audio coding in terms of the following three requirements:

- High efficiency (or low bitrate).
- High signal quality (or low distortion).
- Low complexity in terms of computation, memory, and delay.

In general, not all three goals can be satisfied simultaneously as there always exist trade-offs among the three. The choice of trade-offs in a particular audio coder needs to be dictated by the given application. For example, the application of digital surround

sound system in movie theatres and DVD requires the highest quality, while efficiency and complexity are less of an issue. The application of radio broadcast on the Internet, on the other hand, requires low bitrates and low delay (at least for now), in which case signal quality suffers. Many audio coders available today, e.g. MPEG-1 Layers I, II, III, and MPEG-2 AAC, are already designed to provide flexible trade-offs amng these requirements by offering different algorithms that provide different trade-offs.

The field of perceptual audio coding is still considered to be a young and active field where much of what we know is somewhere between art and science, and where fundamental trade-offs between efficiency, quality, and complexity can still be improved [38]. Some possible future work that has been suggested include the following.

- A number of relatively new filter bank techniques that are currently being explored include wavelet based filter banks and low delay filter banks [45, 2]. As noted in [30], the main requirement for filter banks in delivering high performance is flexibility in terms of its time-frequency resolution and efficiency in terms of how well they can adapt to the requirements of the input signal.

- Limitations of the current psychoacoustic models are well known and well documented. There is still much work to be done in developing more accurate and generalized models that can provide masking information using both spectral and temporal masking, as well as one that is not based on a fixed discrete CB scale.

Chapter 3

Overview of Psychoacoustics

The study of psychoacoustics and the human auditory system (HAS) are of significant importance in the design of a perceptual coder. This is due to the fact that the final quality of the audio is judged at the point where a listener hears the sound and perception is made. Therefore, understanding how sound is processed in the ear and eventually how it is perceived by a listener is required in order to assess which parts of the audio can be discarded. Clearly, the aim in the design of a perceptual model would be to imitate the biological and psychological processes that occur in the HAS and predict how coding distortion can be introduced without introducing perceptual degradation.

Most existing models of human perception are based on the experimental work of Zwicker and Feldtkeller [46] that started in the 1950's, as well as earlier works by Fletcher [47] and Helmholtz [48]. The field of psychoacoustics has since made significant progress in characterizing the process of hearing and how the ear analyzes audio signals in the time-frequency domain [49]. Perceptual audio coders achieve high compression rates by identifying perceptually irrelevant information in the signal by applying psychoacoustic principles such as the absolute threshold of hearing, the critical band resolution, and the spread of masking.

A description of the HAS as summarized from [48, 2, 50, 49, 51] will be given in this chapter as well as some important results from the research that has led to current psychoacoustic models. Section 3.1 covers some basic definitions, section 3.2 looks at the physiological structure of the human ear, section 3.3 covers some important results from psychophysical experiments, section 3.4 discusses the design of a psychoacoustic model, and finally section 3.5 ends with a summary.

3.1 Some Definitions

3.1.1 Sound Pressure Level (SPL)

The SPL is a measure of sound *intensity* defined as

$$L_{\text{SPL}} = 10\log_{10}(I/I_0) \qquad \text{(dB)}, \tag{3.1}$$

where

$$\begin{aligned} I &= \text{Intensity of sound } [W/m^2] \\ I_0 &= \text{Standard reference level intensity} = 10^{-12} W/m^2. \end{aligned}$$

The SPL can also be defined in terms of pressure levels as

$$L_{SPL} = 20\log_{10}(p/p_0) \qquad \text{(dB)}, \tag{3.2}$$

where

$$\begin{aligned} p &= \text{pressure of sound } [N/m^2 or Pa] \\ p_0 &= \text{Standard reference level} = 2 \times 10^{-5} N/m^2. \end{aligned}$$

The dynamic range of the human auditory system spans from the limit of audibility at around 0 dB, to the threshold of pain at around 120 dB. The SPL reference level is calibrated so that the frequency dependent absolute threshold (described next) measures in the vicinity of 0 dB SPL. In addition to sound intensity, other more subjective measures of loudness have also been suggested such as the *phone* and the *sone* [50].

3.1.2 Absolute Threshold and Masking Threshold

The *absolute threshold of hearing* represents the minimum intensity required for a person to detect a pure tone in the absence of any other sound. The *absolute threshold* can be described by a frequency domain curve that is given by

$$T_q(f) = 3.64(f/1000)^{-0.8} - 6.5\exp^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4 \text{ (dB SPL)} \tag{3.3}$$

and also shown in Figure 3.1 [2].

Masking is a process that occurs in the ear where one sound is rendered inaudible

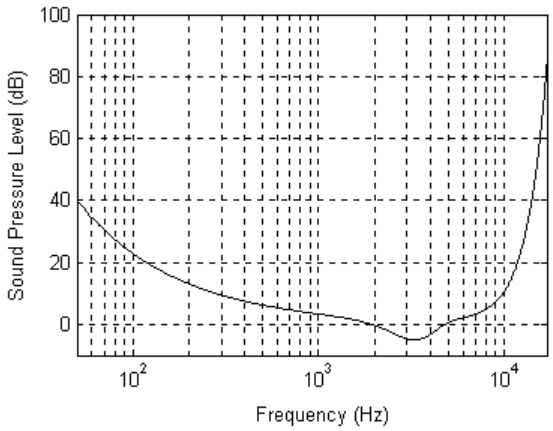


Figure 3.1: Absolute threshold of hearing

or somewhat less audible by the presence of another sound. The *masking threshold* or the *just noticeable difference* (JDN) is the threshold of detection of a given signal, i.e. the *probe* or *maskee*), in the presence of another signal, i.e. the *masker*. The masking threshold can be thought of as a modification of the absolute threshold where the presence of the masker raises the threshold of detection (see Fig. 3.2). Equivalently, the absolute threshold can be thought of as a special case of the *masking threshold* where there is no *masker* present. Naturally, this leads to the idea that masking thresholds are signal dependent and that their shapes are determined by the presence (or absence) of various spectral components. In fact, this is the curve that we wish to compute so that coding noise can be shaped accordingly.

3.2 The Human Auditory System

A closer look at the human ear reveals that there are three separate stages, namely, the *outer ear*, the *middle ear*, and the *inner ear* (see Figure 3.2). The outer ear is the visible part through which sound enters the HAS, the middle ear is the part that lies between the outer and the inner ear, and the inner ear is where the vibrational patterns of sound get converted into neural signals. All three stages of the ear play an important role of in the process of acoustic transduction and deficiency in any one can alter, debilitate, or even cause loss of hearing.

The transmission of sound in the ear can be described as follows. Sound waves trav-

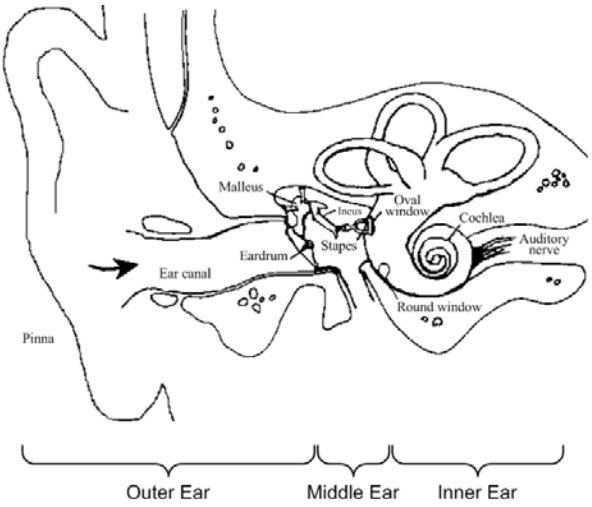


Figure 3.2: The human ear: outer, middle, and inner ear (after [48])

eling in the air enter the ear through the ear canal, some of which are "captured" by the *pinna*, and reach the middle ear at the *eardrum*. The eardrum then vibrates mechanically and sends the mechanical vibrations through the middle ear, which is comprised of three tiny bones called the *ossicles* that lie inside a small air-filled chamber. The middle ear essentially acts as an (acoustic) impedance matching device that transmits the sound from the outer ear to the inner ear. The vibration patterns reach the inner ear at the *oval window*, which is the opening to the *cochlea*. The cochlea is a spiral-shaped structure that is filled with fluid and which represents the main structure of the inner ear. The cochlea is also the most important part of the ear from the point of view of auditory perception as that is where masking occurs. As the oval window gets excited by the incoming vibrations, the cochlear structure induces traveling waves along the length of the *basilar membrane* (BM), which stretches from the oval window to the tip of the cochlear structure. Finally, the pattern of vibrations along the BM causes various auditory nerve fibres found along its length to be triggered and to generate neural signals that are subsequently sent to higher centres of the auditory system.

In general, models of human hearing are divided into three stages, namely, the *analysis*, the *transduction*, and the *reduction* stage [52]. The *analysis* stage covers outer ear, the middle ear, and part of the inner ear, from the point where sound enters the ear up to the point where it reaches the cochlea and sets the BM into motion. The *transduction* stage models the transformation of the mechanical vibrations along the BM

into frequency-dependent electrical activity inside the nerve fibres. Finally, the *reduction* stage models the subsequent processing that occurs inside the cochlea as well as in the higher centres of the auditory system. The transduction stage, in particular, is described further next.

3.2.1 The Inner Ear

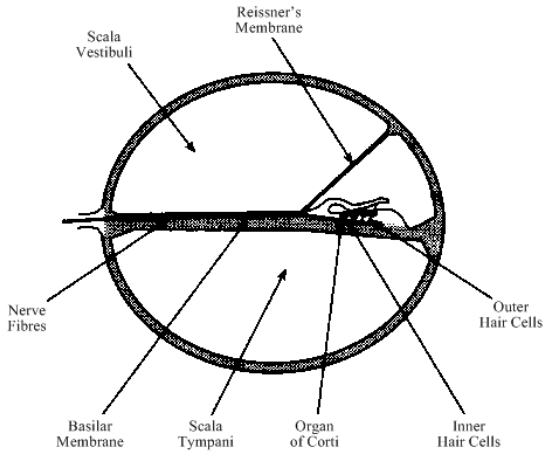
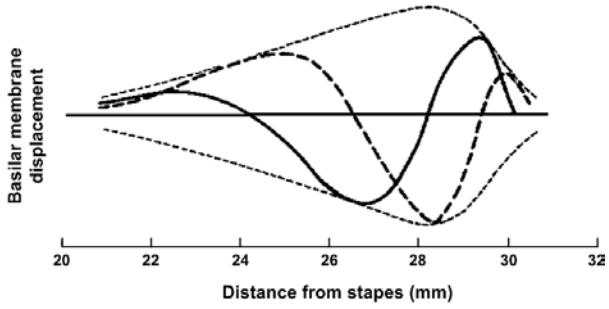


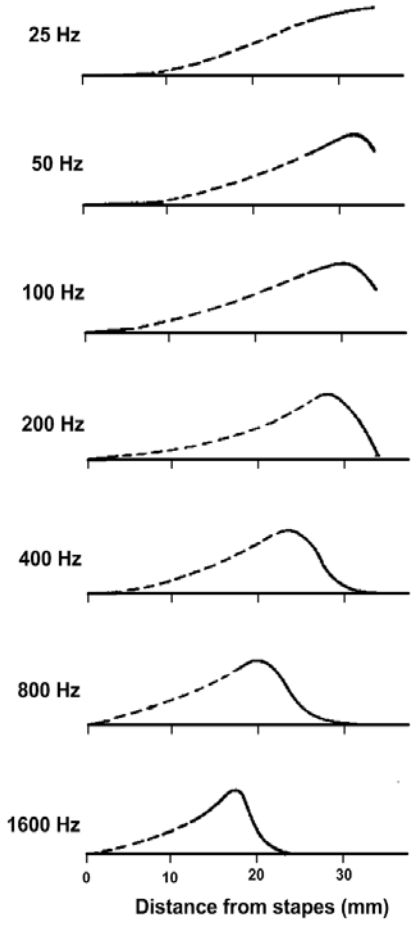
Figure 3.3: Cross section of the inner cochlea (after [51])

The cochlea is a rigid bony structure that is shaped like a snail shell and filled with incompressible fluid. The entrance to the cochlea at the oval window is termed the *base*, while the farthest point in the spiral structure is termed the *apex*. It is divided along its length by two membranes called the *Reissner's membrane* and the *basilar member* (BM) (see Figure lFigInnEar). The two larger chambers separated by the membranes are called the *scala vestibuli* and the *scala tympani*. The oval window at the base is connected to the scala vestibuli and the other window that exits back into the cochlea, called the *round window*, is connected to the scala tympani. The two chambers are connected at the apex of the cochlea, which allows the sound to travel through the scala vestibuli and return through the scala tympani back to the round window. Since the sound gets reflected at the apex, it returns back 180 degrees out of phase only to be released back into the middle ear. Inward movement of the oval window, therefore, results in a corresponding (and rather convenient) outward movement of the round window.

Inside the cochlea, a frequency-to-place transformation occurs along the BM where



(a)



(b)

Figure 3.4: Traveling waves on the basilar membrane (a) The instantaneous displacement of the basilar membrane at two successive instances in time. The pattern moves left to right, building up gradually up until the characteristic frequency and then decaying rapidly. The dotted line represents the envelop. (b) Envelops for a number of low-frequency sinusoids. (after [48])

various regions of the BM are excited by the different frequency components of the input signal. As sound waves enter the cochlea, a pressure difference is applied across the BM and a pattern of motion develops along the BM in the form of traveling waves. These traveling waves move from the base towards the apex of the cochlea with amplitudes that increase gradually up to a certain point and then decrease abruptly. An example of a simple sinusoidal stimulation is given in Figure 3.4. The location where a traveling wave reaches its peak amplitude is strongly correlated to the physical properties of the BM, where high frequency signals peak near the base where the BM is narrow and stiff and low frequency signals peak near the apex where the BM is wider and suppler (Figure 3.5). The frequency of a signal that reaches peak amplitude at a given point on the BM is known as the characteristic frequency (CF). The BM, as a result, can be thought of as a crude Fourier analyzer or a parallel bank of bandpass filters that breaks a complex signal into its frequency components.

Figure 3.5: Physical property of basilar membrane

3.2.2 Neural Responses in the Auditory Nerve

The next step in the transduction process involves the conversion of vibrational patterns on the BM into neural patterns in the auditory fibres. Specifically, information about frequency, magnitude, and time is converted into multiple parallel streams of data by some 2500 *inner hair cells* and 30 000 neurons that are connected along the length of the BM. These hair cells are found within a small structure called the *organ of Corti*, which makes contact with the BM and vibrates as the BM vibrates. The hair cells can be divided into two groups, namely, the *inner hair cells* and the *outer hair cells*. A great deal of the transduction process is still poorly understood, but the general understanding seems to be that the inner and the outer hair cells have different roles. The majority of afferent neurons, which carry information from the cochlea to higher levels of the auditory system, have been found to be connected to the inner hair cells, and most efferent nerve fibres, which carry information from higher centres of the auditory system to the cochlea, are believed to be connected to the outer hair cells. As a result, most of neural information

generated by an input signal is carried by the inner hair cells, while the outer hair cells are believed to be involved in some other processes that actively influence hearing, e.g. in providing sharper tuning and better frequency selectivity. Unfortunately, very little is understood about processing of the neural signals beyond the cochlea at higher centres of the auditory system [48].

3.3 Summary of Relevant Psychophysical Results

Much of our knowledge about the auditory system and the facts that allow us to develop models for audio coding come from psychophysical experiments, rather than our physiological understanding of the HAS [53]. Psychophysical measurements are generally obtained through extensive listening tests that are conducted with people who are representative of the general population. A number of important and well-known results are described next.

3.3.1 Critical Bands and Auditory Filters

As already mentioned, the ear performs a kind-of spectral analysis on the incoming signal where the signal is broken down into the various frequency components. As a result, the frequency selectivity of the peripheral auditory system has been modeled as a bank of bandpass filters where the filters are referred to as *auditory filters*. Auditory filters are found all along the BM and possess a frequency resolution that is characterized by the *critical band* (CB) scale. The CB scale is approximately 100 Hz for frequencies below 500 Hz and 20% of the center frequency for frequencies above 500 Hz. The physiological basis to the CB resolution of the auditory filters is not entirely certain, but it seems that it is largely the result of a frequency-to-place transformation that occurs along the BM. Results from psychophysical measurements indicate that the CB corresponds to a constant distance along the BM, where one critical band equals to 0.9 mm according to Moore, 1.5 mm according to Schroeder, and 1.3 mm according to Zwicker and Scharf [32]. Three well-known masking experiments used to characterize auditory filters and the CB resolution are described next.

3.3.1.1 Fletcher's Band-Widening Experiment

In Fletcher's band-widening experiment [47], the detection threshold of a sinusoidal signal (the probe) was measured in the presence of a bandpass noise (the masker) as shown

in Figure 3.6(a). This was done by centering the bandpass noise with the probe and adjusting the level of the probe until it became just noticeable, i.e. the JND level was found. This was repeated while the masker bandwidth was varied, where an increase in bandwidth resulted in an increase in the total energy of the masker. The results, when plotted, gave a measure of the JND level as a function of masker bandwidth as shown in Figure 3.6(b) for a 2000 Hz probe signal. It would be logical to assume that larger bandwidths implied higher masking power, and therefore higher JND levels. This was shown to be the case, but only up to a certain point, after which the JND level remained constant regardless of masker bandwidth. To explain this, Fletcher suggested that the peripheral auditory system acted like a bank of bandpass filters where, in detecting a signal, the listener made us of the (auditory) filter that was the closest and, furthermore, that only noise passing through this filter contributed to masking. The leveling off of the JND level in Figure 3.6(b) could therefore be explained by the fact that an increase in noise bandwidth, although it increased the total power, did not increase the noise power within the given filter band. This "critical" bandwidth value, at which point an increase in noise bandwidth no longer made a difference, was measured for various frequencies and the resulting curve provided the CB scale.

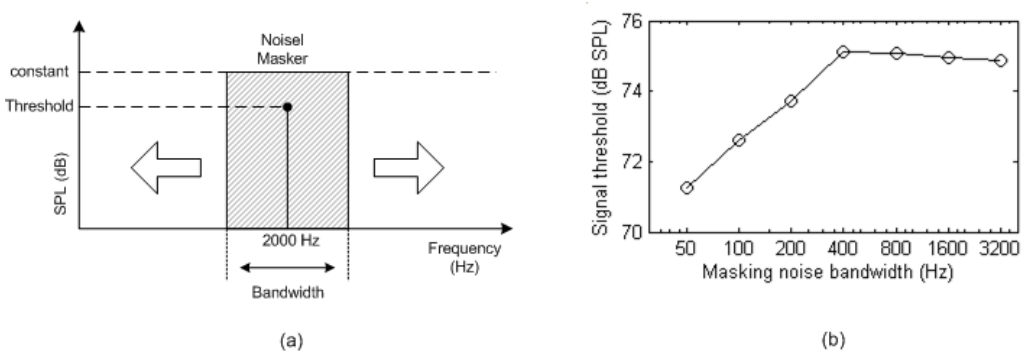


Figure 3.6: Fletcher's band-widening experiment (a) masker and noise (b) result for a 2000 Hz probe

Fletcher's experiment also led to the well-known model of masking known as the *power spectrum model* (PSM). The PSM is a set of assumptions about the process of masking that can be summarized as follows [54]:

1) The peripheral auditory system contains an array of linear and overlapping bandpass filters.

2) When detecting a signal in a noise background, the listener is assumed to make use of just one filter with a center frequency close to that of the signal.

3) Only the noise components that pass through this specific filter have any effect in masking the signal.

4) The masking threshold of the signal is determined by the amount of noise passing through the auditory filter. Specifically, threshold is assumed to correspond to a certain constant signal-to-noise ratio, K, at the output of the filter, i.e. the ratio of the JND level of the signal to the noise passing through the filter is assumed to be constant. Furthermore, the stimuli are assumed to be represented by their long-term power spectra, i.e. the relative phases of the signal components and the short-term fluctuations in the masker are ignored.

In reality, none of the assumptions in the power spectrum model are strictly correct. For example, the auditory filters are not in reality linear, the ear does not always use just one filter in detecting a signal, the noise that falls outside the passband can sometimes contribute to the detection of the signal, and time-dependent fluctuations in the masker can not always be ignored [54]. Nevertheless, the concept of the auditory filter bank is useful and the power spectrum model allows a convenient way of calculating the masking threshold in the frequency domain, e.g. using the power spectral density function in the Fourier domain.

3.3.1.2 Zwicker's Notched-Noise Experiment

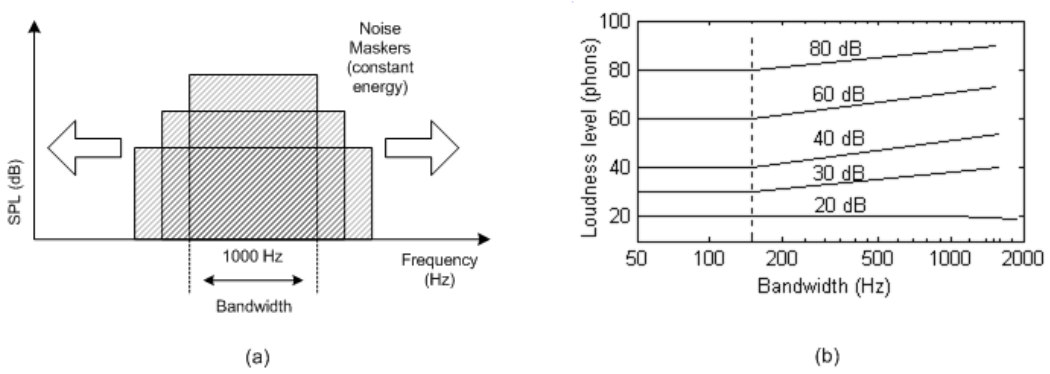


Figure 3.7: Zwicker's notched-noise experiment (a) masker and maskee (b) results

Zwicker in [46] also made measurements of the critical band scale using five different experiments, all of which gave similar results, and one of which is briefly described here. An experiment was set up with a notched noise and a tonal signal placed right in between two noises as shown in Figure 3.7(a). The experiment involved finding the detection threshold of the probe while moving the two notched-noises further apart, and in the process determining the distance between the two notched-noises that caused the threshold level to change. The threshold level was found to remain the same as long as the separation remained within the distance of a critical bandwidth. But when separated beyond that, the threshold level started to decrease rapidly as shown in Figure 3.7(b). Using the PSM, the fourth statement says that a fixed amount of masking energy will always result in a fixed masking level. This is what happens as long as the notched-noises remain within a critical bandwidth. But when the energy of the masker (that falls under the filter) decreases, then masking level also decreases according to the ratio K. This is what happened in the experiment when the notched-noise became separated by more than a critical bandwidth. Using this procedure, Zwicker made measurements of the critical bandwidth across a wide range of frequencies and came up with an expression for the CB scale that is given by

$$\mathrm{BW}_c(f) = 25 + 75[1 + 1.4(f/1000)^2]^{0.69} \qquad \text{Hz.} \tag{3.4}$$

3.3.1.3 Moor and Glasberg's Measurements on the Shape of Auditory Filters

Another experiment by Moore and Glasberg in [55] made attempts to measure the actual shape of the auditory filters. Building on Fletcher's power spectrum model, an experiment was set-up with a tonal probe and a flat notched-noise masker shaped symmetrically around it, as shown in Figure 3.8. According to the power spectrum model, we have the relationship

$$P_s = K \int_0^\infty NW(f)df \tag{3.5}$$

where

$$\begin{aligned} P_s &= \text{power of tonal signal of frequency } f_0 \text{ at the JND level} \\ W(f) &= \text{shape of auditory filter centered at } f_0 \\ N &= \text{energy level of noise masker} \\ K &= \text{constant SNR between masker and maskee according to PSM.} \end{aligned}$$

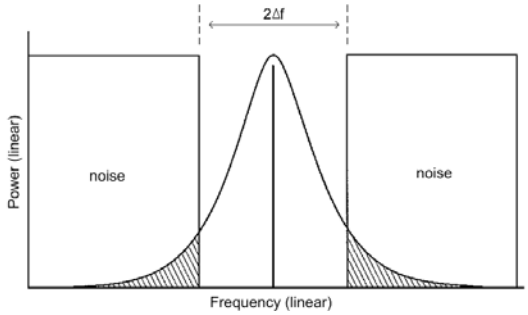


Figure 3.8: Moore's notched-noise experiment (after [48]). The amount of noise passing through the auditory filter centered at the signal frequency is proportional to the shaded area.

Now, since the value of K is a constant and the notched-noise bands were assumed to be symmetrical placed around the maskee, we can manipulate the notched-noise separation, δf, and measure the corresponding change in P_s in order to derive the shape of $W(f)$ from equation 3.5 (detail of this procedure can be found in [54]). From the estimated filter shapes, the critical bandwidth was measured for several frequencies and the results were fitted with a function given by

$$\mathrm{ERB}(f) = 24.7(4.37(f/1000) + 1), \tag{3.6}$$

where ERB represents the *equivalent rectangular bandwidth*. Figure 3.9 shows a plot of both $\mathrm{BW}_c(f)$ of equation 3.4 and $\mathrm{ERB}(f)$ of equation 3.6. Note the slight difference that exists between the two plots where $\mathrm{BW}_c(f)$ remains essentially flat below 500 Hz, but $\mathrm{ERB}(f)$ continues to decrease below 500 Hz.

3.3.1.4 Other Experiments

Other experiments such as two-tone masking, sensitivity to phase, musical consonance, and harmonic discrimination have also demonstrated the existence of this critical band scale [48, 31].

3.3.2 Masking Patterns and Excitation Patterns

The masking described so far represents masking that a probe signal experiences at a fixed frequency, while the frequency of the masker is varied. If we fix, instead, the frequency of the masker and vary the frequency of the probe, then we obtain what are known as

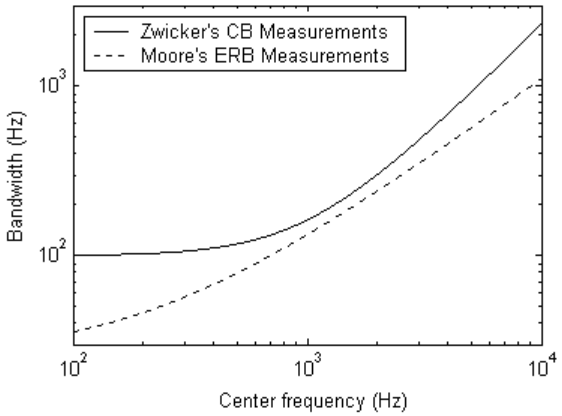


Figure 3.9: CB values from CB and ERB equations

masking patterns or *masked audiograms* [48]. In fact, this is what many of the earlier experiments did in order to characterize the spread of masking. Figure 3.10 shows examples of masking patterns of a narrow-band noise masker and a tonal probe. The masking patterns represent three levels of intensities of the masker centered at 410 Hz. Other curves such as these have been measured for a variety of maskers at different frequencies and with different tonal properties (further described below). Note that masking patterns differ from auditory filters in that a masking pattern describes how a signal as a masker raises the level of the masking threshold, while auditory filters represent the resolution with which the ear experiences masking.

It has been found that masking patterns exhibit various non-linear and signal-dependent properties that reflect the active and complex processes of the ear, e.g. masking patterns are non-symmetric in frequency, non-linear with respect to masker intensity and center frequency, dependent on the tonal qualities of the signal, and non-additive [46]. As a result, masking that results from a typical input signal that contains both tonal and noise-like components at various frequencies and at various intensities is not simply equal to the sum of the individual masking components of the signal. Determining an accurate representation of masking that result from even the simplest of audio signals is therefore a complex and difficult task. In a simple masking model, however, several assumptions need to be made so that the calculation can be simplified. One of those assumptions is to approximate the shape of the masking pattern with a single prototype function called

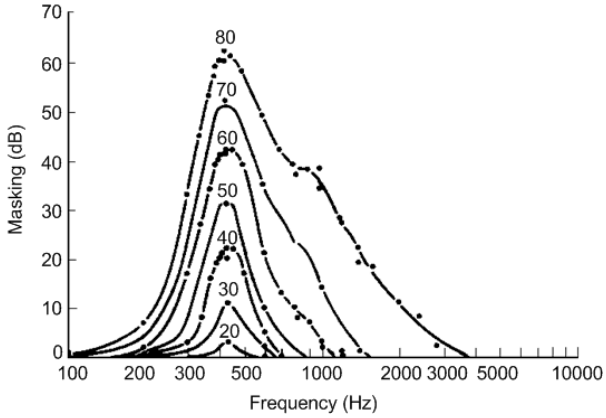


Figure 3.10: masking pattern of a narrow band noise masker centered at 410 Hz

the *spreading function* given by [8]

$$\mathrm{SF}_{\mathrm{dB}}(x) = 15.81 + 7.5(x + 0.474) - 17.5\sqrt{1 + (x + 0.474)^2} \quad \mathrm{dB} \tag{3.7}$$

where x = distance along the Bark scale (the "Bark" unit corresponds to the width of one critical band). Note that the spreading function does not take into account the tonality, non-linearity, or the center frequency of the masker.

It has been further suggested that the masking patterns derived from the masking experiments described above reflect a more basic activity that underlies the process of hearing, which we refer to as *excitation patterns*. An Excitation pattern is essentially a curve that represents the pattern of neural activity beneath the basilar membrane that is evoked by an input signal. It is used as a crude indicator of what the ear senses as opposed to what the ear receives. It has been suggested in [54, 56] that the masking pattern and the excitation pattern are approximately parallel in their shapes and separated by a small distance. Furthermore, the excitation pattern has also been found to be useful in predicting masking in more complex signals when multiple masker and maskee components are present, as will be described in 3.4.2. In addition, a connection between auditory filters and excitation patterns has been made in [54] where a procedure for deriving the excitation pattern from the auditory filters is given.

3.3.3 Asymmetry of Masking

As alluded to above, masking depends on the tonal or noise-like characteristics of the masker as well as the maskee. The differences in masking abilities of tonal signals, e.g. pure sinusoidal, versus noise-like signals, e.g. bandpass noise, has been referred to as the *asymmetry of masking*. Several researchers have studied the asymmetric nature of masking and have provided a number of useful results, some of which are briefly described here [8, 57, 50]. As we are dealing with two types of signals in two types of roles, it is convenient to distinguish four types of masking scenarios, namely, *noise-masking-tone*, *tone-masking-tone*, *tone-masking-noise*, and *noise-masking-noise*.

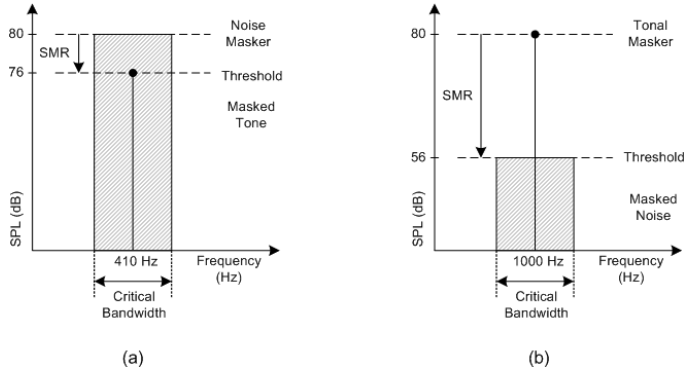


Figure 3.11: Asymmetry of masking (a) noise-masking-tone (b) tone-masking-noise

1) Noise-Masking-Tone (NMT): A typical noise-masking-tone experiment is shown in Figure 3.11(a) where a narrow-band noise centered at 410 Hz represents the masker and a single tone at 410 Hz represents the maskee. The JND level of the probe in general has been found to be directly related to the level of the masker and the shape to be similar to the spreading function of equation 3.7. For the masker and maskee of Figure 3.11(a), the minimum SMR, i.e. the point at which the greatest amount of masking occurs, was found to be 4 dB for an 80 dB masker, and 3 dB for a 60 dB masker [50].

2) Tone-Masking-Tone (TMT): For the scenario where both the masker and maskee are tonal signals, it has been found that masking is greatest for probe frequencies slightly above and slightly below the masker frequency. For a 400 Hz masker signal, the minimum SMR was found to be 19 dB for an 80 dB masker, 15 dB for a 60 dB masker, and 14 dB for a 40 dB masker [50]. When masker and probe frequencies were close together, masking was interrupted by a phenomenon called *beating* where the masker and maskee signal interacted so as to create this fluctuating sense of loudness. When the two signals

were far apart, other non-linear effects came into play that again made the experiment difficult to carry out.

3) Tone-Masking-Noise (TMN): In a tone-masking-noise experiment, the tonal signal is now the masker and the noise signal is now the maskee. Figure 3.11(b) shows a masker tone centered at 1000 Hz and a narrow-band noise also centered at 1000 Hz. Again, the JND curve has been found to be dependent on the level and frequency of the masker signal and can generally be approximated by the spreading function of equation 3.7. But compared to the NMT scenario, masking levels have been found to be a great deal lower due to the inferior masking abilities of tonal signals. The minimum SMR for the signals in Figure 3.11(b) was found to be 21 dB for a 60 dB masker, 24 dB for an 80 dB masker, and 28 dB for a 90 dB masker [50, 57].

4) Noise-Masking-Noise (NMN): Masking of a noise signal by another noise signal is difficult to measure or characterize due to the complex phase relationships that develop between the masker and the maskee. One study done on the intensity discrimination of wide-band noises found minimum SMR values to lie around 26 dB [50].

3.3.4 Temporal Masking

The masking concepts discussed so far were about masking in the frequency domain, also referred to as *simultaneous masking*. In contrast, *temporal masking* or *non-simultaneous masking* is masking that occurs in the time domain before the onset of the masker and following the removal of the masker as shown in Figure 3.12. The former is called *pre-masking* or *backward masking*, and the latter is called *post-masking* or *forward masking*. What this essentially means is that the ear starts to experience masking slightly before the masker signal appears and continues to experience masking slightly after the masker signal disappears.

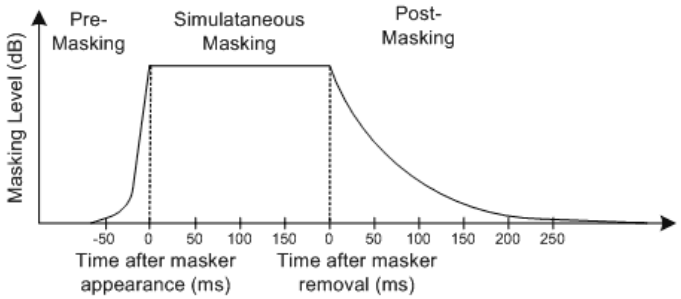


Figure 3.12: Non-simultaneous masking

Although several studies on backward masking exist, no adequate theory has yet been suggested as to what causes it or how it can be predicted, and furthermore, there seems to be a general lack of consensus among various results [54, 2]. For example, the amount of measurable backward masking has been found to be anywhere between 2 and 20 ms [58, 38], and many even suggest that backward masking depends significantly on the training of the individual listener [2].

Forward masking, on the other hand, has been found to be dependent in a predictable way on the frequencies of the signals involved, intensity of the masker, and duration of the masker [2]. In general, post-masking can last anywhere between 50 and 300 ms while exhibiting frequency-dependent behaviours similar to simultaneous masking.

3.4 Design of a Psychoacoustic Model

As stated earlier, the goal of a psychoacoustic model for an audio coder is to determine the masking threshold that is generated by a time-localized input signal where the maskee can be assumed to be (white) quantization noise. Ideally, this would require finding the masking curve of the input signal that takes both spectral and temporal masking into account so that the resulting curve can be represented in the time-frequency plane. Furthermore, the psychoacoustic model would need to handle all types of signals that contain any combinations of tonal and noise components at various frequencies and at various intensities. In reality, however, every psychoacoustic model makes some assumptions and simplifications that make the design practical, while introducing some inaccuracies. As a result, there is always an inherent trade-off between accuracy and complexity that exists in any given model. Most psychoacoustic models can be broadly divided into three categories, namely, physiological models, excitation pattern models, and masking pattern models. Descriptions and examples of these are given next.

3.4.1 Physiological Models

Physiologically based models attempt to simulate the inner-mechanisms of the auditory system based on existing knowledge and understanding of the ear. As mentioned earlier, there is still a great deal about the auditory system that we do not yet understand, particularly the physiological basis to many of the phenomena that are readily noticed in psychophysical experiments, which make the design difficult. But physiological models do represent the ideal way of approaching this problem and can eventually lead to solutions that provide highly accurate results.

An example of a physiological model was proposed by Baumgarte in [59] where the ear was modeled by a series of stages that paralleled the outer, middle, and the inner ear. More specifically, the processing stages were 1) an acoustic filter for the outer and middle ear 2) a BM cochlear filter bank with 103 bands 3) an inner hair cell rectifier/low-pass filter 4) a neural processing stage and 6) a masking curve estimation stage. Other examples have also appeared in [60, 61] where excitation patterns were derived using extensive modeling of the inner ear.

3.4.2 Excitation Pattern Models

Excitation pattern models work by first calculating the excitation pattern that is generated by the masker signal, i.e. the input signal, and the pattern generated by the masker-plus-maskee signal, i.e. the reconstructed signal. Complete masking is, then, said to be achieved if the difference between the two curves differ by no more than a certain threshold across all frequencies. The threshold is usually taken to be between 0.1 and 1 dB. The idea behind this model is that all signals, masker and maskee, generate a certain activity of patterns in the neural fibres according to some complex set of rules and that a maskee signal, e.g. quantization noise, can only be detected in the presence of an existing signal, e.g. input signal, if the additional contribution in the neural activity exceeds a certain amount that the ear is able to detect [54].

In general, the excitation patterns alone do not directly lead to a masking threshold as they are only able to indicate whether a given noise signal can be detected or not. As a result, the excitation pattern model has been suggested more as a tool for evaluating existing psychoacoustic models rather than being used as a model that computes the masking threshold. Examples of excitation pattern models have appeared in [54, 56].

3.4.3 Masking Pattern Models

Masking pattern models are considerably simpler than the previous two as they mainly employ a set of simple rules for deriving the masking threshold. Masking pattern models are also the most common type used in current state-of-the-art audio coders. A typical masking pattern model contains the following steps:

1) The power spectral density (PSD) of the input signal is first computed and divided according to the CB scale.

2) The tonality of each masking component in the signal is identified.

3) Spreading functions are applied to the individual masking components.

4) The overall masking threshold is found by combining the individual masking thresholds while taking into account the tonality of each masking component.

5) The absolute threshold of hearing is also taken into account, i.e. masking that falls below the absolute threshold at any point is raised to the absolute threshold level.

We can see that each step of the model is based on some psychophysical rule already discussed in the previous sections. The first step is based on the power spectrum model of masking (section 3.3.1.1), the second step reflects the asymmetry of masking (3.3.3), the third step uses the spreading function described in section 3.3.2, the fourth step involves an assumption that individual masking components can be added in a simple way, and the fifth step includes the absolute threshold of hearing that was described in section 3.1.2.

However, a number of shortfalls in this model also exist:

1) The model only describes simultaneous masking and results only apply in the frequency domain, i.e. temporal masking details are usually not incorporated into the model.

2) The spreading function is derived from simple single-masker/single-maskee experiments and applied to more complex multiple-masker/multiple-maskee scenarios, which does not take the non-linear and complex effects in the ear into account. Furthermore, the level-dependence and frequency-dependence of masking patterns are usually ignored.

3) The critical band analysis is typically based on a fixed and discretized division of the frequency domain, e.g. 24 CB-division as proposed in [31], and the resulting masking threshold reflects this discretized resolution. In reality, however, the ear performs a spectral analysis of the input signal using a CB scale that is continuous across the entire range of hearing without any artificial boundaries. This means that masking thresholds, in reality, do not exhibit the CB boundaries that are found in the calculated masking threshold. For example, MPEG Psychoacoustic Model 1 is based on a 24-band CB division, and Psychoacoustic Model 2 is based on a 63-band 1/3-octave division.

In spite of these shortfalls, masking pattern models have shown to perform reasonably well with manageable complexity in a number of current audio coders that achieve very

high compression rates, e.g. MPEG-1 Layer III, MPEG-AAC, PAC. Examples of masking pattern models include the MPEG Psychoacoustic Models 1 and 2 [17], the non-linear extension to MPEG Psychoacoustic Model 2 as proposed in [62], a simplified simultaneous masking model in [32], and a time-frequency masking model in [63] that tries to incorporate simple temporal masking.

3.5 Summary

This chapter gave a brief overview of psychoacoustics in the context of perceptual audio coding. A description of the human auditory system and the process of transduction of audio signals were first given. A number of well known and important results from psychophysical experiments were then described. In particular, the concept of the critical band (CB) scale, the modeling of the ear as a bank of bandpass filters, the use of masking patterns to describe the spread of masking, the asymmetric nature of masking, and the presence of temporal masking were some of the main issues described. Finally, existing psychoacoustic models were discussed in terms of three general categories, namely, physiological, excitation pattern, and masking pattern models. Masking pattern-based models were found to be the most popular among the three due to the reasonable trade-off that they offered in terms of complexity and accuracy. A number of limitations in the masking pattern models were also briefly noted.

Chapter 4

Overview of Wavelets and Filter Banks

This chapter presents the theory of the wavelet transform (WT) and its connection to the theory of multirate filter banks. The wavelet transform was first introduced in the mathematical literature by Grossmann and Morlet in 1984, and further treated by Meyer, Daubechies, Mallat, and others in the late 1980's [35]. In particular, works by Daubechies and Mallat established the connection between wavelets and digital filter banks that, as a result, generated much interest and activity in the respective areas. The theory of multirate filter banks, on the other hand, was first developed in the context of coding applications in the late 1970's by Croisier, Esteban, and Galand who introduced a special class of filters called *quadrature mirror filters* (QMF), and also by Crochiere, Webber, and Flanagan who introduced a similar technique in the context of speech coding [37]. Subsequently, solutions to the perfect reconstruction (PR) filter bank for the two-band and the general M-band case were found, and a general theory on the design of multirate filter banks was also established. Some historical perspectives on the development of wavelets and filter banks can be found in [37, 35, 40, 36], and in-depth studies of wavelets and filter banks can be found in [37, 35, 41].

This chapter is organized as follows. Section 4.1 describes maximally decimated two-channel filter banks, section 4.2 presents the wavelet transform in the continuous-time and discrete-time domain and shows its relationship to the two-channel filter bank, section 4.3 covers design issues of the wavelet filter bank, and section 4.4 ends with a brief summary.

4.1 The Two-Channel Filter Bank

Digital Filter banks are commonly used in applications that require a way of transforming the input signal into a frequency or time-frequency domain representation. As the name suggests, this is done through a bank of filters that divides the signal spectrum into approximate frequency *subbands* or *channels* and generates a time-indexed series of coefficients that represent the frequency-localized signal energy within each band [2].

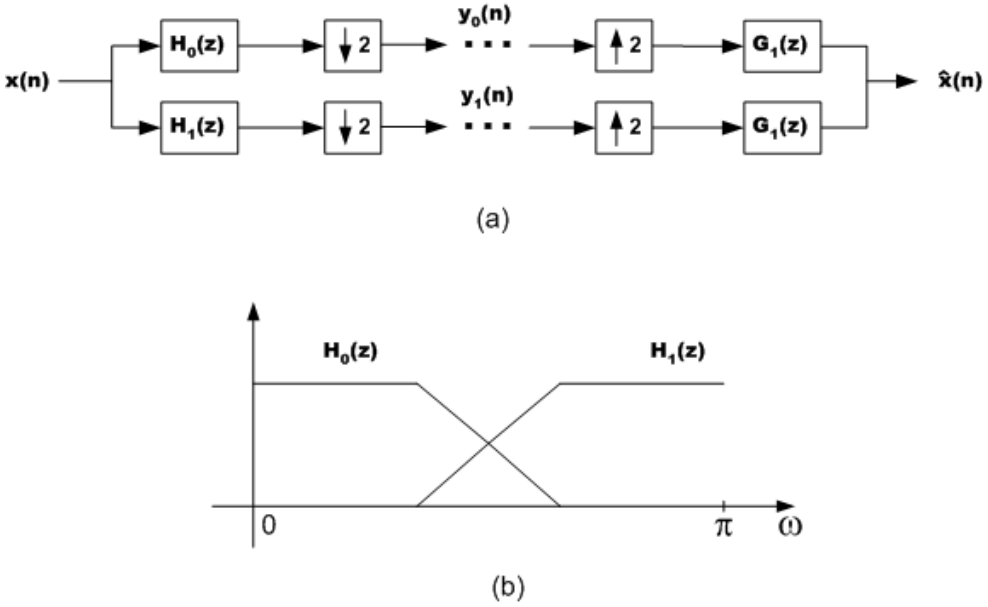


Figure 4.1: Two-channel filter bank (a) analysis and synthesis filter bank structure (b) frequency response of analysis filters $H_0(z)$ and $H_1(z)$

A uniform two-channel filter bank is shown in Figure 4.1(a) and the corresponding magnitude response in Figure 4.1b). In the analysis stage, the input signal $x(n)$ is filtered by the low-pass filter $H_0(z)$ and the high-pass filter $H_1(z)$ and then down-sampled by a factor of 2 to produce subband signals $y_0(n)$ and $y_1(n)$, respectively. In the synthesis stage, the subband signals $y_0(n)$ and $y_1(n)$ are first up-sampled by a factor of 2, then passed through low-pass filter $G_0(z)$ and high-pass filter $G_1(z)$, respectively, and finally added together to produce the reconstructed signal $\hat{x}(n)$.

In the z-domain, the down-sampling and up-sampling operations can be expressed as

[35]

$$g(n) = (\downarrow 2)f(n): \quad G(z) = \frac{1}{2}[F(z^{1/2}) + F(-z^{1/2})] \tag{4.1}$$

$$g(n) = (\uparrow 2)f(n): \quad G(z) = F(z^2). \tag{4.2}$$

Using equations 4.1 and 4.2 and the input-output relationship of the filter bank in Figure 4.1, we obtain

$$\begin{aligned} \hat{X}(z) &= \frac{1}{2}\{H_0(z)G_0(z) + H_1(z)G_1(z)\}X(z) \\ &+ \frac{1}{2}\{H_0(-z)G_0(z) + H_1(-z)G_1(z)\}X(-z) \end{aligned} \tag{4.3}$$

where the first term represents the amplitude and phase distortions that result from the filtering operations and the second term represents the aliasing and imaging distortions that result from the down-sampling and up-sampling operations. The first term is called the *distortion transfer function*, $T(z)$, and the second term is called the *aliasing transfer function*, $A(z)$, i.e.

$$T(z) = \frac{1}{2}\{H_0(z)G_0(z) + H_1(z)G_1(z)\}, \tag{4.4}$$

and

$$A(z) = \frac{1}{2}\{H_0(-z)G_0(z) + H_1(-z)G_1(z)\}. \tag{4.5}$$

Since any distortion caused by the filter bank is undesirable, especially aliasing error [37], the design of the analysis and synthesis filters revolve around the requirements of *alias cancellation* (AC) and *perfect reconstruction* (PR). The conditions for AC and PR can be summarized as follows.

- Alias Cancellation: Choose the synthesis filters as

$$G_0(z) = H_1(-z), \tag{4.6}$$

$$G_1(z) = -H_0(-z). \tag{4.7}$$

  Then,

$$A(z) = \frac{1}{2}\{H_0(-z)H_1(-z) - H_1(-z)H_0(-z)\} = 0. \tag{4.8}$$

  Notice that the AC condition simplifies the design to designing only filters $H_0(z)$ and $H_1(z)$ and minimizing the distortion in $T(z)$.

- Perfect Reconstruction: For PR, we need

$$T(z) = cz^{-l}, \tag{4.9}$$

where $c =$ constant and $l \in \mathbb{Z}$, and

$$A(z) = 0, \tag{4.10}$$

so that (with $c = 1$)

$$\begin{aligned} \hat{X}(z) &= T(z)X(z) + A(z)X(-z) \\ &= z^{-l}X(z) + (0)X(-z) \\ &= z^{-l}X(z), \end{aligned} \tag{4.11}$$

where the reconstructed signal is just a delay of the input signal by z^{-l}.

4.1.1 Classic QMF Filters (non-PR)

The "classic" QMF filters proposed by Croisier, Esteban, and Galand [41] are designed by first imposing the relationship

$$H_1(z) = H_0(-z) \qquad \text{or} \qquad h_1(n) = (-1)^n h_0(n), \tag{4.12}$$

which relates the low-pass and high-pass filter through a simple sign alteration. Equation 4.12 can also be expressed in the Fourier domain as

$$|H_1(e^{jw})| = |H_0(e^{j(\pi - w)})|. \tag{4.13}$$

$H_1(e^{jw})$ in equation 4.13 represents a high-pass filter whose response is a mirror image of the low-pass filter response $|H_0(e^{jw})|$ with respect to the quadrature frequency, $\frac{\pi}{2}$. Using the AC condition of equations 4.6 and 4.7, and equation 4.12 above, the distortion transfer function can now be simplified to

$$\begin{aligned} T(z) &= \frac{1}{2}\{H_0(z)G_0(z) + H_1(z)G_1(z)\} \\ &= \frac{1}{2}\{H_0(z)H_1(-z) - H_0(-z)H_1(z)\} \\ &= \frac{1}{2}\{H_0^2(z) - H_0^2(-z)\}. \end{aligned} \tag{4.14}$$

Note that the design of QMF filters according to 4.14 only involves one filter, $H_0(z)$. Several well known solutions to this exist and a few are described next. First, note that for PR we need

$$T(z) = \frac{1}{2}\{H_0^2(z) - H_0^2(-z)\} = z^{-l}. \tag{4.15}$$

The only solution to 4.15 using an FIR filter is the trivial Haar filter as all other solutions involve some type of distortion in $T(z)$ [64]. Among more practical FIR solutions, Johnston's filters [40] offer small reconstruction error and good overall performance. Johnston's filters are designed to provide high stop-band attenuations and good transition-band characteristics while eliminating phase distortion and minimizing amplitude distortion in $T(z)$. Among IIR solutions, the well known elliptic filters offer a solution where amplitude distortion is eliminated and phase distortion is minimized [65]. Other solutions to (15) can be found in [35, 36].

4.1.2 Smith-Barnwell Filters (PR Orthogonal)

The solution proposed by Smith and Barnwell [66] is based on the AC condition and the relationship

$$H_1(z) = -z^{-N} H_0(-z^{-1}), \tag{4.16}$$

where filters $H_0(z)$ and $H_1(z)$ (as well as $G_0(z)$ and $G_1(z)$) are FIR filters of odd order N. Also called *conjugate quadrature filters* (CQF), these filters provide the quadrature mirror property like QMF filters, but also the perfect reconstruction property as $T(z)$ can now be made to be a pure delay. The distortion function $T(z)$ can be simplified using equations 4.6, 4.7, 4.16 as

$$\begin{aligned} T(z) &= \frac{1}{2}\{H_0(z)G_0(z) + H_1(z)G_1(z)\} \\ &= \frac{1}{2}\{H_0(z)H_1(-z) - H_0(-z)H_1(z)\} \\ &= \frac{z^{-N}}{2}\{H_0(z)H_0(z^{-1}) + H_0(-z)H_0(-z^{-1})\}. \end{aligned} \tag{4.17}$$

Note that the design of CQF filters also involves only one filter, $H_0(z)$, as the other three can be derived using equations 4.6, 4.7, 4.16. To obtain PR in equation 4.17, we need

$$H_0(z)H_0(z^{-1}) + H_0(-z)H_0(-z^{-1}) = 2. \tag{4.18}$$

If we define

$$P(z) = H_0(z)H_0(z^{-1}), \tag{4.19}$$

then we can re-write 4.18 as

$$P(z) + P(-z) = 2. \tag{4.20}$$

$P(z)$ represents a zero-phase *half-band* filter in which all even-indexed terms are zero except the term at z^0. Description and design of half-band filters have already been discussed extensively in the filter bank literature, e.g. [35]. Once half-band filter $P(z)$ is designed, filter $H_0(z)$ can be obtained through symmetrical factorization of 4.19 [67]. In addition to PR, Smith-Barnwell filters also provide the orthogonality property that is described next. First, using equation 4.16 in 4.18, we obtain

$$H_1(z)H_1(z^{-1}) + H_1(-z)H_1(-z^{-1}) = 2. \tag{4.21}$$

Next, we can re-write equation 4.16 and obtain

$$H_0(z) = -z^{-N}H_1(-z^{-1}), \tag{4.22}$$

and using the equality relationship given by

$$H_0(z^{-1})H_1(z) = H_0(z^{-1})H_1(z), \tag{4.23}$$

we can substitute 4.23 in equations 4.16 and 4.22 to obtain

$$\begin{aligned} H_0(z^{-1})H_1(z) &= (-z^N H_1(-z))(z^{-N}H_0(-z^{-1})) \\ H_0(z^{-1})H_1(z) &= -H_1(-z)H_0(-z^{-1}) \\ H_0(z^{-1})H_1(z) &+ H_1(-z)H_0(-z^{-1}) = 0. \end{aligned} \tag{4.24}$$

Equations 4.18, 4.21, and 4.24 represent the orthogonality condition in the z-domain. The term $H_i(z)H_i(z^{-1})$ in equations 4.18 and 4.19 represents the auto-correlation of $H_i(z)$, and the term $H_0(z^{-1})H_1(z)$ in 4.24 represents the cross-correlation between $H_0(z)$ and $H_1(z)$ [68]. Equations 4.18 and 4.21 are also known as the power symmetric property

[35]. In the time domain, equations 4.18, 4.21, and 4.24 can be expressed as

$$\sum_n h_0(n)h_0(n+2k) = \delta(k), \tag{4.25}$$

$$\sum_n h_1(n)h_1(n+2k) = \delta(k), \tag{4.26}$$

$$\sum_n h_0(n)h_1(n+2k) = 0, \tag{4.27}$$

or more succinctly as

$$\sum_n h_i(n)h_j(n+2k) = \delta(i-j)\delta(k) \tag{4.28}$$

where

$$\begin{aligned} h &= \text{analysis filters} \\ g &= \text{synthesis filters} \\ i,j &= 0 \text{ for low-pass, 1 for high-pass} \\ k &\in \mathbb{Z}. \end{aligned}$$

In general, Smith-Barnwell filters provide PR, finite support, and orthogonality, but lack linear phase (except for the trivial Haar filter).

4.1.3 Generalized QMF Filters (PR Linear Phase)

Generalized QMF filters represent PR solutions that sacrifice orthogonality for linear phase. Using equation 4.4 and the AC condition, we obtain

$$\begin{aligned} T(z) &= \frac{1}{2}\{H_0(z)G_0(z) + H_1(z)G_1(z)\} \\ &= \frac{1}{2}\{H_0(z)H_1(-z) - H_0(-z)H_1(z)\} \end{aligned} \tag{4.29}$$

where filters $H_0(z)$ and $H_1(-z)$ can be of even or odd order and the lengths of the two are not necessarily equal. Unlike the CQF filters, the design now involves first designing the two analysis filters $H_0(z)$ and $H_1(-z)$, and then obtaining the two synthesis filters using equations 4.6 and 4.7. To satisfy PR, we impose

$$H_0(z)H_1(-z) - H_0(-z)H_1(z) = 2z^{-2l-1} \tag{4.30}$$

where $l \in \mathbb{Z}$. Note that the delay term on the right-hand side has to be odd since all even terms of $H_0(z)H_1(-z)$ cancel with the even terms of $H_0(-z)H_1(z)$. Defining

$$P(z) = z^{2l+1} H_0(z) H_1(z) \tag{4.31}$$

we can formulate the PR condition as

$$P(z) + P(-z) = 2, \tag{4.32}$$

which again represents a zero-phase half-band filter. However, since orthogonality is no longer required, $P(z)$ in 4.31 is no longer factored symmetrically but factored so as to provide symmetry in $H_0(z)$ and $H_1(z)$ separately. Detail and examples of this procedure can be found in [35, 64]. Similar to the orthogonality condition given in the z-domain and time-domain, we can summarize the biorthogonality condition in the z-domain as [37]

$$H_0(z)G_0(z) + H_1(z)G_1(z) = 2 \tag{4.33}$$

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0 \tag{4.34}$$

and in the time-domain as

$$\sum_n h_i(n) g_j(2k - n) = \delta(i - j)\delta(k)(35?) \tag{4.35}$$

where

$$\begin{aligned} h &= \text{analysis filters} \\ g &= \text{synthesis filters} \\ i, j &= 0 \text{ for low-pass, 1 for high-pass} \\ k &\in \mathbb{Z}. \end{aligned}$$

Note that biorthogonality is a more general condition that provides orthogonality across the analysis and synthesis filters [40], as opposed to within the analysis and synthesis filters, and hence the name “bi-orthogonal”.

4.1.4 Summary and Discussion

Two-channel filter banks, in general, are characterized by the type of errors they introduce into the signal and the properties that the filters provide. Reconstruction error is made up of three components, namely, 1) aliasing distortion, 2) amplitude distortion, and 3) phase distortion. Aliasing (and imaging) distortion is represented by $A(z)$, and amplitude and phase distortions are represented by $T(z)$. Properties of filters that we are particularly interested in are 1) finite support (i.e. FIR) 2) orthogonality, and 3) linear phase. Ideally, all three properties need to be incorporated into the filters as they are considered important in audio coding, e.g. orthogonality ensures that quantization noise in different channels remain independent, linear phase provides constant group delay, and finite support leads to stable and simple implementations [69]. But it has been found that only two out of the three properties can be satisfied simultaneously for any given two-channel PR filter bank [69]. This limitation is illustrated in Figure 4.2 where different solutions to the two-channel PR filter bank are shown. Regions of solutions for the three properties are shown where we find regions that offer two out of the three properties, but none that offer all three, except at the center point where the three properties overlap (i.e. Haar solution).

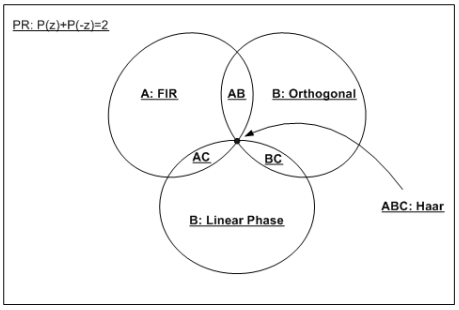


Figure 4.2: Two-channel PR filter bank solutions Venn diagram for 1) finite support, 2) orthogonality, and 3) linear phase (P(z) is rational and real)

We can summarize the two-channel filter bank solutions described in this section according to Table 4.1. Table 4.1 shows a convenient description of the four families of filters using the properties that revolve around PR. Note that, in addition to these properties, filter banks generally need to be designed to provide other important properties such as good stopband attenuation, sharp cut-off rate, low pass-band and stop-band ripples, and short delay [35].

Filter Family	Distortions			Competing Properties		
	ALD	AD	PD	FIR	Orthogonal	Linear Phase
Johnston	None	Min.	None	Yes	?	Yes
Elliptic	None	None	Min.	No	?	No
Smith-Barnwell	None	None	None	Yes	Yes	No
Generalized QMF	None	None	None	Yes	No	Yes

Table 4.1: Two-channel filter bank solutions described in terms of properties that revolve around PR

It is interesting to note that during the development of two-channel (and the more general M-channel) filter banks, the so called *polyphase* representation provided a considerable amount of simplification in theory, design, and implementation. The polyphase representation is essentially a regrouping of terms in the z-domain that allows an efficient representation of the filter bank according to analysis and synthesis polyphase matrices. Some important constraints such as AC, PR, and orthogonality can be rather conveniently expressed using these matrices. As a result, much of the filter bank theory discussed today is based on the polyphase representation [35, 41, 37].

4.2 Wavelets

The origins of wavelets are many and multi-disciplinary. It seems that the idea of wavelet analysis existed even before it was first introduced in the mathematical literature (under its current name) in various forms and in different fields, including pure mathematics, quantum physics, geophysics, artificial vision, and signal processing. For example, the *Haar* wavelet was first proposed as early as 1910 by A. Haar [37]. The name "wavelet" as we use it today derives its meaning from the works of Goupillaud, Morlet, and Grossman from the early 1980's [37], although the term was used by others before. More information on the origins of wavelet and its development can be found in [70, 71, 72].

Due to this unique background, there are many different ways of looking at and interpreting the wavelet transform. The definition of WT as viewed from mathematics is a decomposition of a continuous-time signal in terms of a collection of orthonormal basis functions called *wavelets*. This decomposition can also be thought of as a transformation of a signal from the time to the time-frequency (or time-scale) domain, similar to how we use the (short-time) Fourier transform [68]. In addition, the wavelet domain representation is often viewed as a multiresolution analysis, where one can see the details of a

signal at both coarse and fine scales. From a linear algebra point of view, the collection of wavelet basis functions forms a vector space that can be represented through a matrix. Applying the WT would, therefore, be equivalent to performing a matrix multiplication between the wavelet transformation matrix and the signal vector. In the context of digital filter banks, it has been found that the WT of a discrete-time signal is equivalent to a tree-structured filter bank.

Wavelets will be described in detail next in terms of its definition in continuous-time and discrete-time domain, connection to the two-channel PR filter bank, implementation, and design.

4.2.1 Wavelet in Continuous-Time Domain

4.2.1.1 Continuous Wavelet Transform

The wavelet transform of a square-integrable function $x(t) \in L^2(\mathbb{R})$ is defined as

$$W(a,b) = \int_{-\infty}^{\infty} \psi_{ab}(t)x(t)dt, \tag{4.36}$$

where

$$\begin{aligned} a &\in \mathbb{R}^+ \\ b &\in \mathbb{R} \\ \psi_{ab}(t) &= \sqrt{a}\psi(a(t-b)). \end{aligned} \tag{4.37}$$

$\psi_{ab}(t)$ represents the wavelet basis functions that are derived from a single *mother wavelet* function, $\psi(t)$, through dilations a and translations b according to equation 4.37. The wavelet basis functions represent an orthonormal basis to the space of $L^2(\mathbb{R})$ such that

$$L^2(\mathbb{R}) = \overline{span}\{\psi_{ab}(t); a \in \mathbb{R}^+, b \in \mathbb{R}\} \tag{4.38}$$

Figure 4.3 shows an example of a mother wavelet function and its dilated and translated versions. We can see that as dilation variable a increases, $\psi_{ab}(t)$ becomes more contracted and time-localized and as a decreases, $\psi_{ab}(t)$ becomes more expanded and less time-localized. This allows the basis functions to "see" better the finer details of the signal when a is small and also "see" the coarse shape of the signal when a is large. The factor $\sqrt{a}$ in 4.37 is used to normalize the energy of the wavelet basis function across scales.

Furthermore, assuming that the Fourier transform pair of the mother wavelet is

$$\psi(t) \leftrightarrow \Psi(w), \tag{4.39}$$

the translation and dilation operations can be expressed as

$$\sqrt{a}\psi(a(t-b)) \leftrightarrow \frac{1}{\sqrt{a}}\Psi(\frac{w}{a})e^{-jbw}. \tag{4.40}$$

According to equation 4.40, a contraction by a in one domain results in a dilation by a in the other domain, and a time-translation in the time domain produces the pure delay term e^{-jbw} in the frequency domain. The dilation operation, in particular, shows that an increase in resolution in one domain results in a loss of resolution in the other. This actually reflects the trade-off that exists between time and frequency domain resolution as dictated by Heisenberg's Uncertainty Principle [70].

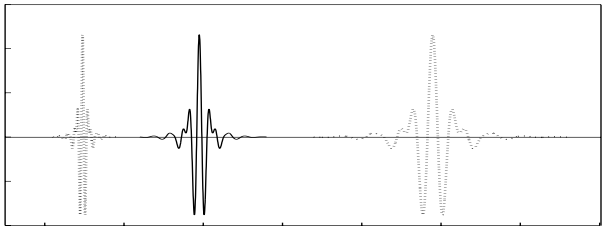


Figure 4.3: Example of a wavelet basis function (Meyer)

Next, the inverse wavelet transform can be defined as

$$x(t) = \frac{1}{C}\int_{-\infty}^{\infty}\int_{0}^{\infty}\frac{W(a,b)\psi_{ab}(t)}{a^2}dadb, \tag{4.41}$$

where

$$C = \int_{0}^{\infty}\frac{|\Psi(w)|^2}{w}dw. \tag{4.42}$$

In order for C in equation 4.42 to be a finite value, we need $\Psi(0)$ to be zero, which means that $\psi(t)$ is a zero mean function in the time domain and resembles a bandpass filter in the frequency domain [73]. Since the forward and inverse wavelet transforms of equations 4.36 and 4.41 are defined using continuous variables a and b, they are referred to as the *continuous wavelet transform* (CWT).

4.2.1.2 Discrete Wavelet Transform

The CWT as defined above is in fact infinitely redundant and not all $W(a,b)$ for $a \in \mathbb{R}^+$ and $b \in \mathbb{R}$ are required for full reconstruction of $x(t)$. A more compact representation can be found in the *Discrete Wavelet Transform* (DWT) where only the required wavelet coefficients for the reconstruction of x(t) are kept. This is done by sampling the dilation and translation variables a and b according to [68]

$$a = a_0^m \qquad \text{and} \qquad b = nb_0a_0^m \tag{4.43}$$

where

$$\begin{aligned} a_0 &> 1 \\ b_0 &\neq 0 \\ m, n &\in \mathbb{Z}. \end{aligned}$$

a_0 is an arbitrary reference scale, b_0 is an arbitrary reference time position, and m and n are the new scaling and shifting variables, respectively. Choosing $a_0 = 2$ and $b_0 = 1$ leads to the well known *Fast Wavelet Transform* (FWT), where

$$a = 2^m \qquad \text{and} \qquad b = n2^m \tag{4.44}$$

and the wavelet basis function $\psi_{ab}(t)$ becomes

$$\psi_{mn}(t) = 2^{m/2}\psi(2^m t - n). \tag{4.45}$$

The forward and inverse DWT can then, respectively, be defined as

$$W(m,n) = \int_{-\infty}^{\infty} \psi_{mn}(t)x(t)dt =< x(t), \psi_{mn}(t) > \tag{4.46}$$

for $m, n \in \mathbb{Z}$, and

$$x(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} W(m,n)\psi_{mn}(t), \tag{4.47}$$

where the operation $< \cdot, \cdot >$ in 4.46 represents the inner product between two functions. The basis functions $\psi_{mn}(t)$ in equation 4.45 now provide an orthonormal basis that is no longer redundant [68]. However, equation 4.47 still requires an infinite number of terms

to describe the infinitely coarse, i.e. $m \to -\infty$, as well as the infinitely fine, $m \to \infty$. Since this is still somewhat impractical, we need to take additional steps to reduce the wavelet domain representation to a finite number of terms. This can be done by defining a new family of basis functions called *scaling functions*, $\phi_{mn}(t)$, that are derived, much like the wavelets, from a single *mother scaling function*, $\phi(t)$, according to

$$\phi_{mn}(t) = 2^{m/2}\phi(2^m t - n) \tag{4.48}$$

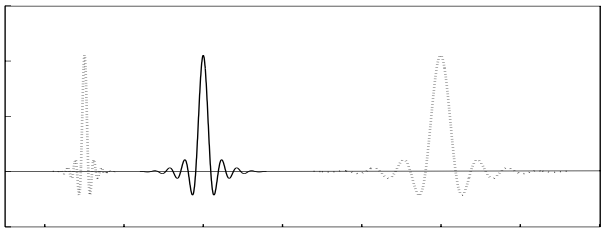


Figure 4.4: Example of a scaling function (complementary to the wavelet in Figure 4.3)

Figure 4.4 shows examples of scaling functions that are scaled and shifted from a mother $\phi(t)$. The scaling functions $\phi_{mn}(t)$ actually represent a complementary basis to the wavelet basis functions such that

$$\sum_n c(l,n)\phi_{ln}(t) = \sum_n \sum_{m=-\infty}^{l-1} d(m,n)\psi_{mn}(t) \tag{4.49}$$

where

$$c(l,n) =< \phi_{ln}(t), x(t) > . \tag{4.50}$$

Equation 4.49 indicates that signal details represented by wavelets at scales $-\infty < m < l$ can be represented by scaling functions at level $m = l$. This means that the scaling functions at level l provides a "complementary" basis to the wavelet basis functions at level l, since together they cover all scales for $-\infty < m \leq l$. As we will see, this naturally leads to the idea of multiresolution analysis. The orthonormal representation and the complementary relationship between wavelet and scaling functions can be summarized as

follows [68]:

$$\begin{aligned} < \psi(x+l), \psi(x+k) > &= \delta(k-l), && (4.51)\\ < \phi(x+l), \phi(x+k) > &= \delta(k-l), && (4.52)\\ < \phi(x+l), \psi(x+k) > &= 0. && (4.53)\\ & && (4.54) \end{aligned}$$

where $l, k \in \mathbb{Z}$. Now, simplifying equation 4.47 can be done in a couple of steps. First, the wavelet representation of levels $m < 0$ in 4.47 can be reduced to one level using the scaling basis functions at level $m = 0$. Second, the wavelet representation of levels $m \geq L$ can be ignored by assuming that function $x(t)$ can be represented by scaling functions at level L, i.e.

$$x(t) = \sum_n c(L,n)\phi_{Ln}(t). \tag{4.55}$$

Obviously, a continuous function can not always be expressed exactly in terms of scaling functions at a finite resolution, but in practice the error can be minimized to an arbitrarily low level by increasing resolution L. Using the above two simplifications, 4.47 now becomes

$$x(t) = \sum_{n=-\infty}^{\infty} c(0,n)\phi_{0n}(t) + \sum_{m=0}^{L-1}\sum_{n=-\infty}^{\infty} d(m,n)\psi_{mn}(t) \tag{4.56}$$

where

$$\begin{aligned} c(m,n) &= < x(t), \phi_{mn}(t) > && (4.57)\\ d(m,n) &= < x(t), \psi_{mn}(t) > . && (4.58) \end{aligned}$$

Coefficients $c(0,n)$ are referred to as *approximation coefficients* and coefficients $d(m,n)$ are referred to as *detailed coefficients*. Equations 4.56, 4.57, and 4.58 represent the finite L-resolution DWT that is commonly found in practice. Note that $x(t)$ in 4.56 still represents a continuous-time function.

4.2.1.3 Multiresolution Analysis

As already mentioned, the decomposition of a signal $x(t)$ in terms of wavelet and scaling functions $\psi_{mn}(t)$ and $\phi_{mn}(t)$ results in a multiresolution analysis. And as the name implies, multiresolution analysis provides a way of breaking down a signal into multiple resolutions so that each resolution captures details that other resolutions are blind to.

The idea of multiresolution relies on the basic notion of subspace, where we introduce two families of subspaces called V_m and W_m next.

V_m is defined as the subspace of $L^2(\mathbb{R})$ spanned by the scaling function $\phi(t)$ at resolution level m, i.e.

$$V_m \triangleq \overline{span}\{\phi_{mn}(t); n \in \mathbb{Z}\}. \tag{4.59}$$

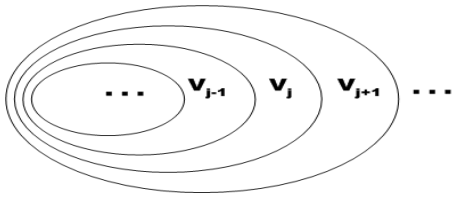


Figure 4.5: Nested resolutions

Equivalently, the set of all translated versions of $\phi(t)$ at resolution level m represents an orthonormal basis for the subspace V_m. Furthermore, the subspaces V_m, for $m \in \mathbb{Z}$, can be represented as nested subspaces in $L^2(\mathbb{R})$ where subspace V_l is a subset of subspace V_{l+1} as shown in Figure 4.5. This follows from the fact that $\phi_{ln}(t)$ as defined in 4.49 is able to represent the signal detail for all resolutions less than l and, therefore, $\phi_{l+1,n}(t)$ at level $l+1$ encompasses the details at level l. Furthermore, we can also see this intuitively from the definition of the scaling function in 4.48 where $\phi_{mn}(t)$ provides greater localization for increasing resolution m. As a result, subspace V_m approaches $L^2(\mathbb{R})$ as $m \to \infty$ and V_m approaches the empty space 0 as $m \to -\infty$. More generally, a multiresolution analysis is required to provide the following properties [73, 74]:

1) **Containment**: The resolution of subspaces are nested such that

$$V_{-\infty} = \{0\} \subset \ldots V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \ldots \subset V_\infty = L^2(\mathbb{R}) \tag{4.60}$$

where finer resolutions see coarser resolutions perfectly.

2) **Completeness**: All V_m are orthogonal to each other and the collection of all subspaces V_m provide a complete space to $L^2(\mathbb{R})$, i.e.

$$\bigcap_{m\in\mathbb{Z}} V_m = \{0\} \quad \text{and} \quad \bigcup_{m\in\mathbb{Z}} V_m = L^2(\mathbb{R}). \tag{4.61}$$

3) **Scaling Property**: If function $f(t)$ is in the subspace V_m, then its scaled version $f(2t)$ is in the subspace V_{m+1}, i.e.

$$f(t) \in V_m \iff f(2t) \in V_{m+1}. \tag{4.62}$$

4) **Shifting Property**: If function $x(t)$ is in the subspace V_m, then all its shifted versions are also in the subspace V_m, i.e.

$$f(t) \in V_m \iff f(t-k) \in V_m, k \in \mathbb{Z}. \tag{4.63}$$

Similar to how we defined V_m, we can also define W_m as the subspace spanned by the wavelet function $\psi(t)$ at resolution level m according to

$$W_m \triangleq \overline{span}\{\psi_{mn}(t); n \in \mathbb{Z}\}. \tag{4.64}$$

The subspace W_m represents an orthogonal complement to the subspace V_m so that the union of V_m and W_m forms the subspace V_{m+1} at one resolution higher, i.e.

$$V_m \perp W_m \qquad \text{and} \qquad V_{m+1} = V_m \oplus W_m. \tag{4.65}$$

In terms of the diagram in Figure 4.5, W_m represents the "difference" between subspaces V_m and V_{m+1}, or the complementary subspace required for V_m to become V_{m+1}. Now, we can setup a multiresolution scheme by choosing reference resolution V_{m_0} and successively adding the subspaces W_{m_0}, W_{m_0+1}, W_{m_0+2}, etc ... to obtain the subspaces V_{m_0+1}, V_{m_0+2}, V_{m_0+3}, etc .... In fact, this is exactly what is done in equation 4.56 for the finite L-resolution DWT. Now, since subspaces V_m and W_m represent subsets of the subspace V_{m+1} according to 4.65, the basis functions $\{\phi_{mn}(t)\}$ and $\{\psi_{mn}(t)\}$ of V_m and W_m, respectively, can be expressed as linear combinations of $\{\phi_{m+1,n}(t)\}$, the basis function of V_{m+1}. For $m = 0$, we then have

$$\phi(t) = \sum_n c_0(n)\sqrt{2}\phi(2t-n), \tag{4.66}$$

$$\psi(t) = \sum_n c_1(n)\sqrt{2}\phi(2t-n). \tag{4.67}$$

where the lengths of c_0 and c_1 are finite since $\phi(t)$ and $\psi(t)$ are of finite support [74]. Equation 4.66 is called the *multiresolution analysis equation* and equation 4.67 is called

the *wavelet equation*. The multiresolution analysis equation relates the scaling function at two different resolutions through $c_0(n)$, or the *scaling coefficients*, and the wavelet equation relates the wavelet function at one resolution to the scaling function at the next resolution through $c_1(n)$, or the *wavelet coefficients*.

4.2.2 Wavelet in Discrete-Time Domain

The description of wavelets so far was given in the continuous-time domain, whereas the filter bank presented in section 4.1 dealt with signals in the discrete-time domain. The connection between wavelets in continuous-time domain and two-channel PR filter banks in discrete-time domain has been first recognized by Mallat [75] and further investigated by Daubechies [68]. Specifically, the scaling coefficients c_0 and the wavelet coefficients c_1 that appear in equations 4.66 and 4.67 are precisely the same filters as h_0 and h_1 of the two channel filter bank in section 4.1. It has been shown that the compactly supported orthonormal DWT necessarily implies an underlying two-channel PR filter bank, and conversely, the filters of a two-channel FIR PR-CQF filter bank can be used to generate the wavelet basis functions under certain conditions. A description of this relationship is briefly described next.

4.2.2.1 Relationship Between Orthonormal DWT and Two-channel CQF Filter Bank

We can show that filters c_0 and c_1 of equations 4.66 and 4.67 satisfy the same orthogonality conditions that h_0 and h_1 satisfy according to equations 4.25, 4.26, and 4.27. First, note that using equation 4.52 we obtain the relationship

$$\begin{aligned} < \phi(2x+l), \phi(2x+k) > &= \int \phi(2x+l)\phi(2x+k)dx \qquad [\text{let } y = 2x] \\ &= \int \phi(y+l)\phi(y+k)\frac{dy}{2} \\ &= \frac{1}{2} < \phi(y+l), \phi(y+k) > \\ &= \frac{1}{2}\delta(k-l). \end{aligned} \tag{4.68}$$

Next, using equation 4.66 in 4.52 we get

$$\begin{aligned}
< \phi(x), \phi(x+k) > &= \delta(k) \\
< \sum_n c_0(n)\sqrt{2}\phi(2t-n), \sum_m c_0(m)\sqrt{2}\phi(2t+2k-m) > &= \delta(k) \qquad [\text{let } m' = m-2k] \\
< \sum_n c_0(n)\sqrt{2}\phi(2t-n), \sum_{m'} c_0(m'+2k)\sqrt{2}\phi(2t-m') > &= \delta(k) \\
\sum_n \sum_{m'} c_0(n)c_0(m'+2k)\cdot 2 < \phi(2t-n), \phi(2t-m') > &= \delta(k) \qquad [\text{using 4.68}] \\
\sum_n \sum_{m'} c_0(n)c_0(m'+2k)\cdot 2 \cdot \frac{1}{2}\delta(n-m') &= \delta(k) \qquad [\text{let } m' = n] \\
\sum_n c_0(n)c_0(n+2k) &= \delta(k) \qquad (4.69)
\end{aligned}$$

In a similar manner, we can use equations 4.66 and 4.67 in equations 4.51 and 4.53 to obtain

$$\sum_n c_1(n)c_1(n+2k) = \delta(k), \qquad (4.70)$$

$$\sum_n c_0(n)c_1(n+2k) = 0. \qquad (4.71)$$

Notice that equations 4.69, 4.70, and 4.71 are just a repeat of equations 4.25, 4.26, and 4.27. This means that sequences $c_0(n)$ and $c_1(n)$ are precisely the same as the low- and high-pass filters that are used in the two-channel orthogonal filter banks. Furthermore, there is a strong connection between the scaling and wavelet functions and the filters $c_0(n)$ and $c_1(n)$ in that one set can be derived from the other. Filters $h_0(n)$ and $h_1(n)$ can be derived from the wavelet function according to (see [74] for derivations)

$$h_0(n) = \sqrt{2}\int_{-\infty}^{\infty} \phi(t)\phi(2t-n)dt, \qquad (4.72)$$

$$h_1(n) = \sqrt{2}\int_{-\infty}^{\infty} \psi(t)\phi(2t-n)dt \qquad (4.73)$$

and the scaling and wavelet functions can be derived from the low- and high-pass filters using an iterative procedure based on equations 4.66 and 4.67 as described in [74]. The iteration has been shown to converge to a continuous wavelet basis function provided that the filter $c_0(n)$ meets a certain *regularity* condition.

4.2.2.2 Computing the Discrete Wavelet Transform Using Filter Banks

Due to the connection between the wavelet transform and the two-channel filter bank, we can now view $c_0(n)$ and $c_1(n)$ in equations 4.66 and 4.67, and $c(m,n)$ and $d(m,n)$ in the expansions 4.56, 4.57, and 4.58 as digital filters and digital signals, respectively [74]. This means that we now have a more practical way of computing the DWT through a filter bank implementation without even requiring $\phi(t)$ and $\psi(t)$. The approximation and detail coefficients at a given level can be derived from the approximation coefficients at the level above using the filtering operations given by [74]

$$c(i,n) = \sum_j c_0(j-2n)c(i+1,j), \tag{4.74}$$

$$d(i,n) = \sum_j c_1(j-2n)c(i+1,j). \tag{4.75}$$

Inversely, the approximation coefficients at a given level can be derived from the approximation and detail coefficients at the level below according to

$$c(i+1,n) = \sum_j c(i,j)c_0(n-2j) + \sum_j d(i,j)c_1(n-2j). \tag{4.76}$$

Equations 4.74 and 4.75 represent the forward DWT and equation 4.76 represents the inverse DWT. Figure 4.6 shows the implementation of the finite L-resolution DWT in terms of a tree structured filter bank. In the analysis stage, the input signal $x(n)$ is first fed into the filter bank at resolution level L where we make

$$c(L,n) = x(n). \tag{4.77}$$

The low- and high-pass filters are given by

$$h_0 = c_0(-n) \tag{4.78}$$

$$h_1 = c_1(-n), \tag{4.79}$$

where filters $c_0(n)$ and $c_1(n)$ are time-reversed in order to make 4.74 and 4.75 agree with the convolution operation. The outputs from the filtering operations at level L are down-sampled by a factor of 2 and we obtain the approximation and detailed coefficients $c(L-1,n)$ and $d(L-1,n)$ at level $L-1$. This procedure is then repeated for levels $L-1$ down to 1. In the synthesis stage, the reverse operation is applied where we

start at resolution level 0 and successively apply equation 4.76 in order to obtain the approximation coefficients at one level above until we get to level L. The synthesis filters are given by the inverse of the analysis filters (according to orthogonal filter bank constraint in section 4.1.2 and according to equation 4.76) as

$$h_0 = c_0(n), \tag{4.80}$$

$$h_1 = c_1(n). \tag{4.81}$$

Finally at level L, we obtain the reconstructed signal according to

$$\hat{x}(n) = c(L, n), \tag{4.82}$$

which in general is a delayed version of 4.77 due to the filtering operations.

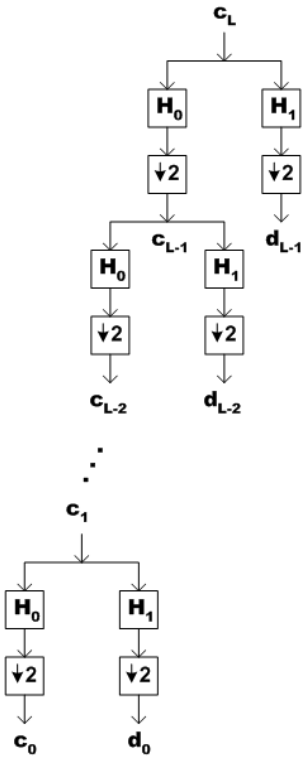


Figure 4.6: Equivalent filter bank structure of the DWT

4.2.2.3 Summary and Discussion

The connection that exists between orthonormal wavelets and orthogonal filter banks can be extended to general biorthogonal wavelets and PR linear phase filter banks [68], and even IIR filters if we wish to have non-compactly supported wavelet basis functions [69]. In general, there is an implied equivalence between two-channel PR filter banks and dyadic DWT as one leads to the other with proper design.

However, we make a note here of some inherent differences that still exist between the wavelet analysis in the continuous-time domain and the finite L-resolution DWT that is implemented in the discrete-time domain. Going back to the derivation of the DWT, recall that we made an assumption in 4.55 by projecting the input signal $x(n) \in L^2(\mathbb{R})$ onto a finite resolution subspace V_L, where the approximation coefficients were defined by

$$c(L,n) =< x(t), \phi_{Ln}(t) >= \int x(t)\phi_{Ln}dt, \tag{4.83}$$

as well as an assumption in 4.77 where we further simplified the definition of approximate coefficients using

$$c(L,n) = x(n). \tag{4.77}$$

Since wavelet analysis was originally defined in the continuous domain, these assumptions were required to extend the analysis to the discrete domain. Equations 4.83 and 4.77 are approximation steps that introduce some discrepancies between the continuous and discrete case. Specifically, the simplification in equation 4.77 does not always hold since the inner product between ϕ_{Ln} and $x(t)$ is not the same thing as the time-sampling of $x(t)$ [73].

4.3 Design of the Wavelet Filter Bank

The design of the Wavelet Filter Bank (WFB) and different existing wavelet schemes are considered in this section. The WFB is a filter bank that offers a great deal of flexibility in terms of the choice of the basis filter and the decomposition tree structure. Additionally, the WFB offers a variety of ways of handling boundary artifacts in the context of block processing. The following sections describe the design of the WFB in terms of these broad design "parameters".

4.3.1 Decomposition Tree Structure

As already shown in Figure 4.6, the standard DWT involves a dyadic tree structure in which the low-channel side is successively split down to a certain depth. We obtain the detail coefficients from the right-leaf node of each level and the approximation coefficients from the left-leaf node at the lowest level. This is illustrated in Figure 4.7(a) where the nodes represent the wavelet coefficients (at various decomposition stages) and the left and right branches represent the low- and high-pass filtering operations, respectively. If we allow the tree to also split on the right-hand side at each node, then we obtain the more general Discrete Wavelet Packet Transform (DWPT) as shown in Figures 4.7(b) and (c). In terms of the wavelet basis function (in continuous domain), the additional degree of freedom in the DWPT comes in the form of frequency, where the standard DWT uses wavelets that are shifted and dilated in time, but the DWPT uses wavelets that can also be modified in terms of the number of oscillations in the basis function.

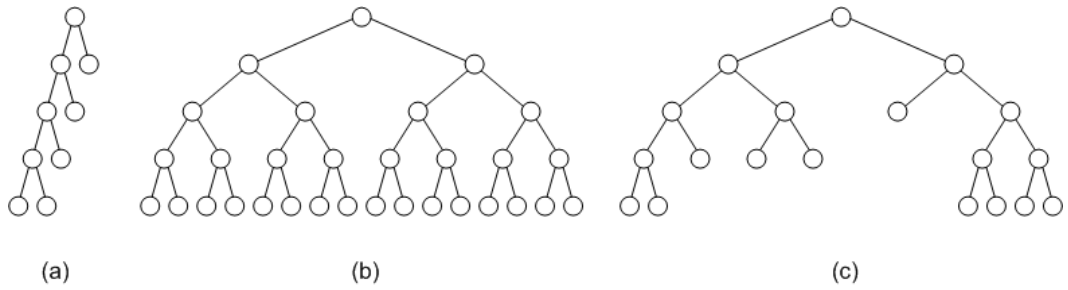


Figure 4.7: Decomposition tree structure for (a) DWT (b) DWPT (complete) (c) DWPT (partial)

From a filter bank point-of-view, the DWT and DWPT represent particular tillings of the time-frequency plane as shown in Figure 4.8(b) and (c). Figure 4.8(a) and (d) also show the time-frequency tiling of the time-domain representation and the Fourier Transform (FT) as well. We can see that at one extreme, time domain provides good time resolution but poor frequency resolution (actually none) and at the other extreme, the Fourier domain provides good frequency resolution but no time resolution. The DWT and DWPT can be thought of as providing a trade-off in resolution between these two extremes.

In Figure 4.9, the time-frequency tiling of the short-time Fourier Transform (STFT) and a fully decomposed DWPT (of depth 2) are shown. This figure shows that the two

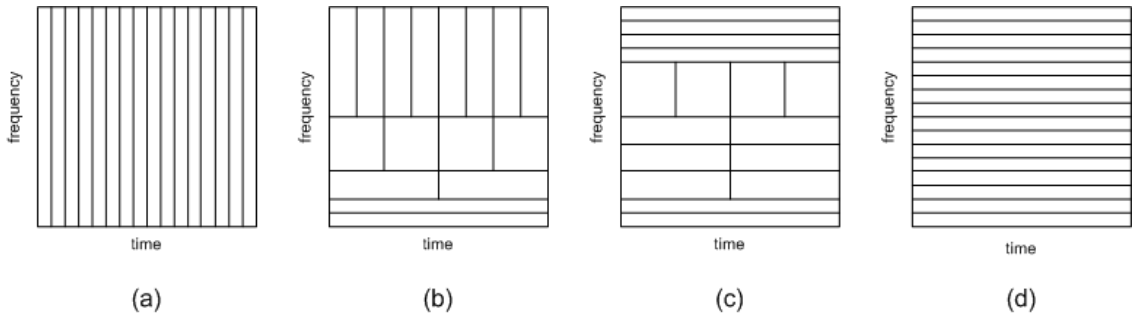


Figure 4.8: Time-frequency tiling of (a) time-domain representation (b) DWT (c) DWPT (partial) (d) Fourier representation

provide the same tiling of the time-frequency plane even though the two differ considerably in their definitions. This is because the tilings in the time-frequency plane represent an idealized situation where each tile is perfectly localized, i.e. there is no overlap between adjacent tiles. The localization property of each tile is actually never ideal (since we are using practical filters) and the localization properties of the DWPT and the STFT are quite different. In particular, the STFT is a modulated filter bank that provides a high frequency resolution representation and the DWPT is an iterated filter bank that provides a flexible time-frequency resolution but suffers from poor localization properties. In general, the localization property of each tile is determined by the choice of the basis filter in the case of DWPT and the window function in the case of STFT.

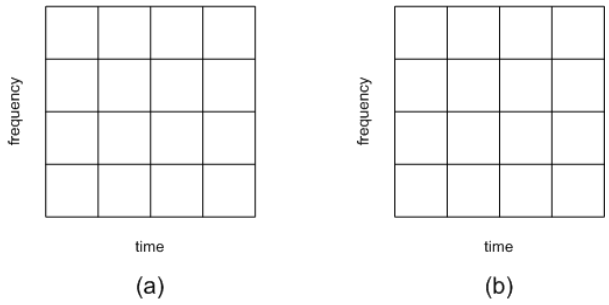


Figure 4.9: Time-frequency tiling of (a) STFT (b) DWPT (Walsh)

4.3.2 Wavelet Basis Filters

The choice of the basis function, or equivalently the wavelet filter, determines the time-frequency localization properties of the individual bands of the WFB as mentioned above. In general, we can also talk about the basis function in terms of other properties such as time-support, number of vanishing moments, various notions of smoothness and regularity, and orthogonality or biorthogonality. In terms of filters $h_0(n)$ and $h_1(n)$, we also have properties such as the transition bandwidth (cut-off rate), stop-band attenuation, passband and stopband ripples, and phase linearity (equivalent to biorthogonality). And in terms of statistical properties, we can use various measures of entropy or coding gain to determine the redundancy extraction that a WFB provides using various basis filters (and tree structures).

There are many existing wavelet basis filters that have appeared in the wavelet literature where each wavelet or family of wavelets is designed to possess certain properties, e.g. high regularity or large number of vanishing moments [37]. Furthermore, there have been many filter design methods developed in the filter bank literature that allow us to construct new wavelet filters based on a variety of design criteria. In general, these design methods revolve around the design of FIR PR-QMF filters and can be divided into three groups [67]. The first group is based on the design of half-band filters (as described in sections 4.1.2 and 4.1.3) followed by spectral factorization, the second group is based on the design using lattice structures that are associated with efficient implementations, and the third group is based on the formulation of the problem in the time-domain and solving it using an optimization algorithm (see [67] for more detail).

4.3.3 Other Wavelet Analysis

In addition to the many choices we have with regards to the basic WFB, i.e. DWPT with the choice of a tree structure and basis filter, other variations and extensions of wavelet analysis have appeared in the literature [70]. For example, basis filters that we usually consider are FIR filters that are either orthogonal or biorthogonal, but we can also employ basis filters that have an infinite support (e.g. IIR filters), that are over-complete and redundant (e.g. Malvar wavelets), or even basis functions that are based on more than one mother wavelet (e.g. *multiwavelets*). Furthermore, algorithms that try to adapt or optimize the choice of the decomposition tree structure and/or the basis filter to a given signal have also been proposed, e.g. *Best Basis*, *Matching Pursuit*, and *Basis Pursuit* [70]. In the context of filter banks, the definition of the WFB has also been

extended to include schemes such as time-varying filter banks and M-channel filter banks [35]. Although some of these analysis schemes offer interesting possibilities, they are not further treated in this thesis.

4.3.4 Boundary Handling

Boundary handling is an issue that arises in practical implementations when we apply the wavelet transform to a finite-length signal. Since the WT is implemented through convolutions between the basis filters and the input signal at each level of the decomposition, a way of properly convoluting at the boundaries is required. General treatment of boundary handling has been covered in [76] and boundary effects that occur in the context of audio coding have been described in [77]. Some techniques for minimizing or eliminating boundary artifacts are briefly described next.

Common methods of boundary handling are:

1) Zero-padding: The signal is padded with enough leading and trailing zeros required to complete the convolution.

2) Symmetric-extension: The signal is symmetrically extended at both ends with the mirror image of the signal.

3) Circular- or Periodic-extension: The signal is periodically extended at both ends, or equivalently, the filter is made to wrap around once it reaches the end of the signal.

Since perfect reconstruction and critical sampling are desirable properties for a filter bank, periodic-extension is typically the method of choice as the other two suffer from slight redundancies when PR is imposed. However, it has been found that periodic-extension suffers from boundary artifacts that spread from one end of the frame to the other end in the context of block processing. Figures 4.10 shows an example of an audio signal that has been encoded with a wavelet audio coder (see Appendix A for detail) based on a WFB with periodic-extension. We can see that the reconstructed signal in Figure 4.10(b) contain some coding artifacts that appear at the frame boundaries as a result of quantization noise spreading from one end of the frame to the other. In listening tests, these boundary artifacts have been perceived as “clicking” noises that were found to be rather objectionable.

An alternate boundary handling scheme was proposed in [77] where a way of performing the WFB analysis without introducing “artificial” boundaries between frames,

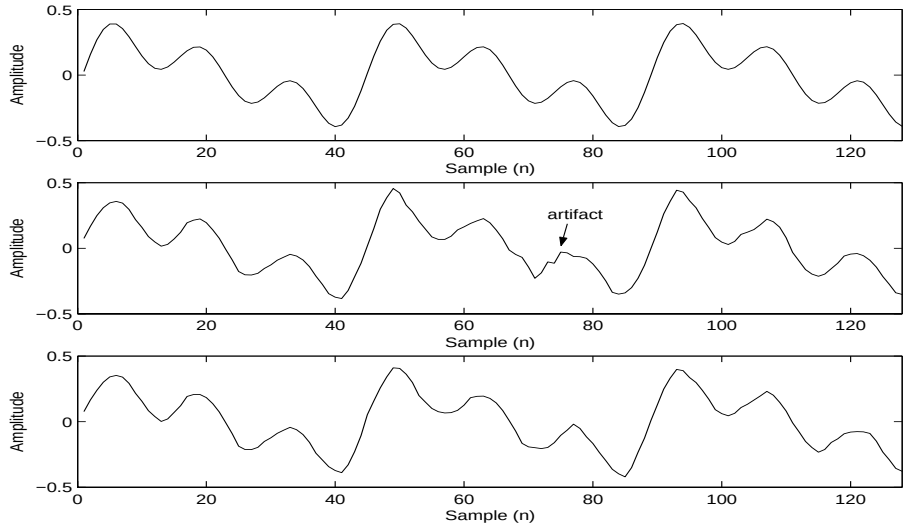


Figure 4.10: Example of a signal encoded with a Wavelet Audio Coder (a) original signal (b) reconstructed signal using circular-extension (c) reconstructed signal using transparent-extension

while maintaining critical sampling and PR, was developed. The scheme essentially involved taking samples from the preceding and proceeding frames so as to make the frame boundaries seem "transparent" when convolution was performed at each stage of the tree structure. An example of this scheme applied to the same signal of Figure 4.10(a) is shown in Figure 4.10(c). The drawback to this scheme was increased complexity and longer delays where the delay was found to be

$$\text{Delay} = (2^d - 1)(L - 2), \tag{4.84}$$

where d = depth of tree and L = length of filter. Another widely used strategy in audio coding for minimizing boundary artifacts is to use an overlap-add window for each frame before the filter bank is applied. This method provides a relatively simple way of minimizing boundary artifacts at the cost of introducing some redundancy. An example of this scheme appears in [34] where a WFB bank with periodic-extension was used in conjunction with an overlap-add window.

4.4 Summary

This chapter presented the theory of the wavelet filter bank (WFB) and showed its connection to the theory of multirate filter banks. In particular, it has been shown that there was an implied equivalence between the two-channel PR filter bank and the dyadic DWPT. As a result, the WFB can be implemented through efficient algorithms that are based on cascaded filter banks and, additionally, new wavelet basis filters can be designed using design methods based on PR-QMF filters.

Furthermore, the WFB has been found to provide a great deal of flexibility through its decomposition tree structure and basis filter. This flexibility can be seen as a way of providing a flexible tiling of the time-frequency plane where the tree structure controls the overall resolution and the basis filter controls the localization properties of each tile. This flexibility in resolution and localization of the WFB are both important features in the context of audio coding and are further considered in the next chapter.

Chapter 5

Wavelets in Perceptual Audio coding

The relatively successful application of wavelets in image coding has led some to also investigate its potential usefulness in audio coding. Many types of wavelet audio coding schemes have been proposed, as a result, and various results have indicated that the WFB provides an interesting and potentially useful way of representing and coding audio signals.

In image coding, the wavelet transform (WT) has been found to be a natural and well suited representation that efficiently captures the important details of an image. The WT captures these details through a multiresolution analysis that can "see" the details or changes in an image signal at various resolutions and at various spatial locations so that lack of change, i.e. small coefficients, can be safely discarded. Image signals are generally characterized by having many regions that contain little or no change, and some regions, especially around image "boundaries", that contain the important changes. These clustered distributions of signal details make the wavelet basis functions, which are localized in space, particularly suitable for analyzing image signals. More generally, the WT has been found to be well suited for analyzing signals that contain abrupt changes, non-stationarities, and points of discontinuities [70].

In audio coding, the signals we encounter are generally made up of many quasi-stationary regions and some non-stationary or transient-like regions (section 2.2.2). Furthermore, significant changes generally occur at all resolutions, i.e. throughout the entire spectral range, and throughout a resolution, i.e. throughout the entire temporal range. Therefore, the coding strategy used in image coding can not be as readily applied in au-

dio coding. A similar strategy, however, might involve adapting the decomposition tree structure and the basis filter so that energy concentration in the wavelet domain can be optimized. This can be thought of as a statistically based coding scheme that tries to maximize (or minimize) the spread of the signal in the wavelet domain according to some statistical measure. Actually, statistically based wavelet audio coders have already been proposed (section 5.1.1) where they were found to provide only modest performances compared to perceptually based schemes. Perceptual coders (as described in chapter 2), on the other hand, rely on a perceptual criteria that is computed by a psychoacoustic model so that coding distortion can be shaped in a way that minimizes perceived distortion. Applying a perceptual criteria to a WFB means that the choice of the tree structure and basis filter, and the way the wavelet coefficients of each band are encoded, need to first consider the perceptual requirements of the audio signal and the characteristics of the human auditory system (section 2.2.1). Moreover, unlike image coding where we think of the wavelet decomposition as a multiresolution analysis, perceptual audio coding treats the WFB as a filter bank that provides a time-frequency domain representation of the input signal.

This chapter examines the application of the WFB in the context of perceptual audio coding. First, an overview of existing wavelet audio coders is given, with particular emphasis on the choice of the tree structure and basis filter. In so doing, we discover that there are some important issues that have not been adequately discussed in the literature, namely, the time and frequency localization (as opposed to resolution, as will be explained) properties of the WFB, as well as the ordering of the subbands in the frequency domain. These represent fundamental issues that need to be addressed when we apply the WFB to a perceptual coding scheme and are, therefore, described in some detail. The frequency localization property of the WFB, in particular, seems to be rather poor when filters are iterated to obtain a subband analysis scheme. One method for improving the channel selectivity in the WFB is explored by making use of a filter design technique that provides some design flexibility. A number of conclusions are drawn from this study and an implementation of a wavelet audio coder is used to compare the newly designed filters with other well known wavelet filters. The sections of this chapter are briefly described as follows. Section 5.1 gives an overview of existing wavelet audio coders, section 5.2 describes some fundamental issues involved in using the WFB in a perceptual audio coder, section 5.3 explores one technique for improving the frequency domain localization property of the WFB, section 5.4 discusses some preliminary results using the proposed technique, and section 5.5 ends with a summary and some concluding remarks.

5.1 Overview of Wavelet-Based Audio Coders

A number of audio coders based on the WFB have been proposed over the past decade in order to demonstrate the feasibility of such a scheme and to explore various configurations that lead to a better design. A brief description of several examples as well as a summary of design approaches for the decomposition tree structure and wavelet basis filter is given next.

5.1.1 Examples of Wavelet Audio Coders

One of the earliest examples of a wavelet audio coder was proposed by Wickerhauser in [78], where the well known Best Basis algorithm was used. The wavelet analysis was done by selecting the "best" tree from a library of tree structures through the use of a simple entropy criterion. The resulting decomposition provided many coefficients that fell below a certain threshold, where such coefficients were simply discarded so that coding requirements were reduced. Furthermore, it was found that Huffman coding in the wavelet domain provided better performances than applying Huffman coding in the time domain, indicating that the wavelet transformation did provide a good decorrelation property. The proposed coder was tested using speech signals only and results indicated that the algorithm provided modest compression ratios of between 2 and 3. Other non-perceptually-based wavelet audio coders have also followed, e.g. [79, 80, 81, 82, 83], but were generally found to provide lower performances compared to the perceptually-based audio coding schemes.

The first extensive study using a perceptually based scheme was done by Sinha and Tewfik in [34]. The coder that they proposed was comprised of two parts, namely, a perceptual part and a dynamic dictionary part. The two were designed to work in conjunction so that one removed the perceptual irrelevancies and the other removed the statistical redundancies. Only the perceptual part will be described here. The perceptual part consisted of a wavelet filter bank, a frequency-domain masking model, and a bit allocation and encoding stage much like the perceptual coder described in chapter 2. The WFB was based on a fixed 29-band CB resolution tree structure and an adaptive basis filter. The filter selection was done by computing the bitrate required for perceptual transparency with each filter from a library of basis filters, and choosing the filter that provided the best performance. The filter library was limited to wavelets with the maximum number of vanishing moments, e.g. Daubechies, which only differed in their phase responses. Filters with the maximum number of vanishing moments were indicated

as being "near optimal" among different classes of filters. Boundary artifacts were minimized by dividing the audio signal into overlapping frames, where the ends of each frame were weighted by a hanning window with an overlap of 128 samples. For filter convolution, periodic-extension was used. The psychoacoustic model was based on frequency domain masking only and was designed in a similar way to MPEG Psychoacoustic Model 2. The bit-allocation and quantization stage was by far the most complex part of the algorithm since it performed the optimization procedure that selected the basis filter for each frame. The bit allocation was done by first translating the masking threshold from the Fourier domain, which was the domain used by the psychoacoustic model, to the wavelet domain and then determining the quantization noise that was allowed in each wavelet band for maintaining perceptual transparency. When translating the masking levels from the Fourier to the wavelet domain, this algorithm made an explicit simplification where out-of-band components that appeared outside of the frequency support of each wavelet band were neglected [34, p. 3469]. The quantization was done using a simple adaptive scalar quantizer. Additionally, a pre-echo protection method was used by adaptively switching the frame size between 2048 and 1024 samples depending on the time-domain characteristics of the input signal. Results indicated that the proposed coder (perceptual part alone) provided "almost transparent" coding at 64-70 kbps.

Other variations on Sinha and Tewfik's wavelet audio coder have also appeared and a few are mentioned here. In [84], Black and Zeytinoglu proposed a simpler wavelet coder based on the same fixed CB tree structure as [34] but using a fixed 16-tap Daubechies filter. The psychoacoustic model employed the output from the WFB stage rather than re-computing a Fourier domain representation, which was essentially less accurate but also less computational load. The coding quality of the algorithm was reported to be comparable to MPEG-1 Layer I algorithm, which provided near-transparency for bitrates above 128 kbps [85]. Other wavelet audio coders that tried to use a wavelet analysis inside the psychoacoustic model appeared in [86, 87, 88]. These audio coders have reported encoding rates that ranged anywhere between 70 and 110 kbps. In [89], Leslie and Sandler proposed a coder that used a fixed uniform 32-band tree structure, similar to the Polyphase filter bank that was used in MPEG-1 Layer I and II algorithm, and a fixed Daubechies filter. Listening results indicated that the coder was comparable to the MPEG-1 Layer I coder even though the frequency localization property of the WFB was found to be poorer than that of the Polyphase filter bank.

In [90], Srinivasan and Jamieson proposed a wavelet audio coder with a fixed basis filter and an adaptive tree structure. The MPEG-1 Psychoacoustic Model 2 [16] was

used for computing the masking threshold. The algorithm essentially tried to adapt the tree structure to match the frequency resolution of the resulting masking threshold while satisfying some computational constraint. This procedure worked as follows. An iterative search was done at each level of the tree to find the nodes that could provide some savings in bits by further splitting them, while an accumulator was used to keep track of the overall computational requirement. For each node that provided a savings in bits, if its computational requirement did not make the overall computational requirement exceed a given computational constraint, then that node was made to split and the accumulator was updated. This was repeated for each level of the tree until all such nodes were found or until all computational load was drained. The idea behind the splitting of a node was that a subdivision of a wavelet band sometimes provided a better match between the subband and the corresponding masking resolution (section 2.2.1), which could result in a more efficient usage of bits. This, of course, depended greatly on the resolution of the psychoacoustic model, which was designed to provide a resolution of 63 1/3-octave bands. Normally, this procedure would produce the same type of tree structure for a given computational constraint (since the masking resolution remained the same) so an additional temporal constraint was included so as to allow the tree structure to vary according to the time-domain properties of the input signal. The resulting algorithm was claimed to provide transparent coding at 45 kbps.

In [91, 92], Philippe et al. experimented with a WFB coding scheme that offered a great deal of flexibility in terms of the tree structure and basis filter. This scheme was used to determine how various choices of the tree structure and basis filter affected the overall performance. To do this, they developed an optimization procedure that estimated the bitrate required to encode a signal based on a perceptual criteria (MPEG Psychoacoustic Model I) and this was used determine the required bitrate for a variety of tree structure and basis filter configurations. More specifically, three features of the WFB were tested, namely, the number of subbands, the resolution of the subbands, and the choice of the filter. They found that for the tree structure, a WFB with 16 channels and a critical band resolution provided the optimal performance and for the basis filter, they found that Onno filters (optimal in an AR(1) coding gain sense) provided the best performance. Among the filters included in the test were maximally regular filters and highly frequency selective filters. A wavelet coder based on their findings was proposed and was assessed through a listening test. Results indicated that its performance was comparable to the MPEG-1 Layer II algorithm at 80 kbps, providing "near-transparency" at that bitrate.

Lucent's EPAC algorithm was an example of a commercial audio coder that utilized

a WFB in its decomposition stage [93]. A switched filter bank was used in the decomposition stage to provide an adaptive scheme, where stationary portions of the signal were analyzed with the high-resolution MDCT and non-stationary portions were analyzed with a WFB. The WFB was found to provide better coding performances for transient signals since it provided a CB resolution that allowed better control of time-domain artifacts. The basis filters were chosen to provide good stop-band and transition-band characteristics and some regularity was also imposed, since this was found to provide "attractive characteristics" that compromised time resolution and frequency resolution requirements. The EPAC algorithm was found to provide slightly better results than the MPEG-1 Layer III algorithm at 64 kbps [2].

In summary, the application of the WFB in perceptual audio coding has been shown to be feasible by several proposed coders and bitrates of between 48 and 110 kbps have been reported. We note that the wide range of performances in these wavelet coders can be attributed to the choice of the WFB, but also to the differences in the other stages of the coder, as well as the testing procedure used to carry out the evaluations. As a result, an objective comparison between various WFB strategies is difficult. But we can still analyze the various strategies and determine if any consensus exists among them.

5.1.2 Wavelet Tree Structure

As described in section 4.3, the WFB tree structure determines the time-frequency resolution of the resulting representation and as described in section 2.2, this resolution needs to be matched to the requirements of the signal. Since perceptual coders are first and foremost designed to eliminate perceptual irrelevancies (and then statistical redundancies), the time-frequency resolution needs to be driven by the resolution of the psychoacoustic model. And since common psychoacoustic models provide masking information in the frequency domain with a critical band resolution (section 3.4.3), the design of most wavelet audio coders has also been based on the use tree structures that provide a similar CB resolution, e.g. the wavelet coders in [34, 87] that use a 29-band CB structure, the coder in [94] that uses a 24-band CB structure, and the coder in [92] that uses a 16-band CB structure (see Figure 5.1). Other types of tree structures that have appeared in some wavelet coders include the standard dyadic WT tree structure [95] and the uniform, i.e. Walsh, tree structure [89].

Wavelet coders that are based on adaptive tree structures have also been explored since they can provide a more flexible way of dealing with the time-varying nature of audio signals. The algorithm proposed by Wickerhauser described above is the earliest example

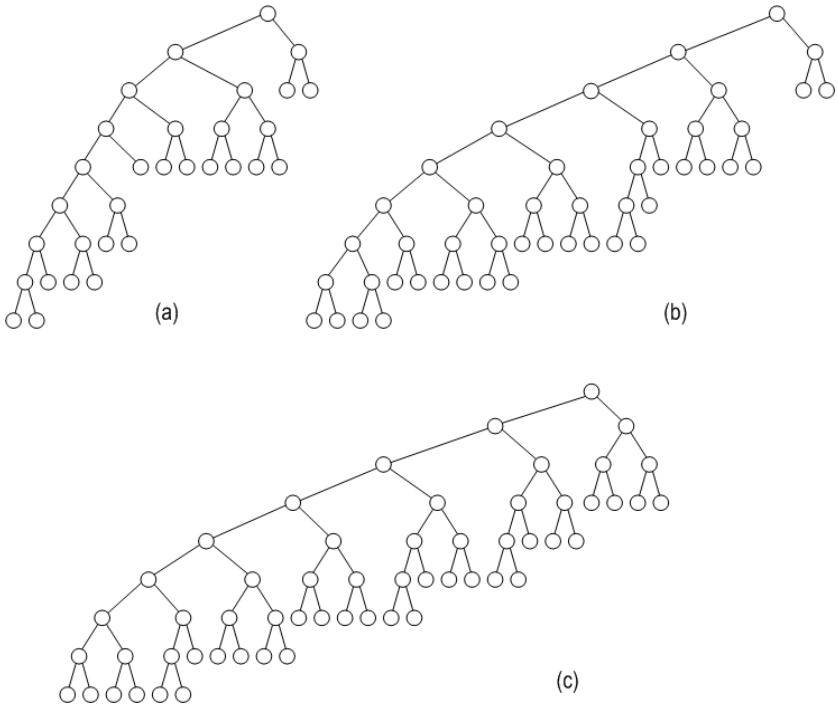


Figure 5.1: Examples of WFB tree structures approximating the CB (a) 16-band (b) 24-band (c) 29-band

(using an entropy-based criteria) and the algorithm proposed by Srinivasan and Jamieson is another example (using a perceptual criteria). A more recent example that appeared in [96] tried to develop an adaptive scheme that took both statistical and perceptual criteria into account, although it was said to be in its developmental stage.

5.1.3 Wavelet Basis Filter

Unlike the tree structure, the choice of the basis filter of a WFB has received much less attention, and even papers that do address it seem to lack any consensus.

Sinha and Tewfik in [34], as described above, have reported some findings from their algorithm during the process of finding the optimal filter for audio coding. They found that filters with the maximum number of vanishing moments provided "near-optimum" results for a given filter length and that longer filters usually provided better results, for lengths of up to 40 and possibly more. Since the algorithm performed a "blind" optimization that selected the filter with the lowest associated bitrate, no clear physical meaning was provided as to why filters with the maximum number of vanishing moments provided

near-optimum results. On the other hand, the increase in performance with longer filters could be explained by the fact that longer filters generally provide better frequency selectivity and even better coding gain. This relationship between filter length and performance was also confirmed in [91, 97, 95], where lengths of up to 32 have typically been considered. Another study done by Kudumakis and Sandler in [95], however, found that four different filter families, namely, Daub-A (minimum phase), Daub-B (maximum symmetry), Johnston filters, and Smith-Barnwell filters, provided very similar performances when assessed with the SSNR measure and with listening tests. Yet another experiment by Philippe et al. in [91, 92] concluded that the coding gain (in AR(1) sense) was the most relevant criteria for selecting the filter, and furthermore, that frequency selectivity (in terms of stop-band attenuation) and regularity (in terms of number of vanishing moments) were less important.

Other wavelet coders that do not really address the selection of the basis filter are usually found to be using Daubechies wavelets, e.g. [84, 88, 94], or biorthogonal wavelets, e.g. [90, 98], with filter lengths that typically lie between 16 and 32.

5.1.4 Discussion

It is interesting to note that in the case of the tree structure, there is somewhat of a consensus, but for the basis filter there seems to be no agreement between the various researchers.

Also, we note that many papers, e.g. [34, 94, 89, 87, 92], have mentioned the poor frequency localization properties of WFB, but none have provided an adequate description of how they arise and how they can be minimized or eliminated, if that is possible to do so. Recall that (section 4.3) the tree structure determines the overall time-frequency resolution, while the basis filter determines the localization properties of the individual wavelet bands. And as discussed in section 2.2.3, the localization properties of each subband is very important in audio coding. This, as a result, motivates us to further explore the localization properties of the WFB, particularly with respect to the choice of the basis filter.

5.2 Audio Representation Using the Wavelet Filter Bank

As already explained above and in section 2.2, the goal of the filter bank is to provide an appropriate representation of the input audio signal with a time-frequency resolution that matches to the characteristics of the input signal. In the context of perceptual coding,

the resolution has to first match the resolution of the masking threshold. Ideally, this resolution would be determined by the psychoacoustic model and vary for each audio frame, but using a fixed-resolution frequency-domain masking model means that:

1) A decomposition with a CB resolution is usually most efficient.

2) If temporal artifacts become important, then they need to be identified by some other means, e.g. over-coding protection (OCP) in PAC [27] or temporal noise shaping (TNS) in AAC [18], since the psychoacoustic model is blind to temporal details within a given frame.

Here, we focus on the first requirement and look at the WFB as a possible solution. In particular, we study the frequency localization properties of the WFB as an iterated filter bank to show that the subbands are not all uniformly shaped and that large out-of-band side-lobes appear for some bands. In addition, the ordering of the wavelet bands in a WFB is shown to be non-sequential and that we need to design the tree structure carefully in order to obtain a subband decomposition that correctly represents the desired CB division. Although the second requirement is not the focus of this study, basic time localization properties of the WFB are also briefly described.

5.2.1 Subband Representation in Frequency Domain

First, consider a wavelet filter whose response is shown in Figure 5.2 and whose scaling and wavelet functions are shown in Fig. 5.3. This is the minimum-phase Daubechies wavelet of length $L = 32$, which is characterized by being orthogonal and having the maximum number of vanishing moments (or zeros) at the Nyquist frequency $w = \pi$ for a given support. As a result, the filters are maximally flat at $w = \pi$ and the stopband response is zero at Nyquist frequency (or close to it in practice). The transition band, however, does not possess a very sharp cut-off rate. The phase response shows that the filters are not linear phase, since orthogonality and linear phase can not be simultaneously satisfied (section 4.1.4).

By taking the given two-channel filter bank and iterating it fully down to a depth of 2, 3, 4, and 5, we obtain uniform 4-channel, 8-channel, 16-channel, and 32-channel filter banks as shown in Figure 5.4. As can be readily seen, the frequency responses of the uniform filter banks are far from being ideal, i.e. channel separation is poor. The shapes of the subband channels are uneven and variable, and become progressively worse for increasing number of channels. There is considerable overlap between some of the

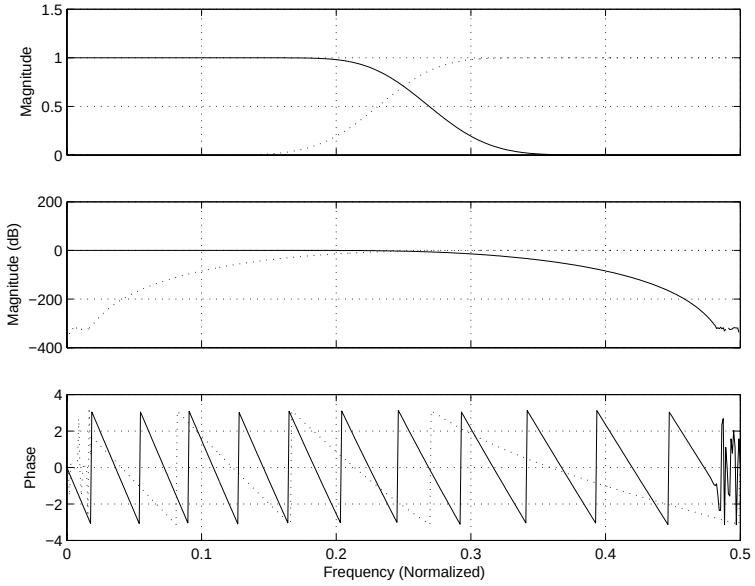


Figure 5.2: Frequency response of Daubechies (minimum-phase) filter of $L = 32$

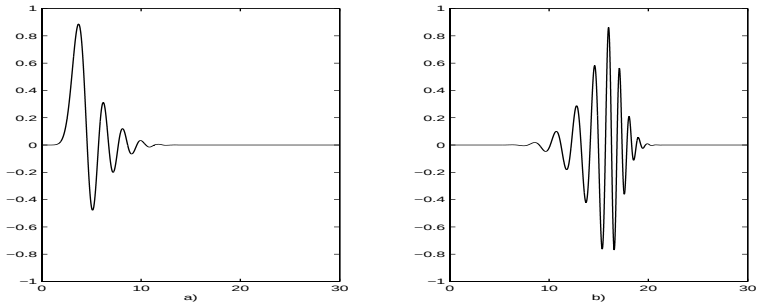


Figure 5.3: Daubechies (minimum-phase) (a) scaling function (b) wavelet function ($L = 32$)

adjacent bands and even between bands that are far apart due to large side-lobes. These side-lobes, which can also be thought of as aliasing components, essentially spread noise from one channel to other channels and can be the cause of much unwanted distortion in audio coding. Furthermore, side-lobes become progressively worse for filter banks with greater numbers of channels, which we can see clearly from the example of a 32-channel filter bank that is shown in Figures 5.5 and 5.6 where the individual channels are plotted separately (only the first 16 bands are shown since the other 16 are symmetrical to the first half). From the figure, we can see that some bands, e.g. bands 6, 9, 12, 13, and 14,

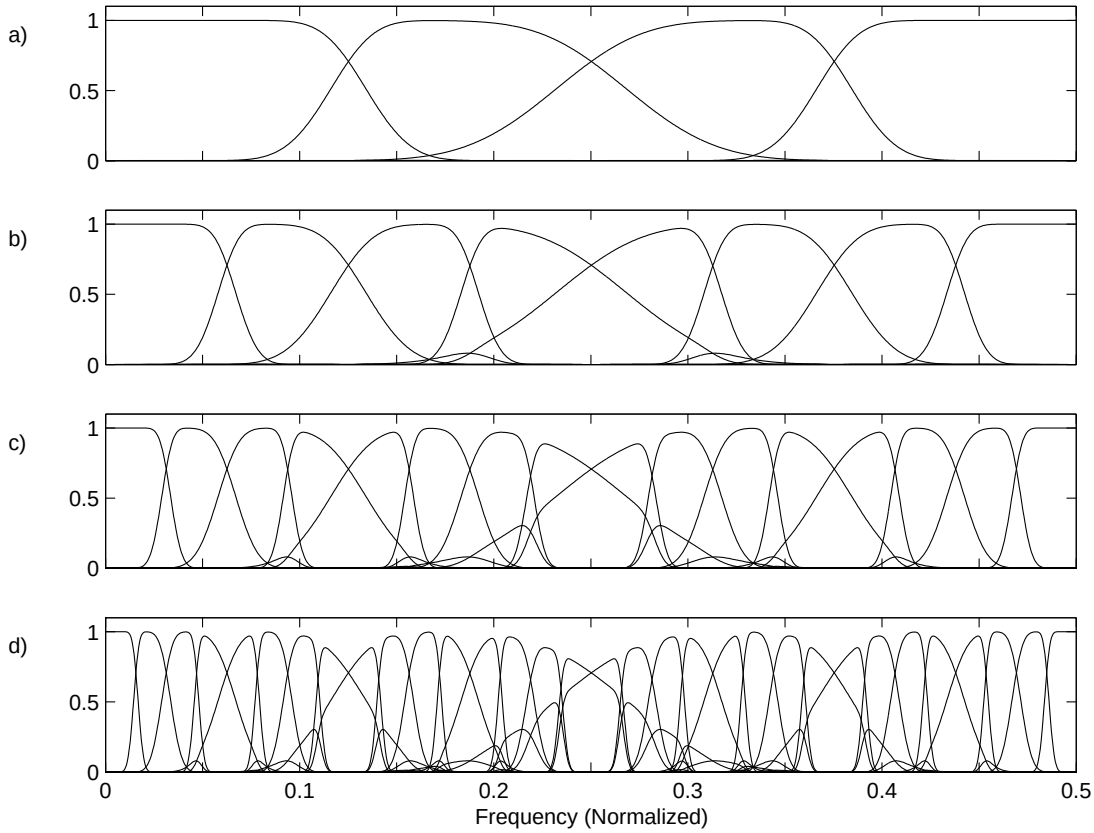


Figure 5.4: Iterated filter banks using Daubechies filters (of Figure 5.2) with (a) 4-channels (b) 8-channels (c) 16-channels (d) 32-channels

are particularly prone to such out-of-band energy components.

To see how these side-lobes arise, Figure 5.7 shows the iteration process (down the tree structure) as it occurs in the frequency domain for band 14, and as a comparison the iteration process for band 15 in Figure 5.8. The channel responses of bands 14 and 15 can be derived by "migrating" all the filters upward through the down-samplers (see Figure 4.6) according to the noble identities [35]. The channel responses are then given by

$$H_{14}(z) = H_0(z)H_1(z^2)H_0(z^4)H_0(z^8)H_1(z^{16}) \tag{5.1}$$

and

$$H_{15}(z) = H_0(z)H_1(z^2)H_0(z^4)H_0(z^8)H_0(z^{16}). \tag{5.2}$$

Figure 5.7 shows plots of depths 1 through 5 (depth 1 is at the top of the tree and depth 5 is

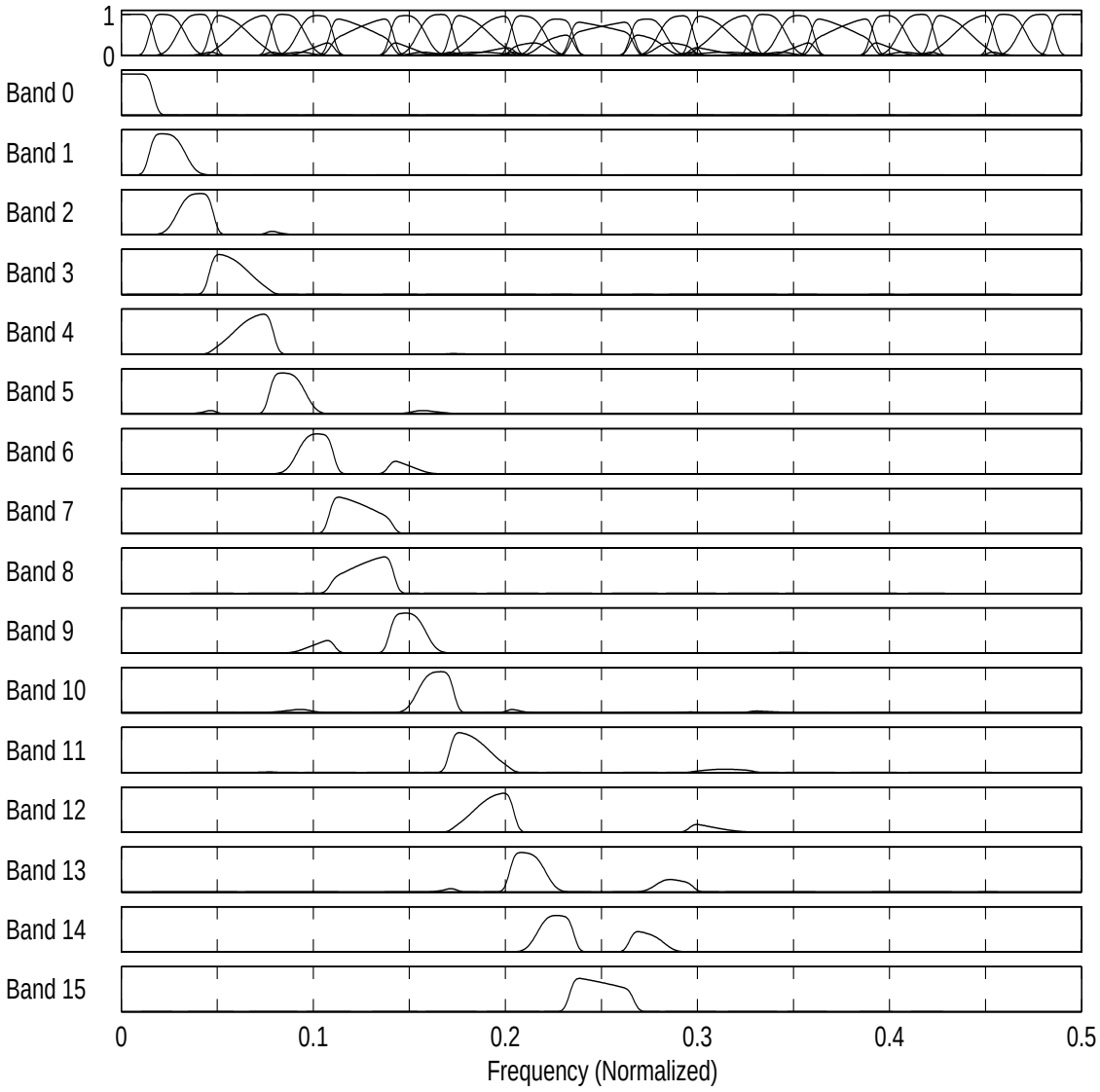


Figure 5.5: 32-channel WFB showing each band individually (linear)

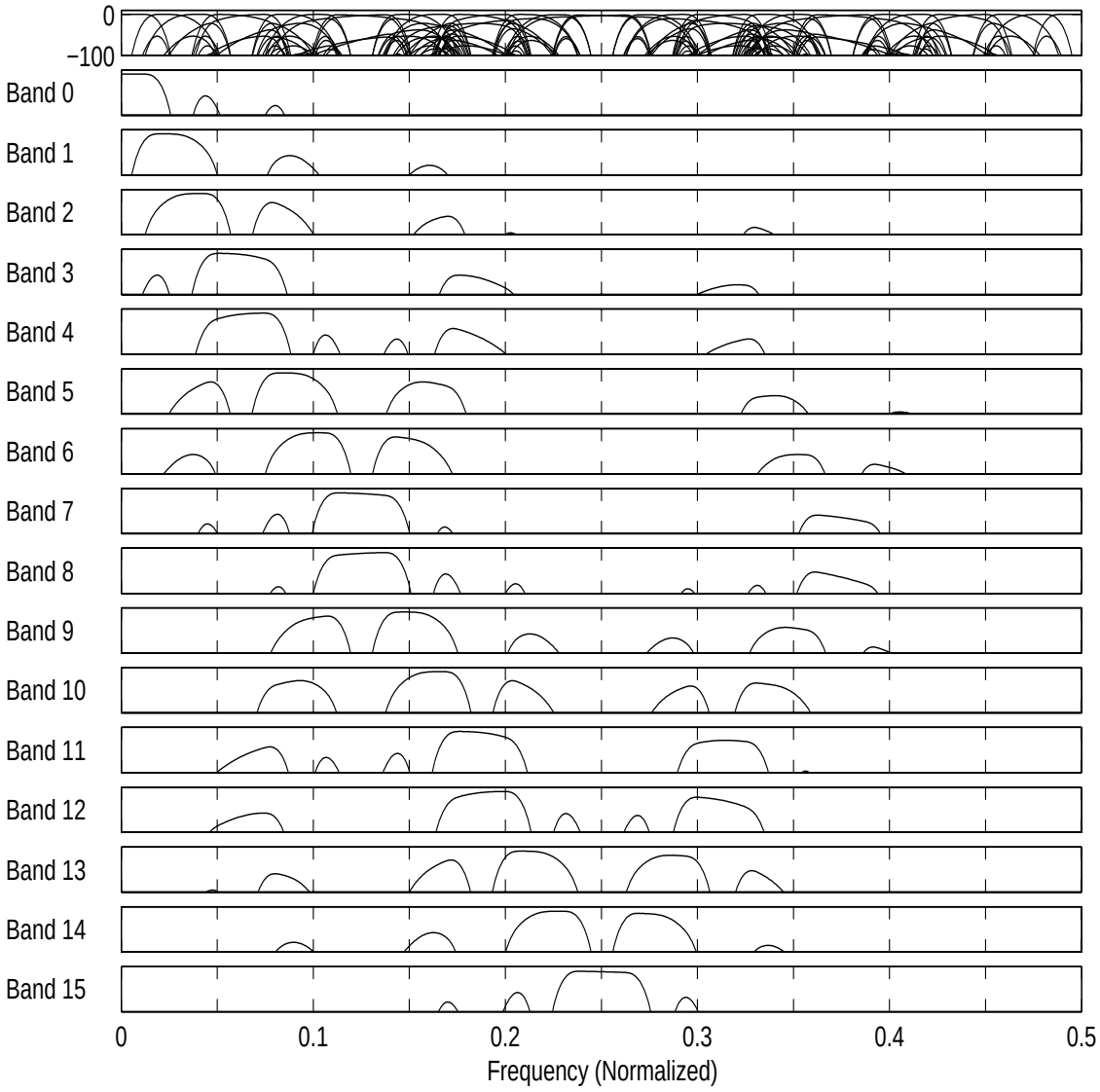


Figure 5.6: 32-channel WFB showing each band individually (dB)

at the bottom of the tree) where at each depth the solid line represents the "accumulated" channel response and the dotted line represents the next filter to be iterated. Iteration in the frequency domain is equivalent to multiplying the two responses (the solid and dotted lines) to obtain the solid line at the next depth. Note that bands 14 and 15 have the same sequence of iterations up until the last depth where the former is iterated with the high-pass filter $H_1(z^{16})$ and the latter is iterated with the low-pass filter $H_0(z^{16})$. Now, looking at this last iteration as shown in depth 4, we see that the accumulated response of band 14 (Figure 5.7) overlaps with two "pass-bands" of $H_1(z^{16})$, where the overlap with the second "pass-band" is what gives rise to the large side-lobe as shown in the depth 5 plot. Similarly, for band 15 (Figure 5.8 we see that the accumulated response only overlaps with one pass-band of $H_0(z^{16})$ and therefore avoids the unwanted overlap. Looking at the sequence of iterations (or multiplications) from one depth to the next, we can see that the side-lobe of band 14 is essentially caused by the large transition bandwidth of filter $H_0(z)$ at depth 1, and furthermore, that the uneven shapes of both bands 14 and 15 are also caused by the poor transition-band characteristics of filter $H_0(z)$ (and $H_1(z)$).

For a given WFB, the shape of each band and the side-lobes that appear will vary depending on the transition-band and stop-band characteristics of the basis filter (and tree structure). But it is clear from the above description that iterated filter banks have an inherent difficulty with providing "clean" frequency separation between subbands, particularly when compared to other filter banks such as the MDCT and pseudo-QMF that are commonly used in audio coding. The presence of such large side-lobes essentially translates into uncancelled aliasing distortions that appear during coding when coefficients are quantized. The way the subbands overlap and spread aliasing errors in this somewhat convoluted way is, in effect, a major drawback in using an iterated filter bank such as the WFB. In perceptual coding, frequency localization and separation are important because (section 2.2.3):

1) The psychoacoustic model provides masking results that only apply to the corresponding subband and any additional quantization error that occurs outside of the band is not guaranteed to be remain transparent.

2) Lots of the audio signals are quasi-stationary, which require good frequency localization.

As a result, these out-of-band aliasing errors need to be eliminated or minimized when designing an audio coder.

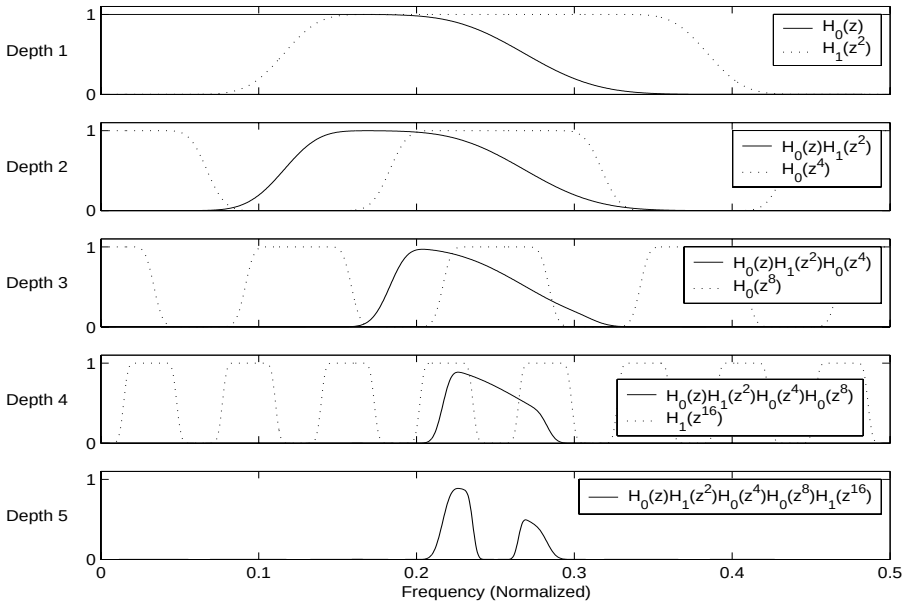


Figure 5.7: Iteration for obtaining band 14 (of a 32-channel uniform WFB)

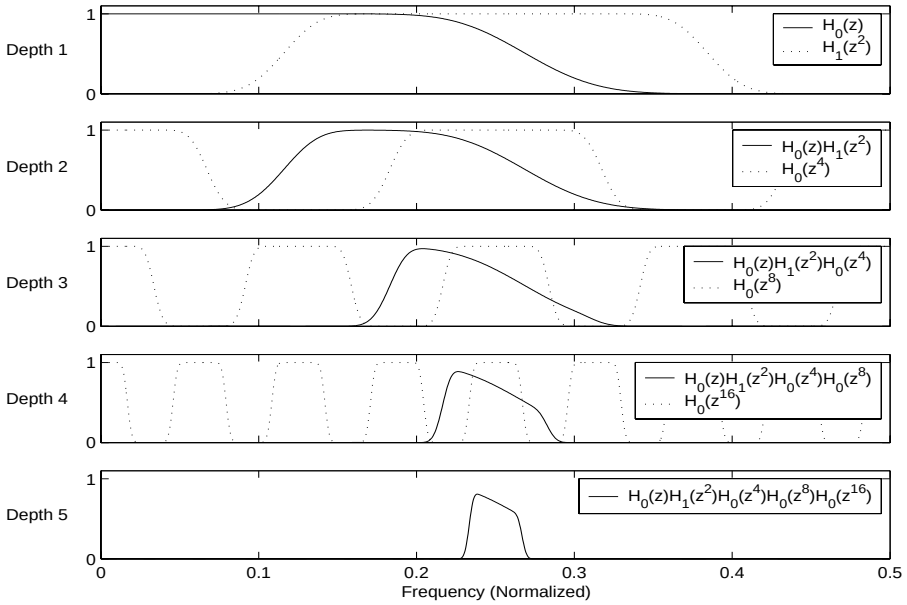


Figure 5.8: Iteration for obtaining band 15 (of a 32-channel uniform WFB)

5.2.2 Coding Examples Using Tonal Signals

The effect of out-of-band energy components in the iterated WFB is further illustrated in Figure 5.9 where tonal, i.e. sinusoidal, signals have been encoded and decoded using a wavelet audio coder described in Appendix A. The WFB uses the $L = 32$ Daubechies wavelet filter and a uniform 32-channel decomposition scheme. The tonal signals are located at the center of each subband. The coding is done by assigning full bits, e.g. 15 bits, to the band where the signal appears and assigning no bits to the remaining 31 bands. This would be equivalent to decomposing the signal and then reconstructing it using only the coefficients from the band where the signal appears. As a result, the reconstructed signal of the given band shows uncancelled aliasing components that appear at the locations where the given band overlaps with other bands. Figure 5.9 and 5.10 shows the original and the reconstructed signals for bands 0 to 15. Note how the distortion for each band is unique in that it reflects the overlapping "characteristics" of the given band. Appendix B also gives references to audio samples of the original and reconstructed signals that are shown in Figures 5.9 and 5.10. As a comparison, Figure 5.11 shows the same 16 sinusoids that are encoded with same audio coder but using an MDCT (with a sine window) instead of a WFB.

5.2.3 Natural Vs. Sequency Ordering of Subbands

First, consider the simple filtering and downsampling operation of the QMF filter bank as shown in Figure 5.12. The input signal $x(n)$ goes through low-pass filter $H_0(z)$ and high-pass filter $H_1(z)$ and then through a down-sampler to produce signals $x_0(n)$ and $x_1(n)$. The filtering operation followed by downsampling can be illustrated in the frequency domain as a multiplication followed by an expansion operation as shown in Figure 5.13. The expansion is done with respect to center frequencies $2\pi m$ for $m \in \mathbb{Z}$, and for convenience the subband responses are shaded to distinguish the left and right regions, i.e. the low and high frequency regions (the shaded region in $H_0(z)$ before down-sampling is the low frequency region and the shaded region in $H_1(z)$ is the high-requency region). Then, for the low-pass filter $H_0(z)$ shown in Figure 5.13(a), the "low-side" before down-sampling becomes the low-side after down-sampling and the "high-side" before down-sampling becomes the high-side after down-sampling. But for the high-pass filter $H_1(z)$ in Figure 5.13(b), the low-side before down-sampling becomes the high-side after down-sampling and high-side before down-sampling becomes the low-side after down-sampling. This essentially means that high-pass filtering followed by down-sampling reverses the frequency

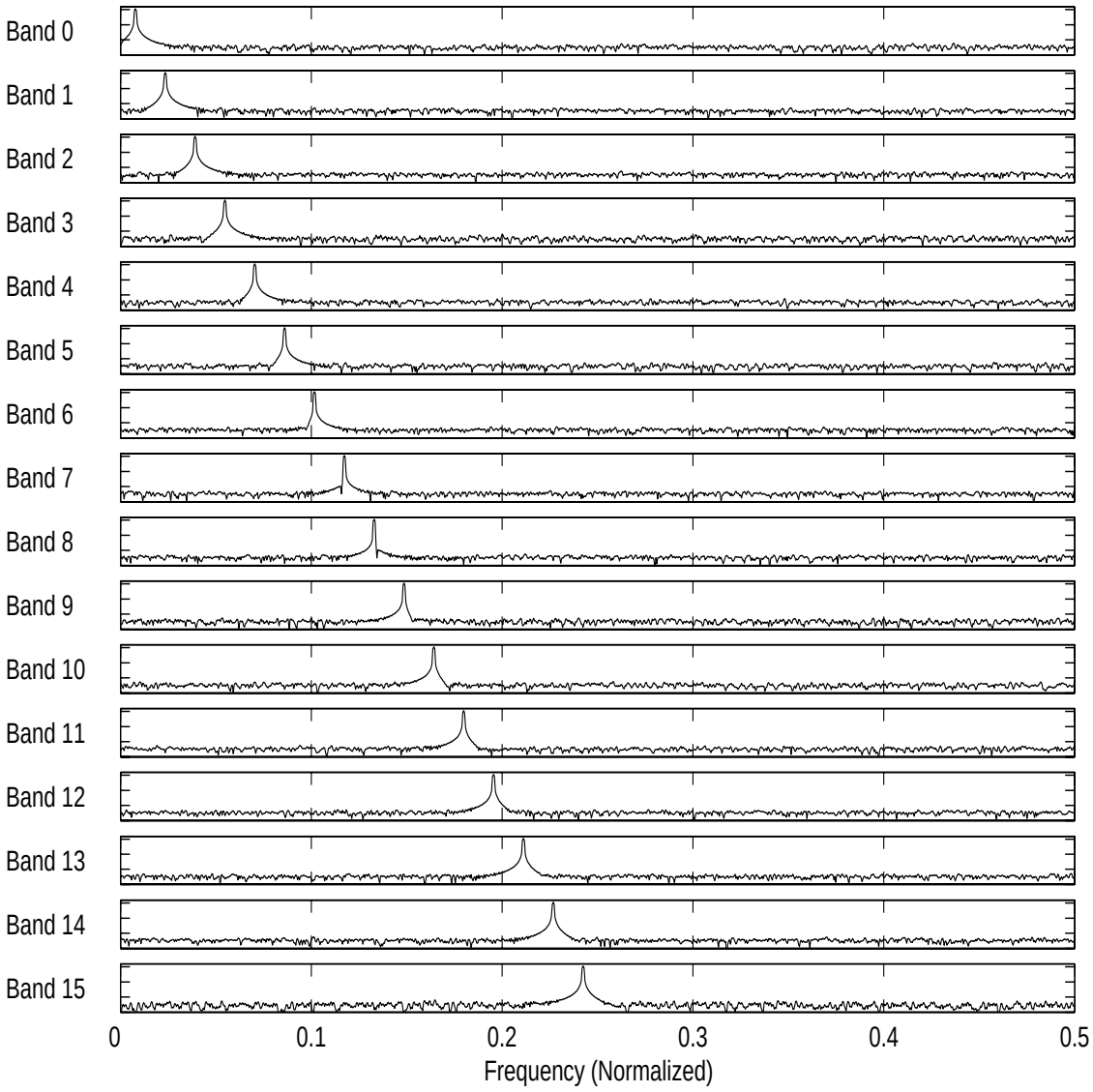


Figure 5.9: Encoding using a WFB: original tonal signals

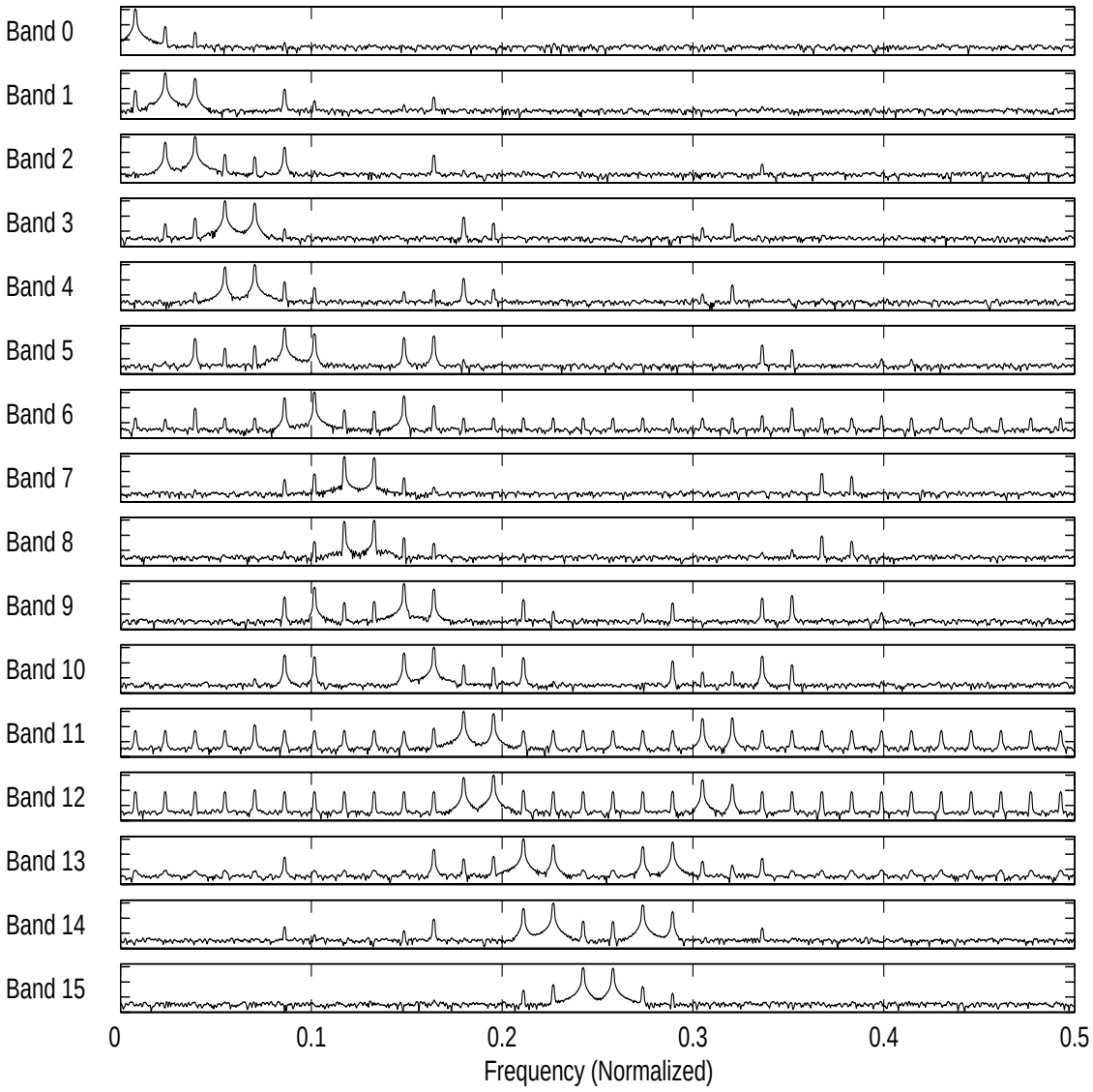


Figure 5.10: Tonal signals encoded by a Wavelet Audio coder

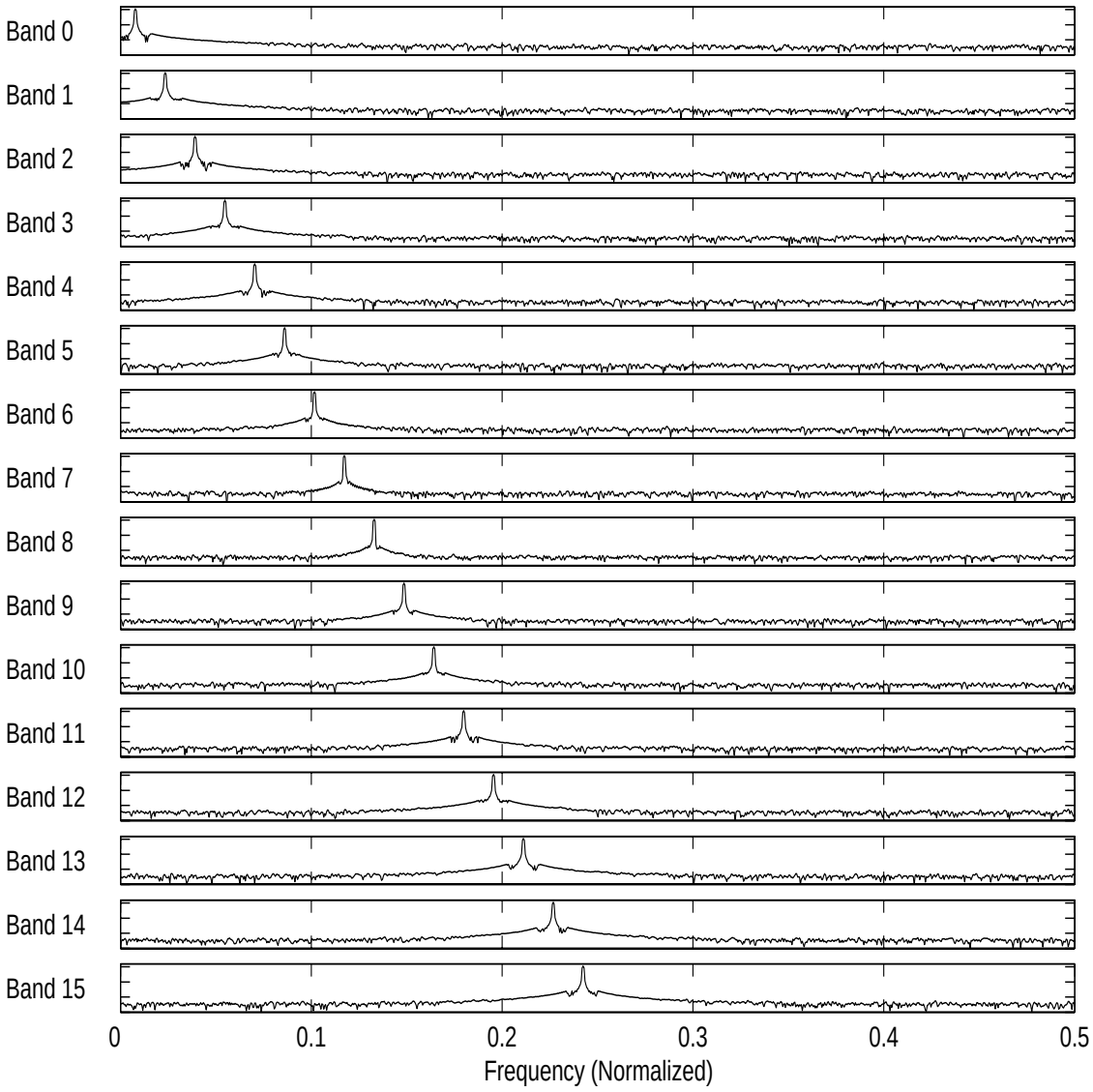


Figure 5.11: Tonal signals encoded by an MDCT Audio coder

order of the resulting signal spectrum.

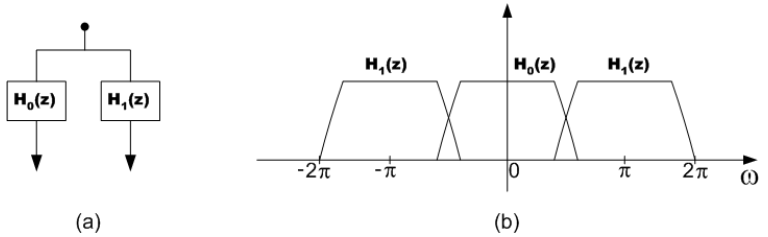


Figure 5.12: A QMF filter bank (a) configuration (b) frequency response

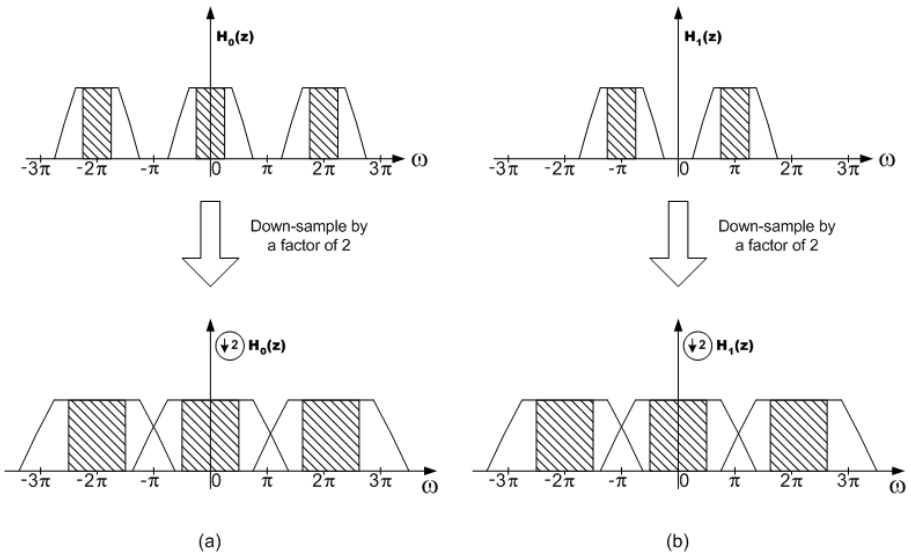


Figure 5.13: Down-sampling of subband channels as expansion in the frequency domain (a) $H_0(z)$ (shaded region is "low-side") (b) $H_1(z)$ (shaded region is "high-side")

For a tree-structured decomposition, a signal may be iterated many times before reaching a leaf node and depending on the path it takes, there many be any number of low-pass and high-pass filtering stages, where each time it encounters a high-pass stage the frequency order of the resulting spectrum is reversed. This means that the leaf nodes of a particular tree structure does not necessarily follow the logical frequency order as we go from left to right, nor does every low-pass branch result in the actual low-frequency side and high-pass branch result in the actual high-frequency side. As an example, a uniform 32-band tree structure is shown in Figure 5.14 where the *natural* and *sequency* ordering [99] of each band (or node) is indicated. Natural ordering represents the node number as

it appears in the tree (at a given level) and sequency ordering represents the band number that we would normally associate with a band if all the bands were sequentially ordered in the frequency domain. Naturally, when we think of subband analysis, we think of each band in terms of the frequency region it occupies and when we design filter banks for audio coding, we expect the bands to reflect a logical sequency ordering in the frequency domain. As a result, we need to apply a decomposition tree structure (in natural order) that reflects the desired sequency order and a way of translating one from the other. An algorithm that provides a WFB decomposition using a sequency ordering specification has been developed by the author and described in Appendix C. Interestingly, this issue never comes up in the standard WT since iteration is only applied on the low-channel side.

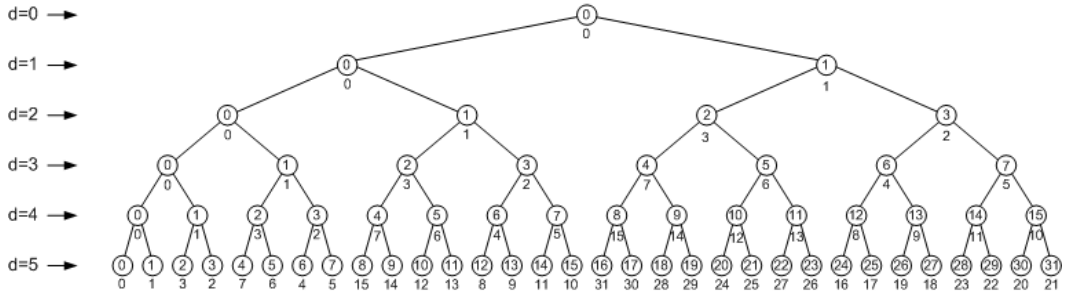


Figure 5.14: Natural and sequency ordering of a uniform 32-channel tree structure. Note, natural ordering is specified inside the node, sequency ordering is specified below the node, and d=depth

5.2.4 Localization in Time Domain

Here we briefly look at the relationship between filter length and time support, or time localization, in the wavelet representation. Since filtering is implemented through a convolution operation, longer filters essentially result in longer convolutions and longer time supports. This is illustrated in Figure 5.15(a) where we try to compute a wavelet coefficient at a tree depth of 2 using a filter with length $L = 6$. The arrays represent the wavelet coefficients at each tree node for depths 1 and 2, and the array at depth 0 represents the input signal with 32 samples. At each iteration, filters h_0 and h_1 are convolved with the coefficients of a given node, which produces two child-nodes of half-length each. This convolution can be thought of as finding the cross-product between the wavelet coefficients and the even-translates of filters h_0 and h_1 as shown in Figure 5.15(b). As a

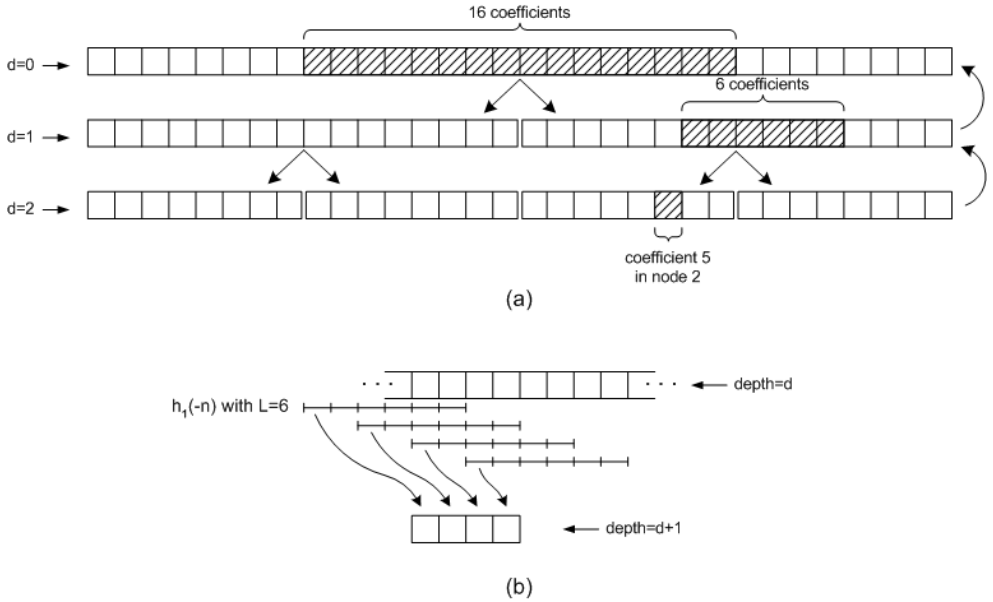


Figure 5.15: The effect on time localization by the WFB tree iteration (a) Time localization of coefficient 5 in node 2 at depth 2 (b) Filtering operation from one level to the next

result, in order to compute coefficient 5 of node 2 at depth 2 (Figure 5.15(a)), we need 6 coefficients from node 1 at depth 1, and similarly, to compute these 6 coefficients of node 1 at depth 1, we need 16 coefficients from depth 0. Thus, we can conclude that we need 16 coefficients from the original signal in order to compute any one wavelet coefficient at depth 2 given a filter length of $L = 6$. More generally, we can derive the number of coefficients we need at depth 0 in order to compute one wavelet coefficient at an arbitrary depth d using an arbitrary filter length L according to

$$T_{\text{support}} = 2^d + (2^d - 1)(L - 2). \tag{5.3}$$

Equation 5.3 essentially represents the time support associated with the wavelet coefficients of a given tree node as a function of tree depth, d, and filter length, L. Some values of these time support have been computed for a number of depths and filter lengths and given in Table 5.1. Note that the ideal time support is given by

$$T_{ideal} = 2^d, \tag{5.4}$$

since down-sampling by 2 at each depth reduces the time-resolution by a factor of 2. From Table 5.1, we can see that only $L = 2$ provides this ideal time localization. We can clearly see that similar to how wavelet bands spread and overlap with adjacent bands in the frequency domain, wavelet coefficients also exhibit this non-ideal behaviour in the time-domain since there will always be some overlap (in time) between adjacent wavelet coefficients. Although the amount of overlap depends on the depth and length of the filter, the way in which they overlap may also depend on the time-domain characteristics of the low- and high-pass filters. For example, if a filter is shaped (in the time domain) so that most of the energy is concentrated near the middle, then its "effective support" can be less than a filter that has its energy more evenly spread-out across its length. Understanding these characteristics can be important in understanding the exact time localization properties of the WFB and could be the topic of further study. But we can use equation 5.3 and Table 5.1 as a general measure indicating the upper bound on the time support provided by wavelet coefficients. From the table, we can already begin to see the trade-offs involved between time localization and frequency localization since ideal time localization requires a filter length of $L = 2$, but this obviously provides poor frequency-domain localization. Conversely, if we start increasing the filter length, frequency-domain localization improves but time-domain localization suffers.

5.3 Minimizing Inter-Band Leakage in the WFB

The problem with inter-band leakages that arises during a WFB decomposition (described above in section 5.2) has been recognized in some audio coding papers, e.g. [34, 92, 94], while most others do not mention it. One possible way of minimizing, or at least dealing with, out-of-band aliasing errors is briefly described in the next section and another method based on the modified Remez exchange algorithm is explored in some detail in the following section.

5.3.1 Method for Minimizing Inter-Band Leakage

One strategy for eliminating or minimizing out-of-band aliasing errors is to calculate the amount of distortion that is introduced into all the bands when one wavelet band is quantized and make sure that the sum of all distortions remains below the masking

Levels (l)	Ideal	L=2	L=4	L=8	L=16	L=32	L=64
1	2	2 (0.05)	4 (0.09)	8 (0.18)	16 (0.36)	32 (0.73)	64 (1.45)
2	4	4 (0.10)	10 (0.23)	22 (0.50)	46 (1.04)	94 (2.13)	190 (4.31)
3	8	8 (0.18)	22 (0.50)	50 (1.13)	106 (2.40)	218 (4.94)	442 (10.0)
4	16	16 (0.36)	46 (1.04)	106 (2.40)	226 (5.12)	466 (10.6)	946 (21.5)
5	32	32 (0.73)	94 (2.13)	218 (4.94)	466 (10.6)	962 (21.8)	1954 (44.3)
6	64	64 (1.45)	190 (4.31)	442 (10.0)	946 (21.5)	1954 (44.3)	3970 (90.0)
7	128	128 (2.90)	382 (8.66)	890 (20.2)	1906 (43.2)	3938 (89.3)	8002 (181.5)
8	256	256 (5.81)	766 (17.4)	1786 (40.5)	3826 (86.8)	7906 (179.3)	16066 (364.3)

Table 5.1: Time support of a wavelet coefficient at depth d with filter of length L. Note: values in parenthesis are duration in ms at 44.1 kHz sampling frequency

threshold for all the bands. This can be expressed as

$$\sum_{i=0}^{M-1} \sigma_i^2 |G_i(w)|^2 \leq T(w) \tag{5.5}$$

for all frequency w where

$$\begin{aligned} \sigma_i^2 &= \text{quantization noise energy in band } i \\ G_i(w) &= \text{response of channel } i \\ T(w) &= \text{masking threshold energy.} \end{aligned}$$

This formulation essentially takes into account the amount of coding noise that is introduced into one band by all the bands (including the given band) and tries to minimize the bit allocation of each band while keeping the overall distortion below the masking threshold. The procedure for obtaining σ_i^2 for each wavelet band is complex and typically done through an optimization process. Examples of this procedure can be found in [92, 94]. Although this method explicitly takes care of the errors associated with all

out-of-band aliasing components, the associated difficulty in designing such a method and the associated increase in bit requirement for some wavelet bands represent drawbacks. Furthermore, if we think of the WFB as a tool that is used for audio coding, then we would expect the tool to "accommodate" to the application and not the other way around.

5.3.2 QMF Design for Minimizing Inter-Band Leakages

Another, rather straight-forward, approach that has not been explored involves designing two-channel filter banks according to conventional filter specifications and determining if overlaps between wavelet bands can be minimized or eliminated altogether. To eliminate the side-lobes, the QMF filters need to have cut-off rates that are sharp enough so that the "multiple band-overlaps" described in section 5.2.1 do not occur when the filters are iterated. Therefore, the transition bandwidth becomes an important criteria in this approach. Generally, QMF filter designs are constrained by the following parameters:

1) Length L

2) Transition bandwidth

3) Stop-band attenuation

4) Pass-band and stop-band ripples

5) Number of vanishing moments

6) Orthogonality or Phase linearity

As already mentioned in 4.3.2, various techniques and algorithms already exist for designing QMF filters that allow one to control some or many of the above design parameters. Here, we focus on one technique proposed by Rioul and Duhamel based on the Remez exchange algorithm that provides orthogonal QMF filter solutions. A short description of this technique is given next.

5.3.3 Modified Remez Exchange Algorithm for Orthogonal QMF filters

The modified Remez exchange algorithm proposed by Rioul and Duhamel [100] provides orthogonal QMF solutions with additional constraints on the filter length (L), transition bandwidth (B), and number of vanishing moments (K). The algorithm also maximizes

the stop-band attenuation level after having satisfied the given requirements. The modified Remez exchange algorithm is based on the design of half-band filters followed by factorization, where the factorized low-pass filter is of the form

$$H_0(z) = (1 + z^{-1})^K Q(z), \tag{5.6}$$

where

$$\begin{aligned} K &= \text{number of vanishing moments} \\ Q(z) &= \text{polynomial with no poles or zeros at } z = -1. \end{aligned}$$

For "wavelet" solutions, filter $H_0(z)$ has to also satisfy the admissibility and the orthogonality conditions [74, p. 73] given by, respectively,

$$\sum_n h_0(n) = \sqrt{2} \qquad \text{and} \qquad \sum_n h_0(n)h_0(n + 2k) = \delta(k). \tag{5.7}$$

The orthogonality condition above is also equivalent to the power symmetry condition in the z-domain (section 4.1.2) and is also referred to as the QMF condition [101]. If $H_0(z)$ is of length L, then the orthogonality condition represents $L/2$ equations (or constraints), which leaves a maximum of $L/2$ degrees of freedom for K. The admissibility condition requires that the zeroth moment exists, i.e. $K \geq 1$, which means that we have

$$1 \leq K \leq L/2. \tag{5.8}$$

The modified Remez exchange algorithm first imposes regularity K on filter $H_0(z)$ and then uses the remaining $L/2 - K$ degrees of freedom to satisfy the transition bandwidth (B) constraint and then to maximize the stop-band attenuation. As a result, regularity, transition bandwidth, and stop-band attenuation represent three competing requirements. Note that for maximum regularity of $K = L/2$, the algorithm provides the Daubechies solution and for minimum regularity of $K = 0$ ("non-wavelet" solution), the algorithm provides the Smith-Barnwell solution (section 4.1.2).

Figure 5.16 shows examples of filters derived from this algorithm. The filters are all designed with $L = 32$ and using various values of B and K. As already mentioned, the filters with $K = 0$ (first column) are Smith-Barnwell filters, while the solution for $K = 16$ (not shown here) results in the Daubechies filter that has already been shown in Figure 5.2. Note that imposing greater constraints on B and K takes away the "degrees of

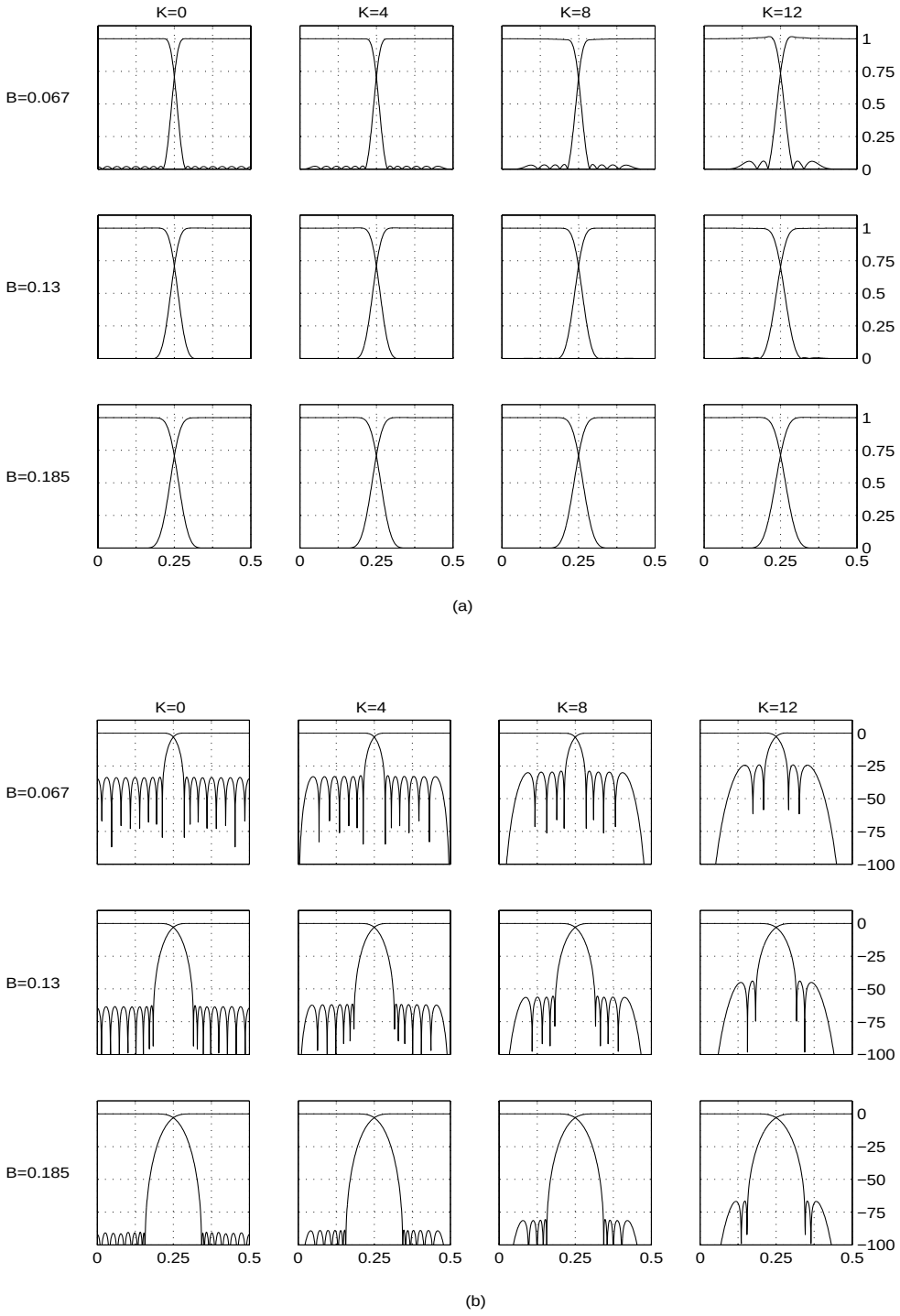


Figure 5.16: QMF filters designed using the modified Remez exchange algorithm with $L = 32$ and various values of B and K (a) linear (b) dB

freedom" for improving the stop-band attenuation, which can be seen in the figure where the filter with the widest bandwidth $B = 0.185$ and smallest regularity $K = 0$ provides the best stop-band attenuation and the filter with the smallest bandwidth $B = 0.067$ and highest regularity $K = 12$ provides the worst stop-band attenuation. For other values of B and K, the stop-band attenuation levels are progressively better for increasing B and decreasing K. Also note that since the regularity constraint puts K zeros at the Nyquist frequency, the filters that have regularity $K > 0$ show responses that approach zero at the Nyquist frequency, and get there quicker if K is larger. Lastly, the number of oscillations in the stop-band is related to the degrees of freedom remaining after regularity K is imposed, i.e. $L/2 - K$, which is used by the alteration theorem within the algorithm to provide the optimized equiripple response.

5.3.4 Filters for Eliminating Side-lobes

We can now generate filters with various transition bandwidths and stop-band attenuation levels and determine the type of channel responses that a particular QMF filter pair provides in an iterated filter bank.

During the experimental stage, it has been found that by using filters with sufficiently sharp cut-off rates the side-lobes could be eliminated altogether and the overlap between adjacent bands could be somewhat improved. The required transition bandwidth for eliminating all side-lobes, called the *critical bandwidth* B_c, for each channel in uniform 4-, 8-, 16-, 32-channel filter banks has been manually determined and results have been summarized in Table 5.2. Since reducing the transition bandwidth effectively raises stop-band attenuation levels, this procedure amounted to manually determining the value of B that made the largest side-lobe level to be equal to the stop-band attenuation level. This procedure was done using filters of lengths $L = 24$ and 32 (both of which provided similar results and where the average was taken) and using a small value of regularity ($K = 2$) so that stopband responses could be maximized.

Two observations can be made from the results in Table 5.2. First, note that B_c values in the lower bands remain the same down the column regardless of the number of channels in the filter bank. For example, B_c values for channel 0 are about 0.202 for all four filter banks and B_c values for channel 2 are 0.120 for the 8-, 16-, 32-channel filter banks. In fact, there is a pattern that appears in the B_c measurements where the values in the lower-half of one filter bank repeat in the "lower-quarter" of the next filter bank, i.e. the filter bank with double the number of channels, and only the remaining values represent new measurements. One way of explaining this is that the lowest bands of

Filter Bank	Channels 0-7							
	0	1	2	3	4	5	6	7
4-channel	0.202*	0.34**						
8-channel	0.202	0.202	0.120*	0.202				
16-channel	0.202	0.202	0.120	0.203	0.177	0.119	0.065*	0.202
32-channel	0.200	0.203	0.120	0.203	0.179	0.120	0.065	0.202

Filter Bank	Channels 8-15							
	8	9	10	11	12	13	14	15
32-channel	0.202	0.065	0.119	0.119	0.066	0.066	0.031*	0.202

Table 5.2: Required bandwidth values, B_c, (normalized bandwidth) for eliminating side-lobes in each channel (in sequency order). Note, only the first half is given as the second half is symmetrical. *Smallest B_c among all channels. **Approximate.

a WFB always split in a similar way regardless of the number of channels in the filter bank. For example, the lowest channel in any filter bank maintains the same shape and subdividing it further is like subdividing the low channel of the original two-channel filter bank. Note that since we are working with uniform filter banks which are symmetrical, the same observation also applies to the second half of the filter bank in a symmetrical fashion. Second, the minimum B_c value for a given filter bank appears near the center bands, which makes sense since the outer bands just repeat the measurements from the "previous" filter bank, and this minimum B_c value is approximately half the value of the minimum B_c from the "previous" filter bank. A somewhat interesting implication of this is that the low (and high) channels of a filter bank can be further split without reducing the minimum B_c.

In order to eliminate all side-lobes in a given WFB, the wavelet filter has to be designed so that it satisfies the minimum B_c value of the given filter bank. The channel responses of uniform filter banks that satisfy this minimum B_c requirement (using $L = 32$) are shown in Figure 5.17. Note that in comparison to Figure 5.4, the side-lobes are now eliminated. However, the stop-band attenuation levels suffer somewhat severely when the transition bandwidth is made small, especially for the 32-channel filter bank as shown in Figure 5.17(d). The stopband levels measured in Figure 5.17 are 97.0, 57.3, 32.3, 17.6 dB for the plots (a), (b), (c), and (d), respectively, which when plotted against their respective B_c values gave a near-linear plot with a slope of approximately 1, i.e. there is a near-proportional relationship between bandwidth B and stopband attenuation in

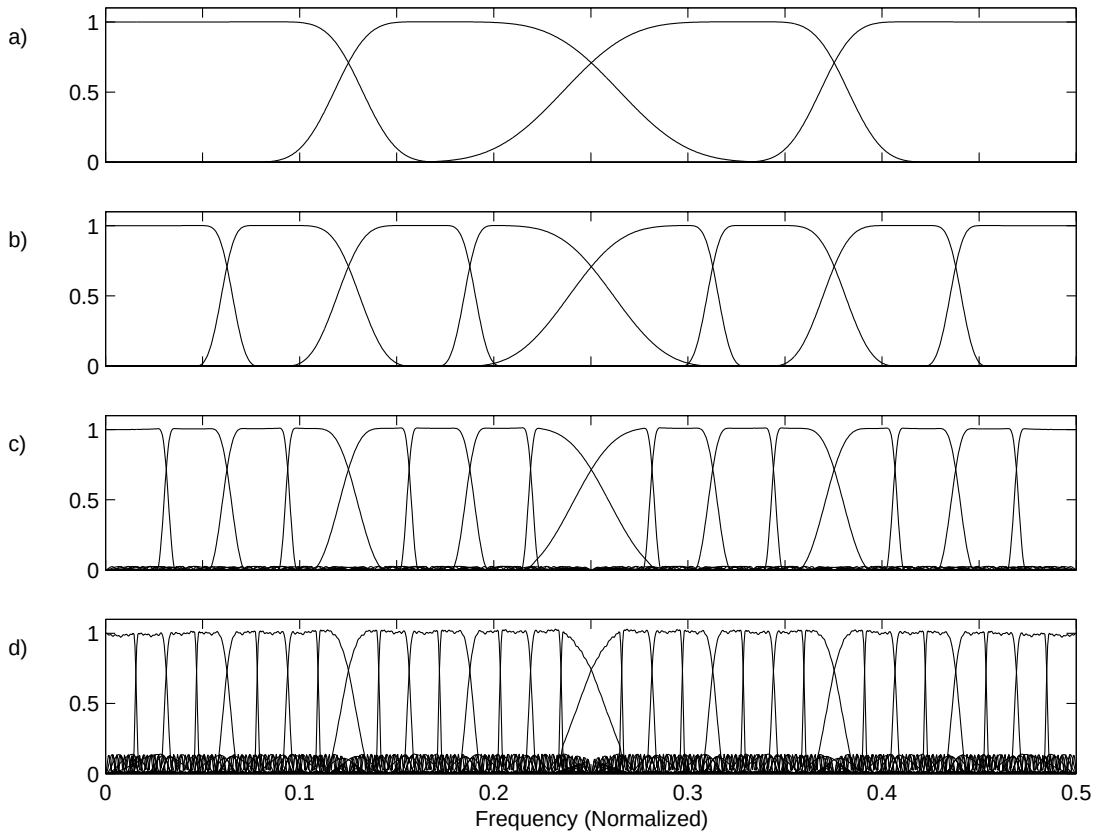


Figure 5.17: Uniform filter banks with L=32, $B = B_c$, and K=2 (a) 4-channels (b) 8-channels (c) 16-channels (d) 32-channels

Remez filters.

It has been suggested that stop-band rejection levels should be better than 96 dB when coding high quality audio [17, p. 784]. Now, in order to satisfy both the required B_c and an attenuation of 96 dB, we need to essentially increase the degrees of freedom available in the design, which can only be done by increasing the filter length L. As a result, the filter lengths required for providing a stopband attenuation of 96 dB and the minimum transition bandwidths given in Table 5.17 were determined and are given in Table 5.3. For a B_c value of 0.031, it has been found that the modified Remez exchange algorithm suffered from numerical instability (as implemented in MATLAB) for filter lengths greater than about $L = 204$, at which length it provided a stopband attenuation

of $A_s = 80$ dB. The required filter length for providing $B_c = 0.031$ and $A_s = 96$ dB, as a result, could not be found but could be assumed to be greater than $L = 204$.

Transition Bandwidth	Required Filter Length
B_c	L
0.202	32
0.178	38
0.120	56
0.065	104
0.031	> 204 (gives $A_s = 80$ dB)

Table 5.3: Required filter lengths for providing B_c and 96 dB stop-band attenuation.

Now, as already described in section 5.2.4, longer filters generally mean poorer time support in the wavelet domain. From equation 5.3, a filter length of 104 will result in a time support of 2.36 ms, 7.03 ms, 16.4 ms, and 35.1 ms for 2-, 4-, 8-, and 16-channel uniform filter banks, respectively. As described in section 3.3.4, temporal masking is shorter for pre-masking than post-masking, which is between 2 to 20 ms. In Sinha and Tewfik's wavelet coder [34], for example, a time support of 4 ms was considered adequate for pre-echo control. It is apparent that a filter length of 104 already provides insufficient time localization for filter banks with more than 4 channels whenever we require better control of time-domain artifacts. A uniform filter bank with 4 channels is rather inadequate since we usually require a WFB with many more channels in audio coding, e.g. CB resolution as described in section 5.1.2.

Consequently, we can now see that it is not possible to design a fixed CB-resolution WFB that satisfies all frequency and time domain localization requirements for audio coding. The design must sacrifice either channel separation, i.e. allow side-lobes to appear, stopband attenuation, i.e. allow attenuation levels less than 96 dB, or time localization, i.e. allow time supports greater than about 5 ms, where each of these represents a source of distortion during the coding process.

5.3.5 Summary and Discussion

We can summarize what we have discussed so far as follows:

1) As explained in section 5.2.1, the iteration process of the WFB results in channel responses that are uneven and sometimes characterized by large out-of-band side-

lobes. Filters with sharp cut-off rates can sometimes improve or eliminate these non-ideal behaviours, but only an ideal filter can provide perfect channel separation.

2) Filters designed with the modified Remez exchange algorithm have indicated that it is possible to design a WFB without any out-of-band side-lobes. Filters that eliminate side-lobes are typically required to satisfy a certain critical bandwidth requirement, B_c, and the value of the critical bandwidth becomes smaller as we decompose the tree further down, i.e. increase the number of channels.

3) Another important requirement in audio coding is to provide high stopband attenuations in the channel responses, typically about 96 dB.

4) It has been found that if we design wavelet filters to eliminate side-lobes and to provide good stopband attenuations, e.g. 96 dB, then time domain localization property greatly suffers. As a result, not all three can be satisfied simultaneously and a trade-off will always exist.

5) The way in which this trade-off is decided needs to take into account the requirements of the input signal, maybe even on a frame-by-frame basis like in an adaptive scheme, but this is a topic of future research.

6) Furthermore, since the modified Remez exchange algorithm provides optimal, or near-optimal, solutions in terms of the constraints in the frequency domain, we can extend these results and generalize them for all wavelet solutions. What this means, essentially, is that the WFB provides a flexible scheme in controlling the time-frequency resolution, but the time-frequency localization of the wavelet bands will always suffer.

5.4 Some Test Results

Some filters have been designed using the modified Remez exchange algorithm according to the findings above and used in the wavelet coder described in Appendix A to determine their performances in comparison to other wavelet filters. The filters have been designed to eliminate side-lobes and the audio coder was designed to provide no additional handling of inter-band leakages. This meant that reconstructed signals contained aliasing errors that resulted from the overlapping of wavelet bands, particularly with adjacent bands. Furthermore, the audio coder was based on frequency domain coding using a fixed CB

resolution WFB and did not provide any control over time-domain artifacts. The tree structure was chosen to provide an approximate CB division with a maximum depth of 5 and with 14 channels where the minimum B_c was 0.120. The resulting 14-channel CB structure is shown in Figure 5.18. This choice was considered to provide a reasonable time resolution and time localization property (although temporal coding was not used in the coder) as well as a transition bandwidth requirement that was not overly restrictive. This structure, however, did not provide a very good approximation of the CB resolution in the low frequency range as that required a much deeper tree depth and a much longer delay when using transparent boundary handling (section 4.3.4).

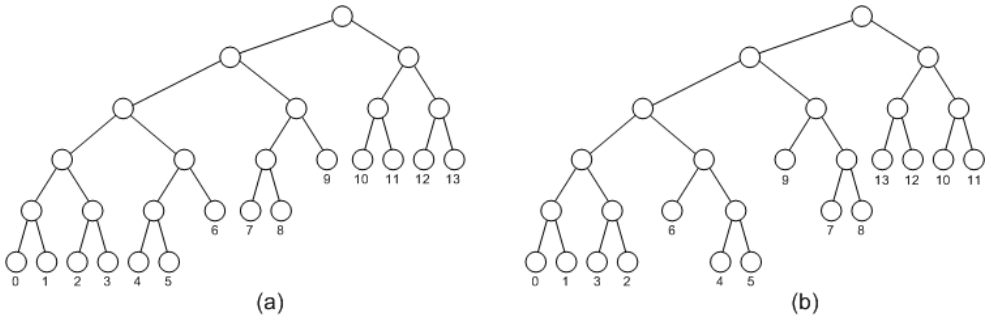


Figure 5.18: 14-channel CB tree structure (a) Desired ordering (sequency) (b) Actual ordering (natural). The numbers under the leaf nodes indicate the band number.

The basis filter was then designed with a transition bandwidth that was better than 0.120 and a length that provided "good" stop-band attenuation levels. The transition bandwidth was chosen as 0.118 and two different filter lengths was used, namely, $L = 34$ and 56, which provided stop-band levels of 61.2 dB and 99 dB, respectively. The channel responses of the WFB using the chosen tree structure and basis filters are given in Figure 5.19. For the purpose of comparison, three other orthogonal wavelet filters of length $L = 34$ and 56 were used, namely, minimum-phase Daubechies, Symmlet, and Battle-Lemarie wavelets. The Symmlet wavelets provided the maximum number of vanishing moments, like Daubechies, but were designed to be as symmetrical as possible. The Battle-Lemarie wavelets are based on spline functions.

An informal listening test was carried out by the author where several audio samples were encoded with each of the four filters at bitrates of 128, 96, 64, 48, and 32 kbps. Results generally indicated that signals with significant amount of temporal events, e.g. music containing lots of percussive sounds, showed similar results among the four filters and signals with lots of spectral "activity", e.g. music containing many instruments and

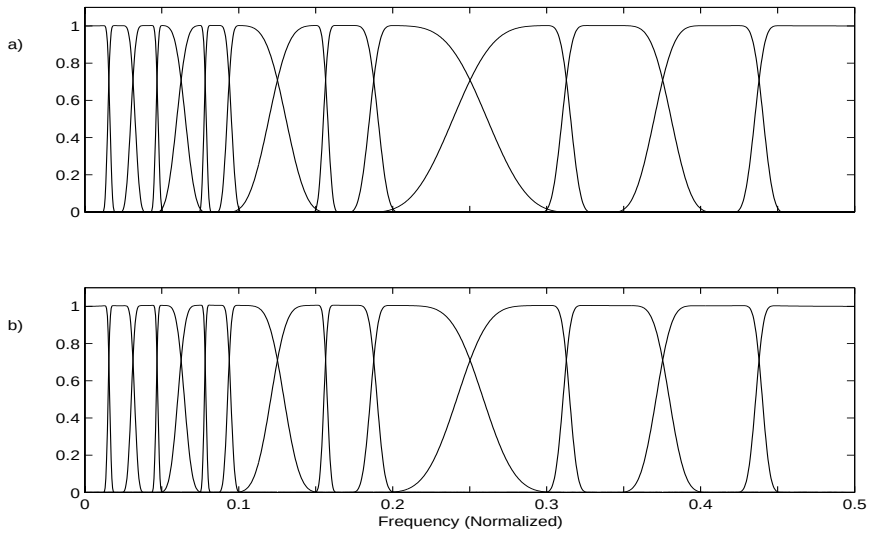


Figure 5.19: Frequency response of WFB with 14-channel CB structure and basis filters of length (a) $L = 34$ (b) $L = 56$

many different parts, also showed similar results among all the filters. On the other hand, stationary-like signals with few spectral components, e.g. classical music with only one or two parts, showed slightly better results when the Remez filter was used. These results seem to indicate that aliasing distortions resulting from the presence of the side-lobes becomes noticeable when there is little temporal activity and when only a few spectral components are present. One possible explanation for this is that the presence of many other spectral components could be masking the differences that might otherwise be noticeable and the amount of temporal activity could also contribute to making the "side-lobe-related" distortions inaudible. In signals where the side-lobe distortions were audible (with the other three filters), they were perceived as high-pitched "whistling" sounds that stood out from the rest of the music. Coding examples have been included in Appendix B where three audio signals have been encoded and decoded with the four given filters. 'corelli_m.wav' is an audio clip of a classical piece that contains almost no transients and few stationary components, 'celine_mono.wav' is a clip of a pop tune that contains many stationary and transient components, and 'percussion.wav' is a percussive clip that contains mostly transient components. For 'corelli_m.wav', audio files containing just the reconstruction error have also been included. The reconstruction error files as

well as the reconstruction files for 'corelli_m.wav' show that the Remez filter provides a minor improvement over the other filters since additional aliasing errors can be easily identified in the reconstructed signals resulting from the other filters. Note that since we are using a perceptually based scheme and are mainly interested in perceptual quality, no additional objective measurement was used to assess the quality of the reconstructed signal.

Two general conclusions can be made from these results:

1) The presence of side-lobes introduces disturbing artifacts that can sometimes be readily perceived and that proper design of the basis filter can reduce or eliminate them.

2) Temporal artifacts become important in signals with significant amounts of transient components and, as a result, additional time domain coding is required in general.

5.5 Summary and Conclusion

This chapter explored the use of the WFB in perceptual audio coding and described, in particular, the non-ideal localization behaviours of the WFB. Although a flexible choice of tiling in the time-frequency plane is provided by the tree structure, the localization properties, as determined by the choice of the basis filter, has been shown to be far from ideal in both time and frequency domain. Localization is particularly important in audio coding since coding distortions need to be carefully shaped in both time and frequency so that their perceived effects can be minimized.

One approach for minimizing the effects of poor frequency localization has been explored by utilizing a filter design algorithm based on the modified Remez exchange algorithm. Filters were designed to provide sharp cut-off rates, while maximizing stop-band attenuations, so that out-of-band side-lobes could be eliminated. Eliminating side-lobes was considered important, or at least attractive, since no additional mechanism would then be required to take care of the inter-band leakages that occur between distant wavelet bands. In designing such filters, some trade-off issues were identified, namely, between the transition bandwidth and the stopband attenuation in frequency domain, and more generally between frequency domain localization and time domain localization. It has been found that it is not possible to satisfy all the given constraints as required in audio coding and that some compromise was always required.

A simple WFB based on a Remez filter designed to eliminate side-lobes was used in an audio coder to determine its performance relative to other wavelet filters. It was found that the Remez filter provided a minor improvement over other wavelet filters when coding audio signals that were stationary-like and had few spectral components. For other audio signals, the Remez filter was found to provide similar results to other wavelet filters.

Chapter 6

Conclusion

This thesis described and explored the use of the Wavelet Filter Bank (WFB) in the context of perceptual audio coding. A brief summary of each chapter is given next, followed by some suggestion for future work.

6.1 Summary of Thesis

Audio coding requires the removal of both perceptual irrelevancy and statistical redundancy in order to provide the kind of compression levels that are required by many band-limited applications. It has been found that a perceptual coding scheme provides the best framework where both perceptual irrelevancy and statistical redundancy can be removed [33]. A perceptual coder is typically comprised of three main stages, namely, a filter bank, a psychoacoustic model, and a coding and quantization stage. The filter bank stage is used to transform the input signal into a domain that is more appropriate for applying perceptual criteria as well as a domain that provides some de-correlation of the input signal. The psychoacoustic stage is used to calculate the perceptual criteria, i.e. masking threshold, using facts from psychoacoustics and psychophysics. And the coding and quantization stage performs the actual bitrate reduction by re-quantizing the transform domain coefficients according to the masking results.

The calculation of the masking threshold is of central importance in a perceptual coder since it indicates how a listener perceives sound and how coding noise can be introduced and shaped without introducing perceived distortion. As a result, the psychoacoustic model dictates either directly or indirectly the design of the filter bank stage and, subsequently, the action of the coding and quantization stage. Common psychoacoustic

models found in coders today are based on the masking pattern model, which utilizes results from simple single-masker and single-maskee experiments. These masking models are attractive due to their relatively simple implementations and reasonably accurate results, although some shortfalls have been noted.

The Wavelet Transform, or more generally the Wavelet Filter Bank, has its roots in various branches of mathematics, physics, and engineering and represents an interesting and potentially very useful tool in many applications, including perceptual audio coding. The WFB can be defined in the continuous domain and also in the discrete domain, where a connection between the two can be established. Furthermore, the WFB provides a great deal of flexibility in its design through the choice of tree structure and basis filter. The tree structure essentially controls the time-frequency resolution, i.e. tiling in the time-frequency plane, and the basis filter controls the localization of each band, i.e. localization of individual tiles.

Various works that have explored the application of the WFB in perceptual audio coding have indicated that the WFB provided a feasible solution to the filter bank stage, where near-transparent bitrates of between 48 and 110 kbps were reported by various coders. In terms of designing the WFB, the tree structure was most commonly found to be based on the critical band (CB) resolution of the human ear, while no general consensus existed for the choice of the basis filter. Although the flexibility of the WFB was an attractive property, as it provided an easy way of designing a CB resolution filter bank (and also the possibility of a signal-adaptive filter bank), work by some researchers as well as the author indicated that the WFB possessed a rather poor frequency localization property. More specifically, the frequency response of the WFB was un-even and variable across bands, and some bands even contained considerable amount of side-lobes. This, in the context of coding, was shown to introduce un-cancelled aliasing components in the reconstructed signal that were audibly disturbing and clearly undesirable. One method in trying to minimize or even eliminate the out-of-band aliasing components, particularly the large side-lobes, was explored by utilizing a filter design method called the modified Remez exchange algorithm. This method allowed filters to be designed with sharp cut-off rates so that overlap between filters during iteration was minimized. The transition bandwidth required to entirely eliminate the side-lobes, i.e. critical bandwidth, was determined for WFB's with various number of channels. In designing filters with sharp cut-off rates, it was found that there was an inherent trade-off between the transition bandwidth, the stopband attenuation, and temporal support. Moreover, it was found that it was not possible to design a WFB that satisfied all the requirements of audio coding,

e.g. eliminated side-lobes, stopband attenuation of better than 96 dB, and temporal localization of 5 ms or less. There was always a trade-off between the three.

Finally, some Remez filters were designed and used in a wavelet audio coder in order to determine their performances in comparison to other wavelet filters. Three other filters, namely, Daubechies, Symmlets, and Battle-Lemarie wavelets, of same length were used to encode a number of audio signals at various bitrates. In general, the four filters were found to provide similar results for most signals, but for signals that contained only a few harmonic components, the Remez filters were found to provide a slightly better result. This was due to the fact that the un-cancelled aliasing components that might have been masked in the other audio signals were audible in audio signals that only contained few masking activities, and only the Remez filter provided a WFB with eliminated side-lobes.

6.2 Future Work

This thesis described the limitation of the WFB by examining the basic localization trade-offs involved in the WFB and exploring one method in minimizing the poor frequency localization of the WFB. As already mentioned, eliminating all undesirable distortions in the WFB is not possible due to the inherent trade-off that exists in the design of wavelet filters. As a result, some areas that can be further explored are:

1) Determining if the stopband attenuation levels of below 96 dB can provide acceptable performances.

2) Experimenting with other tree structures to see how they affect performance.

3) Exploring other filter design techniques to determine how other design parameters, e.g. linear phase, affect performance.

4) Exploring the use of the more general M-channel wavelet transform.

5) Developing a bit allocation procedure that takes care of the overlaps between adjacent bands (which still exists for Remez filters). This can be used in conjunction with the Remez filters designed to eliminate side-lobes.

6) Developing a time domain coding approach that takes advantage of the time resolution of the WFB.

In general, the flexibility of the WFB can be seen as an advantage while the poor frequency localization property represents a drawback. As a result, determining how this flexibility can be fully exploited and the drawback fully minimized needs to be the goal of future research. In trying to minimize the drawback, understanding which of and how these properties, e.g. cut-off rate, stopband attenuation, and time-localization, can be sacrificed without sacrificing performance becomes important. In exploiting the flexibility of the WFB, an adaptive scheme needs to be developed so that the WFB can change according to both perceptual and statistical requirements of the input signal.

We can say that the Wavelet Filter Bank provides a flexible signal analysis scheme in exchange for poor localization properties. Whether the benefits provided by this flexibility outweigh the costs associated with its drawback still remains to be seen.

Appendix A

Wavelet Audio Coder

This appendix describes the Wavelet Audio Coder (WAC) as developed by the author. The WAC was developed under the PC platform using Visual C++ 6.0 and was designed to run through a graphical user interface (GUI) that is shown in Figure A.1. Source code is available for download at 'http://multicom10.uwaterloo.ca/ scplee/thesis/software/' and accompanying documentation is found in the file 'wac_readme.txt'. For execution, the WAC program requires the specification of a number of input parameters through the UI, which can be summarized as follow:

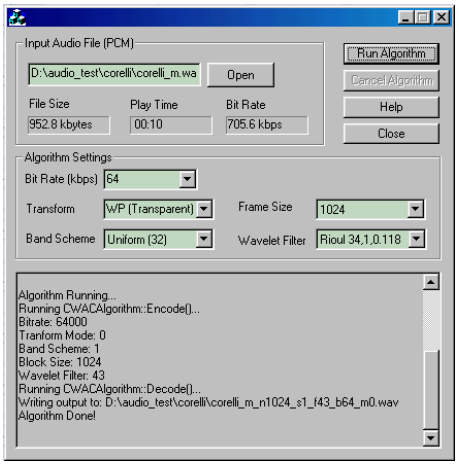


Figure A.1: WAC graphical user interface

1) Input File: Input .wav file in PCM format.

2) Bit Rate: Bitrate of encoded signal.

3) Transform: Type of filter bank used, e.g. WFB or MDCT.

4) Band Scheme: WFB tree structure.

5) Wavelet Filter: WFB basis filter

6) Frame Size: Number of samples in a frame or block.

Once all required fields are specified, the algorithm can be run by pressing on 'Run Algorithm'. The execution will encode and decode the signal and output two files, namely, the encoded file with .wac extension and the decoded .wav file. A diagram of the WAC encoder is given in Figure A.2. Note that the design of the WAC closely follows the structure of the generic perceptual coder as described in chapter 2. A description of each stage is given next.

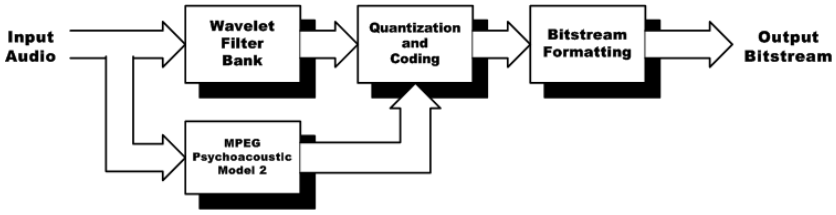


Figure A.2: Structure of Wavelet Audio Coder (WAC)

A.1 The Wavelet Filter Bank (WFB)

The filter bank stage is designed with a WFB that provides a flexible choice of the tree structure and the basis filter. Any number of arbitrary tree structures and basis filters can be specified in the program and made available through the parameter selection boxes in the WAC GUI. In addition, two types of boundary handling are provided, namely, periodic-extension and "transparent-extension" as proposed in [77]. For "transparent-extension", the algorithm takes the required audio samples from the proceeding frame and as a result the number of audio samples required has to be less than the size of a frame. The number of required samples is given by equation 4.84, which is the delay.

A.2 Psychoacoustic Model

The psychoacoustic model used in the WAC coder is based on the Psychoacoustic Model 2 from the MPEG-1 Audio Standard [16]. The MPEG-1 Audio Standard describes two sample psychoacoustic models, the first being computationally simpler and suitable for coding at higher bit rates and the second being more complex but also more reliable at lower bit rates. The Psychoacoustic Model 2 was developed and refined from the psychoacoustic models that appeared in earlier works, namely, from a speech coder developed by Schroeder in [8] and an audio coder developed by Johnston in [27].

The input to the psychoacoustic model is an analysis frame of length 1024 and the output from the model is the resulting masking threshold in the frequency domain (FFT lines) of length 513. The model works as follows [16, 102, 38]:

1) The input signal is transformed into a frequency (Fourier) domain representation.

2) The signal components are mapped into a critical band scale called the *Threshold Calculation Partition* where each partition represents either one FFT line or 1/3 critical band.

3) The tonality of each partition is calculated using an unpredictability measure.

4) A spreading function is applied to each partition, whose relative masking level is determined by the energy and the tonality of the partition.

5) The global masking threshold is computed by combining the masking threshold of the individual partitions.

6) The masking threshold is converted back into the (linear) frequency domain.

7) The absolute threshold of hearing is taken into account.

The masking threshold from the psychoacoustic model can then be applied (by some mapping scheme) to the subband domain that is used by the filter bank stage.

A.3 Coding and Quantization

The coding and quantization stage is based on a simple *bit allocation* scheme that is described as follows. In a bit allocation scheme, the bits are allocated progressively to bands that require it the most, i.e. to the bands that have the highest perceptual distortion. This procedure is illustrated in Figure A.3 and works as follows:

1) Initially, all bands are allocated 0 bits i.e. SNR = 0 dB.

2) For each iteration, the NMR is calculated for each band using NMR = SMR - SNR, and the band with the highest NMR (or worst quality) is allocated additional bits.

3) This process is continued until either all bands reach a NMR value of 0 dB or lower or until no more bits are available to continue the process. The perceptual quality of the decoded signal, as a result, becomes increasingly better at the end of each iteration.

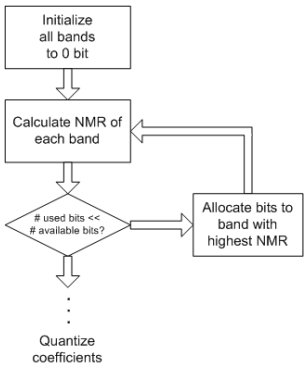


Figure A.3: Bit allocation scheme (after [42])

Note that in calculating the NMR, the value used for SNR is derived directly from the number of bits allocated to each band. This derivation, which is generally implementation specific, has been simplified in this implementation since we are using a uniform scalar quantizer with block companding where each additional bit can be assumed to provide a 6 dB increase in SNR. Furthermore, the bit allocation size is chosen to be 1 bit, and 2 bits if we are allocating bits to a band for the first time, i.e. a band that starts off with 0 bits. However, this simple approach usually only provides a locally optimum solution.

A.4 Bitstream Formatting

The bitstream is comprised of a header and the output frames as shown in Figure A.4. The bitstream header, which appears once at the beginning, contains information about the overall algorithm and the output frame consists of the information required to decode each frame. The frame header consists of a synchronization word that is used to identify the beginning of a frame. The bit allocation is specified for each band, but the scale factors

and wavelet samples are only included if a band has been allocated bits. All encoding is done through bitwise operations in order to make efficient usage of the bitstream.

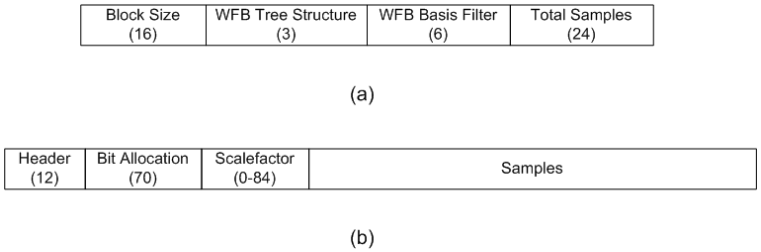


Figure A.4: Bitstream format (a) Bitstream header (b) Output frame format (numbers in parenthesis are the associated number of bits used

A.5 Miscellaneous

The WAC coder was found to provide near-transparent coding at a bitrate of 128 kbps for most audio signals. Furthermore, the WAC coder was designed and implemented so that each stage could be easily extended or modified without affecting the overall algorithm. For example, other filter banks can be added with relative ease through a new C++ class or a new coding and quantization scheme can be added through a new C++ function. Additional detail to the implementation of each C++ class and function are documented within the program.

Appendix B

Audio Samples

The audio samples mentioned in this thesis can be downloaded from 'http://multicom10.uwaterloo.ca/ scplee/thesis/audio/'. All audio files are in PCM format and are briefly described next.

B.1 Audio Samples for Section 5.2.2

* The original samples:

  sine_ch00_345_2sec.wav
  sine_ch01_1034_2sec.wav
  sine_ch02_1723_2sec.wav
  sine_ch03_2412_2sec.wav
  sine_ch04_3101_2sec.wav
  sine_ch05_3790_2sec.wav
  sine_ch06_4479_2sec.wav
  sine_ch07_5168_2sec.wav
  sine_ch08_5857_2sec.wav
  sine_ch09_6546_2sec.wav
  sine_ch10_7235_2sec.wav
  sine_ch11_7924_2sec.wav
  sine_ch12_8613_2sec.wav
  sine_ch13_9302_2sec.wav
  sine_ch14_9991_2sec.wav

sine_ch15_10680_2sec.wav
sine_ch16_11369_2sec.wav
sine_ch17_12058_2sec.wav
sine_ch18_12747_2sec.wav
sine_ch19_13436_2sec.wav
sine_ch20_14125_2sec.wav
sine_ch21_14814_2sec.wav
sine_ch22_15503_2sec.wav
sine_ch23_16192_2sec.wav
sine_ch24_16881_2sec.wav
sine_ch25_17570_2sec.wav
sine_ch26_18259_2sec.wav
sine_ch27_18948_2sec.wav
sine_ch28_19637_2sec.wav
sine_ch29_20326_2sec.wav
sine_ch30_21015_2sec.wav
sine_ch31_21704_2sec.wav

* The reconstructed samples:

sine_ch00_345_2sec_n1024_s1_f36_b0_m0.wav
sine_ch01_1034_2sec_n1024_s1_f36_b0_m0.wav
sine_ch02_1723_2sec_n1024_s1_f36_b0_m0.wav
sine_ch03_2412_2sec_n1024_s1_f36_b0_m0.wav
sine_ch04_3101_2sec_n1024_s1_f36_b0_m0.wav
sine_ch05_3790_2sec_n1024_s1_f36_b0_m0.wav
sine_ch06_4479_2sec_n1024_s1_f36_b0_m0.wav
sine_ch07_5168_2sec_n1024_s1_f36_b0_m0.wav
sine_ch08_5857_2sec_n1024_s1_f36_b0_m0.wav
sine_ch09_6546_2sec_n1024_s1_f36_b0_m0.wav
sine_ch10_7235_2sec_n1024_s1_f36_b0_m0.wav
sine_ch11_7924_2sec_n1024_s1_f36_b0_m0.wav
sine_ch12_8613_2sec_n1024_s1_f36_b0_m0.wav
sine_ch13_9302_2sec_n1024_s1_f36_b0_m0.wav
sine_ch14_9991_2sec_n1024_s1_f36_b0_m0.wav

sine_ch15_10680_2sec_n1024_s1_f36_b0_m0.wav
sine_ch16_11369_2sec_n1024_s1_f36_b0_m0.wav
sine_ch17_12058_2sec_n1024_s1_f36_b0_m0.wav
sine_ch18_12747_2sec_n1024_s1_f36_b0_m0.wav
sine_ch19_13436_2sec_n1024_s1_f36_b0_m0.wav
sine_ch20_14125_2sec_n1024_s1_f36_b0_m0.wav
sine_ch21_14814_2sec_n1024_s1_f36_b0_m0.wav
sine_ch22_15503_2sec_n1024_s1_f36_b0_m0.wav
sine_ch23_16192_2sec_n1024_s1_f36_b0_m0.wav
sine_ch24_16881_2sec_n1024_s1_f36_b0_m0.wav
sine_ch25_17570_2sec_n1024_s1_f36_b0_m0.wav
sine_ch26_18259_2sec_n1024_s1_f36_b0_m0.wav
sine_ch27_18948_2sec_n1024_s1_f36_b0_m0.wav
sine_ch28_19637_2sec_n1024_s1_f36_b0_m0.wav
sine_ch29_20326_2sec_n1024_s1_f36_b0_m0.wav
sine_ch30_21015_2sec_n1024_s1_f36_b0_m0.wav
sine_ch31_21704_2sec_n1024_s1_f36_b0_m0.wav

B.2 Audio Samples for Section 5.4

Note the following conventions (see ‘common.h’ for more info):
‘n1024’ = frame size of 1024
‘s6’ = 14-channel CB resolution tree structure
‘f43’ = Remez filter of L=34
‘f44’ = Daubechies filter of L=34
‘f45’ = Symmlet filter of L=34
‘f46’ = Battle-Lemarie filter of L=34
‘b??’ = bitrate of encoded signal
‘m0’ = WFB with ‘‘transparent-extension’’
‘diff’ = difference between original and reconstructed file

* Pop Clip:

celine_mono.wav
celine_mono_n1024_s6_f43_b96_m0.wav

```
  celine_mono_n1024_s6_f44_b96_m0.wav
  celine_mono_n1024_s6_f45_b96_m0.wav
  celine_mono_n1024_s6_f46_b96_m0.wav

* Percussive Clip:

  percussion.wav
  percussion_n1024_s6_f43_b32_m0.wav
  percussion_n1024_s6_f44_b32_m0.wav
  percussion_n1024_s6_f45_b32_m0.wav
  percussion_n1024_s6_f46_b32_m0.wav

* Classical Clip:

  corelli_m.wav
  corelli_m_n1024_s6_f43_b48_m0.wav
  corelli_m_n1024_s6_f44_b48_m0.wav
  corelli_m_n1024_s6_f45_b48_m0.wav
  corelli_m_n1024_s6_f46_b48_m0.wav
  corelli_f43_b48_diff.wav
  corelli_f44_b48_diff.wav
  corelli_f45_b48_diff.wav
  corelli_f46_b48_diff.wav
```

Appendix C

Sequency Ordered WFB

This appendix describes an algorithm that provides a WFB decomposition according to a sequency-ordered tree structure. A structure called *hedge* used to specify the tree structure is described first and a recursive algorithm that performs the WFB according to a sequency-ordered hedge structure is then described.

C.1 Hedge Tree Structure

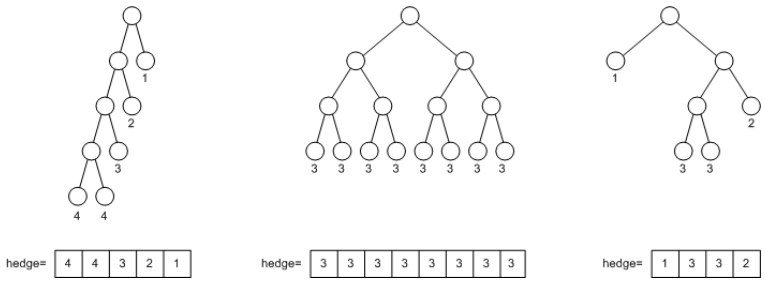


Figure C.1: Examples of hedge structures

The hedge structure as described in [99] is an array of leaf-nodes that can be used to specify the structure of a binary tree. The hedge contains the position of each leaf-node in terms of its depth in the tree as we go from the left-most leaf-node to the right-most leaf-node. Three examples are shown in Figure C.1 where the given tree structures are specified using hedge arrays. Note that providing the leaf-node depths is enough for specifying the exact shape of a tree structure.

C.2 Sequency-Order WFB Algorithm

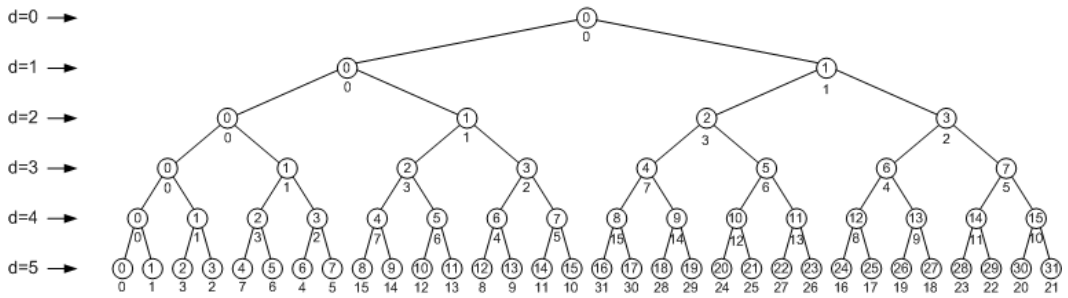


Figure C.2: Natural ordering and sequency ordering in a tree

Now, using a hedge structure that specifies the tree in sequency order, i.e. the tree structure represents the "logical" frequency ordering as we go from the left-most node to the right-most node, an algorithm that performs the desired WFB decomposition can be developed by looking at the relationship between natural ordering and sequency ordering. Figure C.2 shows a tree structure that gives the ordering of the nodes in both natural and sequency order (same as Figure 5.14). As described in section 5.2.3, the low- and high-frequency regions remain the same after a low-pass filtering operation, but switched after a high-pass filtering operation. This can be seen in Figure C.2 where every time a "high-pass child-node", i.e. an odd numbered node in sequency ordering, is divided, the two child-nodes are switched in their order. Therefore, if we travel down the tree according to this sequency-ordered path, then we can decompose a tree so that it results in a sequency ordered decomposition. The simple rule to remember is that if the current node is the child of a low-pass operation, i.e. even-numbered, then its two child nodes will remain in normal order, and if the current node is the child of a high-pass operation, i.e. odd-numbered, then its two child nodes will be in reverse order. Due to the nature of this rule, a recursive algorithm was found to be naturally suited to the problem. A pseudo-code for this recursive algorithm is given next.

```
* Variables:

  wpd = wavelet packet data class
  t   = tree structure (a hedge)
  ti  = tree structure index (one-based)
  d   = tree depth (zero-based)
```

```
  nn  = tree node in natural order (zero-based)
  ns  = tree node in sequency order (zero-based)

* Recursive Function:

  % Travel through the ``actual'' tree using natural order
  function [wpd,ti] = wp_sequency(wpd,t,ti,d,nn,ns)
     if (t(ti)==d) then         % Base case
        ti = ti + 1
     else                       % Recursive step
        t.decompose(d,nn);
        if (ns is even) then
           % If current node (sequency) is even, then take
           % left-branch first
           [wpd,ti] = wp_sequency(wpd,t,ti,d+1,2*nn,2*ns)
           [wpd,ti] = wp_sequency(wpd,t,ti,d+1,2*nn+1,2*ns+1)
        else
           % If current node (sequency) is odd, then take
           % right-branch first
           [wpd,ti] = wp_sequency(wpd,t,ti,d+1,2*nn+1,2*ns)
           [wpd,ti] = wp_sequency(wpd,t,ti,d+1,2*nn,2*ns+1)
        end
     end
  end

* Calling the Algorithm:

  [wpd,ti] = wp_sequency(wpd,t,1,0,0,0)
```

In the algorithm above, 'wpd' is assumed to be a class that is initialized with the input signal but not yet decomposed and a class that provides the necessary function, e.g. decompose(), for performing the decomposition one node at a time. The algorithm works by starting at depth 0 and going down the tree recursively, while splitting each node it visits, until it finds the first leaf-node, and then moves on to the second leaf-node specified in 't'. The second leaf-node is also found recursively in a similar manner, and the algorithm continues until all leaf-nodes have been found. While the algorithm travels

through the tree, variables are used to keep track of both the natural ('nn') and sequency ('ns') order of each node, where the natural order is used to indicate which node is to be split in the "actual" tree and the sequency order is used to decide which branch to take next. Note that in the recursive step, the conditional statement uses the simple rule that has been mentioned above where the left-branch is taken first if the current node is a low-pass child, and the right-branch first if the current node is a high-pass child. The base case simply checks whether or not the current node is the next leaf-node that we are looking for in the hedge array 't'. A similar version of this algorithm also appears in the WAC coder inside the function 'DefineHedge()' under the class 'HedgeSelect'.

Bibliography

[1] N. Jayant and P. Noll, *Digital Coding of Waveform: Principles and Applications to Speech and Video*. New Jersey: Prentice Hall, 1984.

[2] T. Painter and A. Spanias, "Percetual coding of digital audio," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 451–513, April 2000.

[3] N. Jayant, "Signal compression: Technology targets and research directions," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 5, pp. 796–818, June 1992.

[4] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[5] M. Hans and R. Schafer, "Lossless compression of digital audio," *IEEE Signal Processing Magazine*, pp. 21–32, July 2001.

[6] A. Gersho, "Advances in speech and audio compression," *Proceedings of the IEEE*, vol. 82, no. 6, pp. 900–918, 1994.

[7] R. Crochiere, S. Webber, and J. Flanagan, "Digital coding of speech in subbands," *Bell Syst. Tech. Journal*, pp. 1069–1085, 1976.

[8] M. Schroeder, B. Atai, and J. Hall, "Optimizing digital speech coders by exploiting masking properties of the human ear," *Journal Acoust. Soc. Am.*, vol. 66, no. 6, December 1979.

[9] R. Zelinski and P. Noll, "Adaptive transform coding of speech signals," *IEEE Transactions Acoust., Speech, and Signal Processing*, vol. 25, pp. 299–309, 1975.

[10] E. Schroeder and W. Voessing, "High quality deigital audio encoding with 2.0 bits/sample using adaptive transform coding," in *Proc. of the 80th. AES-Convention*, 1986. preprint 2321.

[11] K. Brandenburg, "OCF- A new coding algorithm for high quality sound signals," in *ICASSP-97*, pp. 5.1.1–5.1.4, 1987.

[12] J. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE Journal on Selec. Areas in Comm.*, vol. 6, no. 2, pp. 314–323, 1988.

[13] Y. Mahieux, J. Petit, and A. Charbonnier, "Transform coding of audio signals using correlation between successive transform blocks," in *ICASSP-89*, pp. 2021–2024, 1989.

[14] Y. Dehery, M. Lever, and P.Urcun, "A MUSICAM source codec for digital audio broadcasting and storage," in *ICASSP-91*, vol. 1, pp. 3605–3609, 1991.

[15] K. Brandenburg, J. Herre, J. Johnston, Y. Mahieux, and E. Shroeder, "ASPEC: Adaptive spectral perceptual entropy coding of high quality music signals," in *90th AES Convention*, 1991. preprint 3011 (A-4).

[16] ISO/IEC, JTC1/SC29, *Information technology- Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s-IS 11172-3 (audio)*, 1992.

[17] K. Brandenburg, "ISO-MPEG-1 Audio: A generic standard for coding of high-quality digital audio," *J. Audio Eng. Soc.*, vol. 42, no. 10, pp. 780–792, October 1994.

[18] J. Johnston, S. Quackenbush, G. Davidson, K. Brandenburg, and J. Herre, "Mpeg audio coding," in *Wavelets, Subband, and Block Transform in Communications and Multimedia* (A. Akansu and M. Medlyey, eds.), Kluwer Academic Publishers, 1999.

[19] ISO/IEC, *Overview of the MPEG-4 Standard*. http://mpeg.telecomitalialab.com/standards/mpeg-4/mpeg-4.htm, May 2002.

[20] P. Stokas, *Which is the best low-bitrate audio compression algorithm? OGG vs. MP3 vs. WMA vs. RA*. http://http://ekei.com/audio/, March 2002.

[21] L. Fielder, M. Bosi, G. Davidson, M. Davis, C. Todd, and S. Vernon, "AC-2 and AC-3: Low-complexity transform-based audio coding," in *Collected Papers on Digital Audio Bit-Rate Reduction*, Audio Engineering Society, 1996.

[22] K. Tsutsui and et al., "ATRAC: Adaptive transform acoustic coding for MiniDisc," in *Collected Papers on Digital Audio Bit-Rate Reduction*, Audio Engineering Society, 1996.

[23] J. Johnston and et al., "AT&T Perceptual Audio Coding (PAC)," in *Collected Papers on Digital Audio Bit-Rate Reduction*, Audio Engineering Society, 1996.

[24] J. Moffitt, "Ogg vorbis - open, free audio - set your media free," *Linux Journal*, pp. 146–50, January 2001.

[25] Xiph.Org, *OggVorbis: open, free audio.* http://www.vorbis.com, April 2003.

[26] M. Sablatash and T. Cooklev, "Compression of high-quality audio signals, including recent methods using wavelets packets," *Digital Signal Processing*, vol. 6, pp. 96–107, 1996.

[27] J. Johnston, "Audio coding with filter banks," in *Subband and Wavelet Transforms* (A. Akansu and M. Smith, eds.), pp. 287–307, Kluwer Academic, 1996.

[28] X. Wei, M. Shaw, and M. Varley, "Optimum bit allocation and decomposition for high quality audio coding," in *ICASSP-97*, pp. 315–318, 1997.

[29] J. Herre and J. Johnston, "Continuously signal-adaptive filterbank for high-quality perceptual audio coding," in *Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics*, 1997.

[30] J. Princen and J. Johnston, "Audio coding with signal adaptive filterbank," in *ICASSP-95*, pp. 3071–3074, 1995.

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

[32] A. Ferreira, "Perceptual audio coding and the choice of an analysis/synthesis filter bank and psychoacoustic model," in *104th Convention of the AES*, May 1998. preprint 4671.

[33] M. Bosi, "Filter banks in perceptual audio coding," in *AES 17th International Conference*, pp. 125–136, Audio Engineering Society, 1999.

[34] D. Sinha and A. Tewfik, "Low bit rate transparent audio compression using adapted wavelets," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3463–3479, December 1993.

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

[36] A. Akansu and R. Haddad, *Multiresolution Signal Decomposition*. Academic Press, 2nd ed., 2001.

[37] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Prentice Hall, 1995.

[38] K. Brandenburg, "Perceptual coding of high quality audio," in *Applications of digital signal processing to audio and acoustics* (M. Kahrs and K. Brandenburg, eds.), Kluwer Academic Publishers, 1998.

[39] A. Ferreira, "The perceptual audio coding concept: from speech to high-quality audio coding," in *AES 17th International Conference*, pp. 258–286, 1999.

[40] M. Smith and A. Akansu, "Introduction and overview," in *Subband and Wavelet Transforms* (A. Akansu and M. Smith, eds.), Kluwer Academic Publishers, 1996.

[41] G. strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1997.

[42] J. Herre, "Temporal noise shaping, quantization and coding methods in perceptual audio coding: a tutorial introduction," in *AES 17th International Conference*, pp. 312–325, 1999.

[43] P. Noll, "Wideband speech and audio coding," *IEEE Communications Magazine*, pp. 34–44, November 1993.

[44] T. Ramstad, "Still image compression," in *The Digital Signal Processing Handbook* (V. Madisetti and D. Williams, eds.), CRC Press, 1998.

[45] A. Harma and U. Laine, "Warped low-delay celp for wideband audio coding," in *AES 17th International Conference*, pp. 207–215, 1999.

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

[47] H. Fletcher, "Auditory patterns," *Rev. Mod. Phys.*, vol. 12, pp. 46–65, 1940.

[48] B. Moore, *An Introduction to the Psychology of Hearing*. London: Academic Press, 4th ed., 1997.

[49] J. Allen, "Cochlear modeling," *IEEE ASSP Magazine*, pp. 3–29, January 1985.

[50] J. Hall, "Auditory psychophysics for coding applications," in *The Digital Signal Processing Handbook* (V. Madisetti and D. Williams, eds.), CRC Press, 1998.

[51] D. Mikat, "Human auditory capabilities," in *Advanced Digital Audio* (K. Pohlmann, ed.), SAMS, 1991.

[52] X. Yang and et al., "Auditory representations of acoustic signals," *IEEE Trans. on Information Theory*, vol. 38, no. 2, pp. 824–839, March 1992.

[53] A. Pena, "A global theoretical auditory model for application on audio coders design and objective perceptual assessment," in *Proceedings of EUSIPCO-94: 7th European Signal Processing Conference*, vol. 3, pp. 1457–60, September 1994.

[54] B. Moore, "Masking in the human auditory system," in *Collected Papers on Digital Audio Bit-Ratee Reduction* (N. Gilchrist and C. Grewin, eds.), Audio Engineering Society, 1996.

[55] B. Moore and B. Glasberg, "Suggested formulae for calculating auditory-filter bandwidths and excitation patterns," *J. Acoust. Soc. Am.*, vol. 74, pp. 750–753, 1983.

[56] R. Veldhuis, "Bit rates in audio source coding," *IEEE Journal on Selected Ares in Communications*, vol. 10, no. 1, pp. 86–96, January 1992.

[57] R. Hellman, "Asymmetry of masking between noise and tone," *Perception and Psychophysics*, vol. 11, no. 241–246, 1972.

[58] A. Oxenham and B. Moore, "Modeling the additivity of nonsimultaneous masking," *Hearing Res.*, vol. 80, pp. 105–118, 1994.

[59] F. Baumgarte, "A psychoacoustic model for audio coding based on a cochlear filter bank," in *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, pp. 139–42, October 2001.

[60] C. Colomes and et al., "A perceptual model applied to audio bit-rate reduction," *J. Audio Eng. Soc.*, vol. 43, no. 4, pp. 233–240, April 1995.

[61] B. P. et al., "PERCEVAL: Perceptual evaluation of the quality of audio signals," *J. Audio Eng. Soc.*, vol. 40, no. 1, pp. 21–31, January 1992.

[62] F. Baumgarte, "A nonlinear psychoacoustic model applied to the ISO MPEG Layer 3 Coder," in *99th AES Convention*, Audio Engineering Society, October 1995. preprint 4087.

[63] B. Carnero and A. Drygajlo, "Perceptual speech coding using time and frequency masking constraints," in *ICASSP-97*, vol. 2, pp. 1363–1366, 1997.

[64] S. Mitra, *Digital Signal Processing: A Computer Based Approach*. McGraw-Hill, 2nd ed., 2001.

[65] J. Proakis and D. Manolakis, *Digital Signal Processing: Pinciples, Algorithms, and Applications*. Prentice Hall, 3rd ed., 1996.

[66] M. Smith and I. T.P. Barnwell, "A procedure for designing exact reconstruction filter banks for tree-structured subband coders," in *ICASSP-84*, pp. 27.1.1–27.1.4, 1984.

[67] J. Arrowood and et al., "Filter bank design," in *The Digital Signal Processing Handbook* (V. Madisetti and D. Williams, eds.), CRC Press, 1998.

[68] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Processing*, vol. 40, no. 9, pp. 2207–2232, September 1992.

[69] C. Herley, "Wavelets and filter banks," in *The Digital Signal Processing Handbook* (V. Madisetti and D. Williams, eds.), CRC Press, 1998.

[70] B. Hubbard, *The World According To Wavelets*. Natick, MA: A K Peters, 2nd ed., 1998.

[71] I. Daubechies, "Where do wavelets come from? - a personal point of view," *Proceedings of the IEEE*, vol. 84, no. 4, pp. 510–513, April 1996.

[72] A. Abbate, C. DeCusatis, and P. Das, *Wavelets and Subbands: Fundamentals and Applications*. Boston: Birkhauser, 2002.

[73] A. Akansu and R. Haddad, "Fundamentals and optimal design of subband and wavelet transforms," in *Subband and Wavelet Transforms* (A. Akansu and M. Smith, eds.), Kluwer Academic Publishers, 1996.

[74] C. Burrus, R. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms, A Primer*. Prentice Hall, 1998.

[75] S. Mallat, "A theory of multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

[76] C. Taswell and K. McGill, "Algortihm 735: Wavelet transform algorithms for finite-duration discrete-time signals," *ACM Trans. on Mathematical Software*, vol. 20, no. 3, pp. 398–412, September 1994.

[77] B. Leslie and M. Sandler, "A wavelet packet algorithm for 1-d data with no block end effects," in *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI*, pp. 423–426, 1999.

[78] M. Wickerhauser, "Acoustic signal compression with wavelet packets," in *Wavelets: A Tutorial in Theory and Applications* (C. Chui, ed.), Academic Press, 1992.

[79] S. Chan and et al., "A hybrid coder using the wavelet transform," in *Proceedings of the IEEE International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 463–466, 1992.

[80] W. Kinsner and A. Langi, "Speech and image signal compression with wavelets," in *IEEE WESCANEX 93, Communications, Computers and Power in the Modern Environment Conference Proceedings*, pp. 368–375, 1993.

[81] A. Erdemir and et al., "Data compression using wavelet transforms and vector quantization," in *Proceedings of 1994 Midwest Symposium on Circuits and Systems*, pp. 965–968, 1994.

[82] Y. Karelic and D. Malah, "Compression of high-quality audio signal using adaptive filterbanks and a zero-tree coder," in *1995 Convention of Electrical and Electronics Engineers in Israel*, p. 3.2.4, 1995.

[83] R. Wannamaker and E. Vrscay, "Fractal wavelet compression of audio signals," *J. Audio Eng. Soc.*, vol. 45, no. 7/8, pp. 540–553, 1997.

[84] M. Black and M. Zeytinoglu, "Computationally efficient wavelet packet coding of wide-band stereo audio signals," in *ICASSP-95*, pp. 3075–3078, 1995.

[85] D. Pan, "A tutorial on MPEG/Audio compression," *IEEE Multimedia*, vol. 2, no. 2, pp. 60–74, 1995.

[86] T. Blu, "An iterated rational filter bank for audio coding," in *IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 81–84, 1996.

[87] W. Dobson and et al., "High quality low complexity scalable wavelet audio coding," in *ICASSP-97*, pp. 327–330, 1997.

[88] H. Dongmei and et al., "Complexity scalable audio coding algorithm based on wavelet packet decomposition," in *Proceedings of 5th International Conference on Signal Processing*, pp. 659–665, 2000.

[89] B. Leslie and M. Sandler, "Audio compression using wavelets," in *IEE Colloquium on Audio and Music Technology: The Challenge of Creative DSP*, 1998.

[90] P. Srinivasan and L. Jamieson, "High-quality audio compression using an adaptive wavelet packet decomposition and psychoacoustic modeling," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1085–1093, 1998.

[91] P. Philippe and et al., "On the choice of wavelet filters for audio compression," in *ICASSP-95*, pp. 1045–1048, 1995.

[92] P. Philippe and et al., "Wavelet packet filterbanks for low time delay audio coding," *IEEE Transactions on Speech and audio processing*, vol. 7, no. 3, pp. 310–322, May 1999.

[93] D. Sinha, "The perceptual audio coder (PAC)," in *CRC DSP Handbook* (V. Madisetti and D. Williams, eds.), CRC Press, 1998.

[94] M. Zurera and et al., "A new algorithm for translating psycho-acoustic information to the wavelet domain," *Signal Processing*, vol. 81, pp. 519–531, 2001.

[95] P. Kudumakis, *Synthesis and coding of audio signals using wavelet transforms for multimedia applications*. PhD thesis, King's College, London, April 1996.

[96] M. Erne and G. Moschytz, "Audio coding based on rate-distortion and perceptual optimization techniques," in *17th International Conference of the Audio Eng. Soc.*, pp. 220–225, 1999.

[97] B. Lim and Z. Ying, "Performance analysis of audio signal compression based on wavelet and wavelet packet transforms," in *International Conference on Information, Communications, and Siganl Processing*, September 1997.

[98] P. Chang and J. Lin, "Scalable embedded zero tree wavelet packet audio coding," in *IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications*, pp. 384–387, 2001.

[99] M. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*. A.K. Peters, 1994.

[100] O. Rioul and P. Duhamel, "A remez exchange algorithm for orthonormal wavelets," *IEEE Trans. Circuits Syst. II*, vol. 41, pp. 550–560, August 1994.

[101] G. Freeman, "Wavelet and fractal based methods for image compression," in *ICR Short Course Notes*, University of Waterloo, June 1998.

[102] S. Shlien, "Guide to MPEG-1 audio standard," *IEEE Trans. on Broadcasting*, vol. 40, no. 4, December 1994.