ANALYSIS AND IMPROVEMENT OF THE MPEG-1 AUDIO LAYER III ALGORITHM AT LOW BIT-RATES

by

Ramapriya Rangachar

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

ARIZONA STATE UNIVERSITY

December 2001

ANALYSIS AND IMPROVEMENT OF THE MPEG-1 AUDIO LAYER

ALGORITHM AT LOW BIT-RATES

by

Ramapriya Rangachar

has been approved

December 2001

APPROVED:

_____, Chair

_____

_____

Supervisory Committee

ACCEPTED:

_____

Department Chair

_____

Dean, Graduate College

ABSTRACT

In this digital era, music is "mobile" and the *de-facto* quality of compressed audio is expected to be indistinguishable from a Compact Disc (CD). But, high fidelity and a small bit-stream are two mutually exclusive properties. Designing a compression algorithm that obtains the best compromise between fidelity and size is a challenging undertaking. This has been a subject of extensive research for the past fifteen years and many algorithms, all of them incorporating sophisticated models of human perception, have been proposed. The most popular of these is the MPEG-1 Layer III Audio format, i.e., MP3 for the layman. The contributions of this study are two-fold. First, a graphical simulation tool, implementing the entire standard in MATLAB, has been developed. This is used to introduce perceptual audio coding concepts in senior undergraduate and graduate Digital Signal Processing (DSP) courses. The tool is accompanied by a series of computer experiments and exercises that can be used to provide hands-on training to class participants. The tool may also be used by instructors in a class setting to demonstrate key signal processing concepts associated with the processing of high-fidelity audio. The second contribution is a parametric enhancement model to improve the performance of the algorithm at low bit-rates. The algorithm has been engineered for low complexity. When the model is active, the resulting bitstream provides better spectral matching. Informal listening results indicate a perceptible improvement in signal quality. Also, the resulting bitstream is backward compatible.

In loving memory of *Pati*

ACKNOWLEDGEMENTS

I would like to express my gratitude to Dr. Spanias for having served as my advisor and for giving me the freedom to experiment and digress when necessary. His adherence to excellence has inspired me to bring out the best in myself.

Thanks are due to my colleagues at the Speech Lab and MIDL Lab. Working with them has been a rewarding experience.

My family has in no small part contributed to the culmination of this effort. I am indebted to them for their love, support and patience.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In the early days, researchers resorted to compression as a means to transmit information whose bandwidth was higher than the available channel capacity, for example, transmission of speech over telegraph cables. It was envisioned that as channel bandwidths increased, the need for compression would go away. On the contrary, even with the availability of optical fibers, DSL lines and DVD media, compression today is more important than ever. The main motivation for low bit rate coding is the need to minimize transmission and/or storage costs, the growing demand to transmit rich multimedia content over wireless and wired channels, and to support variable rate coding algorithms in packet-based networks. The compression task is complicated by the fact that the expectations of the consumer are high, for both the basic technology and value added services.

The advent of the Compact Disc (CD) in the early 1980s [12] set the standard for high-fidelity audio. With a sampling rate of 44.1 kHz and 16-bit precision per sample, it truly provided wideband stereo audio, but at the price of a very high bit-rate – about 4.32 Mb/s, with error correction. It has been shown [16] that audio signals have a considerable redundancy and the average entropy is below 2 bits per sample. This has motivated extensive research in compression strategies based on sophisticated models of human auditory perception. It is to be noted that perceptual coding strategies for wideband audio borrow a lot from speech coding in that they try to shape the spectrum of the quantization noise to follow the signal spectrum, only more explicit and involved.

The primary requirement in the design of audio coders is to retain a high perceptual quality of the reconstructed signal with robustness to variations in spectra and levels. Spatial integrity is an additional dimension of quality for stereophonic and multichannel signals. Bandwidth scalability, algorithm-complexity, coding delay and power consumption are vital. Robustness to channel errors (random and burst errors), packet losses, tandeming and transcoding are important in broadcast applications. Graceful degradation of quality in the presence of increased channel errors is also a very important design consideration. For professional applications, the bit-stream syntax should provide for editing, fading, mixing and dynamic range compression.

The state-of-the-art encoders available today [30] [37] [82] go to great lengths to extract perceptual redundancies and can deliver perceptually transparent quality for a wide range of stereo material at modest bit-rates –128 kb/s. There are narrowband coders for network applications that operate at bit-rates as low as 1 bit per sample and provide moderate quality [55] [70]. Expectations over the next decade are that the rates can be reduced by a factor of four.

1.1 Statement of the Problem and Thesis contribution

The MPEG-1 Layer III algorithm, commonly referred to as MP3, was among the first algorithms to be standardized. It has grown to become the most popular and widely used vehicle for delivery of audio over the Internet and in (digital) personal music systems.

A software simulation tool, implementing the MPEG-1 Layer III algorithm as defined in the standard [30] has been developed, primarily as an interactive interface for introducing perceptual audio coding to both novices and advanced students. MATLAB is the tool of choice for implementing the algorithm since it provides a flexible programming syntax along with advanced tools for mathematical analysis, video rendering, memory management, debugging and profiling. The platform-independence of the syntax resolves issues related to specific operating systems, if any. This helps to focus on mathematical and algorithmic intricacies rather than mundane programming issues. An intuitive user-interface helps to walk the reader through the important aspects of the algorithm and also provides ample visual results to enforce the theory. The advanced reader can study the code more closely and can track the finer details by observing and displaying intermediate results.

To improve the performance of the algorithm at low bit-rates, an enhancement layer incorporating a parametric model based on sinusoids and noise is proposed and shown to produce results that are perceptually more pleasing results then the standard algorithm.

1.2 Organization of the Thesis

This chapter presents brief overview of audio coding techniques. The rest of the thesis is organized as follows. To the average student in electrical engineering, who has no background in auditory psychophysics, Chapter 2 serves as a concise introduction to psychoacoustics. Chapter 3 gives an overview of transforms and time-frequency analysis in the context of audio compression. Chapter 4 serves to complement Chapters 2 and 3 by

applying signal processing tools and models to real audio data. Chapter 5 has a detailed analysis of the MPEG-1/Audio Layer III algorithm. Chapter 6 gives a brief description of the tool developed. The enhancement algorithm is proposed in Chapter 7 and conclusions are drawn in Chapter 8.

1.3 An Overview of Audio Coding Techniques

As in speech, audio compression techniques can be broadly classified into three main categories: parametric, waveform and hybrid coders. This section reviews each one of them briefly.

1.3.1 Parametric Coding

The availability of an acceptable model for speech production has resulted in making parametric coders or vocoders (voice coders) very popular in speech coding. Since wideband audio covers diverse categories, everything from chamber music to punk rock, there are no high precision models for audio compression. Parametric audio coders perform satisfactorily for low bit-rate applications. Of course, there are parametric tools that address particular aspects of creation and rendering of musical content.

For delivery of audio content over low bandwidth channels, the MPEG-4 standard (ISO/IEC 14496) provides parametric audio coding tools [74]. The HILN (Harmonic and Individual Lines plus Noise) coder [79] can code audio at bit-rates of 4 kb/s and above using a parametric scheme. The audio signal is essentially decomposed into underlying individual sinusoids, harmonics and noise the parameters of which are coded based on perceptual importance. As the parameters are coded as frequencies and amplitudes, this permits pitch and time-scale modification without additional tools. The HILN and HVXC

(Harmonic Vector Excited Coder) speech coder [57] can be combined to support a wider range of audio material at a variety of bit-rates.

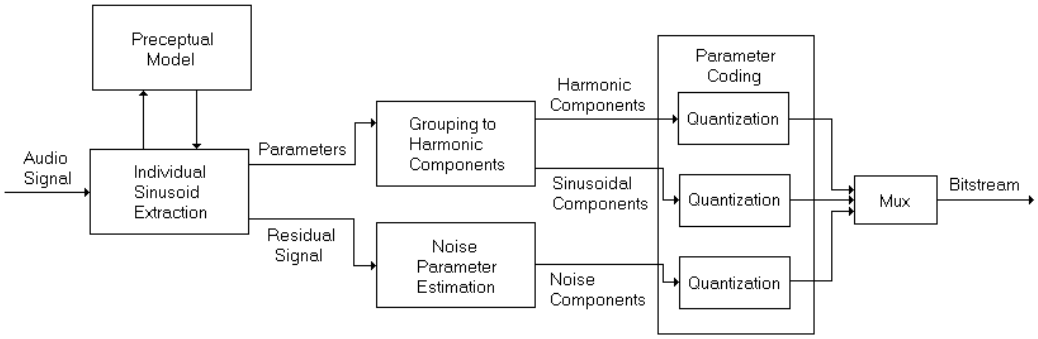


Fig. 1.1 HILN Encoder.

In 1983, the major electronic-instrument manufacturers adopted the Musical Instrument Digital Interface (MIDI) for interconnectivity of electronic instruments. Much in the same way that two computers communicate via modems, two synthesizers communicate via MIDI. The information exchanged between two MIDI devices is musical in nature. MIDI information tells a synthesizer, in its most basic mode, when to start and stop playing a specific note. Other information shared includes the volume and modulation of the note, if any. MIDI information can also be more hardware specific. It can tell a synthesizer to change sounds, master volume, modulation devices, and even how to receive information. In more advanced applications, MIDI information can to indicate the starting and stopping points of a song or the metric position within a song. More recent applications include using the interface between computers and synthesizers to edit and store sound information for the synthesizer on the computer [42].

Csound [76] is a synthesis specification language for music synthesis developed by Dr. Barry Vercoe at the MIT Media Lab, in which sound streams are described by

decomposition into a sound-specification description representing arbitrarily complex signal processing algorithms, and event lists comprising scores or MIDI files. Given two inputs, a score (contains the description of 'notes' and timely events in the composition) and an orchestra (contains a description of how the various instruments will sound like), the Csound engine generates sound (through a file or real-time output) that is a rendering of the score by the orchestra.

NetSound [48] is a sound and music specification protocol oriented towards networked low-bandwidth, native-signal-processing sound synthesis applications like music distribution on the Internet. As a network sound transmission protocol, NetSound has the advantage of being able to transmit a wide selection of sounds using a descriptive format that does not require a high-bandwidth channel. Since description-based audio represents acoustic events as parameterized units, a great deal of control over the resulting sound is offered. In order to time-compress a sound stream, for example, a scalar multiplier can be applied to all event duration values, or a synthesis algorithm such as phase-vocoder resynthesis can be specified and appropriate time-frequency modifications made from a simple control function. The use of complex instrument descriptions and appropriately parameterized score makes it possible to specify descriptions of complete sound tracks or musical pieces using a very small amount of data. Other synthesis languages' instruments, such as the MUSIC-N languages, and commercial synthesizer implementations can be translated into Csound syntax. On the note level, NetSound has its own event-specification format but is also capable of reading and playing MIDI files.

The process of designing a sound stream using NetSound comprises the specification of the required sound synthesis algorithms or selection from pre-existing synthesis units, such as wavetable synthesis, FM synthesis, phase-vocoder or additive synthesis. A standard sequencing program is used to construct the temporal structure of the required sound stream as a MIDI file or the readable Csound score format. Sound streams are computed in real time and synthesized buffer-by-buffer by a network client- i.e. an executable on the network user's computer. The resulting audio sample data is not stored or transmitted; only the descriptions and the necessary sampled sounds or synthesis data are stored and transmitted by the network server. It is important to note that NetSound is not a compression protocol; the process does not include a transcription from mixed audio to NetSound format. NetSound can be considered as a distribution tool that reflects the manner in which music and sound tracks are constructed for multimedia applications. That is, a small number of sounds or algorithms are utilized for generating a large amount of audio data. NetSound renders the data into sound without requiring large storage or throughput capacity.

The MPEG-4 standard [82] introduces Structured Audio tools [64] [73] for describing semantic and/or model-based representations of multimedia content. This has applications in Internet karaoke, virtual gaming, multimedia presentations, and effects processing primitives like filters, reverbs and chorus effects – to help render an "audio scene" that is made up of both natural and synthetic audio objects. This is discussed in more detail in Ch. 4.

1.3.2 Waveform Coding

As the name suggests, waveform coders try to produce a signal that matches the input waveform as closely as possible. They do not rely on explicit source models and generally perform well for both speech and audio content. From a signal processing perspective, they can be further divided into Time Domain and Frequency Domain coders.

1.3.2.1 Waveform Coders

Waveform coders operate on time domain data. Typical examples are Pulse Code Modulation (PCM), Adaptive Pulse Code Modulation (ADPCM), Delta Modulation (DM), Adaptive Delta Modulation (ADM) and Adaptive Predictive Coding (APC).

PCM is widely used for quantizing speech and audio. The CD uses linear PCM with 16-bit resolution to store music. Non-uniform PCM algorithms (ITU G.711) like the A-law and μ-law that quantize the linear PCM sample into 8 bits using a logarithmic quantizer. The DPCM coder extracts correlation between adjacent samples and quantizes the difference between. The DPCM decoder reconstructs the signal by adding the difference signal to the predicted signal. The ADPCM coder adapts the predictor and quantizer to the local statistical characteristics of the signal.

1.3.2.2 Frequency-Domain Coders

There is a class of coders that operate on a frequency-domain representation of the signal. Frequency domain coders can be further classified as subband and transform coders. Subband coders divide the input signal into subbands and allocate bits to each subband independently. At the decoder, the subbands are recombined to reconstruct the

signal. Transform coders operate on blocks of data and identify statistical redundancies by the application of a high-resolution transform. They can be considered to be subband coders with a large number of bands. The most successful audio coders operate in the transform domain as it permits easier tracking of the signal characteristics and provides for complete control of noise shaping over the entire spectrum.

1.3.3 Hybrid Coding

Hybrid coders blend the best of both the parametric and waveform matching coders. A hybrid coder consists of a parametric core that models the signal to an extent. The un-modeled part or residual is transmitted as side information to the receiver. The receiver uses this information to enhance the quality of the reconstructed signal.

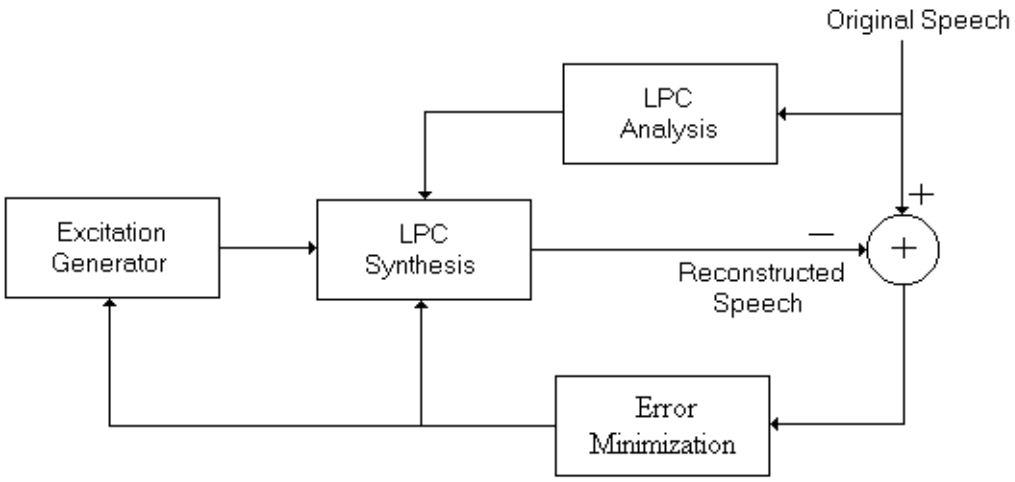


Fig. 1.2 Generalized LPAS.

Systems based on the Analysis-by-Synthesis paradigm fall into this category, where, the decoder is embedded within the encoder. These systems have parametric cores that model (analyzes) the signal. Based on a predefined distance metric (mean square error, perceptual error), a matching pursuit algorithm is used to obtain a model that is

close to the original. Generalized Linear Predictive Coding by Analysis-by-Synthesis (Generalized LPAS) employing linear predictive analysis is the most prevalent analysis-by-synthesis method in speech coding. The Generalized LPAS system for speech can be depicted as shown in Fig. 1.1. Code Excited Linear Prediction (CELP) and Multi-Pulse Linear Predictive Coding (MP-LPC) are state-of-the-art speech coders employing LPAS. The MPEG-4 standard provides for CELP based speech coders that can also be used for low bit-rate audio coding applications [74].

1.3.4 Perceptual Coding

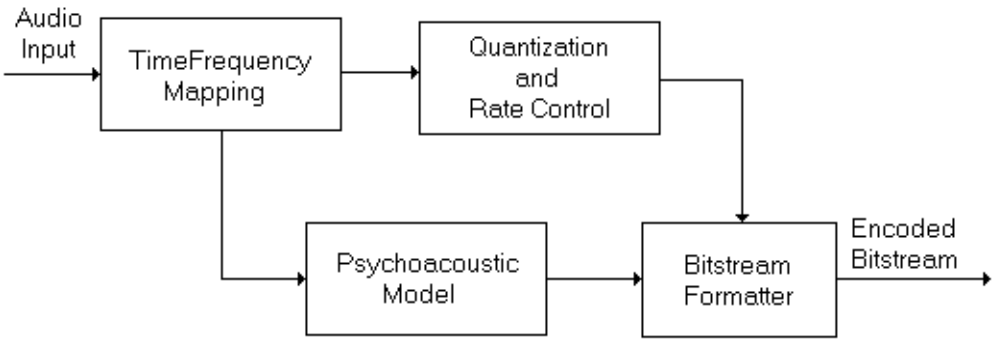


Fig. 1.3 Generalized Perceptual Audio Coder.

Perceptual audio coders take into account explicit models of human perception for purposes of quantization and noise shaping. As a result, they extract both statistical and perceptual redundancies in the signal. Wideband audio coders that fall into this category are basically transform coders that employ high resolution transforms (> 500 lines) to extract statistical redundancies in the signal. The basic structure of a perceptual audio coder is depicted in Fig. 1.3. Parametric audio coders like the HVXC and HILN coder also that have quantization strategies based on perceptual criteria.

CHAPTER 2

HEARING, PERCEPTION AND PSYCHOACOUSTICS

An understanding of the human auditory mechanism is the key to developing a working model for it. It is known that the ear has finite resolution in both time and frequency domains. This is to be expected as the neuro-mechanical processes and the higher level cognitive processes do take finite amounts of time to respond to stimuli. It can be observed that some characteristics of the sound reaching the ears are rendered inaudible, and are masked. For example, a weak signal (maskee) in the temporal or spectral proximity of a stronger signal (masker) can be masked (rendered inaudible/drowned out) [1] [2]. The goal of an engineering model for hearing is to quantify the limitations of the human ear, so that perceptual irrelevancies can be extracted.

Hearing is the process by which sound is received and converted into nerve impulses. Perception implies the post-processing within the brain by which sounds are heard and interpreted and given meaning.

2.1 Hearing

The sensation of hearing results from the interaction between the neuro-mechanical processes in the ear, the higher processes along the auditory pathway and the brain. The anatomy of the ear can be divided into three main parts: the outer ear, the middle ear and the inner ear, as depicted in Fig. 2.1.

The outer ear consists of a convoluted cartilage (pinna), the external canal (external auditory meatus) and the eardrum (tympanic membrane). The pinna protects the

opening of the ear; its convoluted shape is thought to provide directional cues. The external auditory meatus is an approximately cylindrical tube, about 2.7 cm long and 0.7 cm in diameter. As a result of being tube-like, it has many resonant modes, one of which is approximately 3 kHz, falls in the frequency range of speech. The tympanic membrane is a stiff, conical membrane at the end of the external auditory meatus. It vibrates in response to the sound impinging on it and is the first link in the chain of structures that form the biomechanical sound transducer.

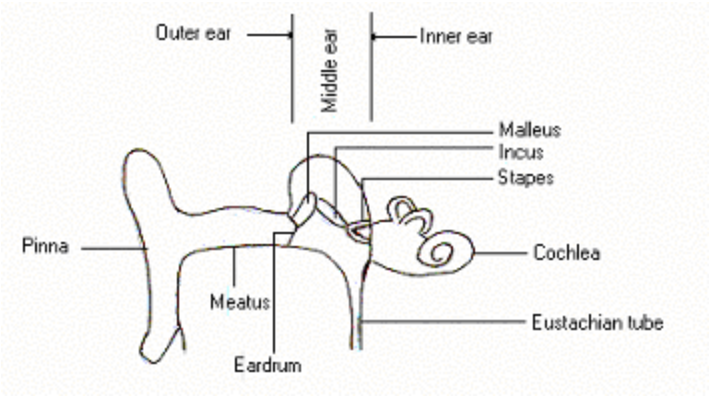


Fig. 2.1 Cross-section of the human ear.

The middle ear is an air-filled cavity. It connects to the inner ear by two apertures called the oval and round windows. It is also connected to the outside world through the eustachian tube, which permits the equalization of sound pressure between the middle ear and the surrounding atmosphere. The middle ear consists of three tiny bones or ossicles viz., hammer (malleus), anvil (incus) and stirrup (stapes). The ossicles provide acoustical coupling between the oval window and the tympanic membrane. The function of the ossicles is two-fold: impedance matching and amplitude limiting.

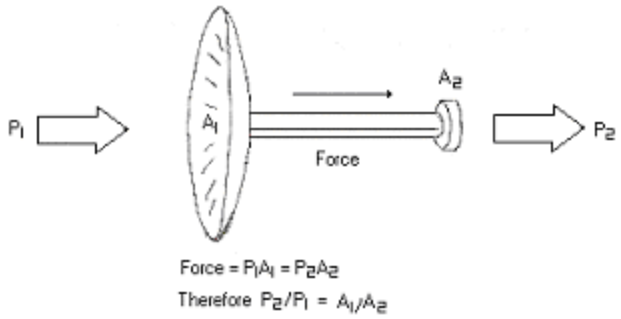


Fig. 2.2 The impedance transformation effected by the middle ear.

The vibrations of the tympanic membrane must be transduced by the inner ear for the sensation of hearing to occur. If the airborne sound were to be incident on the oval window directly, much of the sound would be reflected due to impedance mismatch. The middle ear provides the necessary impedance transformation for maximum power transfer. The ossicles act as a lever system such that the large displacement of the tympanic membrane results in a smaller displacement of the oval window but with greater force, as shown in Fig. 2.2.

The inner ear consists of the vestibular apparatus, the round and oval windows and the cochlea. The vestibular apparatus comprises the semicircular canals and associated organs, used for balance and sensing orientation and of no interest here. The cochlea is a snail-shaped organ connected to the middle ear via the round and oval windows. It contains the neural transducers that convert acoustical vibrations into nerve firings. If the cochlea were uncoiled, it looks like Fig. 2.3. The widest part is called the base and the narrowest part at the opposite end is called the apex. In cross-section, a single turn would like Fig. 2.4. The cochlea is divided down its middle by a partition bounded by a flexible sheet called the basilar membrane (BM) and by a thinner

membrane called Reissner's membrane. The partition divides the cochlea into two passages, the scala vestibuli and the scala tympani. The two passages are connected to each other at the apex of the cochlea by an opening called helicotrema. The acoustical energy enters the cochlea by way of the round window, which is driven by the stapes. The acoustical energy is converted into fluid pressure variations, which travel down the scala vestibuli, through the helicotrema into the scala tympani and finally exits by way of the round window. As the BM is in series with this fluid motion, it is driven by it. The resonance at any particular point of the BM is a function of the frequency of the stimulus and hence an indicator of the spectral content of the stimulus. High frequencies cause resonance near the oval window while low frequencies cause resonance near the apex. The distance from the apex where maximum resonance occurs is a logarithmic function of frequency. Thus the BM acts as a neuro-mechanical spectrum analyzer. This is known as the Place Theory of frequency resolution.

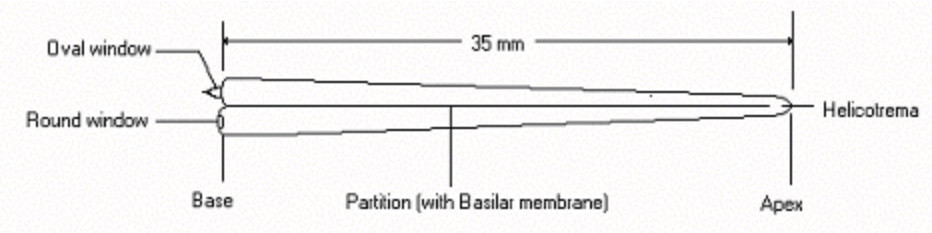


Fig. 2.3 The uncoiled cochlea.

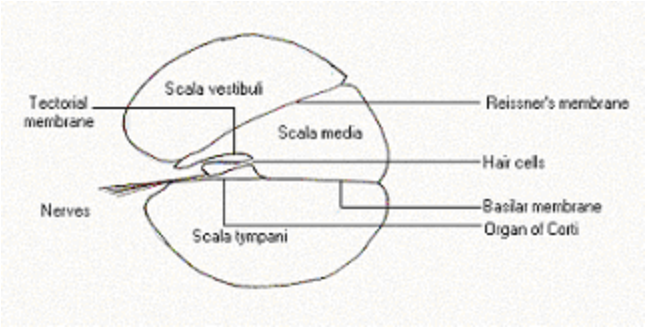


Fig. 2.4 One turn of the cochlea in cross-section.

2.2 Perception

Determining the performance of the human auditory system is one of the classical experiments in psychology. The ear can detect sound pressure variations as low as $2x10^{-5}$ Pascal r.m.s. This is called as the *threshold in quiet* or *absolute threshold of hearing* and used as a reference against which sound pressure level (SPL) is measured. The dynamic range of the ear is about 130dB SPL. At the high end, sound turns to pain, while at the low end it becomes silence. This function exhibits a strong dependency on frequency and is well approximated by the function

$$Tq(f) = 3.64\left(\frac{f}{1000}\right)^{-0.8} - 6.5e^{-0.6\left(\frac{f}{1000}-3.3\right)^2} + 10^{-3}\left(\frac{f}{1000}\right)^4 \quad (dB\ SPL) \qquad (2.1)$$

Perceived loudness is a function of both frequency and level. By comparing tones at different frequencies and amplitudes, contours of equal subjective loudness can be found. These contours take the general shape shown in Fig. 2.6. These are sometimes called as Fletcher-Munson (1933) curves. The frequency range of human hearing is approximately 16 Hz to 16 kHz. The upper limits falls off with increasing age, among the young it occasionally reaches 20 kHz.

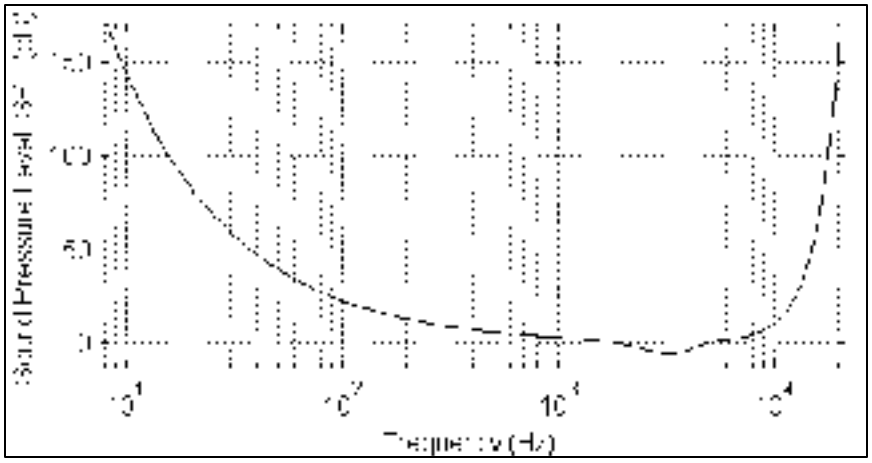


Fig. 2.5 The Absolute Threshold of Hearing.

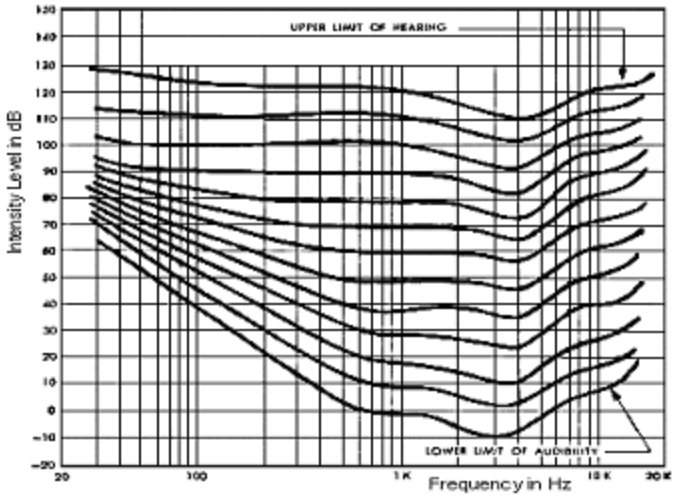


Fig. 2.6 Contours of equal loudness.

2.3 Frequency Selectivity and the Critical Band

According to the *Place Theory*, every point on the BM responds only to a particular frequency. The frequency that gives maximum response at a particular point on the BM is known as the *Characteristic Frequency* (CF) for that place. But, the vibration of the membrane to a single frequency cannot be localized to an infinitely small area, and

nearby areas also show response to the same frequency, but with an amplitude that decreases with distance. So, each point on the BM can be considered as a band-pass filter with a certain center frequency (corresponding to the CF) and a bandwidth. The bandwidth of these 'auditory filters' is not constant but increases with CF, but the Q factor of these filters remains approximately the same

Fig. 2.7 shows the uncoiled BM with the apex on the left so that the logarithmic frequency scale can be applied. The envelope of displacement and hence the position of maximum displacement (i.e., CF) of the BM for two different frequencies is different. Also, since the BM is continuous, the CF is infinitely variable it allows for extremely good pitch discrimination of about one-twelfth of a semitone, this limit being set by the distance between individual hair cells. It is to be noted that the envelope of vibration is asymmetrical about the CF due the fact that the BM is tapering and also due to the frequency-dependent losses in the cochlear fluid as flows through the scalae. The envelope of vibration is also affected by the intensity of the sound stimulus.

The frequency selectivity of the auditory nerve fibers (that synapse with the hair cells) can be quantified in the same manner as the BM. Every nerve fiber is assumed to derive its output from a particular point on the BM. Frequency selectivity of individual nerve fibers is indicated by *Frequency Threshold* C*urves* (FTC) or 'tuning curves'. FTCs show the threshold of each fiber as a function of frequency. On a logarithmic frequency scale, these curves are steeper on the high-frequency side than the low-frequency side-analogous to the envelope of vibration of the BM. At the CF, the threshold of the fiber is lowest and the corresponding point on the BM exhibits resonance. In response to a

complex stimulus, different points on the BM would vibrate simultaneously, in response to the individual component frequencies of the signal. The envelopes of vibration would also depend on the intensity of these component frequencies, as shown in Fig. 2.8. Thus the envelope of basilar vibration is a complex function. If the complex has two very closely spaced frequency components, it would excite two very closely spaced points on the BM. In such a case, the vibration of the BM is influenced by the stronger of the two components - the weaker of the two components is *masked.* The response of the auditory neurons to a complex stimulus is analogous to that of the BM.

The finite width of the envelope of vibration of the basilar membrane is called as *Critical Bandwidth* (CB). Thus each point on the basilar membrane has a CF and a corresponding *critical band*.

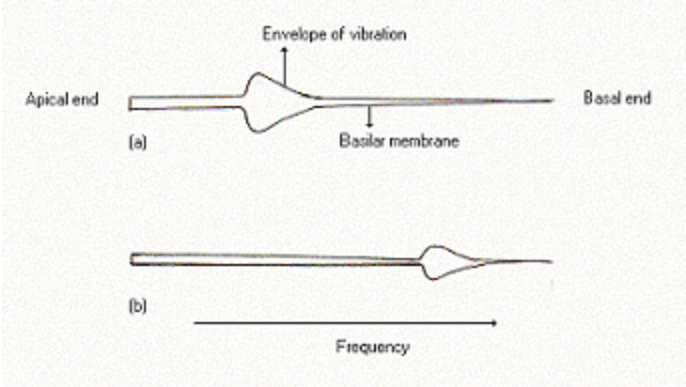


Fig. 2.7 The basilar membrane uncoiled. (a) Vibration envelope for a single frequency. (b) Vibration envelope for a higher frequency.

One approach to identify the CB is the following setup: a narrowband noise source is presented at a constant intensity and the width of the noise spectrum is gradually increased (effectively increasing the power of the noise). The idea here is that the perceived loudness of the stimulus will increase when the width of the noise spectrum

exceeds that of a critical band. So, loudness remains constant as long as the noise energy is restricted to one critical band and increases as the energy spreads into adjacent critical bands.

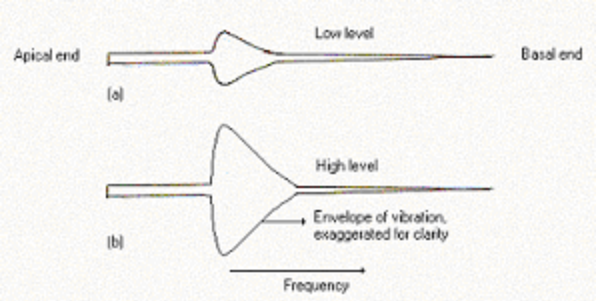


Fig. 2.8 The effect of SPL on critical bandwidth.

CBs can also estimated by masking experiments. The detection threshold for a narrowband noise presented between two tonal maskers is constant as long as the frequency separation of the maskers is within one critical band and drops rapidly otherwise. The efficacy of detecting the probe (the narrowband noise in this case) can be explained in terms of the SNR criterion. As long as the tonal maskers are within the critical band, the SNR presented to the auditory filter is constant and so is the threshold for the probe. As the maskers move out of the pass-band of the auditory filter, the SNR for the probe improves (note that the noise-probe is the target while the tonal maskers are the 'noise' in the conventional sense) and the detection threshold falls.

A notched-noise experiment, with the roles of the masker and maskee reversed, can also be devised. And similar conclusions can be drawn.

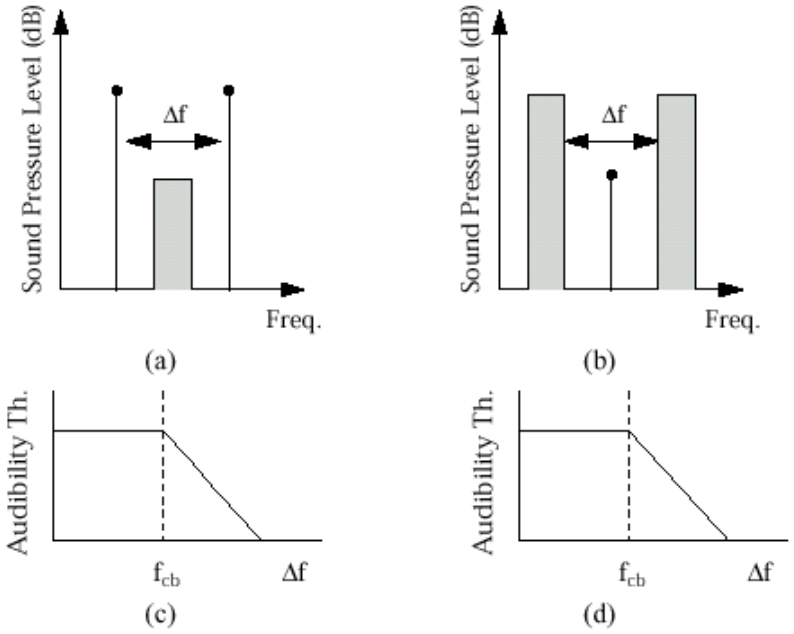


Fig. 2.9 Critical Band Measurement procedures for tonal maskers (a, c) and noise maskers (b, d), after [78].

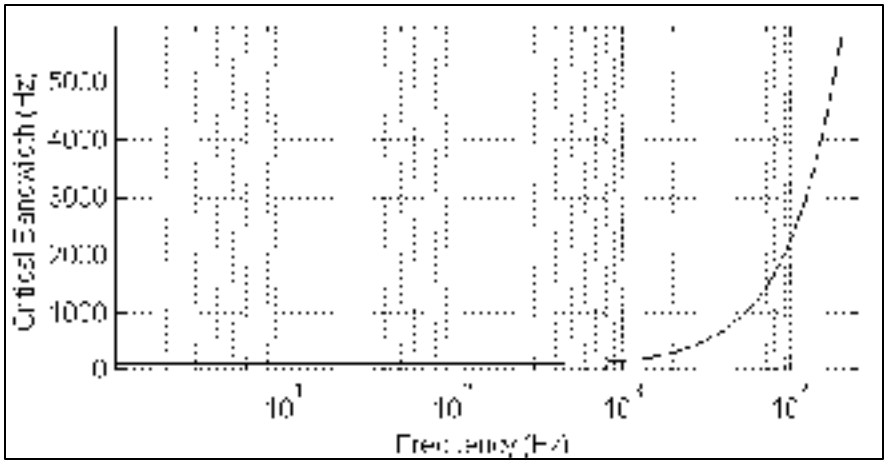


Fig. 2.10 Critical bandwidth.

As discussed earlier, the CB of the auditory filters increases with CF, but the Q factor remains approximately the constant. The CBs are less than 100 Hz at the lowest audible frequencies and more than 4 kHz at the highest. They can be approximated by

$$BWc(f) = 25 + 75\left[1 + 1.4\left(\frac{f}{1000}\right)^2\right]^{0.69} \quad (Hz) \tag{2.2}$$

In practical applications, the ear is modeled to have a fixed number of CBs, by a discrete version of $BWc(f)$. The distance of one CB is referred to as one *Bark*. Conversion from the linear frequency scale to the Bark scale is effected by the function

$$z(f) = 13\arctan(.00076f) + 3.5\arctan\left[\left(\frac{f}{1000}\right)^2\right] \quad (Bark) \tag{2.3}$$

2.4 The Masking Phenomenon

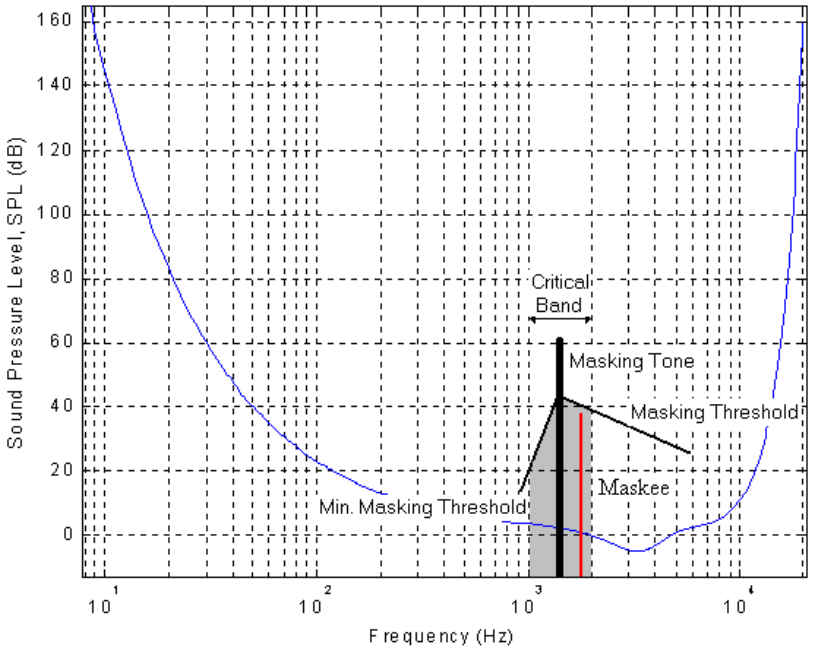


Fig. 2.11 Schematic representing Simultaneous Masking (not to scale).

Masking is a phenomenon by which weaker signals in the spectral or temporal proximity of stronger ones are rendered inaudible. *Simultaneous masking* is used to

describe circumstances where the masker is present throughout the presentation time of the signal.

Audio records contain complicated spectra and a combination of innumerable masking scenarios. In the framework of audio compression, where compression can be achieved by preferentially discarding perceptually irrelevant spectral components, the maskers are strong spectral components of the music, while the maskee is either weak spectral components or quantization noise. For such applications, the following scenarios are most useful.

2.4.1 Noise-Masking-Tone (NMT)

A narrow-band noise signal of bandwidth 1 Bark masks a tone in the same CB. Signal-to-Mask Ratio (SMR) is minimum when the frequency of the tone is equal to the center frequency of the noise. The minimum SMR is in the range of –5 to +5 dB.

2.4.2 Tone-Masking-Noise (TMN)

A pure tone occurring in the center of a CB masks noise of any sub-critical bandwidth or shape. It has been found that the SMR is minimum when the frequency of the tone is close to the center frequency of the probe noise. The minimum SMR tends to be in the range of 21 – 28 dB.

2.4.3 Noise-Masking-Noise (NMN)

Noise-masking-noise scenarios, where one narrow-band noise masks out another, are more difficult to characterize due to the complicated phase relationships between the masker and maskee. Some results have shown about 26 dB SMRs.

The effect of masking is not only felt in the current CB, but also in the adjacent bands. Based on psychoacoustic results, this inter-band masking is a function of the frequency and level of the masker, as indicated in Fig. 2.7 and Fig. 2.8. In the Bark frequency domain, this can be approximated by a triangular function, with a steep slope on the low frequency side and a shallow slope on the high frequency side. The spread of masking, represented by a triangular function that is independent of level and frequency is given by

$$SpF(z) = 15.81 + 7.5(z + 0.474) - 7.5\left[1 + (z + 0.474)^2\right]^{0.5} \tag{2.4}$$

There are other models that account for the level dependence of the masker.

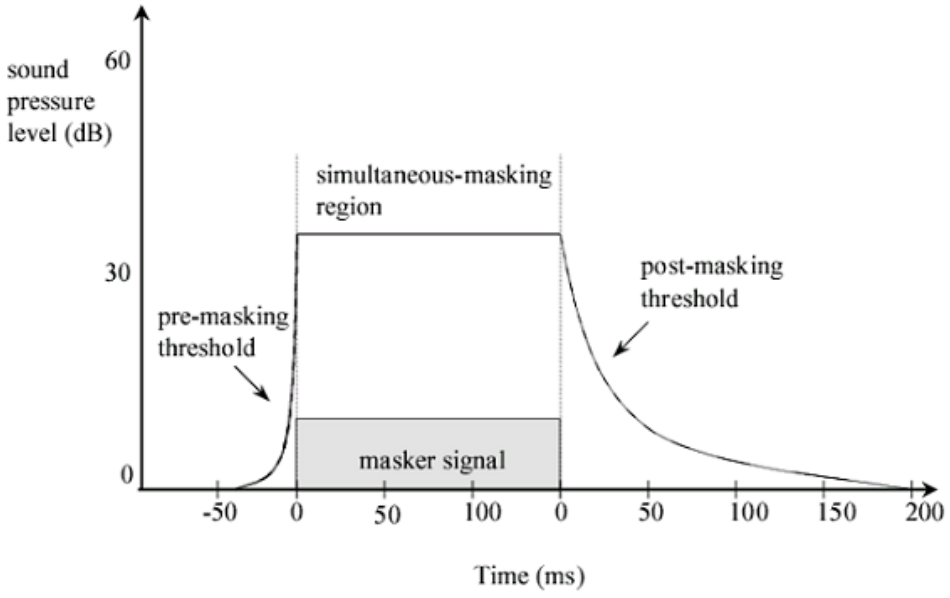


Fig. 2.12 Temporal masking in the human ear [78].

Being a physical mechanism, the BM vibration takes finite time to reach a steady-state response and also to stop responding. As a result, it is difficult to detect short

interruptions to a continuous tone – masking can take place even when the masking tone begins after and cease before the masked sound. This is referred to as *forward* and *backward masking* respectively; they fall under the category of *Temporal Masking*.

CHAPTER 3

SIGNAL PROCESSING WITH LAPPED TRANSFORMS

Transformations are a very powerful tool for signal compression. Unitarity, energy compaction, signal de-correlation, critical sampling and perfect reconstruction are some of the desirable properties that motivate coding in the transform domain.

A unitary transform preserves energy. Signal de-correlation and energy compaction are the basis of all compression schemes. For subband based compression methodologies, critical sampling is very essential requirement to stem the data rate at the output of the analysis filterbank. Perfect reconstruction (PR) ensures that the signal can be recovered exactly, in the absence of quantization noise. In lossy compression schemes especially, where transform coefficients are preferentially discarded, it is essential that the retained coefficients be reconstructed exactly. Choosing the right transform provides one or more of these properties.

3.1 Lapped Transforms

In traditional transform domain coders, a block of samples of a signal $x(n)$ are transformed by application of a transformation matrix H, whose rows contain the basis-vectors. In matrix notation

$$X = Hx \tag{3.1}$$

where X is the transform coefficients, H is the transformation matrix applied to the signal in the vector x. In practical applications, x needs to be windowed to mitigate the boundary effects of the transform. Also, overlapping between adjacent blocks is common to avoid discontinuities in the reconstructed signal at the transform boundaries.

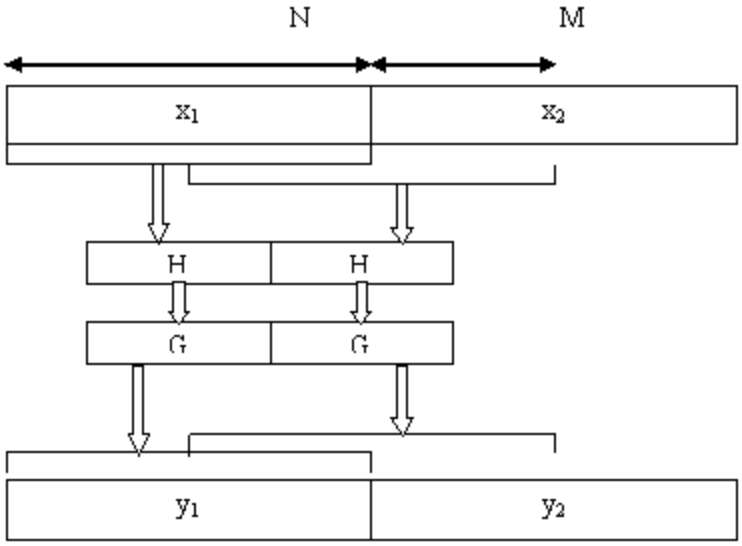


Fig. 3.1 Lapped Transform with 50% overlap.

The primary aim of designing and employing lapped transforms [25] [11] is the reduction of blocking artifacts. The basis functions of lapped transforms are longer than the length of the transform itself. As a result, the basis functions (and the samples) of adjacent blocks overlap to reduce blocking artifacts. More importantly, they achieve a higher coding gain.

To compute a lapped transform of length M using a basis with time support N, it is necessary to create a dataset consisting of M new samples and N-M previous samples.

Fig. 3.1 represents the computation of a lapped transform with 50% overlap. Block x_1 can be transformed into the transform domain as defined in Eq. 3.1, except that, now, the transformation matrix H is $M \times 2M$ and each x_i is $2M \times 1$ Dividing H into two $M \times M$ matrices and x_1 into two $M \times 1$ matrices, we can denote the result as

$$x_1 = H_1 x_1^1 + H_2 x_1^2 \tag{3.2}$$

Similarly, block x_2 can be represented as

$$x_2 = H_1 x_2^1 + H_2 x_2^2 \tag{3.3}$$

On the synthesis side, the $2M \times 1$ reconstructed signal y_1 can be represented in matrix notation as

$$y = GX \tag{3.4}$$

As before, splitting each output vector into two sub-vectors results in

$$y_1 = y_1^1 + y_1^2 = G_1 X_1 + G_2 X_1 \tag{3.5}$$
$$y_2 = y_2^1 + y_2^2 = G_1 X_2 + G_2 X_2 \tag{3.6}$$

The reconstructed signal in the overlapping parts of y_1 and y_2 can be expressed as

$$\begin{aligned} y_{overlap} &= y_1^2 + y_2^1 \\ &= G_2 X_1 + G_1 X_2 \\ &= G_2 \left[H_1 x_1^1 + H_2 x_1^2 \right] + G_1 \left[H_1 x_2^1 + H_2 x_2^2 \right] \\ &= G_2 H_1 x_1^1 + G_2 H_2 x_1^2 + G_1 H_1 x_2^1 + G_1 H_2 x_2^2 \end{aligned} \tag{3.7}$$

It is to be noted that

$$x_1^2 = x_2^1 = x_{overlap} \tag{3.8}$$

For perfect reconstruction, the sum of overlapping parts $y_{overlap}$ should return the corresponding part of the input signal. This results in the following constraints:

$$G_2 H_1 = G_1 H_2 = O_M$$
$$\textit{and} \qquad (3.9)$$
$$G_1 H_1 + G_2 H_2 = I_M$$

where O_M is an $M \times M$ zero matrix and I_M is an $M \times M$ identity matrix.

In the special case when $G = H^t$ we have

$$H_2^t H_1 = H_1^t H_2 = O_M$$
$$\textit{and} \qquad (3.10)$$
$$H_1^t H_1 + H_2^t H_2 = I_M$$

This is referred to as a *Lapped Orthogonal Transform* (LOT). In this case, H_1 and H_2 are orthogonal. Therefore, the overlapping parts of the basis functions are also orthogonal.

3.1.1 Filterbank Interpretation of the Lapped Transforms

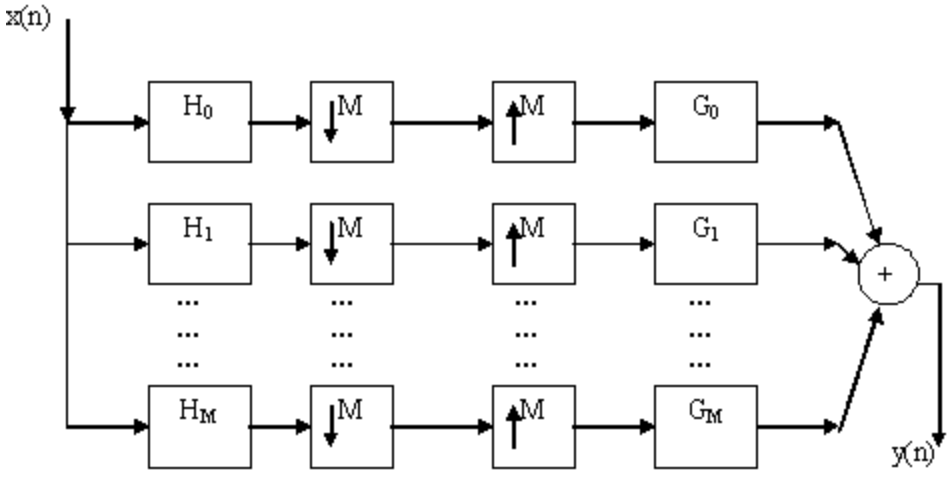


Fig. 3.2 Filterbank interpretation of the Lapped Transform.

The lapped transform can also be interpreted as a filterbank by considering each row of the analysis matrix to be the time-reversed impulse responses of a bank of

bandpass filters. The synthesis filterbank is made up of the columns of G. Such a filterbank interpretation of the lapped transform is given in Fig. 3.2.

3.2 Modulated Lapped Transforms (MLT)

By modulating a low-pass prototype filter, the design of the analysis and synthesis filterbanks can be simplified. Modulated Discrete Cosine Transforms (MDCT) or Modulated Lapped Transforms (MLT) is a family of lapped transforms generated by modulating a lowpass prototype filter. The basis functions of the MDCT have a length $N = 2M$, where M is the number of subbands. Perfect reconstruction can be achieved by an appropriate choice of the phase of the modulating cosine function and the lowpass prototype window.

Given a lowpass window $h(n)$, the MDCT basis functions are defined by the equation

$$H_i(n) = h(n)\sqrt{2/M}\cos\left[\left(n+\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{p}{M}\right] \qquad (3.11)$$
$$k = 0,1,2,\ldots,M-1$$
$$n = 0,1,2,\ldots,2M-1$$

$$h(n) = \sin\left[\left(n+\frac{1}{2}\right)\frac{p}{2M}\right] \qquad (3.12)$$

The frequency response of the MDCT for the sine window defined by Eq. 3.12 is shown in Fig. 3.3.

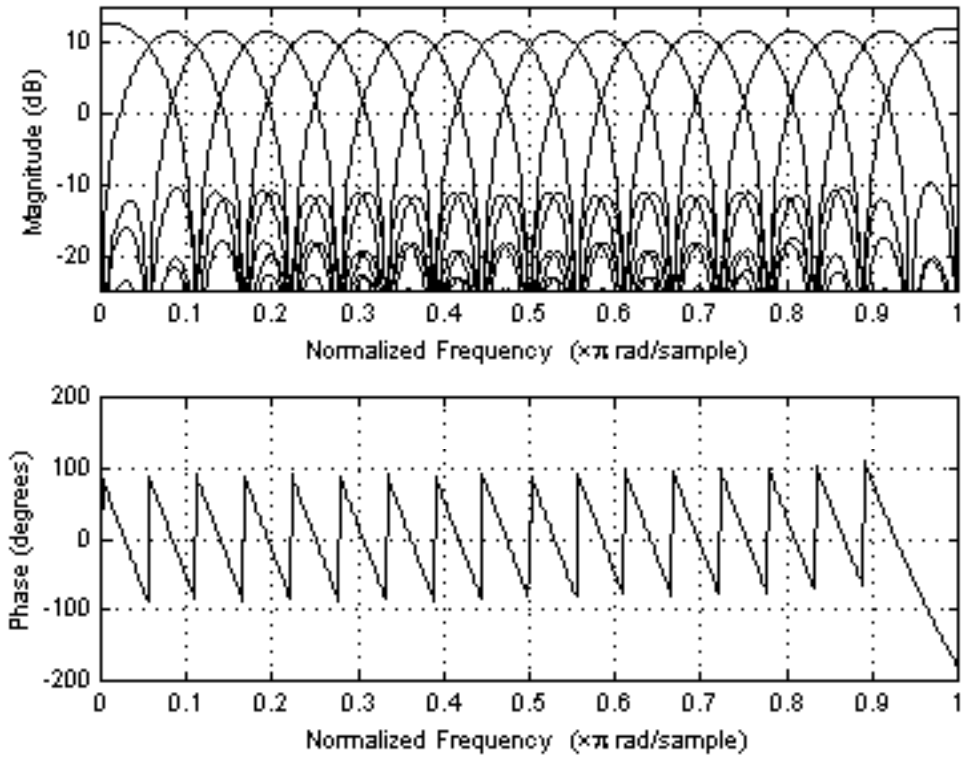


Fig. 3.3 MDCT Frequency Response (M=18).

3.2.1 Perfect Reconstruction Conditions for an MDCT

Given an analysis window $h(n)$, the output of the analysis filterbank can be represented as

$$X(k) = \sqrt{\frac{2}{M}} \sum_{n=0}^{N-1} x(n)h(n) \cos\left[\left(n + \frac{M+1}{2}\right)\left(k + \frac{1}{2}\right)\frac{p}{M}\right] \qquad (3.13)$$

Similarly, the output of the synthesis filterbank can be expressed as

$$y(n) = \sqrt{\frac{2}{M}} \sum_{k=0}^{M-1} g(n)X(k) \cos\left[\left(n + \frac{M+1}{2}\right)\left(k + \frac{1}{2}\right)\frac{p}{M}\right] \qquad (3.14)$$

Substituting Eq. 3.12 in 3.13 and simplifying, we get

$$y(n)=\left\{\begin{array}{l}\frac{1}{M}g(n)\sum_{m=0}^{N-1}x(m)h(m)\sum_{k=0}^{M-1}\cos\left[(m+n+M+1)\left(k+\frac{1}{2}\right)\frac{p}{M}\right]\\ +\frac{1}{M}g(n)\sum_{m=0}^{N-1}x(m)h(m)\sum_{k=0}^{M-1}\cos\left[(m-n)\left(k+\frac{1}{2}\right)\frac{p}{M}\right]\end{array}\right\} \quad (3.15)$$

On further observation,

$$\begin{array}{l}y(n)=g(n)h(n)x(n)-g(n)h(M-1-n)x(M-1-n),\quad n=0,1,\ldots,M-1\\ y(n)=g(n)h(n)x(n)-g(n)h(3M-1-n)x(3M-1-n),\quad n=M,M+1,\ldots,2M-1\end{array} \quad (3.16)$$

Now, the reconstructed signal in the overlapping parts is given by

$$y_{overlap}=y_1^2+y_2^1 \quad (3.17)$$

Taking y_2 as the time reference,

$$\begin{array}{l}y_{overlap}=y_2(n)+y_1(n+M),\quad n=0,1,\ldots,M-1\\ =\left\{\begin{array}{l}g(n+M)h(n+M)x_1(n+M)+g(n+M)h(2M-1-n)x_1(2M-1-n)\\ +g(n)h(n)x_2(n)+-g(n)h(M-1-n)x_2(M-1-n)\end{array}\right\}\end{array} \quad (3.18)$$

Using a common time reference for the input blocks results in

$$\begin{array}{l}x_1^2=x_2^1=x_{overlap}\\ =x_1(n+M)=x_2(n),\quad n=0,1,\ldots,M-1\end{array} \quad (3.19)$$

and

$$x_1(2M-1-n)=x_2(M-1-n),\quad n=0,1,\ldots,M-1 \quad (3.20)$$

Therefore, for perfect reconstruction, we need

$$\begin{aligned} &h(n)g(n) + h(n+M)g(n+M) = 1 \\ &g(n)h(M-1-n) - g(n+M)h(2M-1-n) = 0 \end{aligned} \tag{3.21}$$

Using the same window for both analysis and synthesis results in the *Modulated Lapped Orthogonal Transform* (MLOT). The use of a symmetric window $h(n)$ results in the following constraints for perfect reconstruction

$$\begin{aligned} &h(n) = h(N-1-n) \\ &h^2(n) + h^2(n+M) = 1 \end{aligned} \tag{3.22}$$

3.3 Adaptive Filterbanks

For a relatively stationary signal, a higher coding gain can be achieved with better frequency resolution (long windows). On the other hand, it is preferable to have better temporal resolution (short windows) for transients and signal attacks to localize the spread of quantization noise. Almost all audio coders switch between a set of available filterbank resolutions to match the signal. The switching criterion is based on a measure of information content in the signal, like energy or perceptual entropy. Some of the more advanced coders use Temporal Noise Shaping (TNS) to continuously adapt to the temporal and spectral resolution of the filterbank [50].

In order to maintain perfect reconstruction, switching between windows has to be smooth. Therefore, a set of transition windows is used to shift from one resolution to another gracefully. A start window is used to switch from a long window to a short one

and a stop window is used to switch back. The start window is defined as

$$h_{start}(n)=\begin{cases} h_{long}(n), & 0\le n\le M-1 \\ 1, & M\le n\le M+\dfrac{M}{3}-1 \\ h_{short}(n-N), & M+\dfrac{M}{3}\le n\le M+\dfrac{2M}{3}-1 \\ 0, & M+\dfrac{2M}{3}\le n\le 2M-1 \end{cases} \tag{3.23}$$

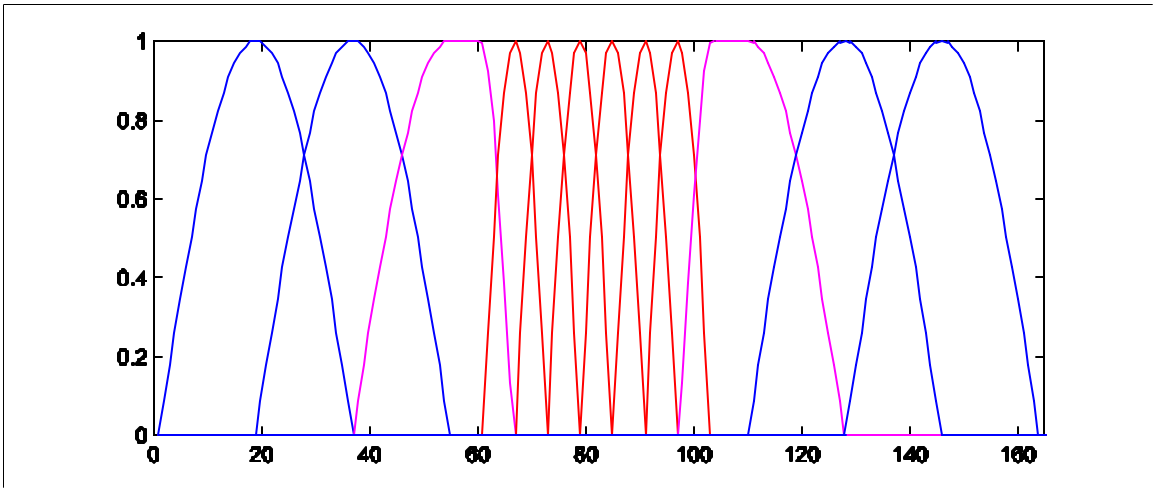


Fig. 3.4 MDCT Window Switching (M=18).

Fig. 3.4 graphically depicts the switching from a long to short window and back. In this particular case, there are 4 short windows between two long windows, with transition windows in between.

3.3.1 Perfect Reconstruction Conditions for Window Switching

Assuming that the analysis and synthesis filers are the same, from Eq. 3.14 becomes

$$y(n)=\left\{\begin{array}{l} \dfrac{1}{M}h_{start}(n)\displaystyle\sum_{m=0}^{N-1}x(m)h_{start}(m)\sum_{k=0}^{M-1}\cos\left[(m+n+M+1)\left(k+\frac{1}{2}\right)\frac{p}{M}\right] \\ +\dfrac{1}{M}h_{start}(n)\displaystyle\sum_{m=0}^{N-1}x(m)h_{start}(m)\sum_{k=0}^{M-1}\cos\left[(m-n)\left(k+\frac{1}{2}\right)\frac{p}{M}\right] \end{array}\right\} \tag{3.24}$$

For $0 \le n \le M-1$, we have

$$y(n) = h_{long}^2(n)x(n) - h_{long}(n)h_{long}(M-1-n)x(M-1-n) \tag{3.25}$$

In a lapped transform, the first half of the output of the synthesis filterbank will have the same terms as the latter half of the previous block, except that the time-reversed terns have opposite signs. Thus the overlap and add operation results in perfect reconstruction of the signal.

For $M \le n \le M + \frac{M}{3} - 1$, $h(n) = 1$ and $y(n)$ exists only when $n = m$ Therefore

$$y(n) = x(n) \tag{3.26}$$

For $M + \frac{M}{3} \le n \le M + \frac{2M}{3} - 1$, we have

$$\begin{aligned} y(n) &= h_{start}^2(n)x(n) - h_{start}(n)h_{start}(3M-1-n)x(3M-1-n) \\ &= h_{short}^2(n-M)x(n) - h_{short}(n-M)h_{short}(2M-1-n)x(3M-1-n) \end{aligned} \tag{3.27}$$

The second term is cancelled by a similar term in the next short block and perfect reconstruction is maintained.

For $M + \frac{2M}{3} \le n \le 2M-1$, $h(n) = 0$ and so is the output of the synthesis filtertbank. The signal is perfectly reconstructed from the outputs of two successive short frames.

Perfect reconstruction can also be proved for transition from the short window back to long ones.

CHAPTER 4

PERCEPTUAL AUDIO CODING: APPLICATION OF PSYCHOACOUSTICS TO AUDIO COMPRESSION

As a consequence of the finite time-frequency resolution of the human auditory mechanism, the ear perceives only a part of the information present in the stimulus. This is called as *Perceptual Entropy* (PE) [16] [17] [19]. From a compression standpoint, this is the *critical-mass* of the signal, the minimum number of bits required to represent the perceptually relevant information in the signal. Any extra information can be safely discarded without affecting the perceptual quality of the signal reconstructed from a compact representation of this critical mass. Decidedly, the scheme is *lossy*, but perceptually *transparent*.

A model for computing the perceptual entropy mimics the working of the auditory system and computes a Just Noticeable Distortion (JND) profile for a given frame of audio data - a measure of the maximum quantization noise that can be injected for perceptually lossless signal recovery. The JND profile can then be used to shape the spectrum of the quantization noise to make it inaudible. The first part of this chapter concentrates on the application of psychoacoustic principles to audio compression in general. For the sake of illustration and completeness, references are made to MPEG Audio Psychoacoustics Model 2. The latter part of this chapter is devoted to the review of some of the most prominent perceptual audio codecs.

4.1 Perceptual Entropy

A high-resolution spectral estimate of the data is essential for an accurate estimation of the masking thresholds in the critical bands. The MPEG-1 standard uses a 1024-point time-to-frequency mapping via a fast Fourier Transform (FFT) for spectral estimation (Johnston originally used a 2048-point transform). A window is usually applied to the time domain data to reduce the edge effects of the transform window. The power spectrum (PSD) is estimated as

$$P(f) = \mathrm{Re}^2(f) + \mathrm{Im}^2(f) \tag{4.1}$$

To emulate the critical band analysis of the basilar membrane, the spectral components are transformed to the Bark-frequency domain by summing up power spectral components as follows

$$B_z = \sum_{f=bl_z}^{bh_z} P(f) \tag{4.2}$$

where bl_z and bh_z represent the lower and higher frequency limits for the z^{th} critical band.

The BM vibrates in response to the stimulus. The distribution of energy along the vibrating BM is called as an *excitation pattern*. The excitation pattern due to a single spectral component is called as a *Spreading Function*. The spreading function has a triangular shape, with a steep slope on the low-frequency side and a shallow slope on the high frequency side. If the model used for the inner ear is linear, the global excitation pattern can be computed by convolving the bark energy spectrum with the spreading

function that is independent of frequency and the level of the masker.

$$C_z = B_z * SpF_z \tag{4.3}$$

The spreading function used in the MPEG Psychoacoustics Model 2 has a constant shape (independent of frequency and the level of the masker), with slopes of 25 dB/Bark on the low frequency side and –10 dB/Bark on the high frequency side. At critical band z it is defined by the following equation

$$SpF(z) = 15.81 + 7.5(z + 0.474) - 7.5\left[1 + (z + 0.474)^2\right]^{0.5} \tag{4.4}$$

The masking threshold is determined by subtracting an offset from the excitation pattern. From Ch. 2, it is known that the masking threshold due to a tonal masker is less than that of a narrow-band noise masker. Said otherwise, narrowband noise maskers produce more masking than tonal maskers. Therefore, the value of the offset is strongly dependent on the tonal or noise-like nature of the masker. The threshold can be calculated as

$$\begin{aligned} O_T &= [14.5 + z] \\ O_N &= K \\ TH_N &= E_T - O_T \\ TH_T &= E_N - O_N \end{aligned} \tag{4.5}$$

where O_T is the offset for a tonal masker, O_N is the offset for a noise masker, TH_N is TMN, TH_T is NMT, E_T is critical band tone masker energy, E_N is the critical band noise masker energy, K is between 3 and 5 dB and is the critical band number.

The *Spectral Flatness Measure* (SFM) is defined as

$$SFM = \frac{m_g}{m_a} \tag{4.6}$$

where m_g is the geometric mean and m_a is the arithmetic mean of the signal PSD respectively. SFM is an indicator of the nature of the masker. SFM values close zero indicates a narrowband spectrum while values close to one indicate a flat spectrum. The SFM lies between zero and one. Translating this to the dB scale results in a more intuitive 'coefficient of tonality', defined by

$$a = \min\left(\frac{SFM_{dB}}{-60}, \quad 1\right) \tag{4.7}$$

As is obvious from the choice of name, tonal components have a values close to unity while noise-like components reveal a values nearer to zero. This coefficient of tonality can be used to compute the offset as

$$\begin{aligned} O_z &= aO_T + (1-a)O_N \quad dB \\ &= a(14.5+z) + (1-a)K \quad dB \end{aligned} \tag{4.8}$$

It can be seen that the masking offset is a function of the critical band number and of both the tonal and noise maskers, geometrically weighted by the tonality index. A frame of audio contains both tonal and noise-like maskers.

If the auditory system is modeled as a bank of linear, overlapping bandpass filters, the global masking pattern (JND) is determined by summing up individual masking thresholds. The ISO/MPEG models are based on this assumption. On a dB scale, the JND estimate is obtained as

$$T_z = 10^{\log_{10}(C_z) - \frac{O_z}{10}} \tag{4.9}$$

There is evidence thresholds generated by a non-linear additive model for masking better fits the human psychophysical system [46]. Also, a non-linear model results in higher global masking threshold.

The global threshold so generated is compared against the absolute threshold of hearing and the final JND estimate is arrived at as follows

$$T_z = \max\left[T_z, T_q(z)\right] \tag{4.10}$$

This JND estimate drives the quantization stage of the audio coder. Assume that a uniform quantizer is used for quantization of the spectral components. Let the quantizer step size be denoted by Δ. If the number of quantization levels is large enough to assume that the quantization noise has a uniform distribution, the power in the quantization noise is given by

$$s_n^2 = \frac{\Delta^2}{12} \tag{4.11}$$

Since the masking threshold is calculated in the Bark domain, the masking power per spectral line is obtained by dividing the energy in each critical band among its constituents.

$$P_m(f) = \frac{T_z}{2\left(bh_z - bl_z\right)} \tag{4.12}$$

The FFT being a complex transform, the additional factor 2 in the denominator

further divides the power among the real and imaginary components. For perceptually lossless reproduction, the quantization noise must be below $P_m(f)$, i.e.,

$$\begin{aligned} \frac{\Delta_n^2(f)}{12} &\le P_m(f) \\ \Delta_n(f) &\le \sqrt{\frac{6T_z}{bh_z - bl_z}} \end{aligned} \tag{4.13}$$

This rule can be used to weight the quantization error and iteratively vary the step size of the quantizers till the shape of the quantization noise lies below the perceptual threshold.

The requisite quantization levels are

$$L_{\mathrm{Re}}(f) = \left| n \operatorname{int}\left(\frac{\mathrm{Re}(f)}{\Delta_n(f)} \right) \right| \tag{4.14}$$

Using mid-tread (uniform) quantizers, this translates into

$$\begin{aligned} b_{\mathrm{Re}}(f) &= \log_2\left(2L_{\mathrm{Re}}(f)+1\right) \\ b_{\mathrm{Im}}(f) &= \log_2\left(2L_{\mathrm{Im}}(f)+1\right) \end{aligned} \tag{4.15}$$

bits for the real and imaginary components. For an N-point transform, the total number of bits required to quantize all components with the noise below audible threshold is given by

$$\sum_{f=1}^{N} \left(b_{\mathrm{Re}}(f) + b_{\mathrm{Im}}(f) \right) \tag{4.16}$$

Since PE is defined as the number of bits per component, we have

$$
\begin{aligned}
PE &= \frac{1}{N}\sum_{f=1}^{N}\left(b_{\mathrm{Re}}(f)+b_{\mathrm{Im}}(f)\right) \\
&= \frac{1}{N}\sum_{f=1}^{N}\left[\log_2\left(2\left|n\,\mathrm{int}\left(\frac{\mathrm{Re}(f)}{\Delta_n(f)}\right)\right|+1\right)+\log_2\left(2\left|n\,\mathrm{int}\left(\frac{\mathrm{Im}(f)}{\Delta_n(f)}\right)\right|+1\right)\right] \\
&= \frac{1}{N}\sum_{z=1}^{25}\sum_{f=bl_z}^{bh_z}\left[\log_2\left(2\left|n\,\mathrm{int}\left(\frac{\mathrm{Re}(f)}{\sqrt{\frac{6T_z}{bh_z-bl_z}}}\right)\right|+1\right)+\log_2\left(2\left|n\,\mathrm{int}\left(\frac{\mathrm{Im}(f)}{\sqrt{\frac{6T_z}{bh_z-bl_z}}}\right)\right|+1\right)\right]
\end{aligned}
\tag{4.17}
$$

This procedure is applied to a wide variety of audio material, on a frame-by-frame basis and a long-term histogram can be obtained. The worst-case value is selected as the PE. In his seminal paper, Johnston determined the PE of audio signals to be in the neighborhood of 2.1 bits/sample. Fig. 4.1 is a collection of PE estimates for different kinds of audio material.

4.1.1 Alternative options for Quantization

4.1.1.1 Uniform Scalar Quantization

In the computation of perceptual entropy detailed above, uniform scalar quantizers were considered. The quantizer step size Δ is given by

$$\Delta = \frac{x_{\max} - x_{\min}}{L} \tag{4.18}$$

where $x_{\max}$ and $x_{\min}$ are the maximum and minimum values of the input and L is the number of quantization levels. There are other alternatives, as listed below.

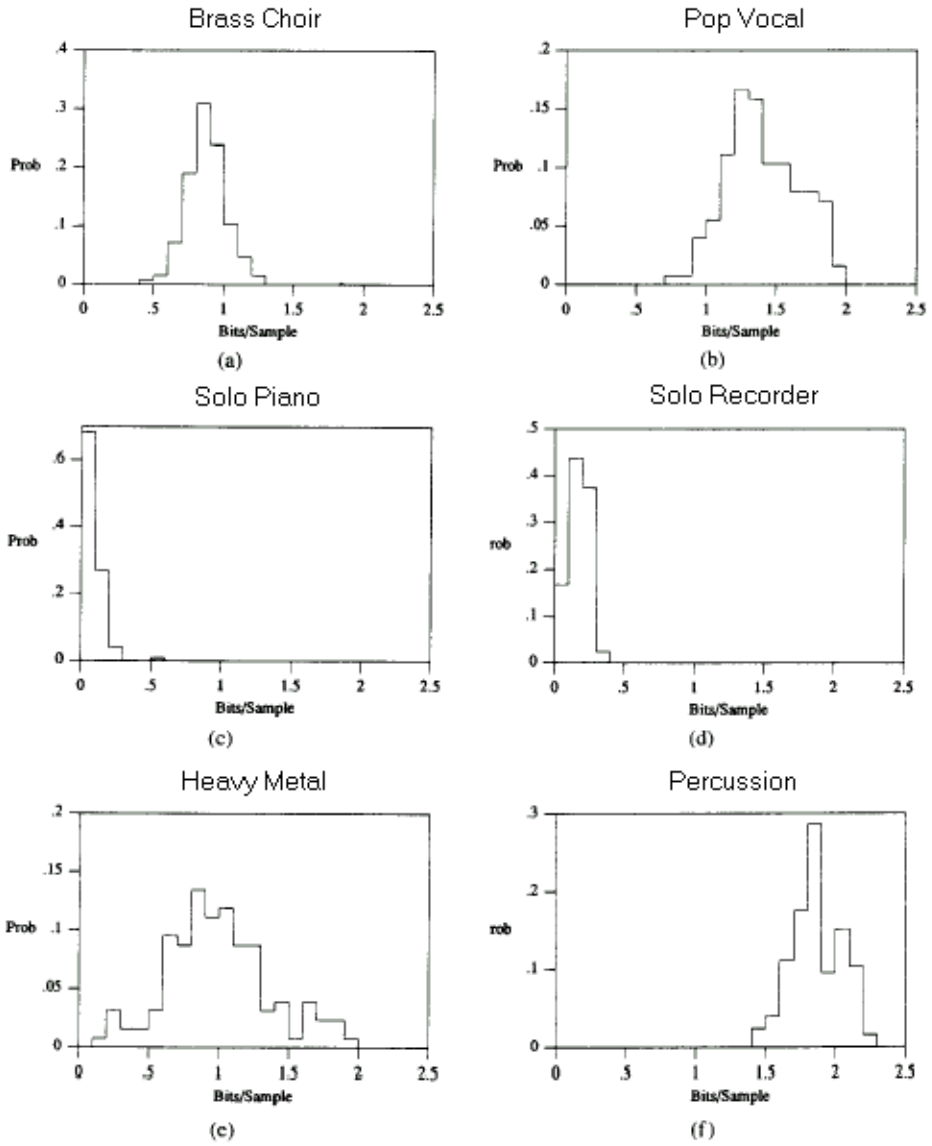


Fig. 4.1 Individual PE histograms for some audio sources (after Johnston)..

4.1.1.2 Non-Uniform Scalar Quantization

For a signal with a non-uniform distribution, the quantizer step sizes can be optimized for minimum Mean-Square-Error (MSE) performance by minimizing the total distortion as follows [9] [77]

$$D = \sum_{i=1}^{N} \int_{R_i} d\left(x_i, Q_i(x)\right) p_x(x) dx \tag{4.19}$$

where $p_x(x)$ is the probability density function of the input values, $Q_i(x)$ is the i^{th} quantization level, R_i denotes the i^{th} Voronoi cell and *d(. , .)* is the distortion measure. Since Eq. 4.20 does not yield a closed form solution in most cases, iterative algorithms like the Lloyd's Algorithm [77] are used to determine the actual quantizer levels.

For quantization of speech signals, the A-law and m–law are very popular. The MPEG-1 Layer III (MP3) [40] [41] [44] and MPEG-2 Advanced Audio Coding (AAC) [54] architectures use a similar non-uniform power law quantization scheme for quantizing the transform (MDCT) coefficients. This law is given by

$$X_q(i) = n\,\text{int}\left[\left(\frac{|X(i)|}{\Delta}\right)^{3/4} - 0.0946\right] \tag{4.20}$$

where $X(i)$ and $X_q(i)$ are the i^{th} transform coefficient and its quantized value. To emphasize low amplitude components, the quantizer raises its input to the ¾ power before quantization. As a result, larger amplitudes are quantized roughly and smaller amplitudes are quantized more finely. This provides a more consistent SNR over the range of quantizer values.

4.1.1.3 Perceptually Weighted Vector-Quantization

It is straightforward to include the masking threshold into the distortion measure to train the codebooks of a vector quantizer [77]. The same error criterion can be used to pick codebook indices in the actual signal compression process. Given a *K*-dimension

vector of input spectral components X, a vector of corresponding masking thresholds M and a codebook c , a measure of distortion of the k^{th} component can be defined as

$$d(k) = \left| X(k) - c^{j}(k) \right|^2 - M(k) \tag{4.21}$$

The energy of the audible noise can be calculated as

$$D(X, c^{j}) = \sum_{k=1}^{K} \max\left(d(k), 0\right) \tag{4.22}$$

The centroid of each Voronoi cell is determined by minimizing the energy of the audible noise as

$$c_{opt}^{j} = \underset{c^{j}}{\arg\min} \sum_{i=1}^{I} D\left(X^{i}, c^{j}\right) \tag{4.23}$$

where I is the number of vectors in region j.

4.1.2 Example

The application of the psychoacoustic rules defined in MPEG-1 Psychoacoustic Model 2 (Ch. 5) to a block of audio data is graphically depicted in the figures that follow.

Fig. 4.2 depicts the spectrum of the data as computed by a 1024-point windowed FFT. The spectral energy is mapped into the perceptual domain (for a sampling frequency of 44100 Hz, 63 one-third critical band partitions). This mapping is non-linear, expanding the low frequency region, while compressing the high frequency region, as shown in Fig. 4.3.

The *spreading function* shown in Fig 4.4 is then applied to spread the energy of each partition into the adjacent partitions. This simulates the masking effect on the BM.

The shape of this function is constant as a function of the partition number.

The spreading of energy into the adjacent critical bands is shown in Fig. 4.5. The masking threshold computed from the spread energy is shown in Fig. 4.6.

In Fig. 4.7, the computed thresholds for the partitions are spread over the spectral lines. The effect of stronger components on adjacent frequencies is clearly visible as an increase in the threshold. In a typical perceptual coder, this JND estimate is used to shape the floor of the quantization noise to meet the target bit-rate.

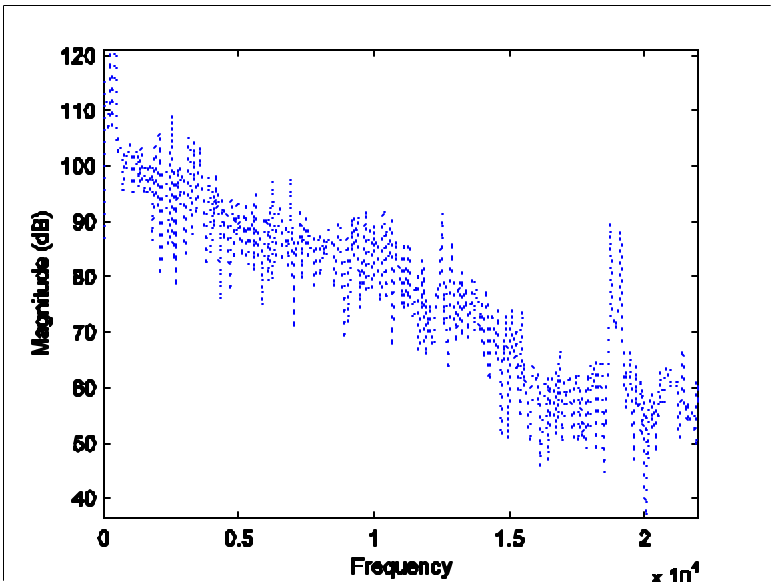


Fig. 4.2 The energy of a frame of audio data.

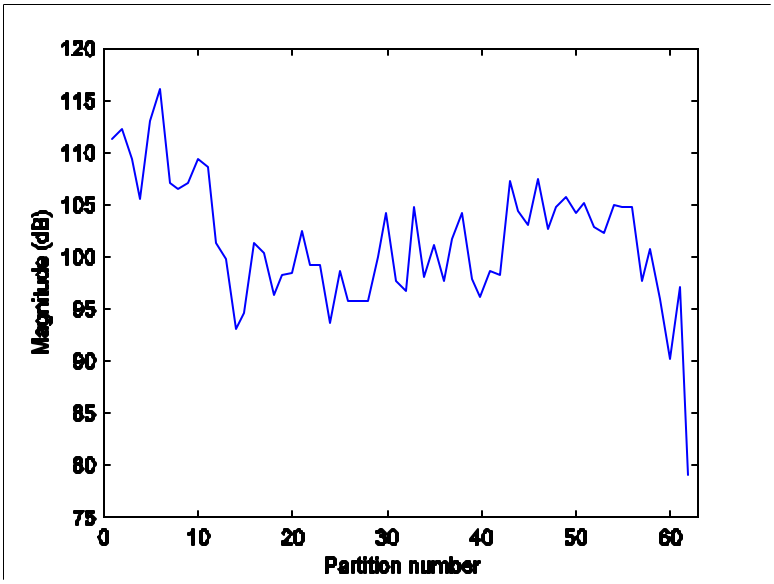


Fig. 4.3 The energy in the Perceptual domain.

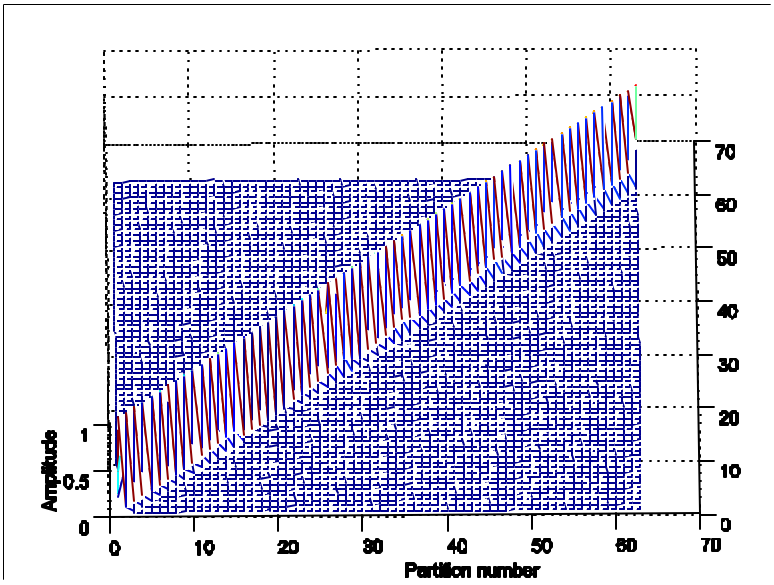


Fig. 4.4 The spreading function that simulates the effect of masking on the BM.

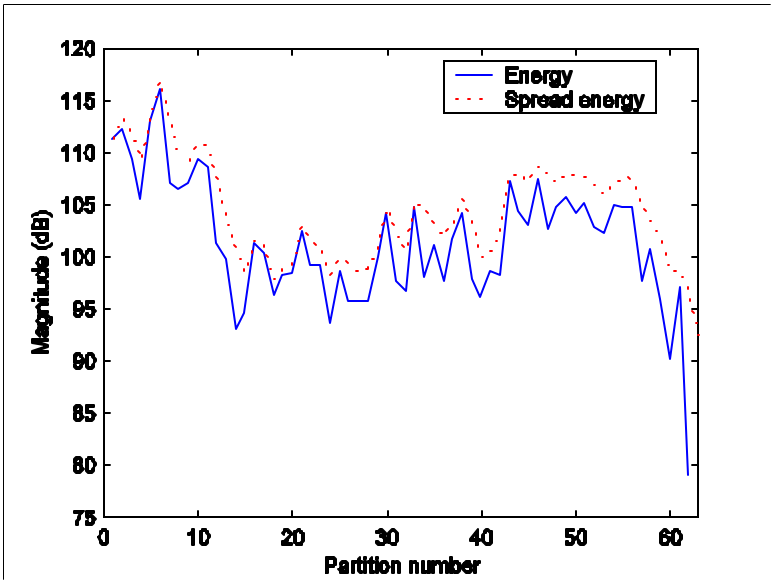


Fig. 4.5 The audio energy and the spread energy in the Perceptual domain.

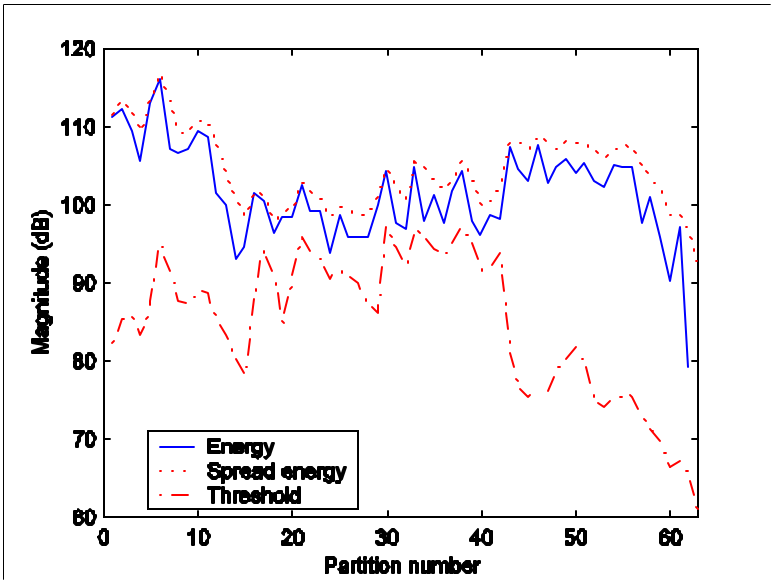


Fig. 4.6 The masking thresholds in the partition domain, as computed by the model.

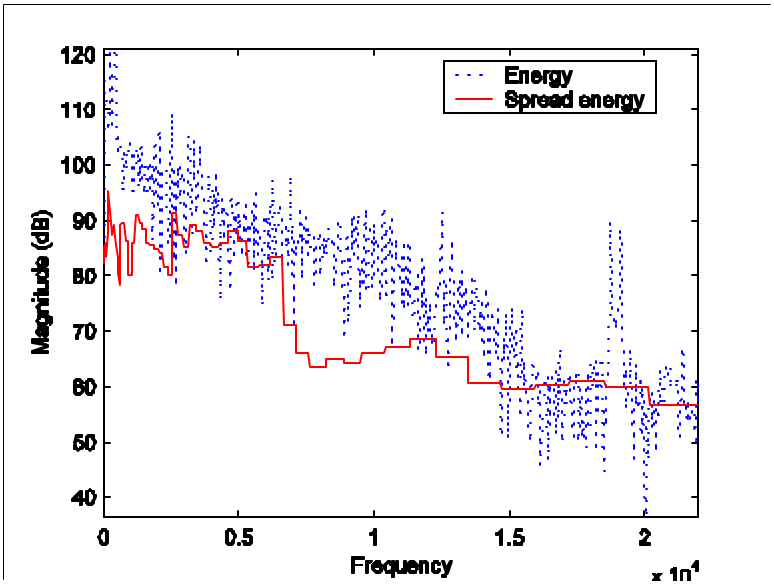


Fig. 4.7 The masking thresholds spread over the FFT lines.

4.2 A Review of Perceptual Audio Coders

This section is an overview of some of the most successful perceptual audio coders. It is to be reiterated that these audio coders operate in the transform domain because tracking of the signal characteristics is straightforward and provides for complete control of noise shaping over the entire spectrum. The early efforts in audio coding were based on principles borrowed from speech coding [4] [5] [6].

4.2.1 Optimum Coding in the Frequency Domain (OCF-1, OCF-2, OCF-3)

In 1987 Brandenburg proposed the Optimum Coding in the Frequency Domain (OCF) audio coder [13] that was based on the Adaptive Transform Coder (ATC) [4] for speech. The coder operates on 512-point audio samples and transforms them into the DCT domain. The DCT spectrum is perceptually weighted and iteratively quantized till the target bit-rate is achieved (inner loop). In the outer loop, the quantization noise resulting from the inner loop is compared with the JND thresholds derived from the

psychoacoustic model; if the noise exceeds the threshold, the process repeats till convergence is achieved or a time-out is reached. This coder could achieve high quality at 132 kb/s.

Brandenburg reported enhanced versions of OCF-1, viz., OCF-2 and OCF-3 in 1988, where several enhancements were made; the DCT was replaced by the MDCT, the psychoacoustic model was improved to include a model of temporal masking, pre-echo control, improved rate control loops, differential coding of spectral components and reduction in computational complexity. The OCF-3 could achieve higher quality at around 64 kb/s.

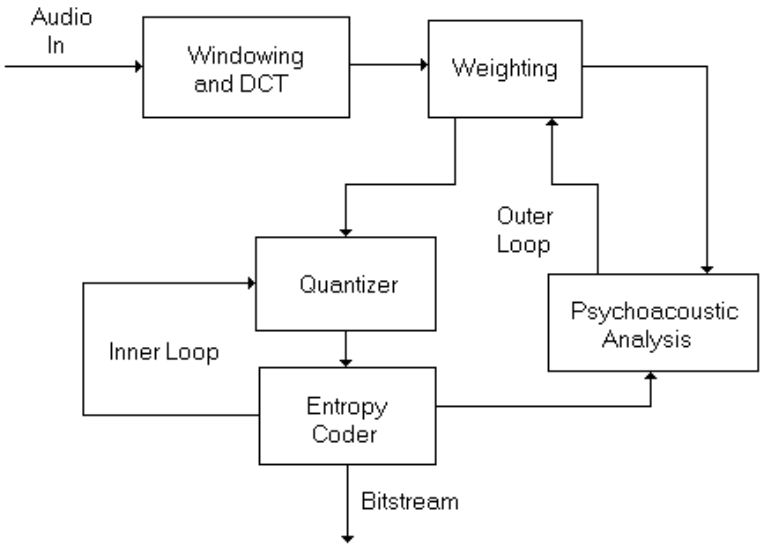


Fig. 4.8 The OCF Coder.

4.2.2 Perceptual Transform Coder (PXFM)

Around 1988 Johnston developed a series of DFT-based transform coders at the AT&T Bell Labs, Perceptual Transform Coder (PXFM) and Stereo Extended Perceptual Transform Coder (SEPXFM) that became an integral part of the ASPEC proposal [27].

The PXFM works to quantize the complex valued DFT samples based on the perceptual entropy criterion developed by Johnston [16]. The algorithm operates on 2048-poiint windowed segments of the signal, with (1/16) overlap between successive segments. Based on the JND estimate from the perceptual model, the rate control loop divides the transform components into 128 bands and quantizes them to meet the desired bit-rate, quantization being performed by variable radix bit packing.

The SEPXM exploits stereo redundancy and can achieve transparent coding at 192 kb/s. It has other refinements such as rate-optimized entropy codebooks for lossless coding of the quantized coefficients.

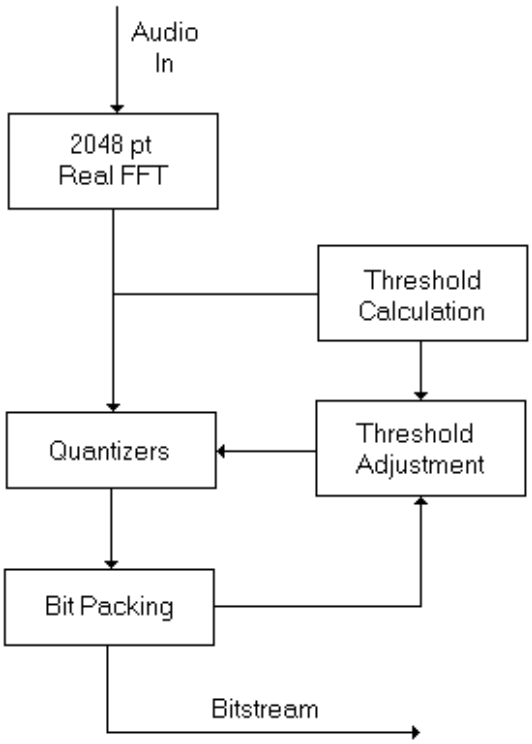


Fig. 4.9 Block diagram of the PXFM Coder.

4.2.3 Transform-domain Weighted Interleaved Vector Quantization (TWIN-VQ)

The architecture of the TWIN-VQ coder [43] [55] is depicted in Fig. 4.10. The coder seeks to extract statistical redundancy by parametric modeling and code the residual efficiently by interleaved vector quantization. It provides high quality for wideband audio below 64 kb/s. The audio data first transformed into the MDCT domain. The spectral components are divided their respective LPC coefficients to flatten the envelope. The signature of the fine structure is still present in the residual. Backward prediction is used to predict the fine structure from the previous three frames and a second stage residual is extracted. Interleaved VQ is applied to the residual to achieve a high coding gain. The performance of the coder exceeds that of the MPEG-2 AAC coder at low bit-rates – around 8 kb/s. This inspired the inclusion of a combined AAC/TWIN-VQ coder in the MPEG-4 standard. More details can be found in [75] [83].

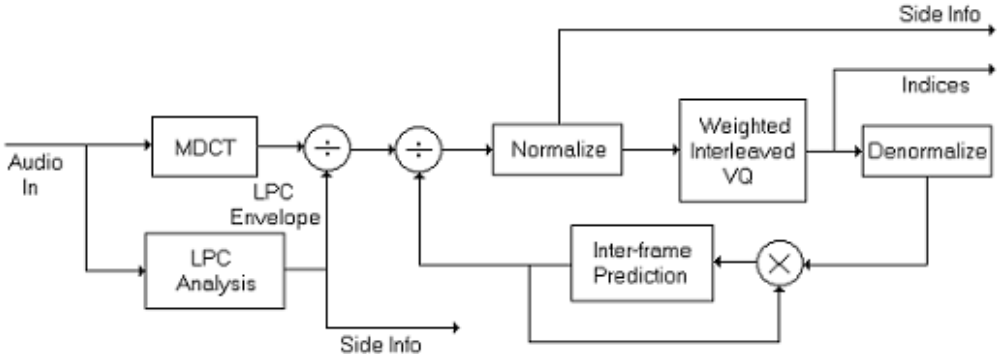


Fig. 4.10 Block diagram of the TWIN-VQ Coder.

4.2.4 Dolby AC-3

The AC-3 algorithm developed by Dolby Laboratories is a high quality transform coder for multichannel applications. The audio data first transformed into the MDCT

domain. There is a choice of two window sizes – 512 or 256 points, for signal dependent time-frequency analysis. The window is based on a proprietary Kaiser-Bessel Design (KBD) optimized for good stopband attenuation. Suitable phase shifts of the MDCT basis vectors during short windows results in perfect reconstruction without the need for transition windows. The transform coefficients are converted into a binary exponential notation as a binary exponent and mantissa. The set of exponents is encoded into a coarse representation of the signal spectrum that is referred to as the spectral envelope. The spectral envelope is processed by a bit allocation routine to calculate the amplitude resolution required for encoding each individual mantissa. The use of a forward-backward adaptive perceptual model reduces the side-information significantly and also provides for transmitting differences in modeled and actual masking thresholds explicitly - *deltas* [53]. Unlike other coders, the AC-3 algorithm does not have an entropy coder at the backend. The spectral envelope and the quantized mantissa for 6 blocks (1536 audio samples) are formatted into one AC-3 synchronization frame. The AC-3 bitstream is a sequence of consecutive AC-3 frames.

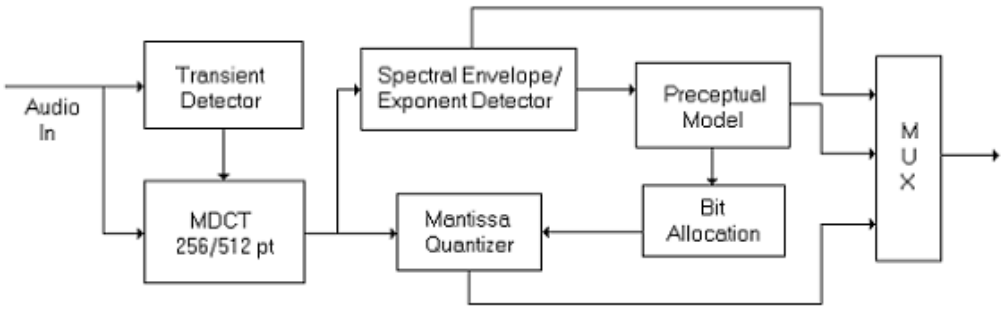


Fig. 4.11 Architecture of the AC-3 Coder.

4.2.5 MPEG Audio Coders

A discussion of audio compression would be incomplete and indeed impossible without a reference to the MPEG-Audio group. MPEG is an acronym for *Moving Pictures Expert Group*. The MPEG committee, established in 1988, works under the direction of the International Standards Organization (ISO) and the International Electrotechnical Commission (IEC) to standardize audio and video compression algorithms.

4.2.5.1 MPEG-1

Eureka 147 DAB is a reliable, multi-service, digital radio broadcasting system, designed specifically for robust reception by mobile, portable, and fixed receivers, using simple non-directional antennas. CCETT, IRT and Philips jointly developed the MUSICAM (Masking pattern adapted Universal Sub-band Integrated Coding and Multiplexing) [29] algorithm for the Eureka 147.

Around the same time, AT & T Bell Labs, Thomson, Fraunhofer Society and CNET jointly proposed the ASPEC (Adaptive Spectral Perceptual Entropy Coding) [27], an algorithm for transmitting audio over the Internet.

Both systems were subjected to comprehensive listening tests. It was found that the MUSICAM algorithm has higher complexity and coding delay than the ASPEC coder. But the ASPEC coder performed better at lower bit-rates. The MPEG/Audio group combined the attributes of both into a draft standard (ISO/IEC JTC1/SC2/WG11, Committee draft 11172) [30] having three levels of complexity and performance. This was standardized at the end of 1992.

The three different levels offer increasing levels of compression at the cost of higher computational requirements. The standard supports three sampling rates of 32, 44.1 and 48 kHz and output bit-rates from 32 to 448 kb/s for Layer I, from 32 to 384 kb/s for Layer II, and from 32 to 320 kb/s for Layer III. The transmission can be mono, dual channel (e.g. bilingual), stereo or joint stereo (where the redundancy between left and right channels can be exploited).

MPEG-1 Layer I audio algorithm is a simplified version of the MUSICAM algorithm, tailored for mild compression and low cost applications. The Philips Digital Compact Cassette (DCC) uses this scheme at a rate of 192 kb/s per channel.

Layer II is identical to MUSICAM and has been engineered for target bit-rates around 218 kb/s per channel. Applications include DAB, storage of synchronized video-and-audio sequences on CD-ROM and Video-CD.

Layer III combines the best attributes of both the MUSICAM and ASPEC coders and hence the most complex of the three. It provides high compression factors, with target bit-rates as low as 64 kb/s per channel, required for low bandwidth applications like audio transmission over ISDN channels.

The coded bitstream also provides for an embedded error-detection code by way of cyclic redundancy checks (CRC). The algorithms are asymmetrical in the sense that the encoder is more complicated and computationally expensive than the decoder. All three layers are simple enough to allow single-chip, real-time decoder implementations.

MPEG-1 was the first phase of an international effort at standardizing audio and video compression technologies. There have been others since then.

4.2.5.2 MPEG-2

The MPEG-2 [37] standard provides enhancements and additional tools to target a wider range of applications. The MPEG-2 BC (ISO/IEC 13818-3) [38] provides for an extension of MPEG-1 towards lower sampling rates for low bit-rate applications. Support for 16, 22.05, and 24 kHz sampling rates is provided. Bit-rates from 32 to 256 kbit/s for Layer I, and from 8 to 160 kbit/s for Layer II & Layer III are supported. It also provides a backward compatible multichannel extension to MPEG-1 for surround sound applications: up to five main channels, *Left*, *Right*, *Center*, *Left Surround*, *Right Surround*, and an additional *Low Frequency Enhancement* (LFE) or *Sub-Woofer* channel. The upper limit on the bit-rate is 1 Mbit/s. For the bitstream to be backward compatible with MPEG-1, a two channel signal is derived for the five channel signal by matrixing. These two channels are encoded into a standard MPEG-1 audio frame, while the remaining three channels are encoded in the ancillary frame. As a result, an MPEG-1 decoder will decode the main frame and discard the ancillary frame, while an MPEG-2 decoder is smart enough to decode the additional channels.

MPEG-2 AAC (ISO/IEC 13818-7) [49] provides a very high-quality audio coding standard for 1 to 48 channels at sampling rates of 8 to 96 kHz, with multichannel, multilingual, and multiprogram capabilities. AAC works at bit-rates from 8 kbit/s for a monophonic speech signal up to in excess of 160 kbit/s/channel for very-high-quality coding that permits multiple encode/decode cycles.

AAC is organized as a collection of tools. Three complexity profiles, namely main, low and scalable sampling rate (SSR) profile, providing varying levels of

complexity and scalability are defined. MPEG-2 AAC is not backward compatible. A specific combination of tools is recommended for each profile.

The AAC uses an MDCT filterbank that has signal adaptive resolution – 2048-point transforms for stationary signals and 256-point transforms for transients. There is also a choice of MDCT windows, sine window for passband selectivity or Kaiser-Bessel Design (KBD) window for stopband attenuation, depending on the statistics of the signal. The filterbank is continuously signal-adaptive as it uses Temporal Noise Shaping (TNS).

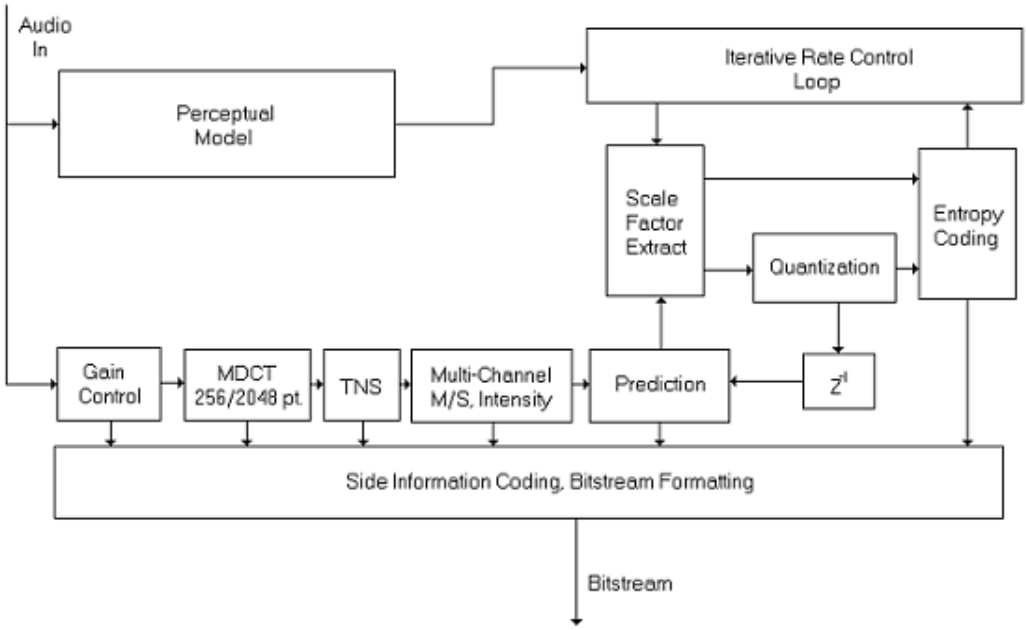


Fig. 4.12 Block diagram of the MPEG-2 NBC/AAC Coder.

In TNS [50], transform coefficients are predicted over time and replaced by the predicted residuals. TNS provides for a better encoding of “pitch based” signals. It also reduces the high bit-rate demand for signal attacks by reducing pre-echo conditions. TNS can be applied over the entire spectrum or selective parts, which ever is deemed necessary. Time-domain noise control can be applied in a frequency-dependent fashion.

4.2.5.3 MPEG-4

MPEG-4 (ISO/IEC 14496-3) [82] [83] provides an integrated family of tools for coding and composition of natural and synthetic audio-visual objects. The transmitter codes and transmits audio-visual objects and an associated scene-description. The scene-description describes how the objects interact to form a scene. The decoder would reconstruct the audio-visual scene from the primitives decoded from the bitstream. The standard also provides for bit-rate, bandwidth and complexity scalability. Besides speech and perceptual audio coding, the audio coding tools support Structured Audio, a universal language for score-driven sound synthesis, and TTSI, a text-to-speech conversion interface.

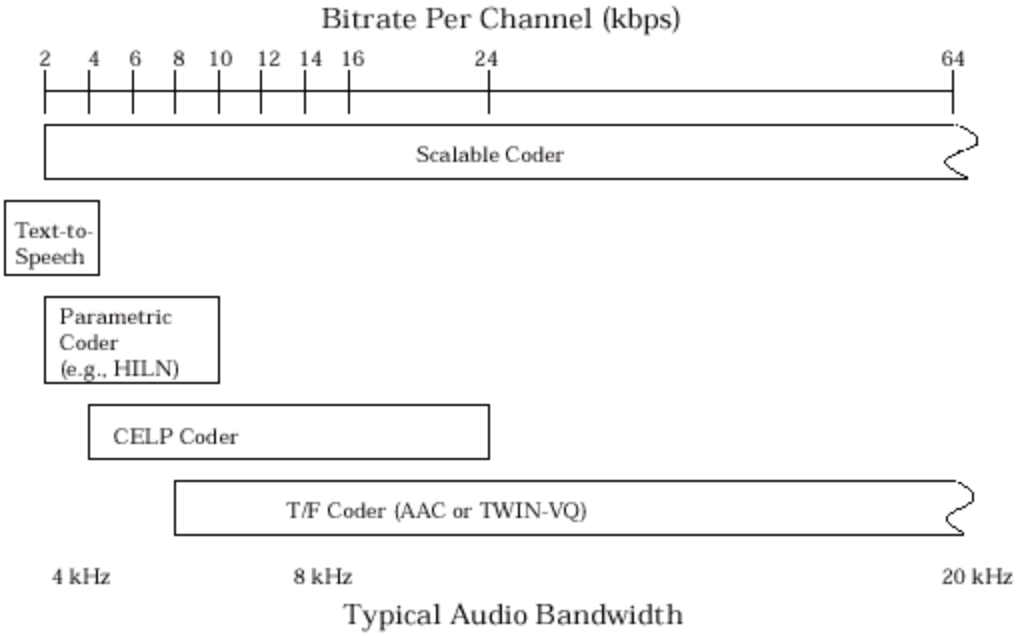


Fig. 4.13 The MPEG-4 Audio tools.

Speech and audio coding have matured into a fine art over the past decade and play a key role in enabling communication services. Today, in addition to high coding

gain, operation over low bandwidth channels, new functionalities like flexible access to coded data and manipulation by the recipient and interoperability are of prime concern in coding audio-visual objects. The MPEG-4 standard recognizes this trend and provides tools for coding natural and synthetic objects efficiently. Natural audio objects like speech and music can be coded at bit-rates ranging from 2 kb/s to 64 kb/s using Parametric Speech Coding, CELP-based Speech Coding or transform based General Audio Coding. Synthetic audio objects can be efficiently coded with Text-To-Speech Interface or the Structured Audio tools. These tools are used to add effects like echo, reverb, chorus and environmental spatialization to the final 'audio scene' that is rendered on an MPEG-4 compliant terminal. There are advanced tools for bit-rate and bandwidth scalability, pitch and time scale modification of speech, low-delay coding and error resilience.

The standard provides tools to code speech from 2 to 24 kb/s. Harmonic Vector eXcitation Coding (HVXC) [71] is a parametric technique used for bit-rates up to 4 kb/s. HVXC inherently supports pitch and speed modifications. CELP coders provide support for the remaining range of bit-rates.

For coding of audio objects up to 16 kb/s, TWIN-VQ [43] [55] tool is used. For higher bit-rates, an extended version of the MPEG-2 AAC [54] is used. Both TWIN-VQ and AAC provide bit-rate scalability. The scalable AAC scheme also has provision for using a CELP core for the base layer.

Low-delay audio coding, with an algorithmic delay not exceeding 20 ms is achieved by modifying the General Audio Coder as follows: the transform size is halved

to 512 samples, there is no block switching and the use of the bit reservoir is minimized or totally abandoned.

For signals coded with AAC or TWIN-VQ, the MPEG-4 standard introduces a Long Term Predictor (LTP) to improve the quality for stationary harmonic signals.

Perceptual Noise Substitution (PNS) is based on the observation that one noise sounds like the other. This implies that the actual fine structure of a noise signal is of minor importance for the subjective perception of such a signal. Consequently, instead of transmitting the actual spectral components of a noisy signal, the bitstream would just signal that this frequency region is a noise-like one and give some additional information on the total power in that band. PNS can be switched on a scale-factor band basis so that even if there are a few spectral regions with a noisy structure, PNS can be used to save bits. In the decoder, a randomly generated noise will be inserted into the appropriate spectral region according to the power level signaled within the bitstream.

Support for rendering synthetic audio objects is provided by the Structured Audio Orchestra Language (SAOL). This language provides the syntax for defining an 'orchestra' of 'instruments' which create and process control data. Control of the synthesis is accomplished by using a score described in the Structured Audio Score Language (SASL) or MIDI.

The Error Resilience tools provide for both error robustness and protection. Virtual Codebooks (VCB11), Reversible Variable Length Coding (RVLC) and Huffman Codeword Reordering (HCR) tools are used to improve error robustness. Unequal Error Protection (UEP) is an efficient technique to protect data. Cyclic Redundancy Checks

(CRC), Systematic Rate-Compatible Punctured Convolutional Codes (SRCPC) and Shortened Reed-Solomon Codes are the used for error detection, correction and concealment. More details on the MPEG-4 standard can be found in [82] [84].

4.2.5.4 MPEG-7

MPEG-7 (ISO/IEC 15938), the current standardization effort, will provide standardized descriptions and description schemes of audio structures and sound content and a language to specify such descriptions and description schemes.

The MPEG standard mandates the syntax of the coded bitstream, defines the decoding process, and provides compliance tests for assessing the accuracy of the decoder. There are no compliance requirements for the encoder except that it should generate a legal bitstream. This guarantees that, regardless of the origin, any fully compliant MPEG/audio decoder will be able to decode the MPEG/audio bitstream with predictable results. But system designers are free to try improved or novel implementations, within the bounds of the standard. So, the standard strives to maintain interoperability while promoting improvement and ingenuity at the same time.

CHAPTER 5

ANALYSIS OF THE MPEG-1 LAYER III ALGORITHM

This chapter concentrates on the MPEG-1 Layer III algorithm [24] [30] [41] [44], popularly known as MP3, which has become the *de-facto* standard for multimedia applications, storage applications and transporting audio over the Internet. Also, the launch of portable MP3 players like the Diamond RIO and it clones, have made its appeal truly universal.

A very basic functional block diagram of the MPEG-1 audio codec is as shown in Fig. 5.1. The algorithm operates on blocks of data. The input audio block to be encoded passes through a filterbank that divides it into multiple frequency subbands. The same chunk of data is also fed to a psychoacoustics model that determines the ratio of signal energy to the masking threshold for each subband. Based on the result of the psychoacoustics analysis and the available bits (target bit-rate), the quantization block iteratively allocates bits to the various subbands to minimize the audibility of the quantization noise. These quantized subband samples and the side information is packed into a coded bitstream by *entropy coding*. Every block of data, thus operated on, is represented as a *frame* in the coded bitstream. There is also provision for inserting ancillary data, not necessarily related to the audio stream, into the frame; but this reduces the number of code bits that can be devoted to the audio.

The decoder parses through the bitstream and extracts the quantized subband values. It then dequantizes these values and reconstructs the audio signal frame by frame.

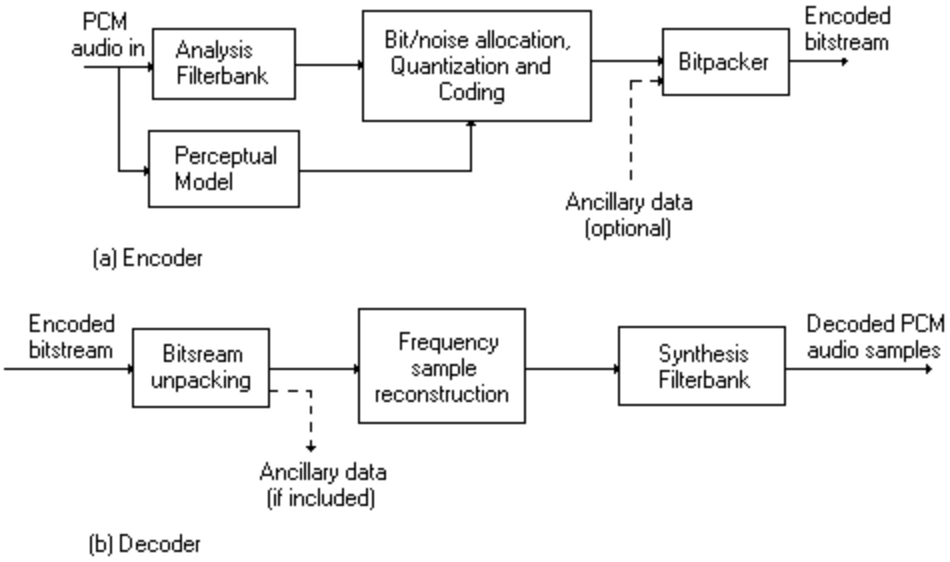


Fig. 5.1 MPEG/Audio codec. (a) Encoder. (b) Decoder.

5.1 The Analysis Filterbank

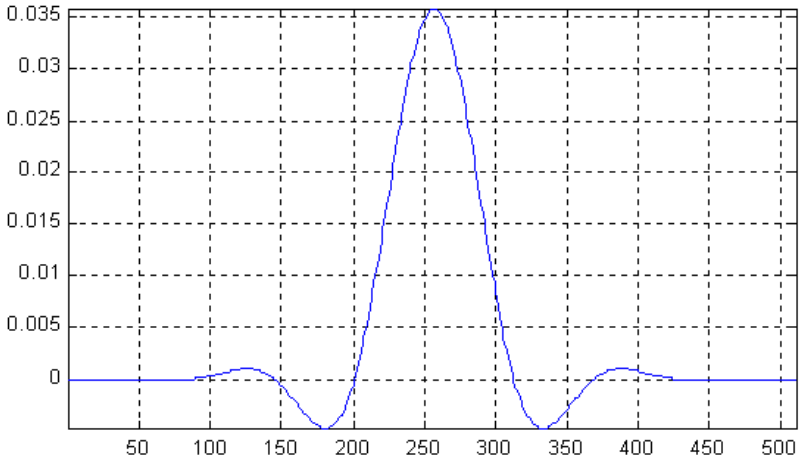


Fig. 5.2 Coefficients of the prototype filter.

The analysis filterbank is common to all the three layers of the algorithm. This *critically sampled* filterbank divides the block of audio into 32 bands, each of a nominal bandwidth $p/(32T)$, where T is the sampling interval. The 512 coefficients of the lowpass prototype filter are plotted in Fig. 5.2. The corresponding impulse response

plotted in Fig. 5.3, attenuates the side-lobes by more than 96 dB. This lowpass filter is cosine modulated to obtain a bank of filters with center frequencies at odd multiples of $p/(64T)$, depicted in Fig. 5.4. For any time instant t, that is an integral multiple of 32 audio sample intervals, the subband outputs are given by

$$s_t[i] = \sum_{n=0}^{511} x[t-n] H_i[n] \tag{5.1}$$

where x is the buffer of input samples and H_i is a bank of bandpass filters obtained by modulating the lowpass prototype as follows

$$H_i[n] = h[n] \cos\left[\frac{(2i+1)(n-16)p}{64}\right] \tag{5.2}$$

By defining

$$C(n) = \begin{cases} -h(n), & n \,\text{int}\,(n/64)\ is\ odd \\ h(n), & else \end{cases} \tag{5.3}$$

we have

$$s_t[i] = \sum_{k=0}^{64} \sum_{j=0}^{7} M[i][k] \times \left[C(k+64j) \times x(k+64j)\right] \tag{5.4}$$

where $n\,\text{int}\,(.)$ is the nearest integer operator and M is a 32×64 matrix for cosine modulation.

The delay through the filterbank, 256 samples, is tolerable and the computational requirement moderate - implementation of the cosine-modulated filterbank as a *polyphase* filterbank, as in Eq. 5.4, requires about 80 multiplies and 80 additions per

output sample. It is also amenable to FFT-like methods that are considerably faster.

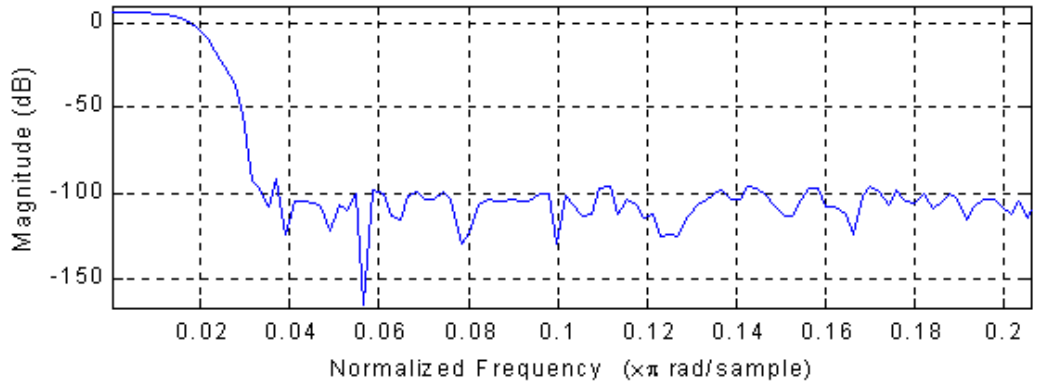


Fig. 5.3 Magnitude response of the lowpass prototype filter.

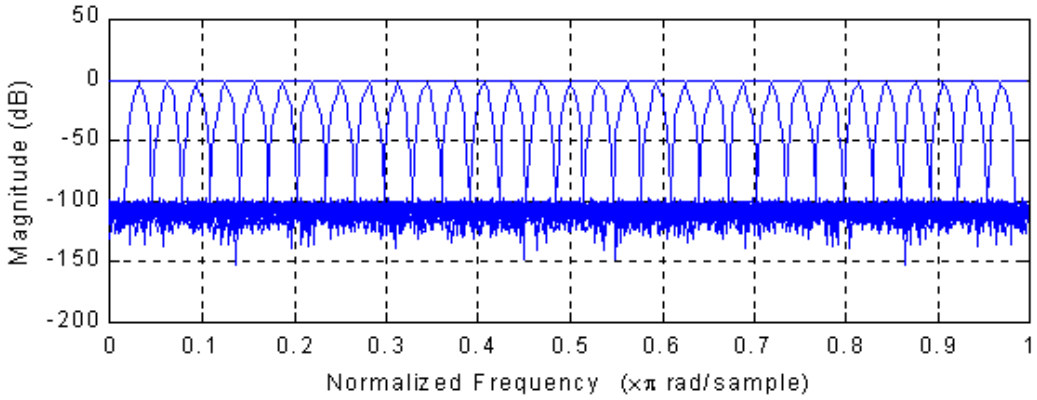


Fig. 5.4 Magnitude response of the analysis filterbank.

Even though the response of the prototype filter is admirable, the filterbank has three significant shortcomings.

Firstly, the lack of a sharp cut-off at the nominal bandwidth can cause a single tone to simultaneously affect the output of two adjacent subbands. Consider a synthetic audio signal made up of a combination of two sinusoids at 675 Hz 11,100 Hz. At a sampling rate of 44.1 kHz, the nominal bandwidth of the prototype filter is about 689 Hz. If the filterbank responses were ideal, only the 0^{th} and 15^{th} subbands will produce outputs.

Since there is significant overlap between adjacent channels, as is obvious from Fig. 5.4, some of the energy leaks into the neighboring bands. The resulting response is as illustrated in Fig. 5.5. To complicate matters further, sub-sampling results in significant aliasing. To partly mitigate the problem, algorithm resorts to explicit alias-reduction once the subband components are transformed into the frequency domain, as discussed in Sec. 5.3.

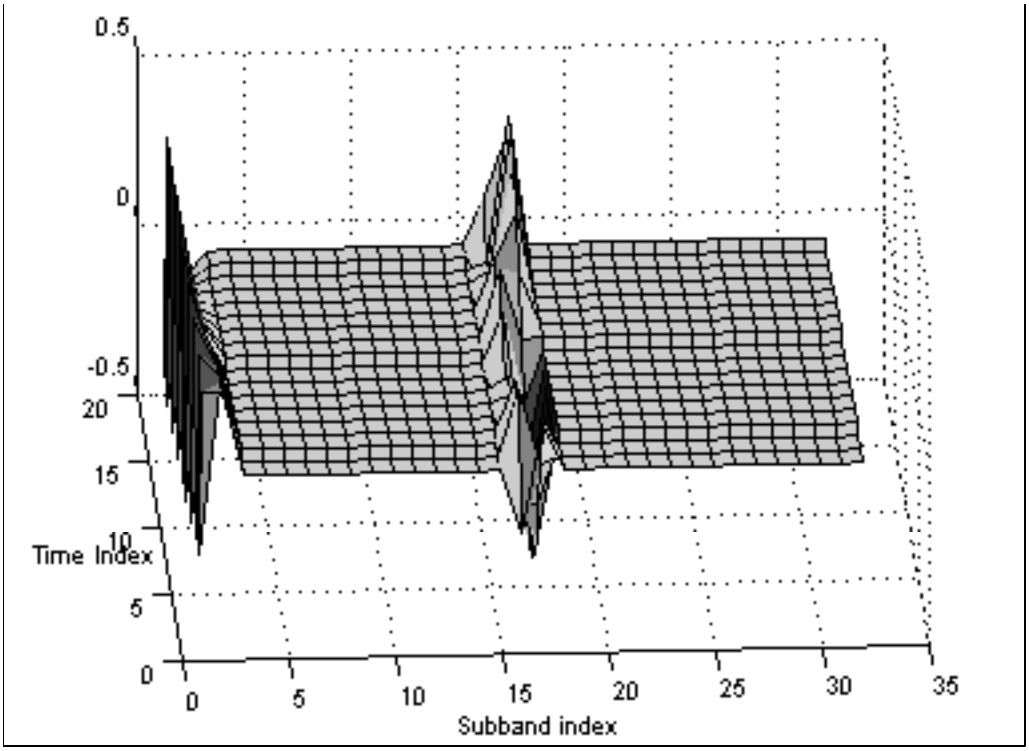


Fig. 5.5 Response of the analysis filterbank for the combination of tones at 675 Hz and 11,100 Hz.

Secondly, the division of the frequency content into subbands of equal width is so unlike the response of the BM. As a result, at low frequencies, a single subband spans many critical bands. This makes the computation of masking thresholds inaccurate. Masking thresholds computed in such a way can be used to steer the psychoacoustic

model in low-complexity applications (Layer I), the penalty being a high bit-rate. For low bit-rate applications, a high-resolution spectral estimate is necessary to compute accurate masking thresholds.

Thirdly, the filterbank and its inverse (the synthesis filterbank) are not lossless transformations. The filterbank and its inverse in tandem, without a quantization process in-between, will not reconstruct the signal exactly. However, the filterbank has been designed to introduce imperceptible error (less than 0.07 dB ripple).

5.2 The Psychoacoustic Model

The psychoacoustic model calculates just-noticeable distortion (JND) profiles for each band in the filterbank. This noise level is used to determine the actual quantizers and quantization levels. There are two psychoacoustic models defined by the standard. They can be applied to any layer of the MPEG/Audio algorithm. In practice however, Model 1 has been used for Layers I and II and Model 2 for Layer III. Both models compute a signal-to-mask ratio (SMR) for each band (Layers I and II) or group of bands (Layer III).

The more sophisticated of the two, Model 2, will be discussed. The steps leading to the computation of the JND profiles is outlined below.

5.2.1 Time-align audio data

The psychoacoustic model must estimate the masking thresholds for the audio data that are to be quantized. So, it must account for both the delay through the filter bank and an additional offset to center the relevant data within the psychoacoustics analysis window. For the Layer III algorithm, the model is computed twice in parallel. One computation is for data buffered by 576 samples. The data is further time-aligned for an

additional 192 samples for use with short blocks (Sec. 5.2.9, Sec 5.3). Therefore, time-aligning the psychoacoustic model with the filterbank demands that the data fed to the model be delayed by a total of 768 samples.

5.2.2 Spectral Analysis and Normalization

A high-resolution spectral estimate of the time-aligned data is essential for an accurate estimation of the masking thresholds in the critical bands. The low frequency resolution of the filterbank leaves no option but to compute an independent time-to-frequency mapping via a fast Fourier Transform (FFT). A Hann window is applied to the data to reduce the edge effects of the transform window. The power spectrum (PSD) is estimated as

$$P(f) = R(f)e^{f(f)} \tag{5.5}$$

where is $R(f)$ the magnitude and $f(f)$ is the phase of each spectral component.

Layer III operates on 1152-sample data frames. Model 2 uses a 1024- point window for spectral estimation. Ideally, the analysis window should completely cover the samples to be coded. The model computes two 1024-point psychoacoustic calculations. On the first pass, the first 576 samples are centered in the analysis window. The second pass centers the remaining samples. The model combines the results of the two calculations by using the more stringent of the two JND estimates for bit or noise allocation in each subband.

When window switching (Sec. 5.2.9, Sec 5.3) is active, a 256-point FFT is computed, with the data centered appropriately.

The relative loudness of different frequency components of a complex sound changes as a function of the overall level. So, unless sounds are reproduced at the same level as the original, the 'tonal balance' is altered. For example, when human voices are reproduced via loudspeakers at high levels, they sound boomy because the ears are very sensitive to low frequencies at high intensities. Since playback levels are unknown to the encoder, the sound-pressure level (SPL) needs to be normalized. This implies clamping the lowest point in the *absolute threshold of hearing* curves to +/- 1-bit amplitude.

5.2.3 Spectral Prediction and Unpredictability Measure

The magnitude $\hat{R}_t$ and phase $\hat{f}_t$ of the spectral components of the current frame t are predicted from those of the previous two frames $t-1, t-2$ as

$$\begin{aligned} \hat{R}_t(f) &= 2 \bullet R_{t-1}(f) - R_{t-2}(f) \\ \hat{f}_t(f) &= 2 \bullet f_{t-1}(f) - f_{t-2}(f) \end{aligned} \tag{5.6}$$

Tonal components are more predictable than broadband signals. They also have different masking characteristics. A measure of unpredictability of each spectral component can be computed as

$$c(f) = \frac{\sqrt{\left[\begin{array}{l} \left(R_t(f)\cos(f_t(f)) - \hat{R}_t(f)\cos(\hat{f}_t(f))\right)^2 + \\ \left(R_t(f)\sin(f_t(f)) - \hat{R}_t(f)\sin(\hat{f}_t(f))\right)^2 \end{array}\right]}}{\left(R_t(f) + \left|\hat{R}_t(f)\right|\right)} \tag{5.7}$$

For best performance, $c(f)$ should be computed for all spectral lines up to 20 kHz. Computing the unpredictability measure only for the low-end spectrum reduces

computational burden at the cost of sacrificing performance. It should be computed from DC to at least 3 kHz and preferably up to 7 kHz. An upper limit of less than 5.5 kHz may considerably reduce performance from that obtained during subjective testing of the audio algorithm. In any case, when $c(f)$ is computed for only a part of the entire spectrum, the remaining values should be set to 0.3.

When Model 2 is used for Layer III compression, the unpredictability is computed for the first 206 spectral lines. For the remaining lines, the value is set to 0.4. The unpredictability of the first 6 lines is calculated from a long FFT (window length = 1024, delay = 576 samples). For the remaining spectral lines up to 205, the unpredictability is computed from the short FFT (window length = 256, delay = 192 samples).

$$c(f) = \begin{cases} c_l(f) & for \quad 0 \le f < 6 \\ c_s\left(\frac{f+2}{4}\right) & for \quad 6 \le f < 206 \\ 0.4 & for \quad f \ge 206 \end{cases} \tag{5.8}$$

where $c_l(f)$ is the unpredictability calculated from the long FFT and $c_s(f)$ is the unpredictability calculated from second short block out of three short blocks in a granule (= a group of 576 samples).

5.2.4 Grouping of spectral values into threshold calculation partitions

The uniform frequency decomposition and poor selectivity of the filterbank do not reflect the response of the BM. To accurately model the masking phenomenon of the BM, the spectral values are grouped into a large number of partitions. The exact number of threshold partitions depends on the choice of sampling rate. This transformation

provides a resolution of approximately either 1 FFT line or 1/3 critical band, whichever is smaller. At low frequencies, a single line of the FFT will constitute a partition, while at high frequencies many lines are grouped into one. The spectral components are transformed to the Bark-frequency domain by summing up power spectral components as follows

$$e(z) = \sum_{f=bl_z}^{bh_z} R^2(f) \tag{5.9}$$

where bl_z and bh_z represent the lower and higher frequency limits for the z^{th} critical band and $e(z)$ is the energy in each threshold calculation partition.

The weighted unpredictability of each partition is given by

$$c(z) = \sum_{f=bl_z}^{bh_z} R^2(f)c(f) \tag{5.10}$$

5.2.5 Simulation of the spread of masking on the BM

A strong signal component affects the audibility of weaker components in the same critical band and the adjacent bands. Model 2 simulates this phenomenon by applying a *Spreading function* to spread the energy of any critical band into its surrounding bands. On the Bark scale, the spreading function *SpF* has a constant shape as a function of partition number, with slopes of +25 and –10 dB per Bark, as defined in Eq. 4.4. In Layer III applications, only values of the spreading function greater than 60 dB are used. The basilar excitation pattern per partition is given by

$$ec(z) = e(z) * SpF(z)$$
$$= \sum_{b=1}^{z_{max}} e(z_b) Spf(zm_b, zm) \tag{5.11}$$

where zm is the median bark value of the partition z and is z_{max} the largest partition index for the particular sampling rate. The unpredictability of each partition is also convolved to get

$$ct(z) = c(z) * SpF(z)$$
$$= \sum_{b=1}^{z_{max}} c(z_b) SpF(zm_b, zm) \tag{5.12}$$

Since $ct(z)$ is weighted by the signal energy, it must be renormalized to $cb(z)$ as

$$cb(z) = \frac{ct(z)}{ec(z)} \tag{5.13}$$

Ideally, the effect of the spreading function on the energy should also be reversed. Implementing this as a standard deconvolution problem would result in numerical problems such as negative and/or zero energy in some regions. These problems manifest themselves because the deconvolution process seeks a strictly numerical solution that disregards the physical and acoustic realities of the situation. So, a renormalization process is used instead.

The spreading function, because of its shape, increases the energy estimates in each band due to the effects of spreading. The renormalization takes this into account, and multiplies each partition by the inverse of the energy gain, assuming a uniform energy of 1 in each partition. While this is not the most accurate procedure, it accounts

for very little error in the bit-rate estimation process.

$$en(z) = ec(z) \bullet rnorm(z) \tag{5.14}$$

and the normalization coefficient is defined as

$$rnorm(z) = \frac{1}{\sum_{b=0}^{Z_{max}} SpF(zm_b, zm)} \tag{5.15}$$

5.2.6 Estimation of tonality indices

It is necessary to identify tonal and non-tonal (noise-like) components because the masking abilities of the two types of signals differ. Model 2 does not explicitly separate tonal and non-tonal components. Instead, it computes a tonality index as a function of frequency. This is an indicator of the tone-like or noise-like nature of the spectral component. The tonality index $tb(z)$ is based on a measure of predictability. Linear extrapolation is used to predict the component values of the current window from the previous two analysis windows. Model 2 uses this index to interpolate between pure tone-masking-noise and noise-masking-tone values. Tonal components are more predictable and thus have a higher tonality index. As this process has memory, it is more likely to discriminate better between tonal and non-tonal components.

$$tb(z) = -0.299 - 0.43\log_e[cb(z)], \quad 0 < tb(z) < 1 \tag{5.16}$$

At this juncture, it is necessary to point out that Model 1 explicitly labels the components as tonal or non-tonal based on the local peaks of the audio power spectrum. After labeling the tonal components, it sums the remaining spectral values into a single non-tonal component per critical band whose frequency index is closest to the geometric

mean of the critical band. This approach works well for low frequency subbands where the subband is narrow relative to the corresponding critical band. But it is inaccurate for higher frequency subbands because critical bands span several subbands – all non-tonal components within a critical band are concentrated into a single component at a single frequency, in essence taking the form of a tonal component. Consequently, a subband within a wide critical band but far from this concentrated non-tonal component will not get an accurate non-tonal masking assessment.

5.2.7 Calculate the required SNR in each partition

The masking threshold is determined by subtracting an offset from the excitation pattern. The value of the offset is strongly dependent on the tonal or noise-like nature of the masker. For Layer III applications, the NMT (noise-masking-tone) is set to 6 dB for all threshold calculation partitions. Similarly, TMN (tone-masking-noise) is set to 29 dB for all partitions. The offset is determined by weighting the maskers with the tonality index as

$$\begin{aligned} O(z) &= tb(z) \bullet TMN(z) + (1 - tb(z)) \bullet NMT(z) \\ &= 29 \bullet tb(z) + 6 \bullet (1 - tb(z)) \quad dB \end{aligned} \tag{5.17}$$

The required SNR in each partition is estimated as

$$SNR(z) = \max[\min val(z), O(z)] \quad dB \tag{5.18}$$

where $\min val(z)$ is the lower limit for the SNR in the partition that controls the stereo unmasking effects. It is predetermined and stored in tables for every sampling rate supported. Transforming the SNR into the power domain gives

$$bc(z) = 10^{\frac{-SNR(z)}{10}} \tag{5.19}$$

5.2.8 Calculate the threshold for each partition

The actual energy threshold in each partition is given by

$$nb(z) = en(z) \bullet bc(z) \tag{5.20}$$

5.2.9 Pre-echo detection and window switching

Audio coding algorithms transform blocks of data and code them efficiently using the energy compaction properties of the transformation, supplemented by psychoacoustic analysis to extract perceptual redundancies. The Layer III algorithm uses the MDCT to transform the subband data.

The longer the block length, the better is the frequency resolution of the transform but the poorer is its time resolution. For relatively stationary signals, long blocks provide better compression (coding gain). On the other hand, the characteristics of transients are best captured with short time windows. For best results, the size of the block has to be adapted to the statistics of the signal.

If a sharp attack occurs at the end of a long block, the psychoacoustic model would be misled to derive a higher masking threshold for that entire block. As a result, the signal is coarsely quantized. In the time domain, the quantization noise is spread over the entire block and it would be higher than the signal level at the beginning of the block, manifesting itself as a perceptible *pre-echo* just before the attack of the signal.

To control pre-echo, the first problem is to detect the occurrence of such transients. The decision to switch to short windows is derived from the calculation of the masking threshold by calculating the estimate of psychoacoustic entropy (PE) and switching when it exceeds an empirically determined value.

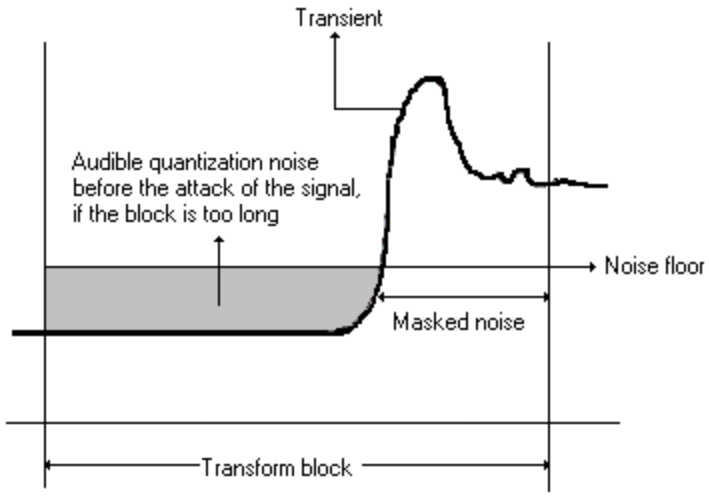


Fig. 5.6 Pre-echo distortion for long blocks.

An empirically determined absolute masking threshold, the *threshold in quiet* $Tq(f)$, as defined in Eq. 2.1, is used as a lower bound on the audibility of sound. The threshold derived for the current frame is compared with that of the previous frame and the threshold in quiet. The maximum of these values is chosen as the threshold.

$$thr(z) = \max\left[Tq(z), \min\left[nb(z), nb_{t-1}(z), nb_{t-2}(z)\right]\right] \tag{5.21}$$

where $nb_{t-1}(z)$ and $nb_{t-2}(z)$ are the energy thresholds from the previous two frames, that are computed as

$$\begin{aligned} nb_{t-1}(z) &= 2 \bullet nb(z) \\ nb_{t-2}(z) &= 16 \bullet nb(z) \end{aligned} \tag{5.22}$$

For Layer III, the perceptual entropy is determined as outlined below

$$PE = -\sum_{b=1}^{Z_{\max}} \left[cbwidth(b) \bullet \log\left(\frac{thr(b)}{eb(b)+1} \right) \right] \tag{5.23}$$

where $cbwidth(b)$ is the width of the threshold calculation partition.

The Layer III algorithm also incorporates other measures to compensate for pre-echo such as a 'bit-reservoir'. The bit reservoir is a pool of extra coding bits that are saved when the target quality is met below the estimated bit-rate. These extra bits are used in compensating for the demand of additional bits while coding transients (Sec 5.5.4).

5.2.10 Calculate the JND estimate

For use of Model 2 with Layer I or II, the JND estimate computed (in the threshold partition domain) above is spread over the spectral lines before comparing against the absolute threshold of hearing.

$$nb(f) = \frac{nb(z)}{f_{high} - f_{low} + 1}$$
$$thr(f) = \max\left[nb(f), Tq(f)\right] \tag{5.24}$$

For Layer III encoding, the threshold is not spread over the FFT lines. Instead, the threshold calculation partitions are converted directly into scale-factor bands. There are tables that indicate the number of partitions that go into each scale-factor band. The first and last partitions in each scale-factor band are weighted by $w1$ and $w2$ respectively. For each sampling frequency supported, there are 21 bands for long (and transition) windows and 12 bands each for short windows.

The energy in each scale-factor band is given by

$$en(sb) = w1 \bullet eb(bu) + \sum_{b=bu+1}^{b=bo-1} eb(b) + w2 \bullet eb(bo) \qquad (5.25)$$

The parameters bo and bu, used for converting threshold calculation partitions to scale-factor bands, are listed in tables. So are the weights $w1$ and $w2$, for each scale-factor band.

The threshold in each scale-factor band is given by

$$thrn(sb) = w1 \bullet thr(bu) + \sum_{b=bu+1}^{b=bo-1} thr(b) + w2 \bullet thr(bo) \qquad (5.26)$$

5.2.11 Calculation of the signal-to-mask ratio (SMR)

For Layers I and II, SMR is calculated as a ratio of signal energy within the subband. Each subband is identified as a psychoacoustically narrow or wide scale-factor band. A psychoacoustically narrow scale-factor band is one whose width is less than approximately $\frac{1}{3}$ critical band.

The energy in each scale-factor band $epart_n$ is

$$epart_n = \sum_{f=flow_n}^{fhigh_n} R^2(f) \qquad (5.27)$$

For a narrow scale-factor band, the noise level $npart_n$ is calculated as

$$npart_n = \sum_{f=flow_n}^{fhigh_n} thr(f) \qquad (5.28)$$

For a psychoacoustically wide scale-factor band, the noise level $npart_n$ is

calculated as

$$npart_n = \min\left[thr(flow_n), \ldots thr(fhigh_n)\right](fhigh_n - flow_n + 1) \tag{5.29}$$

The SMR is defined as

$$SMR_n = 10\log_{10}\left[\frac{epart_n}{npart_n}\right] \tag{5.30}$$

For Layer III, the SMR in each scale-factor band is defined as

$$SMR(sb) = \frac{thrn(sb)}{en(sb)} \tag{5.31}$$

This is the final output of the psychoacoustic model.

5.3 MDCT and the Hybrid Filterbank

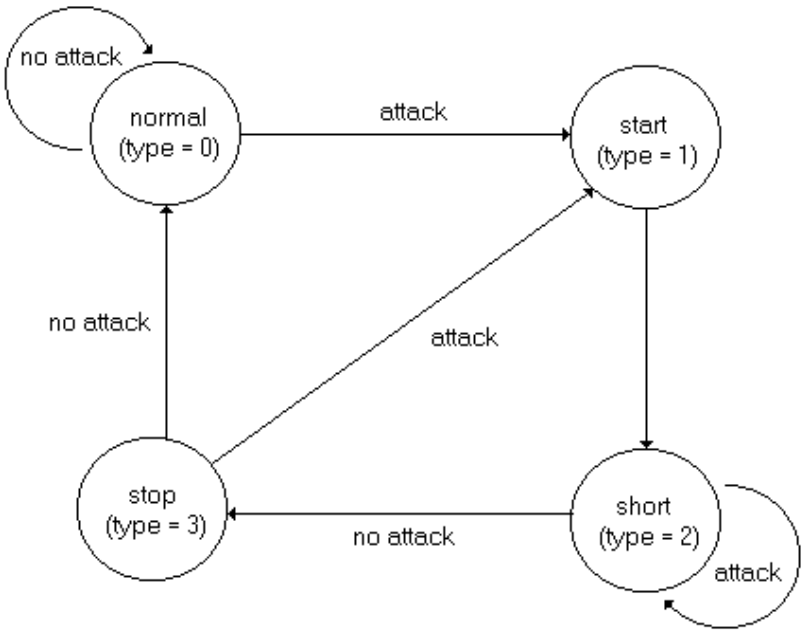


Fig. 5.7 Window-switching State Machine.

The filterbank outputs are processed using a Modified Discrete Cosine Transform (MDCT) as it has very good energy compaction properties. Unlike the filterbank, the MDCT is a lossless transformation. The MDCT further subdivides the filterbank outputs in the frequency domain. Layer III specifies two block sizes for the transform, a short block of 6 samples and a long block of 18 samples. The MDCT has been designed such that there is a 50% overlap between adjacent time windows. As a result in the short block mode, it takes 12 time domain samples and gives 6 frequency domain samples. Similarly for the long block mode it results in 18 frequency lines for 36 time domain samples. Prior to cascading the subband outputs with the MDCT, each of the odd subbands must undergo a frequency inversion correction so that the spectral lines will appear in proper monotonic ascending order. The frequency inversion consists of multiplying each odd sample by –1.

The psychoacoustic model detects conditions of pre-echo and triggers short blocks for transients (better time resolution) or long blocks (better frequency resolution) for signals with stationary statistics. When the perceptual entropy exceeds the value 1800, an empirically determined constant, the MDCT filterbank is switched to short windows. To maintain perfect reconstruction properties of the MDCT, switching between short and long blocks cannot be instantaneous. Long-to-short and short-to-long transition windows are provided for this purpose. Fig. 5.7 illustrates the possible state transitions for the window switching logic.

The size of the short block is one-third the size of a long block. In the short block mode, three short blocks replace one long block so that irrespective of the kind of

window applied, the number of MDCT lines remains constant. For a particular block of data, all the filterbank channels can have the same MDCT block-type (short or long) or a mixed mode where the two lower frequency subbands have long blocks while the remaining 30 upper bands have short blocks. The mixed mode provides better frequency resolution for the lower frequencies, while maintaining a high time resolution for the higher frequencies.

The polyphase filterbank and the MDCT are together called as the *Hybrid Filterbank* as they adapt to signal characteristics.

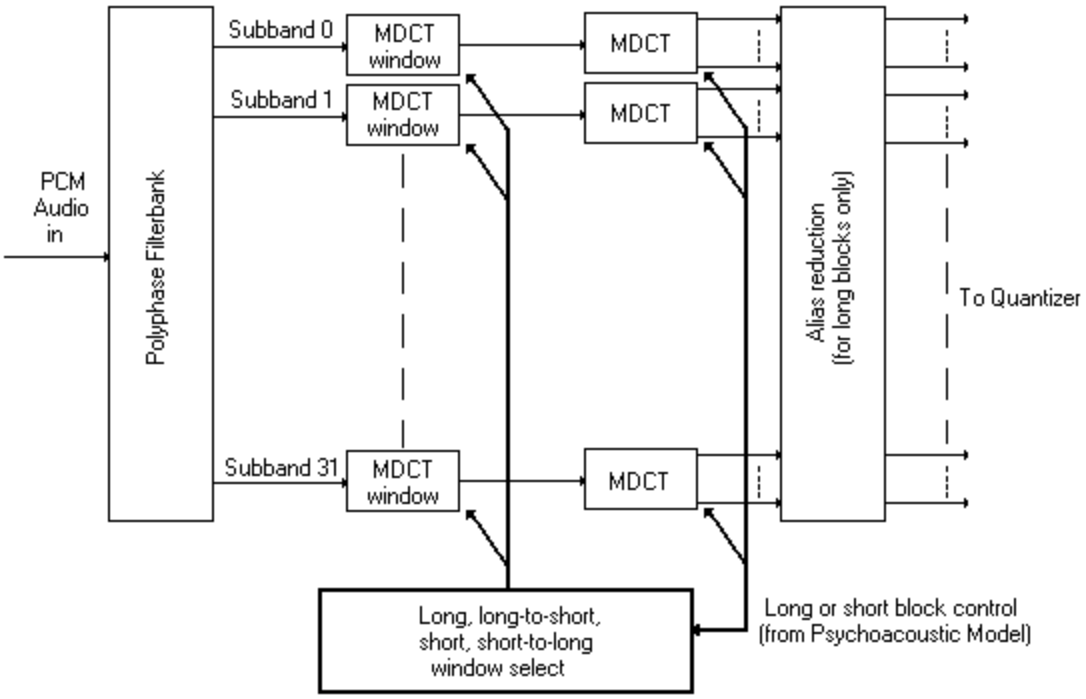


Fig. 5.8 Hybrid Filterbank.

Once the MDCT converts the audio signal into the frequency domain, the aliasing introduced by the subsampling in the filterbank can be partially cancelled. Alias reduction is applied for long blocks only.

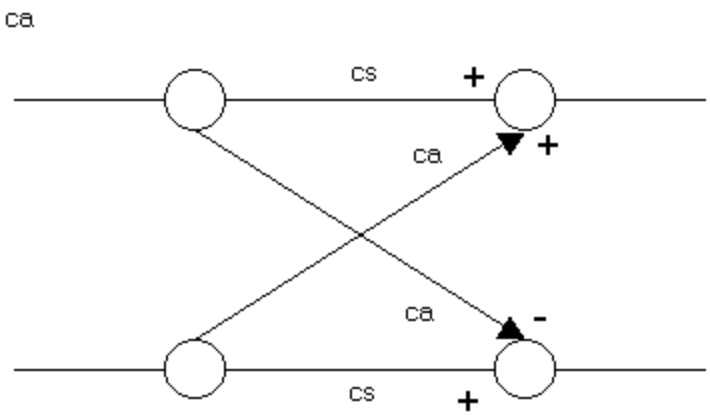


Fig. 5.9 Alias reduction butterfly for the Encoder.

The anti-aliasing butterfly for the encoder is as shown in Fig. 5.9. Each anti-aliasing butterfly is an orthonormal transformation (rotation) applied to one of the eight designated pairs of spectral lines. They do not affect the perfect reconstruction properties of the filterbank, but improve the compression factor of the coder by trying to contain the energy within each subband.

As can be seen in Fig. 5.10, a granule (group of 576 samples) is the time-frequency mapping of half a frame of the input data. Unlike the usual time-frequency plane, they are now rearranged for alias-reduction - first in the order of frequency (subband) and then in time. So, 18 time-domain samples of each subband are grouped together. The butterflies are applied to every alternate sample in every alternate subband.

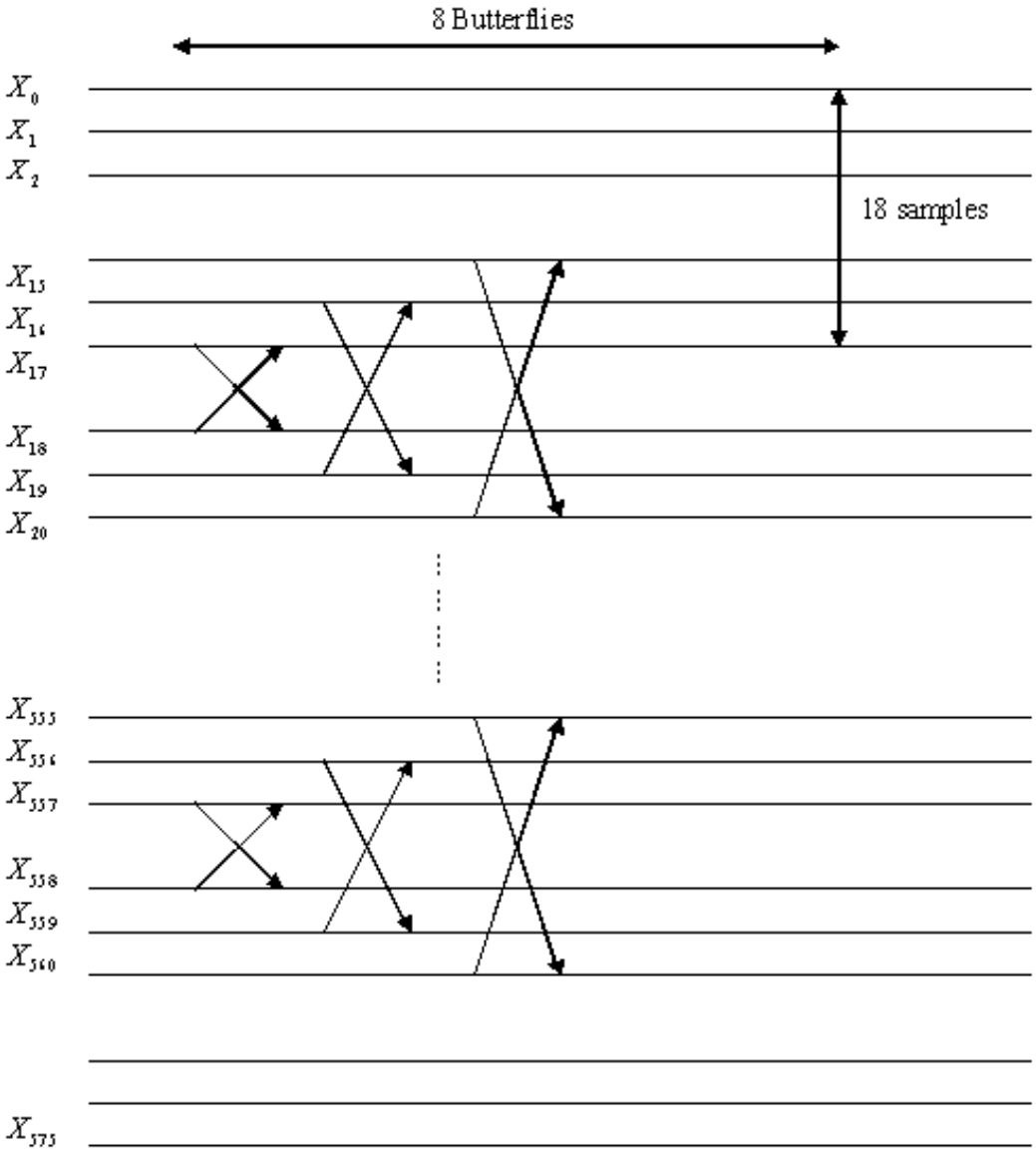


Fig. 5.10 Alias reduction operations for a granule of MDCT data.

The decoder has to undo this in order for the inverse MDCT to reconstruct the subband samples in their original aliased form for reconstruction by the synthesis filterbank. The decoder butterflies are similar, except for changes in sign.

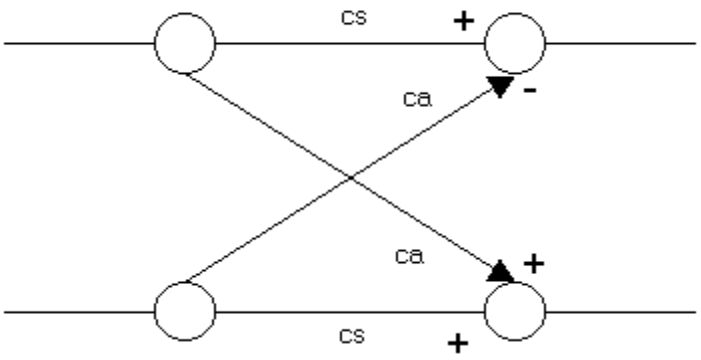


Fig. 5.11 Alias reduction butterfly for the Decoder.

5.4 The Noise Allocation, Quantization and Coding

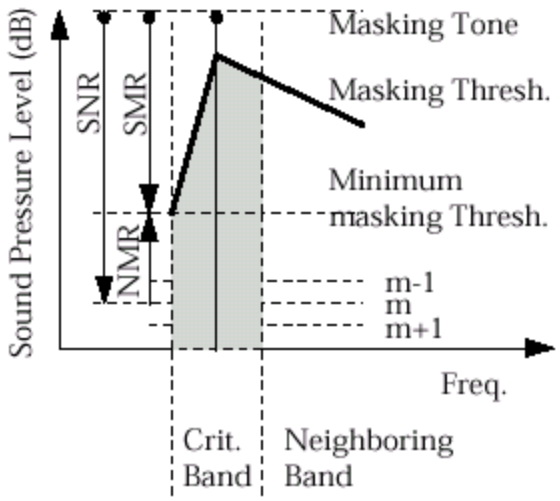


Fig. 5.12 Calculation of mask-to-noise ratio based on simultaneous masking [78].

Consider the tone in Fig 5.12. The masking produced by this tone is determined by simulating the excitation pattern on the BM (by applying a spreading function), deleting an offset and finally comparing it with the threshold in quiet. Assuming that the masker is quantized using an m-bit uniform scalar quantizer, noise might be introduced at level *m*. *Signal-to-mask ratio* (SMR) and *Mask-to-Noise ratio* (MNR) represent distances

from the minimum masking threshold to the masker and noise levels respectively, in the log domain.

The mask-to-noise (noise-to-mask) ratio can be computed as

$$MNR_{db} = NMR_{dB} = SNR_{dB} - SMR_{dB} \tag{5.32}$$

where SNR is the signal-to-noise ratio and SMR is the signal-to-mask ratio from the psychoacoustic analysis.

The Layer III encoder quantizes the spectral values by allocating just the right number of bits to each subband to maintain perceptual transparency at a given bit-rate. It controls and shapes the spectrum of the quantization noise to lie below audible levels. So, the scheme is called as noise allocation, as opposed to bit allocation.

Rate control (quantization of spectral values) is realized using two nested loops. The outer loop is called as the *distortion control loop* while the inner loop is called *rate control loop*. The outer loop controls the quantization noise produced by the quantization of the frequency lines within the inner loop. The inner loop does the actual quantization of the spectral values to meet the desired bit-rate.

So, the inner loop chooses a quantizer, quantizes the spectral values and counts the number of bits required for Huffman coding. If the resulting bit-rate is higher than the desired rate, it iteratively increases the quantization step-size, recomputes the resulting bit-rate; and continues to do so until the target rate is met. The outer loop computes the quantization noise resulting from this coarse quantization. If some of the scale-factor bands have more than the permissible distortion, it amplifies the corresponding values and as a result decreases the quantization step-size for those scale-factor bands.

This process continues till

- There is no scale-factor band with more than the allowed distortion.
- All scale-factor bands are already amplified
- The amplification of at least one band exceeds the upper limit, which is determined by the transmission format of the scale-factors.
- A time-out is reached (for real-time implementations).

5.5 Other refinements in Layer III

In addition to the MDCT window switching and alias reduction, the Layer III algorithm includes other refinements like

5.5.1 Non-uniform Quantization

The quantizer raises its input to the ¾ power before quantization to provide a more consistent SNR over the range of quantizer values. This is undone at the decoder. The encoder the samples *xr[i]* are quantized to *ix[i]* according to the equation

$$ix[i] = nint\left[\left(\frac{|xr[i]|}{2^{\frac{quantizerstepsize}{4}}}\right)^{\frac{3}{4}} - 0.0946\right] \qquad (5.33)$$

where *nint()* represents the nearest integer. The maximum allowed quantized value is limited to constrain the size of the tables used for lookup at the decoder.

5.5.2 Scale-factor bands

The scale-factor bands cover several MDCT coefficients and have approximately critical-band widths. . Scale-factor values are adjusted in the noise allocation loop to fit the masking threshold.

5.5.3 Entropy coding of quantized values

Similar to the perceptual model, the lossless coding backend also divides a frame into two granules. A granule is defined as a set of 576 frequency lines that carry their own side information. Due to energy compaction and quantization, the higher frequency components are zero or have negligible energy. The quantized samples are ordered by increasing frequency, to get a string of zeros at the end of the spectrum. For short blocks, the coefficients are arranged in ascending order, first by block and then by frequency number. Starting from the Nyquist frequency, the longest possible stretch of pairs of zeros is identified. Their number is named *rzero*. Next, quadruples of quantized values with absolute value not exceeding 1 (3 possible quantization levels) are identified. They are named *count1*. The remaining part of the spectrum may contain spectral amplitudes upto 8191 (13 quantization levels). This part of the spectrum, called *big values* is divided into three significant regions and coded using Huffman tables optimized for the statistics of each region.

Fig. 5.13 depicts the unquantized MDCT for a sub-frame of audio data. Since this has been flagged as a short block by the perceptual model, the coefficients are reordered for entropy coding. The rate-control loop iteratively changes quantizers till the target rate of 128 kb/s is met while maintaining the noise-floor below the masking threshold. The resulting coefficients are as shown in Fig. 5.14. It can be seen that some of low-energy coefficients present at higher frequencies are discarded in the process.

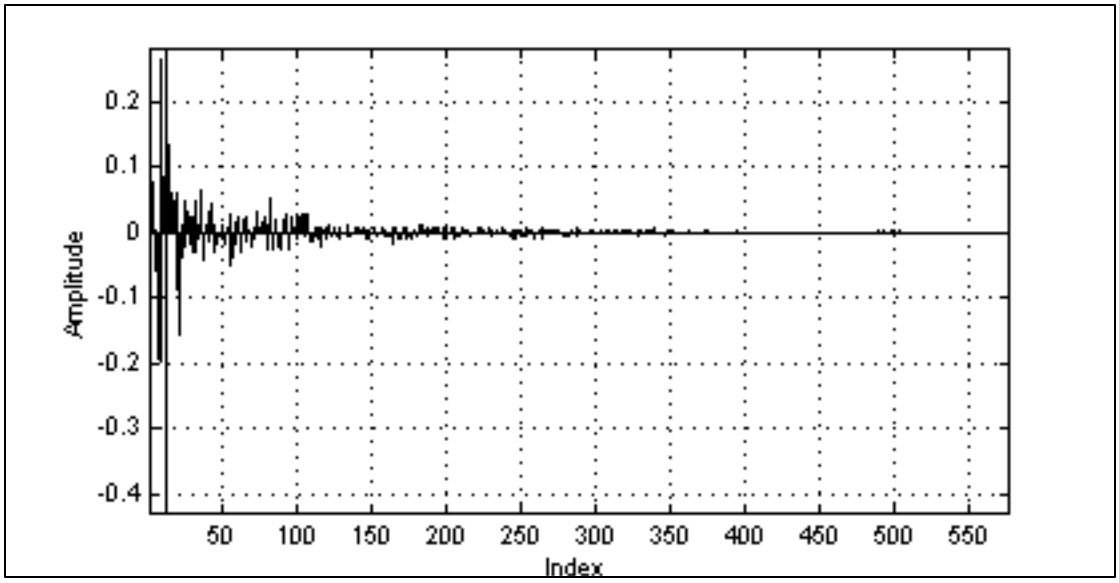


Fig. 5.13 MDCT coefficients for a short block.

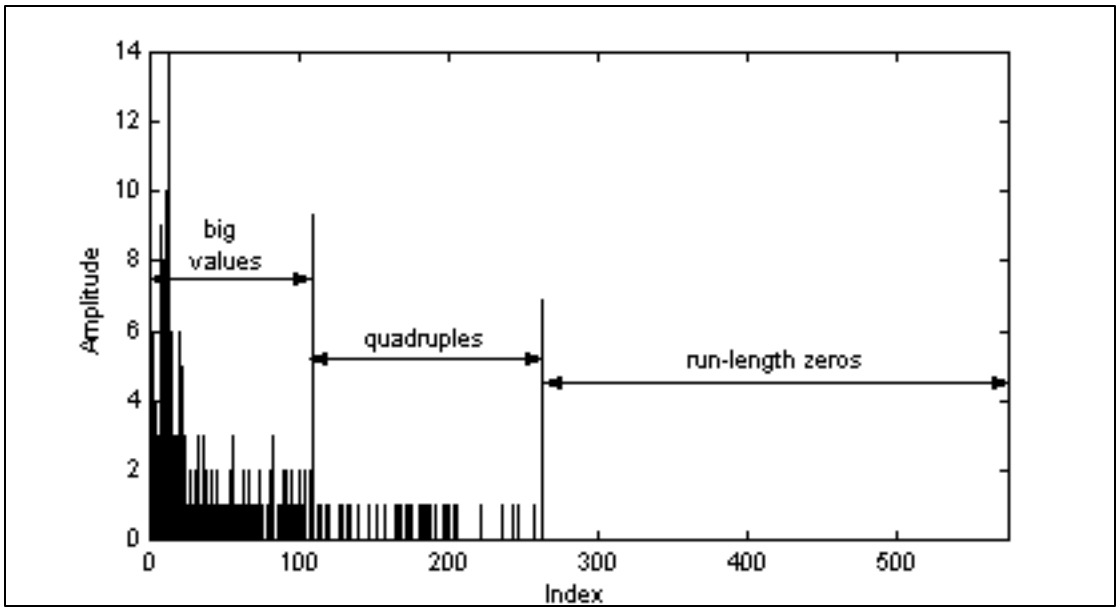


Fig. 5.14 MDCT coefficients (magnitude) quantized to meet a target bit-rate of 128 kb/s.

5.5.4 Bit reservoir

The encoder operates on blocks of data 1152-samples long. When a frame of data is coded with less than the average number of bits necessary, the encoder donates the

extra bits to a reservoir. It can borrow these extra bits when the coding gain is low, especially during transients, to maintain perceptual quality. The encoder can borrow bits donated from past frames; it cannot borrow bits from future frames. As a result, the coded representation for a block of data need not necessarily be confined to one frame in the bitstream; it can start from previous frames.

5.6 Error Sensitivity, Detection and Concealment

5.6.1 Bit-error Sensitivity

Sensitivity of individual bits of the various logical elements of the bitstream can be indicated on a 6-point scale defined as below.

Sensitivity Index	Description
5	Catastrophic
4	Very Annoying
3	Annoying
2	Slightly Annoying
1	Audible
0	Insensitive

Table 5-1 Index of Bit-error Sensitivity.

The sensitivities of the various elements of the Layer III algorithm are indicated in

Table 5-1. The values are not results of precise measurements; rather they rely upon the knowledge of the codec. They assume that an error detection scheme is not in use. Some fields in the bitstream are variable length and all bits in these fields are rated for error sensitivity, even if not in use.

The header and error check information are the first logical elements in the frame. The decoder identifies a new frame by seeking for the synchronization sequence at the

beginning of the frame. Loss of synchronization means the loss of an entire frame of compressed data. The CRC word, if present is the next logical element. Therefore, the header and error check information are considered to have the highest sensitivity.

Parameters	**#bit**	**Sensitivity**
scfsi	all bits	5
part2_3 _length	all bits	4
big_values	all bits	3
global_gain	all bits	5
scalefac_compress	all bits	5
window_switching_flag	0	5
block_type	all bits	4
mixed_block_flag	0	4
table_select	all bits	5
region0_count	all bits	3
region1_count	all bits	3
preflag	0	2
scalefac_scale	0	2
count1_table_select	0	3
subblock_gain	2(msb) 1 0	4 3 2
scale_fac [1]	3(msb) 2 1 0	3(2) 3(2) 2(1) 2(1)
Huffmancodebits() [2]	0….n-1	3-0

Table 5-2 Table of Bit-error Sensitivity.

5.6.2 Huffman Codeword Reordering

Interleaving Huffman codewords as opposed to logical ordering provides implicit error robustness for the low frequency spectral components. If *max_hlen* is the maximum

[1] The scalefac length depends on scalefac_compress. The bit sensitivity values refer to the scalefac_scale value 1 (if 0, the value in parenthesis)

[2] If n is the number of bits for Huffman coding in one block the bit sensitivity decreases linearly from 3 to 0 as the bit number varies from 0 up to n (from low to high frequency)

length of a Huffman codeword (over the tables which are used to code the particular block) and n is the number of bits used for Huffman coding of data in the block (not frame), then $\text{int}\left(\frac{n}{\max\_ blen}\right)$ slots are filled with the codewords, starting at low frequencies. And the remaining codewords are filled into the remaining place, again in the order of increasing frequency. After interleaving, the sensitivity of bit $k + i \bullet \text{int}\left(\frac{n}{\max\_ blen}\right)$ decreases linearly from 3 to 0 as k varies from 0 up to $\left[\text{int}\left(\frac{n}{\max\_ blen}\right) - 1\right]$, where *i=0, …, max_hlen-1*, and n is the number of bits for Huffman coding of one block. This is the recommended practice for Layer III data for all channels where error robustness is important.

5.6.3 Error Concealment

Source-coded bitstreams are very susceptible to channel errors, both random and burst errors, when directly used in transmission applications. In such situations, it is always recommended to protect them with channel codes. The MPEG-1 algorithm provides an optional CRC to provide some error detection facility to the decoder. The CRC check diagram is shown in Fig. 5.15. The Hamming distance of this detection code is $d = 4$. This permits to detect up to 3 single bit errors or for the detection of one error burst of up to 16-bit length. The amount and the position of the protected bits within one encoded audio frame generally depend on the layer, the mode, data rate and sampling frequency.

When critical parts of the bitstream are corrupted, a simple method of error concealment is to repeat the previous frame, if it is error-free. Muting is another option. Error in specific parts of the decoding process can be handled by substitution of average values. For example, if there is an error in decoding Huffman-coded values, the corrupted value can be replaced by an average value.

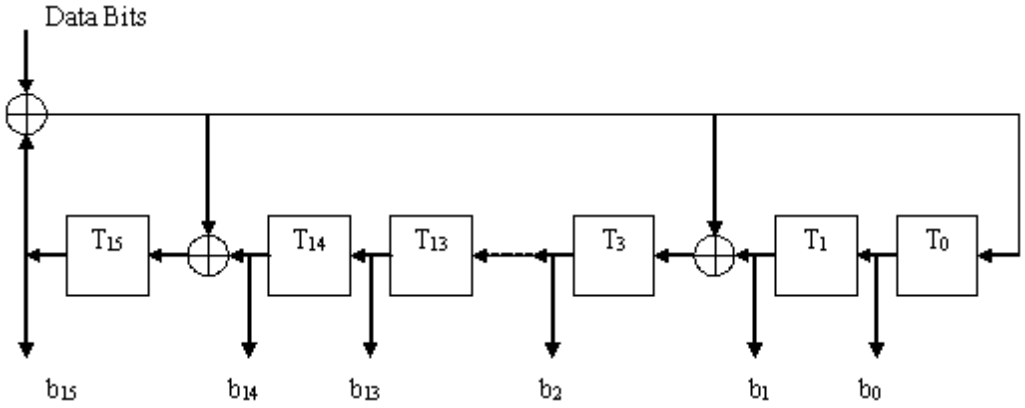


Fig. 5.15 CRC check diagram.

CHAPTER 6

THE ASU MP3TOOL: IMPLEMENTATION OF THE MP3 ALGORITHM IN MATLAB

The Arizona State University MATLAB MP3Tool is a graphical user-interface (GUI) based tool for introducing audio-DSP concepts to both undergraduate and graduate students. This chapter briefly reviews this simulation tool. The tool consists of a user-friendly graphical interface along with a complete MATLAB realization of all aspects of the audio MPEG-1 Layer 3 (MP3) algorithm. The tool is accompanied by a series of computer experiments and exercises that can be used to provide hands-on training to class participants. The tool may also be used by instructors in a class setting to demonstrate key signal processing concepts associated with the processing of high-fidelity audio. The MATLAB MP3 tool has been used in Arizona State University undergraduate DSP courses as well as in a graduate course on speech and audio coding and in a continuing education short course. A complexity profile for the implementation is also presented

6.1 Description of the ASU MP3TOOL

The tool is invoked by running the MP3Encoder.m script. A copyright notice is first displayed, as shown in Fig. 6.1. The user has to accept the terms of the copyright to use the software. Next, the tool asks the user if he would like to visualize and plot the results at crucial stages of the algorithm. On an affirmative response, the main user-interface for the tool will be enabled; else the visual cues are disabled. It is recommended to enable the GUI.

The *Encoder Configuration* menu is shown in Fig. 6.3 provides a graphical method to input the encoding parameters. Essential inputs are the names of the source and destination files and the target bit-rate. Other optional parameters include

- Mode selection: select from stereo, intensity and/or ms-stereo, stereo and mono
- De-emphasis: select from none, 0-15 μs and CCITT J.17
- Private Bit: bit for private use. This will no longer be used in future by ISO/IEC
- Error Protection: information for CRC-based error checks.
- Copyright: Is this material copyrighted ?
- Original: Is this material original ?

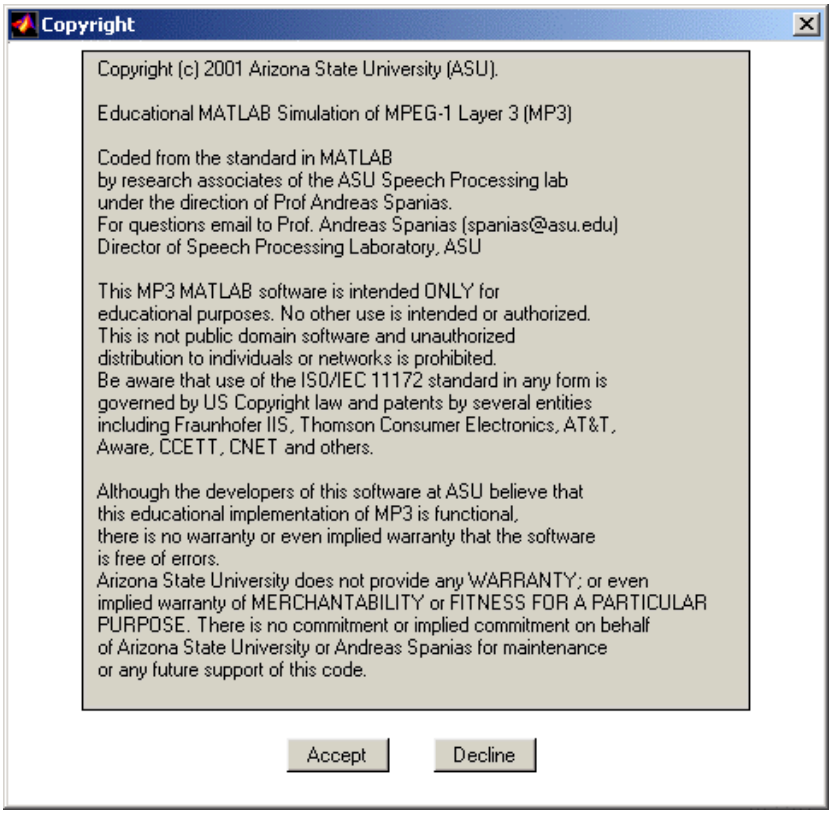


Fig. 6.1 The copyright notice.

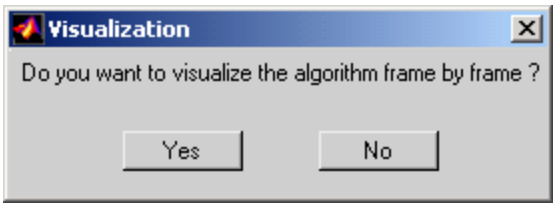


Fig. 6.2 The modal dialog to enable/disable the GUI.

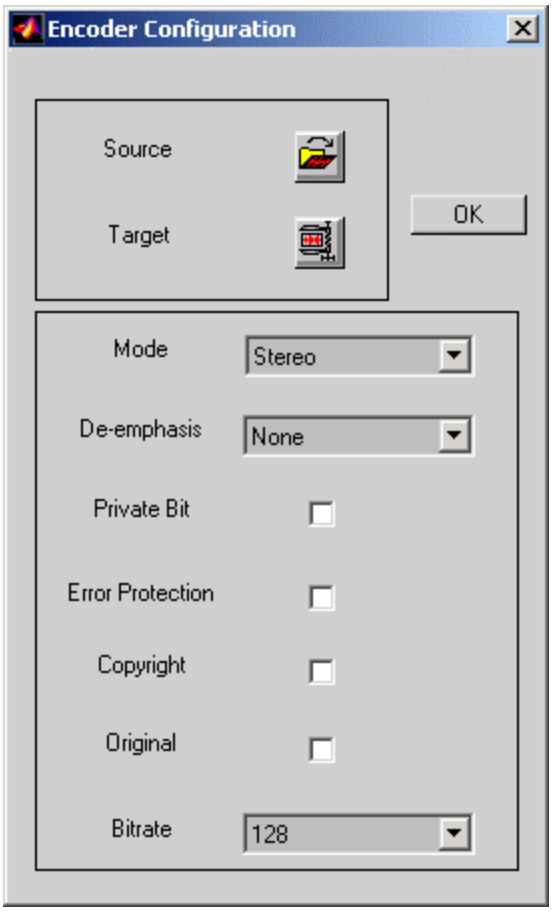


Fig. 6.3 The menu for determining the encoder configuration.

The main graphical user interface of the MATLAB MP3 tool is shown in Fig. 6.4. This consists of a block-by-block graphical representation of the MP3 algorithm. Each of the blocks is associated with the relevant signal processing functions that can be activated by opening the corresponding context menus.

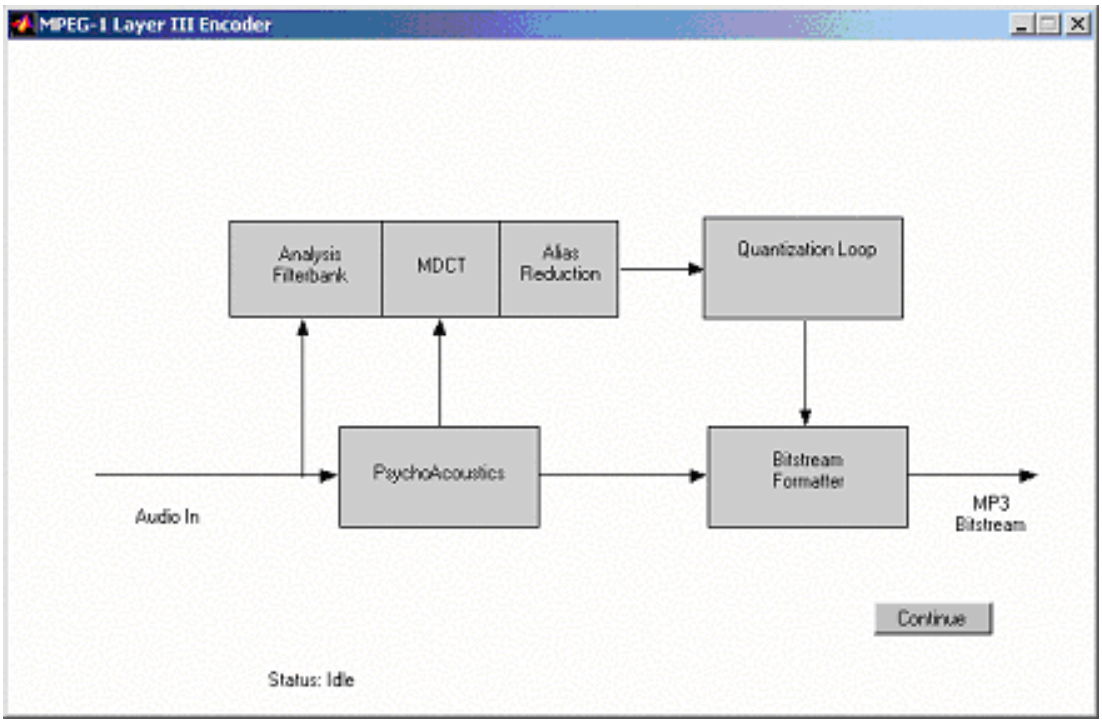


Fig. 6.4 The main user interface for the MP3Tool.

The algorithm operates on blocks of data. The block of PCM audio data to be encoded passes through a filterbank that divides the spectrum into 32 uniform subbands. The energy in each subband is further compacted by the application of the MDCT. The same frame of audio data is also fed to a psychoacoustics model that determines the ratio of signal energy to the masking threshold for each subband. Based on this threshold and the target bit-rate, the quantization block iteratively allocates bits to the MDCT spectral components of the various subbands to minimize the perception and audibility of the quantization noise. These quantized subband samples and the side information are packed into a coded bitstream by entropy coding. Every block of data, thus operated on, is represented as a frame in the coded bitstream. The algorithm steps through each frame of data and displays corresponding results.

The context menu of the analysis filterbank block provides for the visualization of not only the actual filterbank outputs, but also frequency responses of the prototype filter and the entire cosine modulated filterbank, as shown in Fig. 6.5 – 6.7.

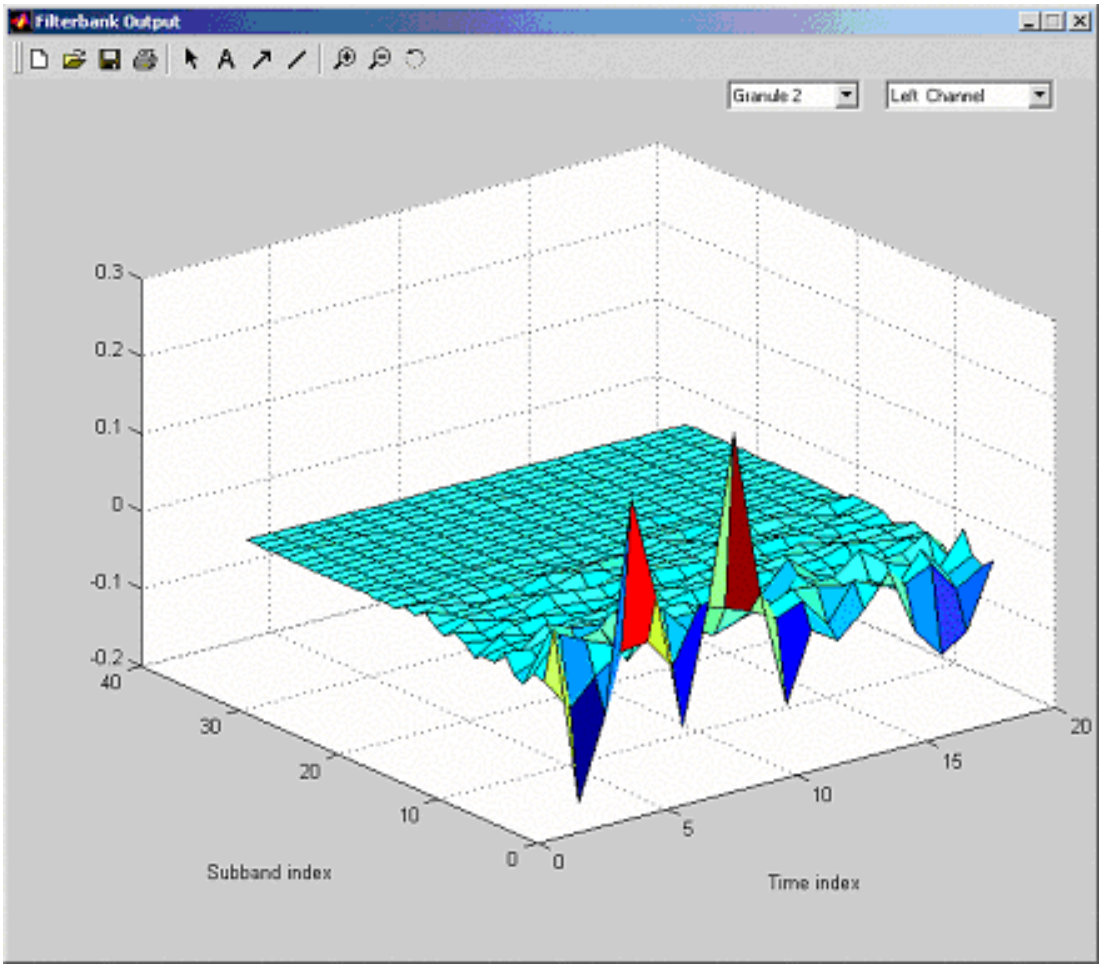


Fig. 6.5 The output of the Analysis Filterbank.

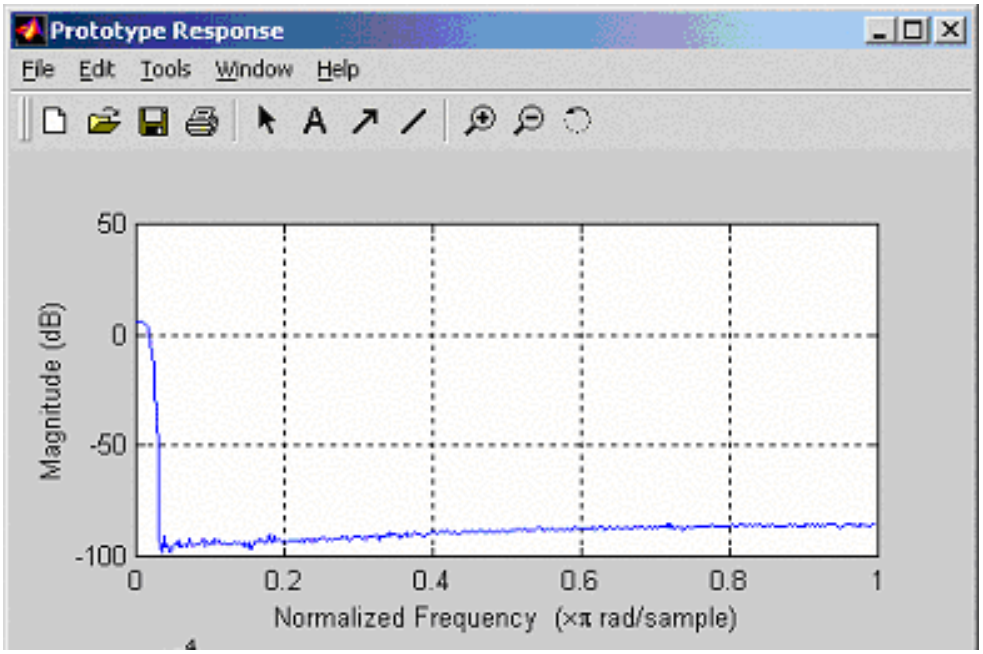


Fig. 6.6 The response of the Prototype filter.

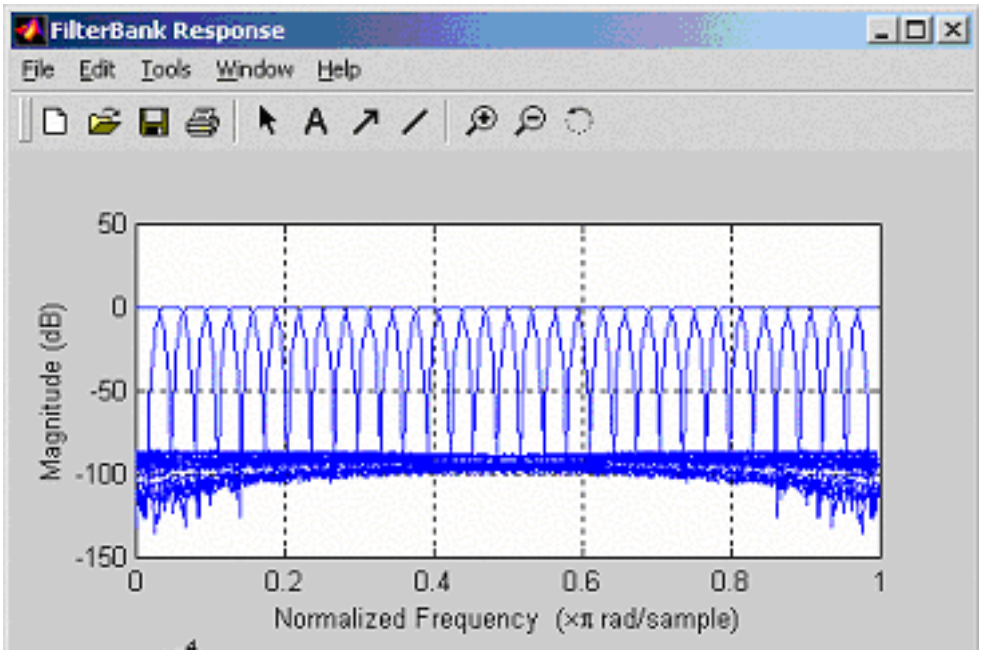


Fig. 6.7 The response of the Analysis filterbank.

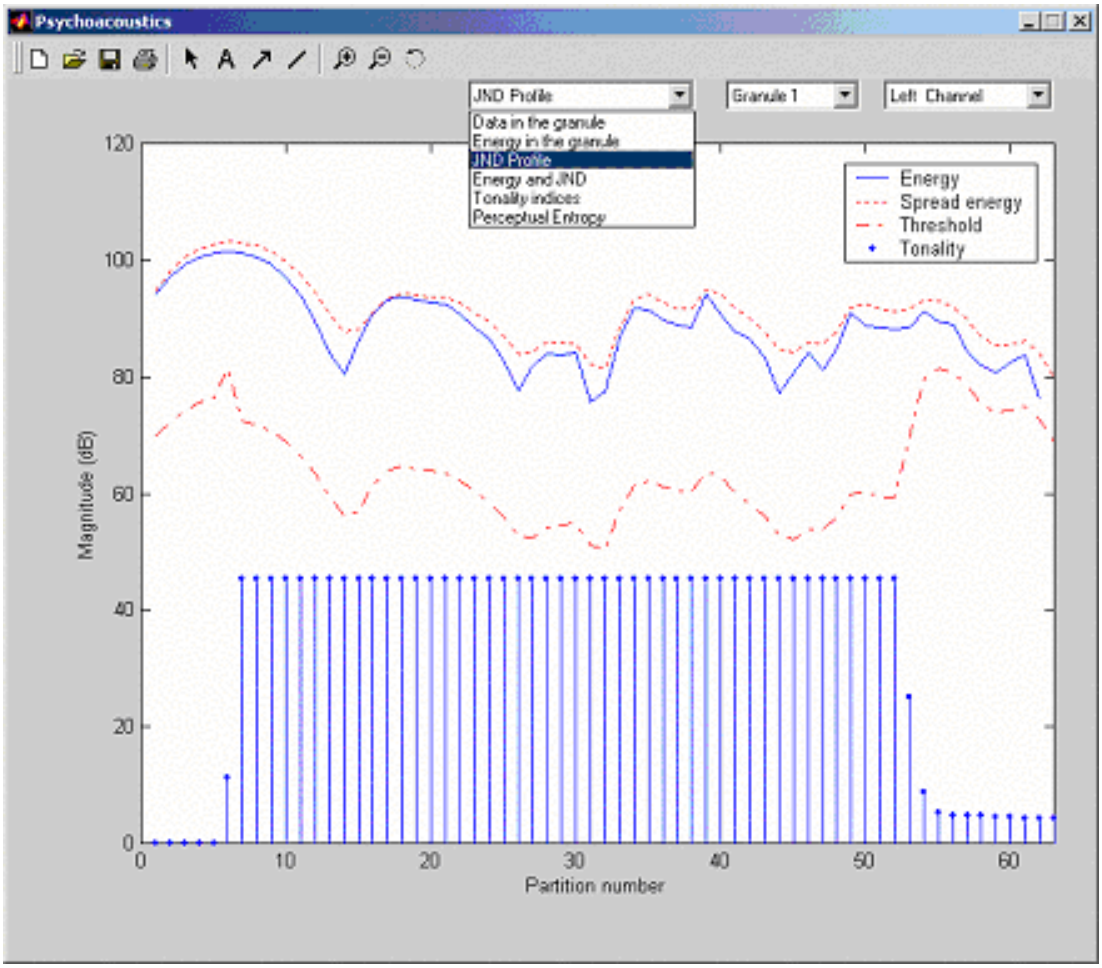


Fig. 6.8 The Psychoacoustics user interface.

Fig. 6.8 represents the user interface for the psychoacoustics model. A drop-down list-box provides many results to choose from; these results can be viewed for every granule of every channel. The data and its spectrum can be viewed. The energy in the partition domain, the resulting JND profile and an indicator of tonality of each of the components are provided on a single plot to elucidate the differences in the masking properties of tonal and noise maskers.

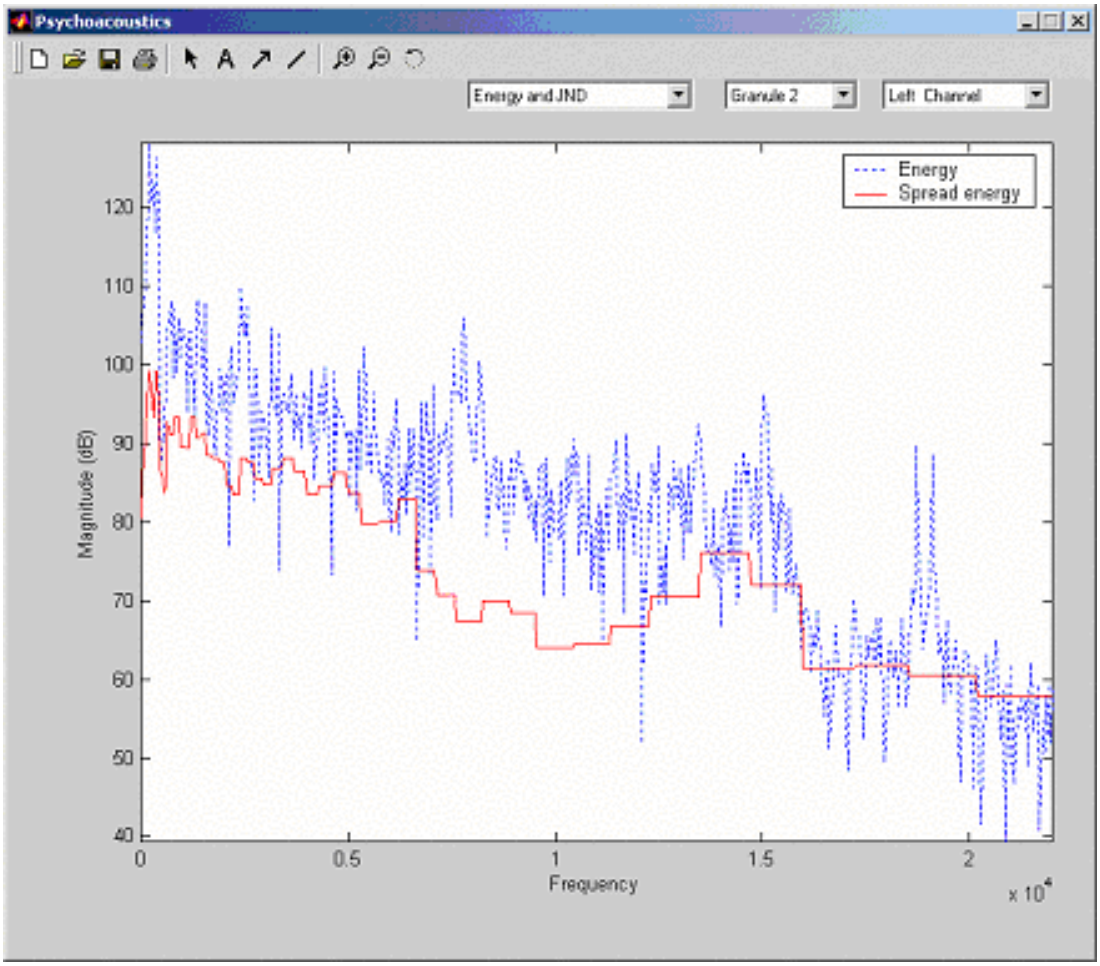


Fig. 6.9 The masking phenomenon.

Even though spreading the JND profile over the FFT components is not part of the coding strategy of the MP3 algorithm, it is a powerful demonstrator of the concept of masking, as shown in Fig. 6.9.

The MP3 algorithm tracks changes in the PE of the signal and switches the time-frequency resolution of the hybrid filterbank accordingly. Non-stationary spectra correspond to instances of high energy that have a potential to trigger a reorganization of the time-frequency plane. The tool uses a buffer to track and display the PE of the previous 10 frames and bring such events to the attention of the user.

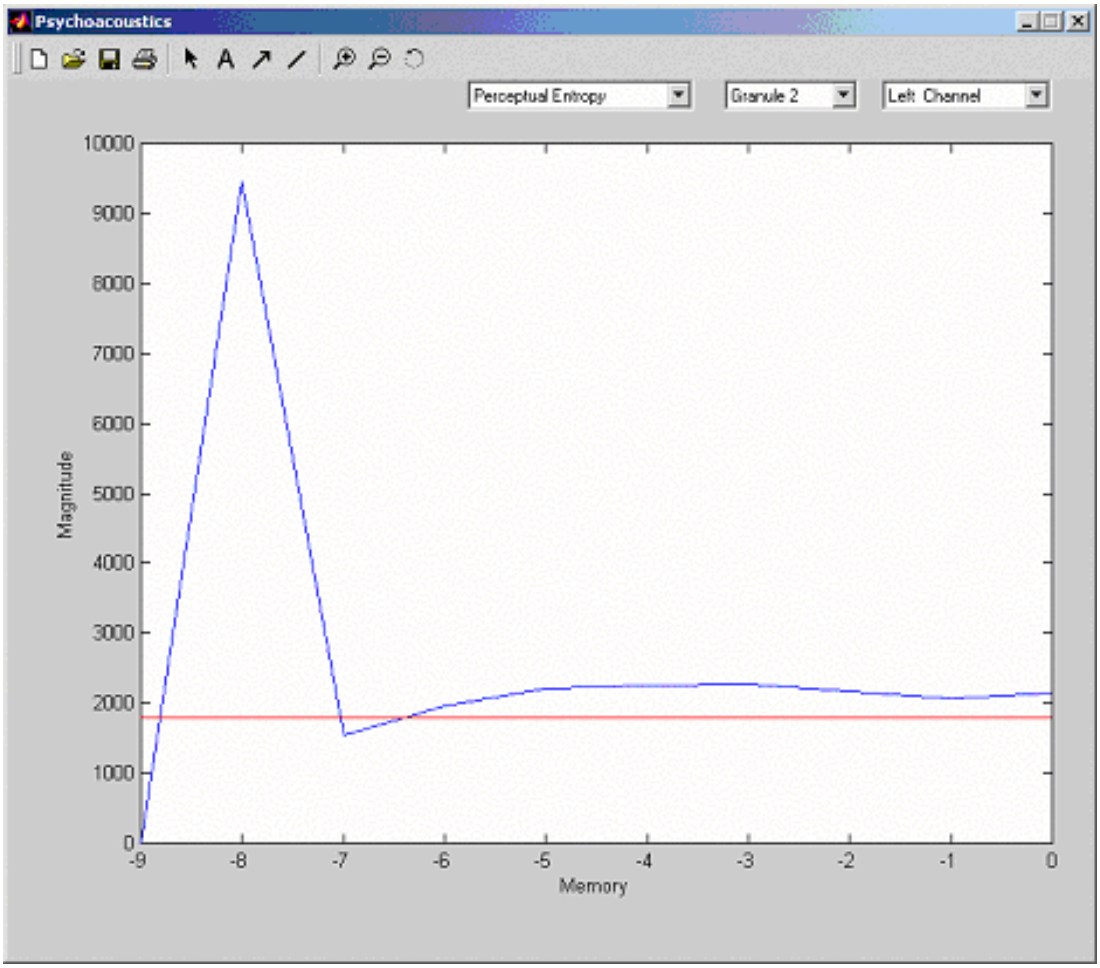


Fig. 6.10 The PE tracker.

The MDCT is a lossless transformation that further subdivides the filterbank outputs in the frequency domain. The algorithm specifies two block sizes for the transform, a short block of 6 samples and a long block of 18 samples. In the MDCT domain, the aliasing introduced by the subsampling in the filterbank can be partially cancelled by the application of alias reduction butterflies. This helps to contain the energy within each subband and increase the compression factor of the coder. The MDCT spectra are displayed for both the aliased and alias-reduced case, as shown in Fig. 6.11 and Fig. 6.12.

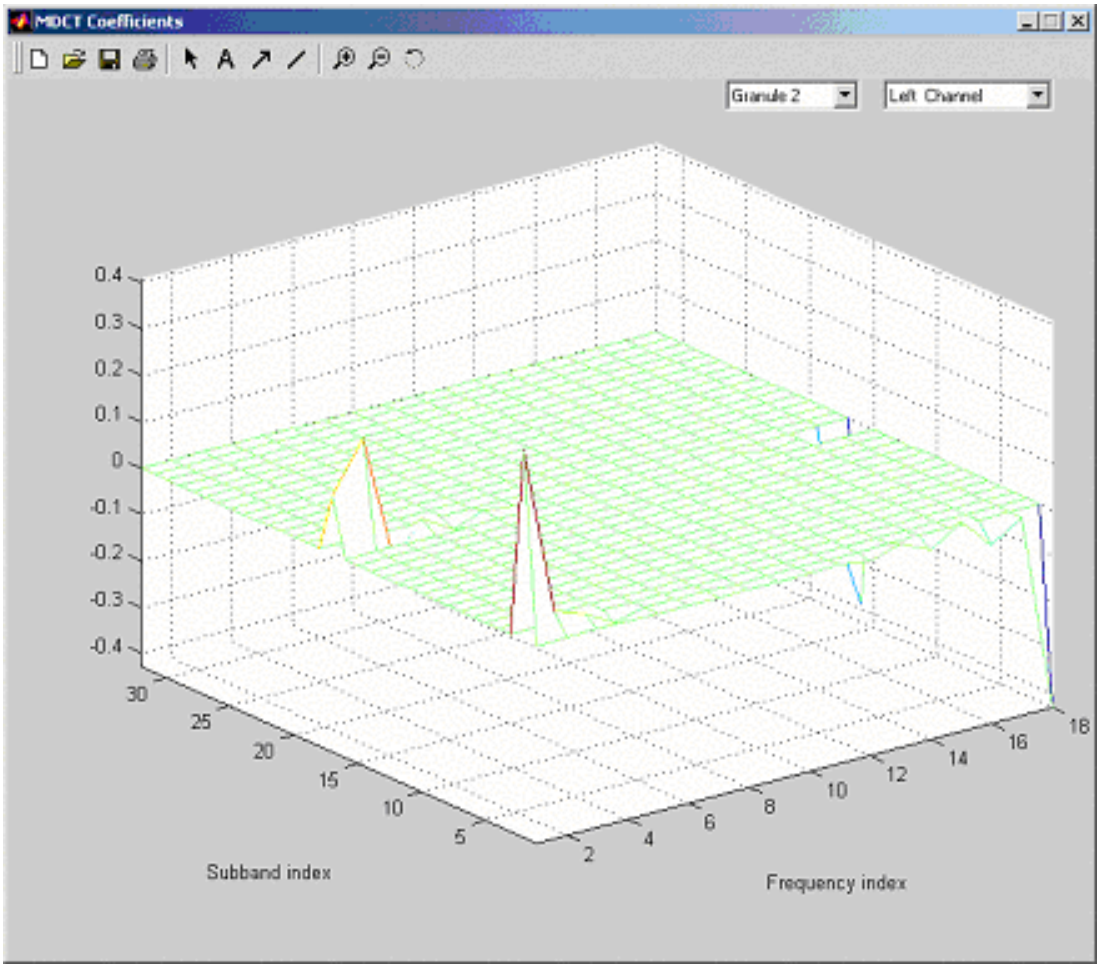


Fig. 6.11 The MDCT outputs with aliasing.

The quantized MDCT components are displayed as the output of the rate-control loop. The quantized spectrum is basically divided into three categories: run-length zeros, values in the range [-1, 1] and *big-values*. The *big-values* spectrum is further divided into three regions and coded with Huffman code-books designed for each region. Observation of the quantized spectrum shows the performance of the algorithm at various bit-rates. The degradation in signal quality at high compression ratios can, in part, be attributed to the truncation of high frequency spectral during rate-control.

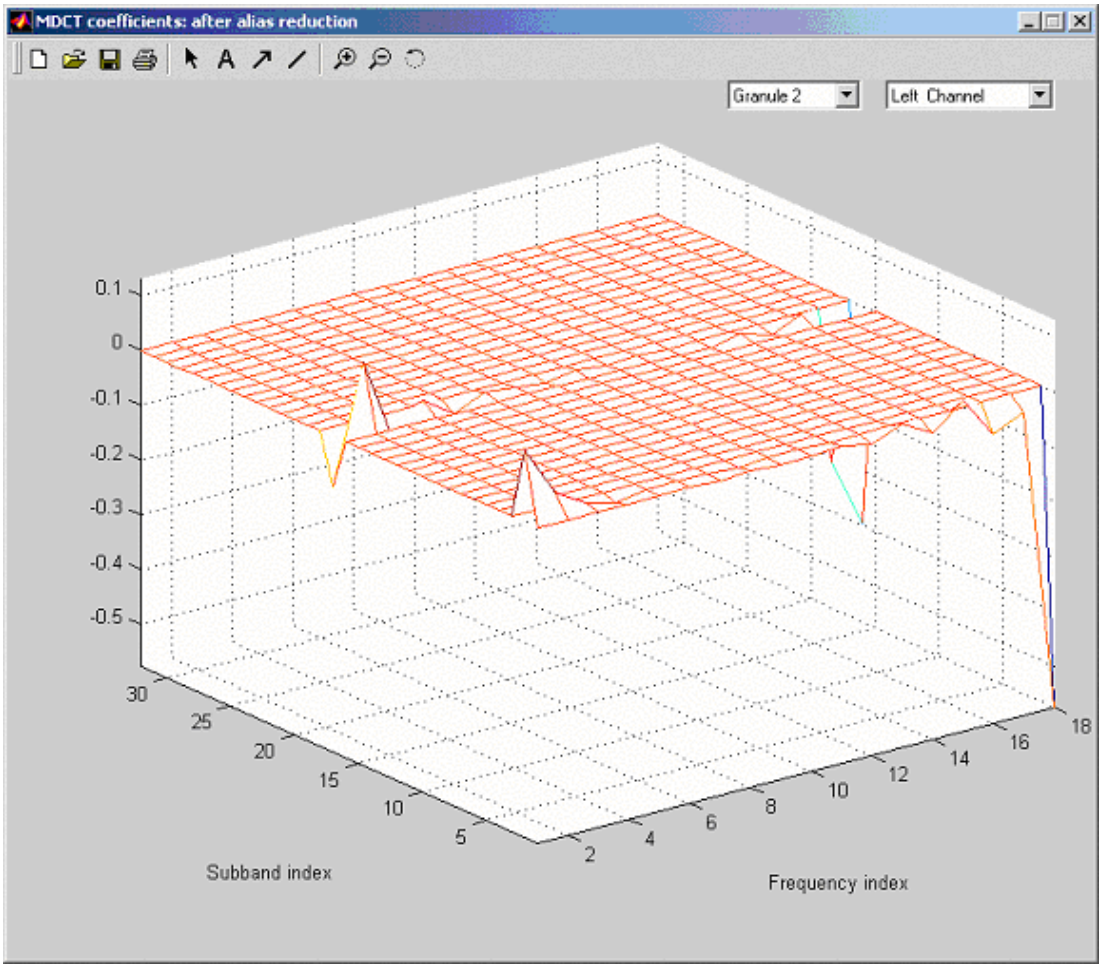


Fig. 6.12 The alias-cancelled result for the MDCT components in Fig. 6.8.

The most interesting part of the bitstream formatter is the bit-reservoir. The encoder operates on blocks of data 1152-samples long. When a frame of data is coded with less than the average number of bits necessary, the encoder donates the extra bits to a reservoir. It can borrow these extra bits when the coding gain is low, especially during transients, to maintain perceptual quality. A logical representation of the bitstream is provided to convey the idea and its parameters are updated every frame to demonstrate the bit-reservoir at work.

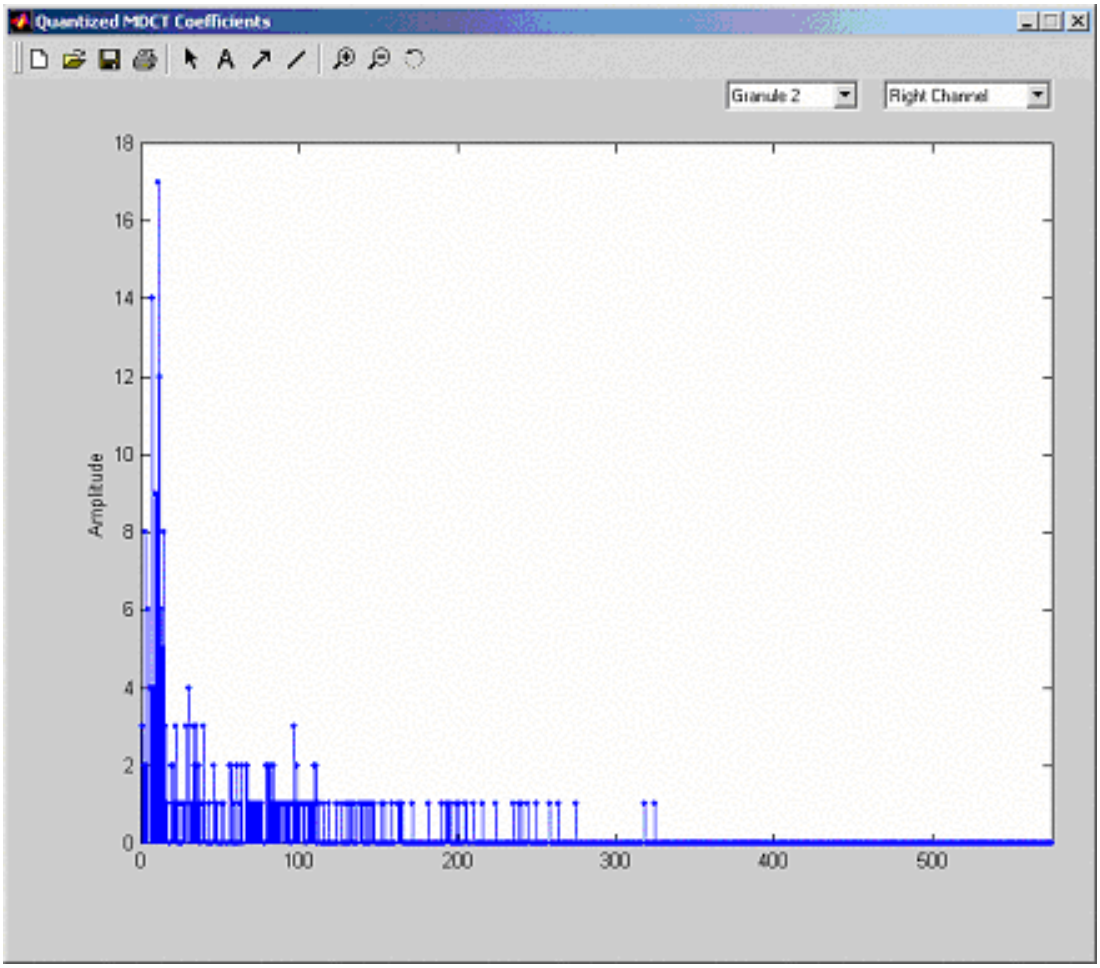


Fig. 6.13 The quantized MDCT coefficients at the output of the rate control loop.

6.1.1 Tutorial Exercises

A set of experiments have been designed to provide hands-on experience with the various aspects of the algorithm. Select exercises have been given to undergraduate students at ASU as part of a computer project in a DSP class. The students performed the experiments and responded to the questions in the quiz section. They then submitted a typed report that described the results obtained in the computer exercise along with several relevant figures and graphs that are copied in their report using a drag-and-drop process. Select exercises for our graduate class in speech and audio coding included

subjective evaluations based on records that are obtained by modifying audio parameters directly in the MATLAB MP3 code. A set of experiments in the form of a tutorial have been included in the Appendix.

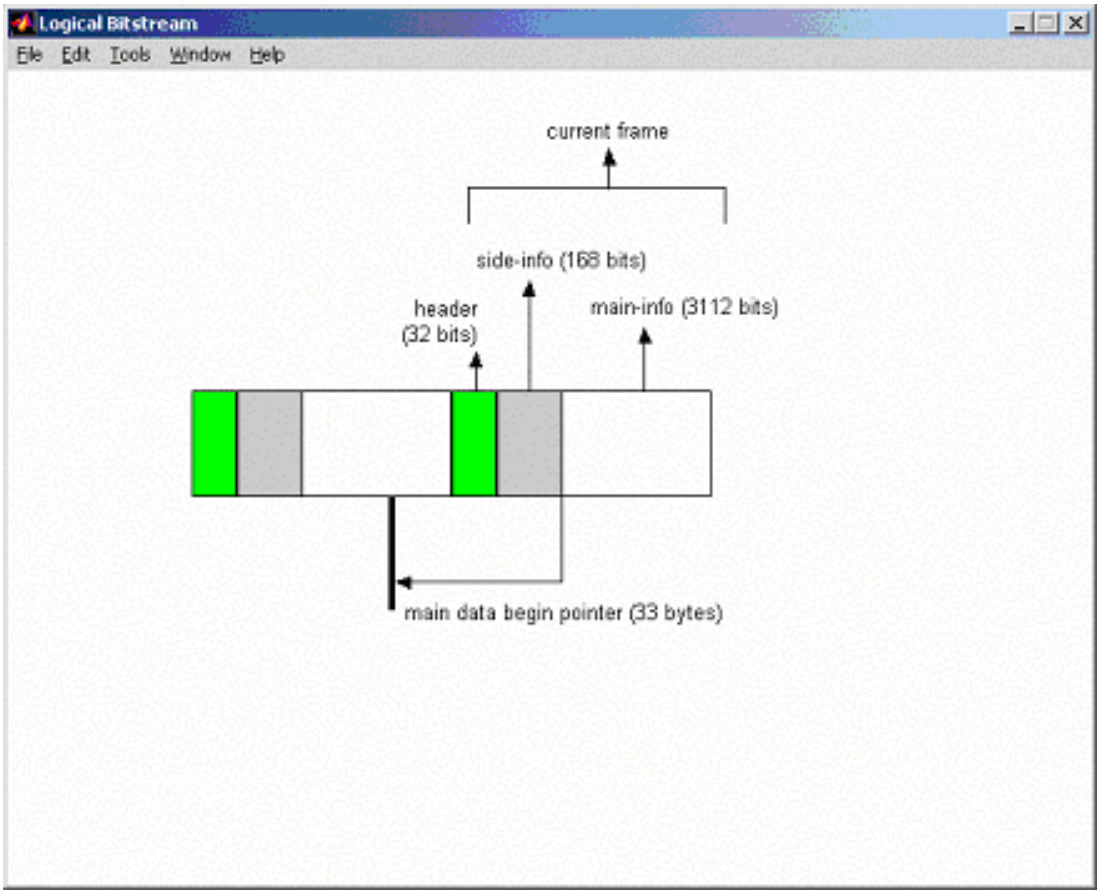


Fig. 6.14 The bit-reservoir in action.

6.2 Complexity Profile of the MATLAB implementation

To analyze the distribution of computational complexity, the encoder and decoder were profiled. The results are as shown in Fig. 6.15 – Fig. 6.17. It is to be noted that complexity profile is highly dependent on the tool (in this case, MATLAB), the Operating System (OS) and the underlying hardware architecture. MATLAB is a high level double-precision floating-point engine that provides a highly advanced and flexible

interface for testing and visualization. On the flip side, execution is slower than the more traditional compiled binaries and libraries. And bit-manipulation routines are definitely slower. The profile of the reference C-language implementation is also shown for comparison.

Bit-manipulation operations, nested for-loops and dynamic memory allocation are very expensive in MATLAB. Therefore, the bitstream formatting takes up most of the computational horsepower for the encoder. Most of the remaining processing bandwidth is used up in the rate-control loop. At the decoder, Huffman decoding is costly in terns of bit-manipulation and as expected, is the most expensive part of the algorithm.

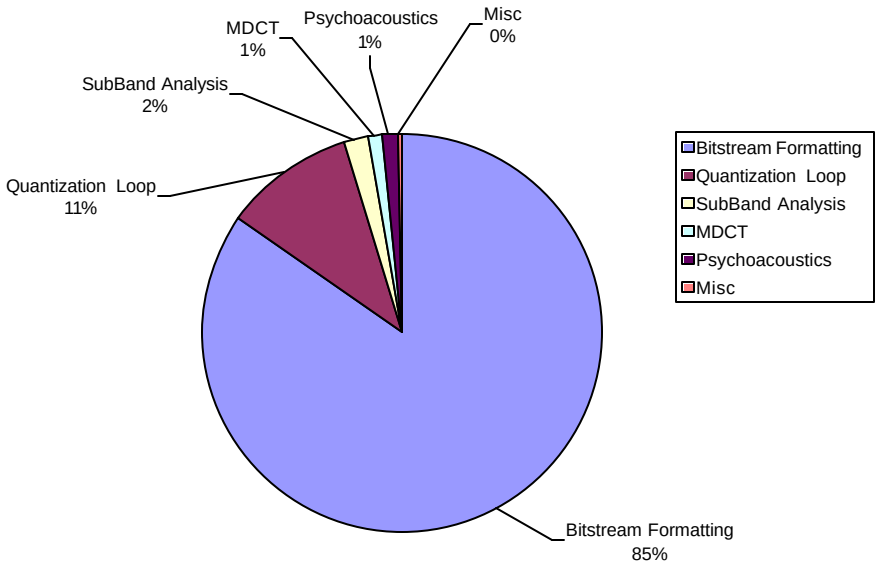


Fig. 6.15 Encoder Complexity Profile for the MATLAB implementation.

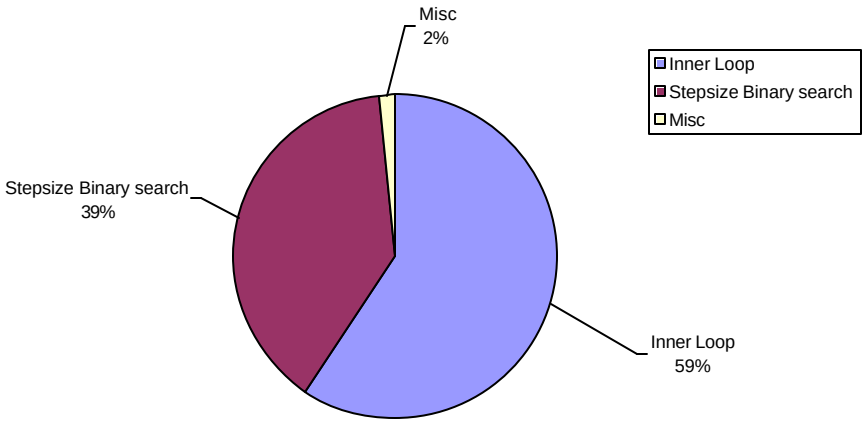


Fig. 6.16 Profile details of the Quantization Loop for the MATLAB implementation.

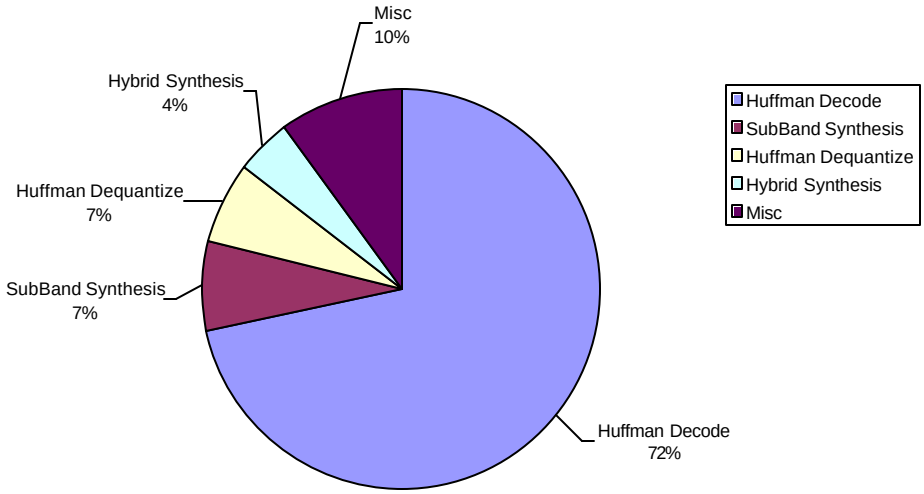


Fig. 6.17 Decoder Complexity Profile for the MATLAB implementation.

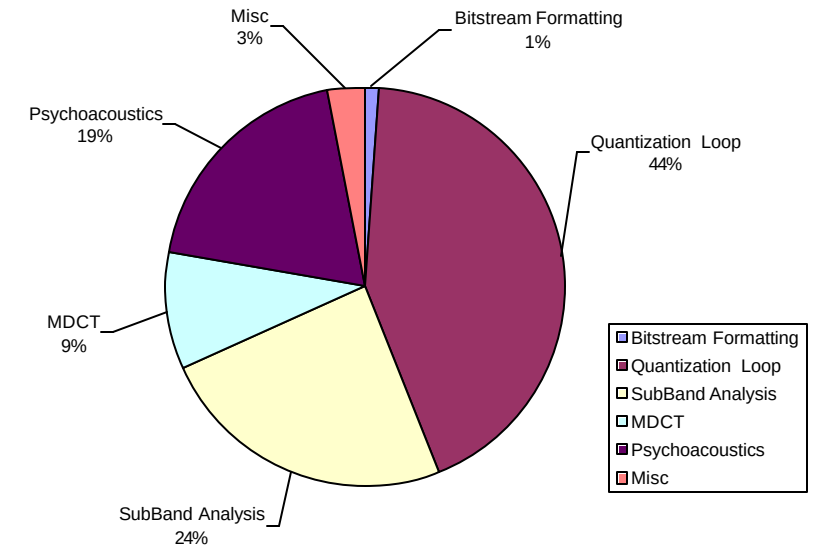


Fig. 6.18 Encoder Complexity Profile for the C implementation.

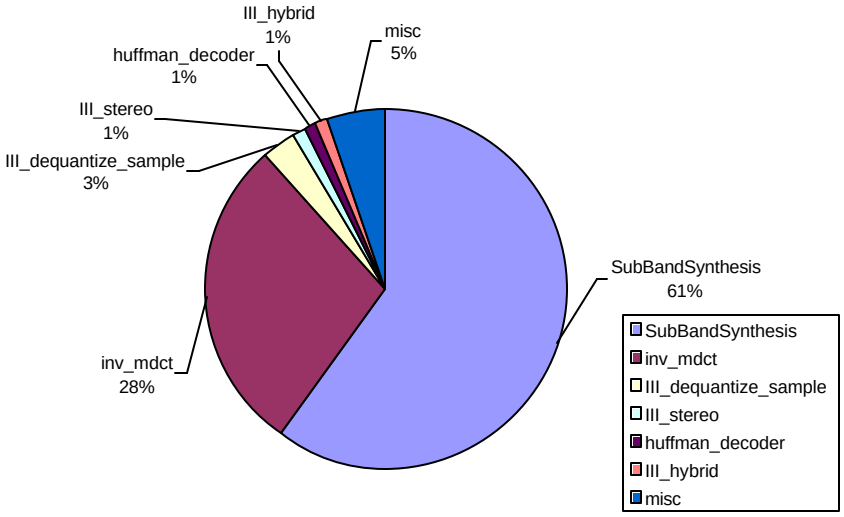


Fig. 6.19 Decoder Complexity Profile for the C implementation.

Optimization of the MATLAB encoder is an important issue, especially for running on slower hardware. Hot-spots in the code were identified from the profile in Fig. 6.15. Three critical functions in the bitstream formatter were optimized using the MATLAB Compiler to improve speed and still maintain a high readability of the code. The resulting code is 4.5 times faster than the unoptimized version. Fig. 4.20 shows the distribution of computational complexity for this version of the code.

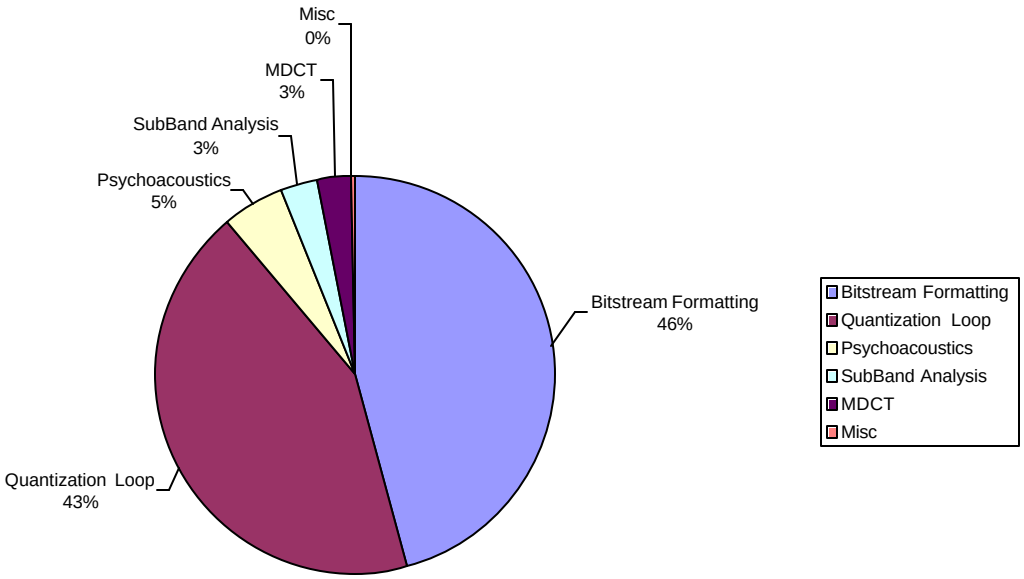


Fig. 6.20 Encoder Complexity Profile for the optimized MATLAB implementation.

CHAPTER 7

IMPROVING PERFORMANCE OF THE MP3 ALGORITHM AT LOW BITRATES

The MPEG-1 standard [30] embodies first generation coders that provide modest coding gains at the cost of fine-grain scalability of bit-rate, bandwidth and complexity. They were not designed for streaming applications. The MPEG-2 standard [37] addressed some of these issues. For audio coding, MPEG-2 LSF [38] extends stereo and mono coding of MPEG-1 standard to halved sampling rates (16, 22.05 and 24 kHz) for improved quality for bit-rates at or below 64 kb/s per channel. Nevertheless, it is to be pointed out that retrofitting a high fidelity audio coder for lower sampling frequencies and consequently lower bit-rates is not the best solution to the problem at hand. On the contrary, parametric and analysis-by-synthesis coders perform very well at very low bit-rates [82]. In the light of these issues, to improve quality at high compression ratios, this chapter concerns itself with the proposal of an algorithm for embedding a parametric model as an enhancement layer in the standard MP3 bitstream.

7.1 Motivation

All high-fidelity audio coders rely upon a model of human auditory masking for shaping quantization noise in the frequency domain. In the case of MP3, rate control is achieved by iteratively changing quantizers till the (quantization) noise is below the JND for all scale-factor bands. This process is non-linear and aims to maximize SMR at the expense of pruning higher frequencies. Fig. 7.1 shows the original and decoded MP3 bitstream at 64 kb/s. Fig. 7.2 shows the original and decoded MP3 bitstream for more

severe case, 48 kb/s. The decoded signals have a marked low-pass filtered effect.

On closer observation, it can be seen that the design of the lossless coding (Huffman) backend of the encoder is based on the fact that, statistically, higher frequency components have low energy. So, as the bit rate control loop iteratively increases quantizer step-sizes, part of the quantized high-frequency spectrum will contribute to the string of run-length zeros. The bits thus saved are distributed among the stronger signal components.

This study proposes a scheme for embedding information about the truncated spectrum to enhance the quality of the decoded bitstream, as outlined below.

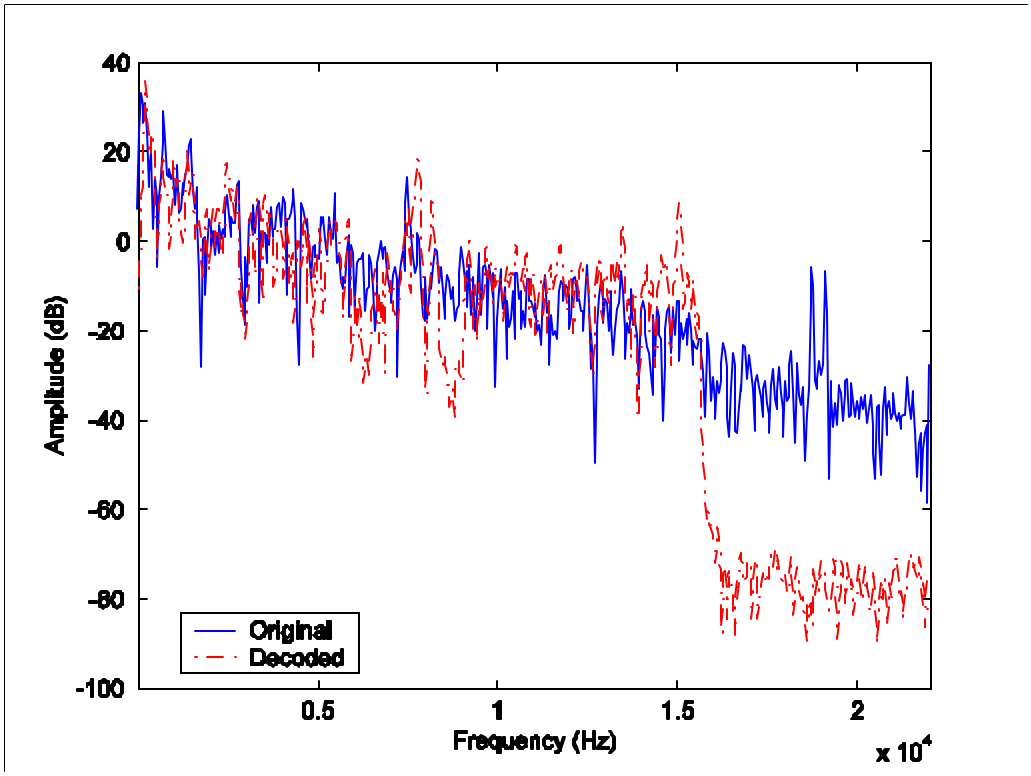


Fig. 7.1 High frequencies are sacrificed for compression at low bit-rates (64 kb/s).

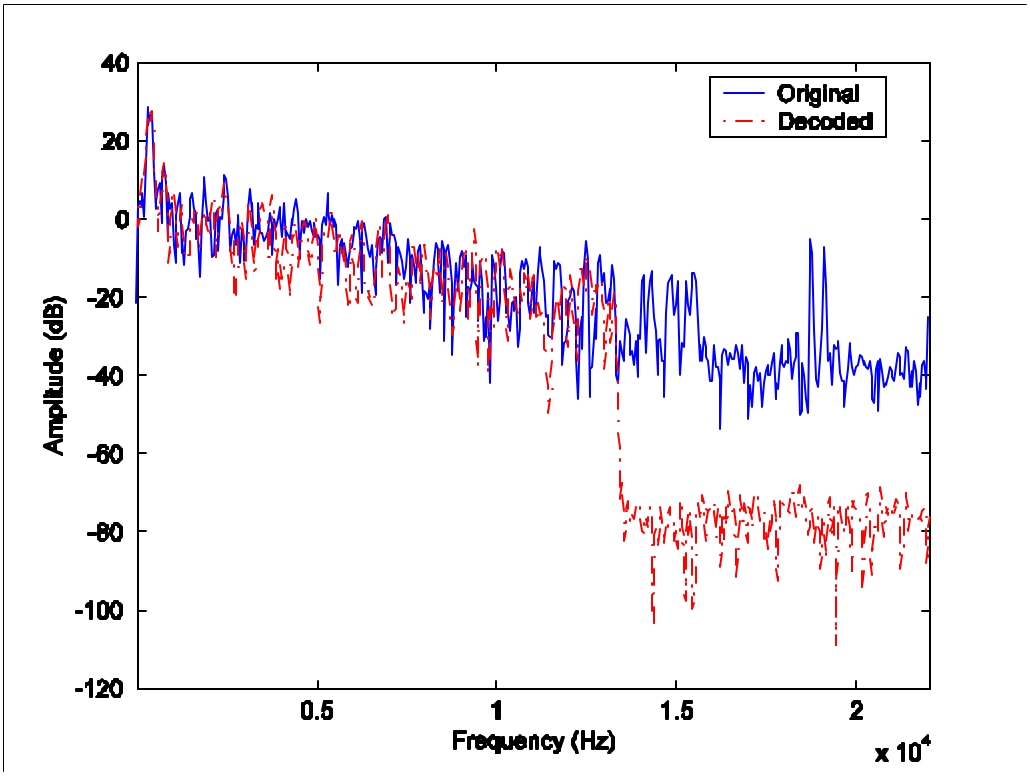


Fig. 7.2 The lowpass filtering effect is even more severs at 48 kb/s.

Header	CRC	Audio Data	Ancillary Data

Fig. 7.3 Logical representation of the MP3 bitstream.

Fig. 7.3 is a logical representation of the MP3 bitstream. Also, it can be seen from Fig. 5.1 that the inclusion of ancillary data is optional. The ancillary bitstream may be used to encode information not necessarily related to the audio data being compressed. On the contrary, the ancillary bitstream may in some way be related to the data being compressed; for example, an enhancement layer. For a particular target bit-rate, ancillary data may be included only be reducing the number of bits for encoding the audio data.

Given that the encoder is working at very high compression ratios and that the enhancement layer can only be supported at the cost of compromising the primary bitstream, it is of paramount importance to use a minimal number of bits in the ancillary data. Typically, at bit-rates below 64 kb/s, the quality is not expected to be transparent and it is desirable to have any noticeable improvement in audio quality with only a moderate increase in computational complexity. Parametric models can be engineered for low complexity and they can inherently be represented by a compact set of parameters.

Sampling Rate (kHz)	Bit-rate (kb/s)	Average PE (%) in the run-length zeros	Average number of scale-factor bands in the run-length zeros
44.1	64	42.32	2
	56	48.77	3
	48	56.18	4
	32	73.33	8
32	64	32.56	2
	56	39.41	3
	48	47.86	3
	32	69.42	6

Table 7-1 Average PE contained in the run-length zeros of the Huffman spectrum and the equivalent number of scale-factor bands.

As part of this study, an experiment was devised to calculate the average perceptual entropy in the run-length zeros of the Huffman spectrum for a combination of sampling rates and bit-rates. The corresponding number of scale-factor bands was also determined as an average. Table 7-1 lists the results. It is obvious from the table that at higher compression ratios, a significant part of the PE is contained in the truncated spectrum. Representing this part of the spectrum by a single coding gain for the quantizer undermines the quality of the reconstructed signal.

7.2 The Enhancement Algorithm

A block diagram of the proposed enhancement model that aims to model the truncated spectrum by a combination of sinusoids and noise is as shown in Fig. 7.4.

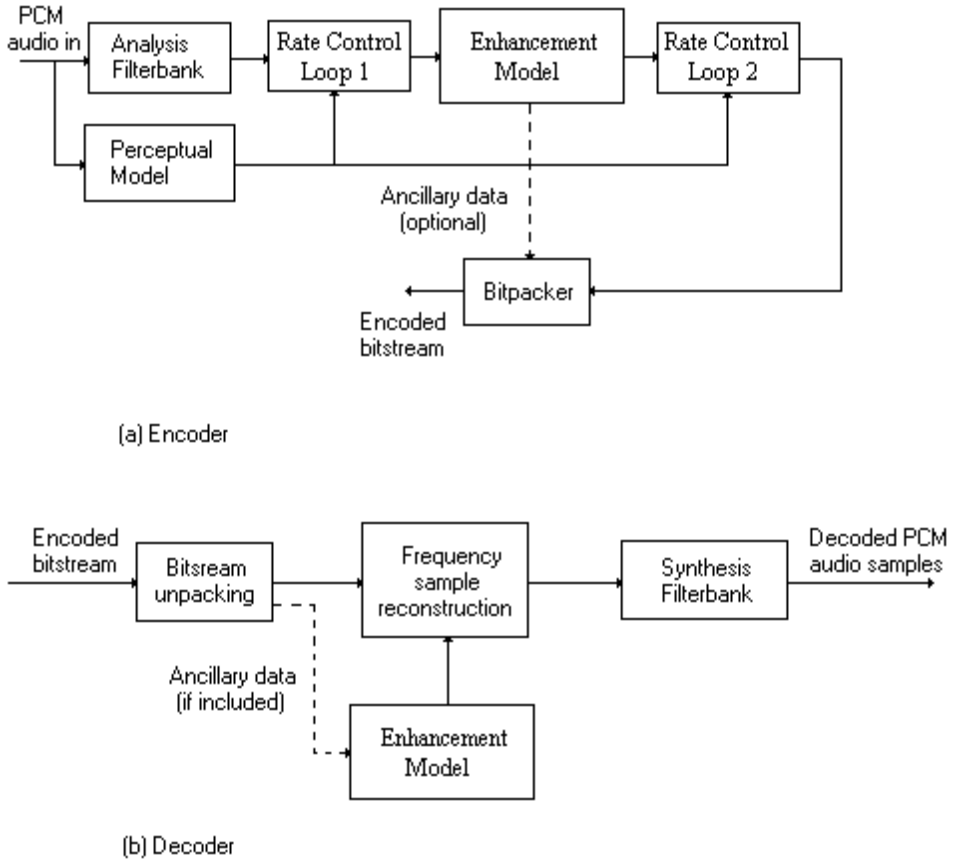


Fig. 7.4 The Enhanced MPEG/Audio codec. (a) Encoder. (b) Decoder.

The scheme is based on a two-pass rate-control loop for dynamically determining the number of bits required for the model. In the first pass, the MDCT components are quantized and the number of scale-factor bands in the truncated spectrum is counted. This is compared against the statistical average determined in the table above and the minimum of the two is chosen. In effect, the scheme preferentially models higher scale-

factor bands. Once the actual model parameters are determined, a small part of the audio-data bits are apportioned to the ancillary bitstream and the spectral components of the hybrid filterbank are finally quantized by a second pass of the rate-control loop. Since the MP3 algorithm switches to short blocks when the PE is high and utilizes the bits in the bit-reservoir (if necessary), the model is applied only for long blocks.

At the decoder, the model parameters are extracted and the parametric spectrum is appended to the standard MDCT spectrum. Given the way the model is computed, it never overlaps with the original spectrum; even in the best of cases, this results in a small gap in the reconstructed spectrum between the two regions. The advantage of this method is that there is no need for smoothing of any potential spectral overlap. The time-domain signal is reconstructed from the standard synthesis filterbank.

7.2.1 Details of the Sines and Noise Model

Sampling Rate (kHz)	sfb 15	sfb 16	sfb 17	sfb 18	sfb 19	sfb 20	sfb 21	sfb 22
48	3	5	6	6	6	6	6	8
44.1	5	5	6	6	6	6	7	8
32	5	6	6	6	7	7	7	5
24	5	6	6	6	6	7	7	6
22.05	5	6	6	6	6	7	7	5
16	5	6	4	7	6	7	6	5

Table 7-2 Bit-allocation for the differential encoding of the sinusoidal frequencies.

To keep the bit-rate requirements and complexity at a minimum, every scale-factor band in the model has a single sinusoid. The remaining energy in the scale-factor band is modeled as bark-band noise. From the results obtained in Table 8.1, only the last eight scale-factor bands are modeled. The sinusoidal and noise amplitudes are quantized by a logarithmic quantizer with a fixed bit-allocation of 4 bits per element. The sinusoidal frequencies are encoded as differences from the lower boundaries of the scale-factor bands. The scale-factor band boundaries are defined in Table B.8 of the standard. The bit-allocation for the differential encoding of the frequencies is depicted in Table 8.2.

7.3 MP3PRO

A recent commercial product to improve the MP3 codec at low bit-rates is the MP3PRO [80] [81]. It has also been designed to mitigate the lowpass filter effect. The technique used is called Spectral Band Replication (SBR). SBR, as the name suggests, does not really encode the high-frequency spectrum; it reconstructs it from the lower frequencies. The SBR encoder stores information about the part of the original band-limited signal from which the upper frequencies should be replicated in the ancillary data of the standard MP3 file. At 64 kb/s, around 4 kb/s are used for the SBR model. The spectrum up to 8 kHz is encoded in a conventional way. This part of the resulting MP3 file can be decoded by any MP3 decoder, so compatibility is kept with conventional decoders. The SBR technique reconstructs the missing high frequency part, from 8 kHz up to 16 kHz, by duplicating the spectrum. The quality of the MP3PRO codec comes at the price of 300% increase in computational complexity of the decoder, not to mention additional licensing costs in a price sensitive market.

7.4 Results

Since the enhancement model parameters are transmitted as ancillary data, the bitstream is compatible with legacy MP3 decoders. To get the benefits of the parametric model, the decoder has to be re-engineered to decode the model, replace the truncated MDCT spectral components with those derived from the model and then synthesize the time-domain waveform.

7.4.1 Subjective Evaluation

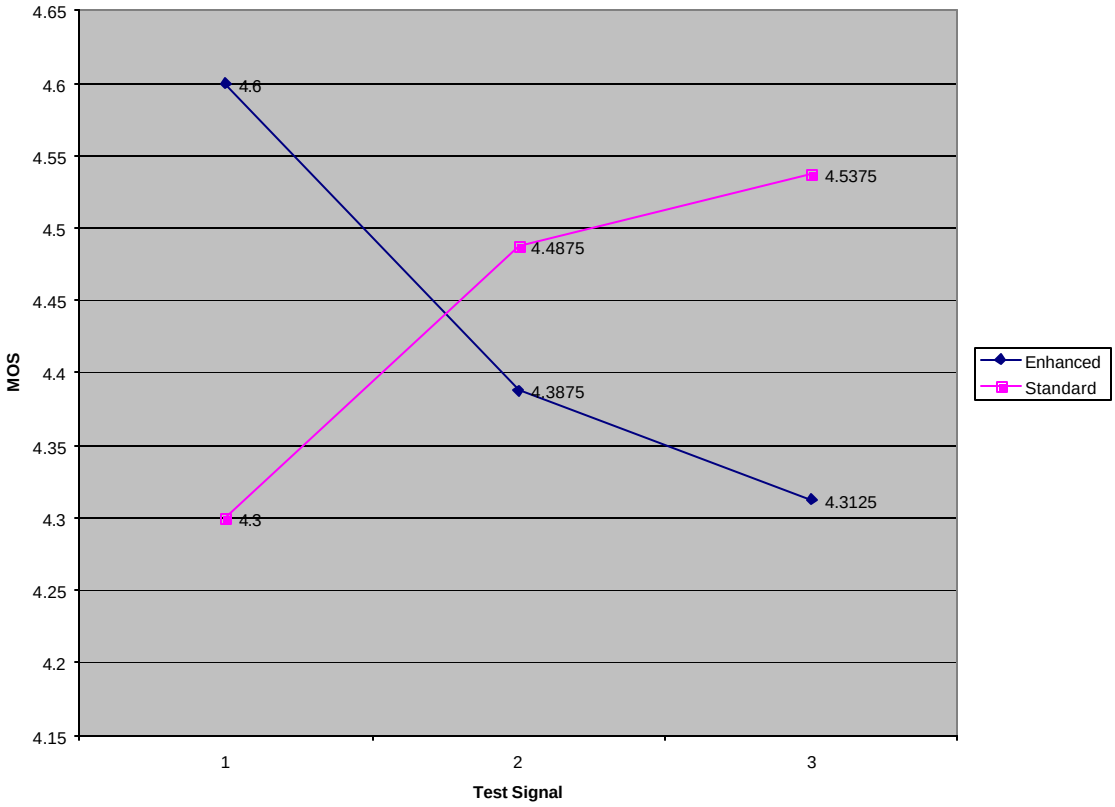


Fig. 7.5 MOS results for the original and decoded signals at 64 kb/s.

Informal subjective evaluation of the enhanced codec was conducted at 64 kb/s and 48 kb/s. There were three test signals and eight listeners. The Mean Opinion Scores

(MOS) for both the tests are indicated in Fig. 7.5 and Fig. 7.6.

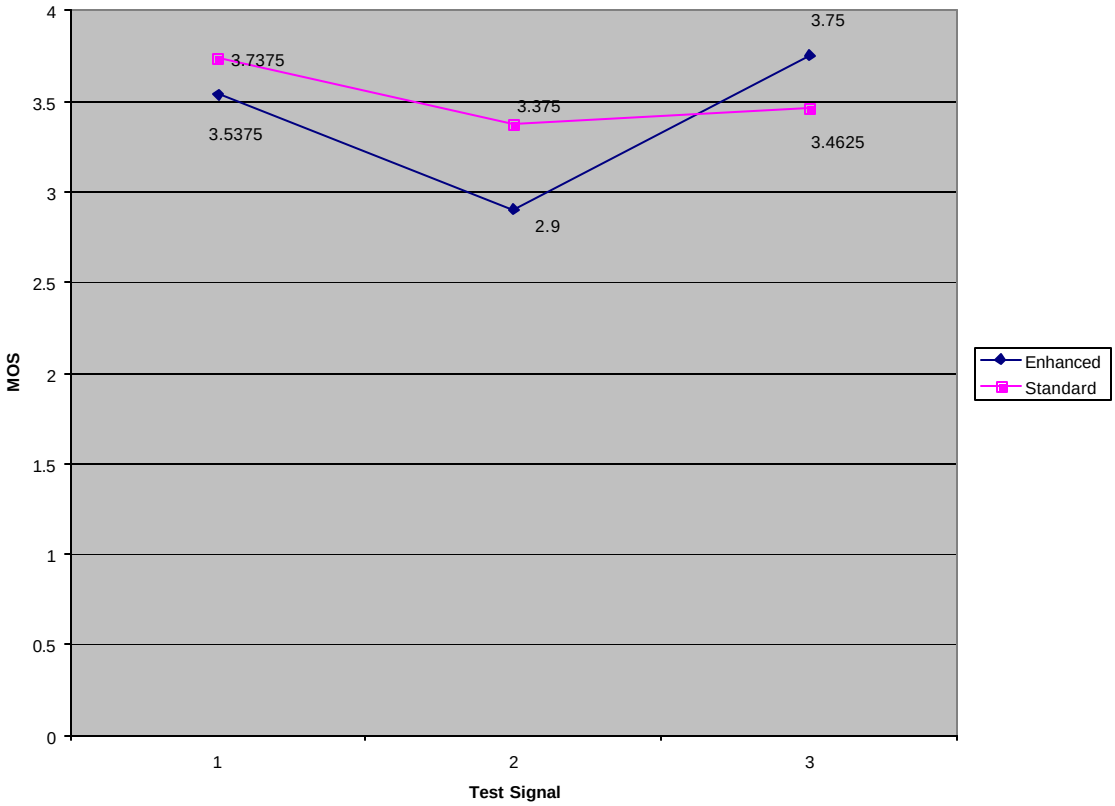


Fig. 7.6 MOS results for the original and decoded signals at 48 kb/s.

7.4.2 Objective Evaluation

Fig. 7.7 and Fig. 7.8 depict the objective results obtained from the standard and enhanced decoders. It is obvious that diverting a small percentage of the bits to the parametric model dramatically improves the spectral matching at high frequencies. Informal listening tests also indicated that the enhanced model contributes to an improvement in perceptual quality.

The designed enhancement model was also compared against the MP3PRO codec. Informal investigation of the MP3PRO codec revealed that it does quite a good job of tracking the spectral envelope, as shown in Fig. 7.9. At times, the reconstructed

signal suffers what appears to be excess harmonic excitation that results in a “tinny” output. It was also observed that the MPPRO codec does not capture sinusoidal components at high frequencies, confirming that it does not explicitly model the high frequency spectrum.

The proposed algorithm is of very low complexity when compared to the MP3PRO; a direct comparison would be unfair. The output of the designed model is closer to the standard MP3 algorithm than to the MP3PRO, with a slight improvement in quality.

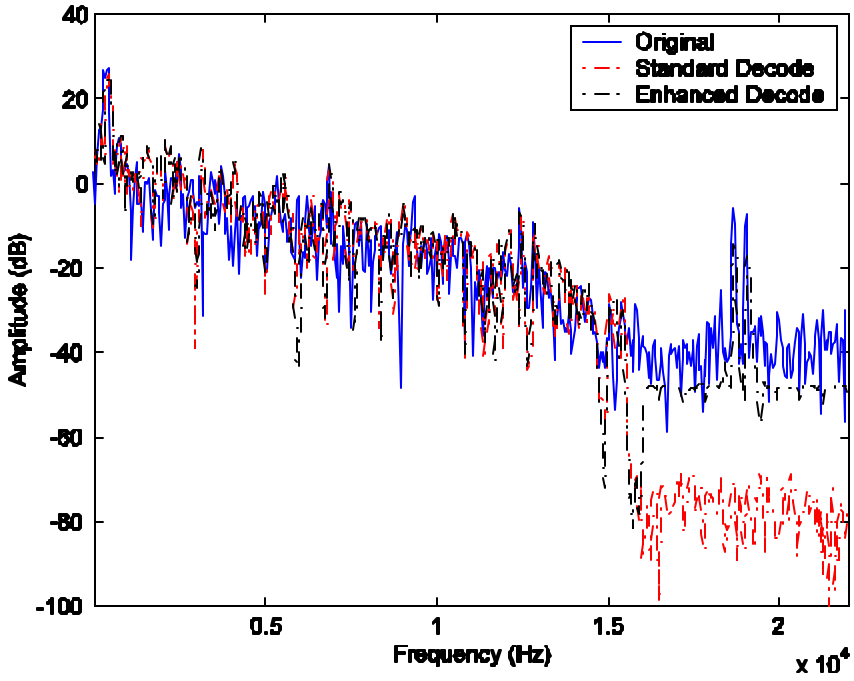


Fig. 7.7 Original and decoded signals at 64 kb/s.

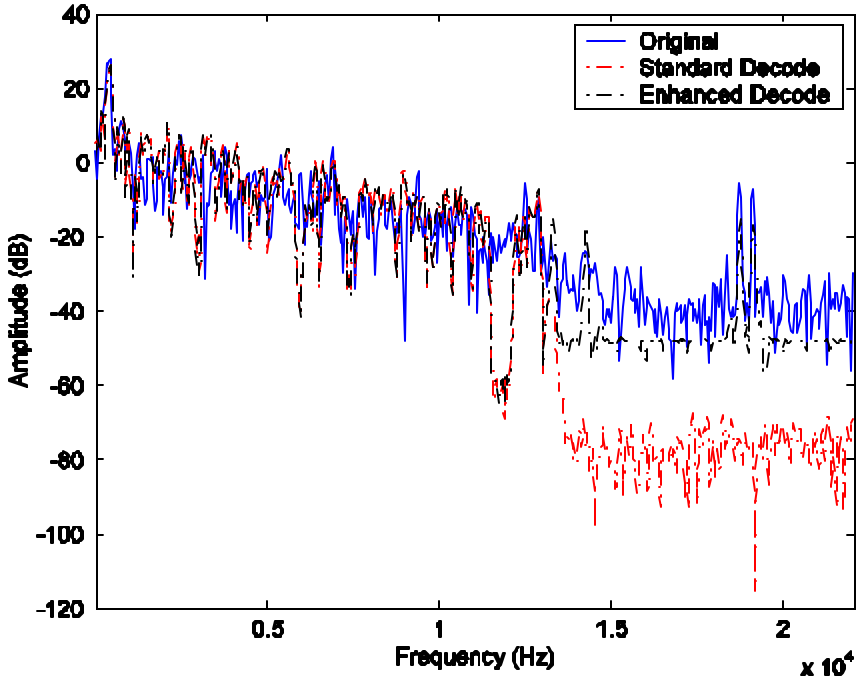


Fig. 7.8 Original and decoded signals at 48 kb/s.

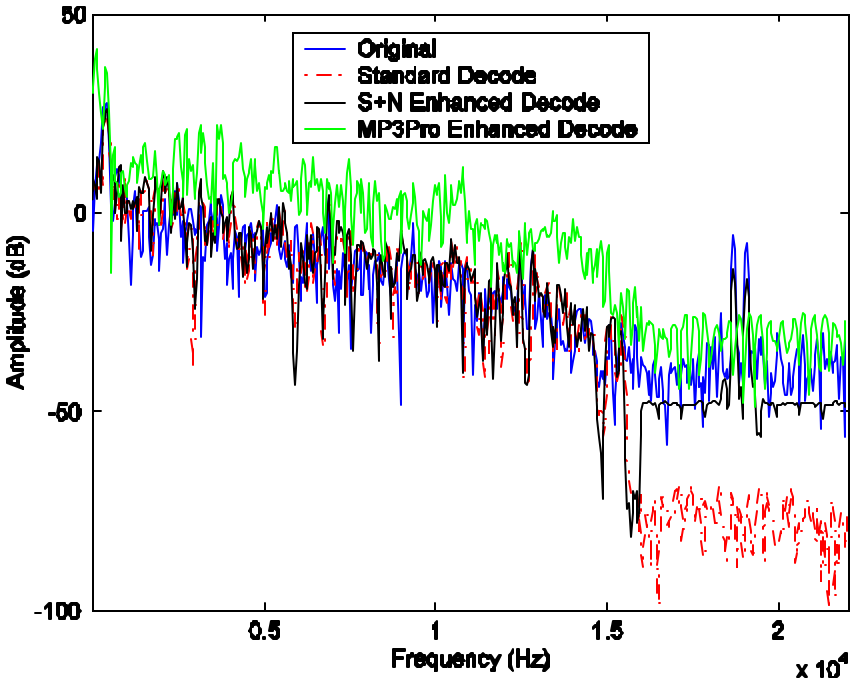


Fig. 7.9 Comparison of the designed model with MP3PRO at 64 kb/s.

CHAPTER 8

CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

The MPEG-1 Layer III (MP3) algorithm is almost a decade old and has proliferated into desktop computers and embedded devices alike. Even though newer algorithms based on the MPEG-2 and MPEG-4 standards perform better, they are definitely more complex; the MP3 algorithm is a better choice for low complexity implementations. Besides, the normative description of the MPEG-1 encoder provides ample scope for improving its performance.

8.1 Summary of Contributions

8.1.1 ASU MATLAB MP3Tool

As part of this research, a software simulation tool was developed for introducing perceptual audio coding concepts in senior undergraduate and graduate DSP courses. The tool consists of a user-friendly graphical interface along with a MATLAB realization of the audio MPEG-1 Layer III (MP3) algorithm. The tool is accompanied by a series of computer experiments and exercises that can be used to provide hands-on training to class participants. The tool may also be used by instructors in a class setting to demonstrate key signal processing concepts associated with the processing of high-fidelity audio. The MATLAB MP3 tool has been used in Arizona State University undergraduate DSP courses as well as in a graduate course on speech and audio coding and in a continuing education short course. The experiments designed are listed in Appendix-A.

8.1.2 Development of an algorithm to improve performance at low bit-rates

For the MP3 algorithm, the best compromise between bit-rate and quality is obtained at around 128 kb/s. Bitstream syntax issues aside, for streaming applications, typical bit-rates are at or below 64 kb/s. At these bit-rates, the algorithm is not transparent; it is a well known fact that the MP3 algorithm has a marked low-pass effect at lower bit-rates. In such cases, any incremental improvement in quality is definitely desirable. This thesis also concerned itself with the development of a parametric signal model to improve the performance of the MP3 algorithm at very low rates. The designed model has a very low complexity and demonstrates the viability of the solution.

8.2 Directions for future research

Feedback obtained from a wider user-base can help improve the quality of the MP3Tool by incorporating small enhancements. The tutorial can be further expanded to include a more detailed description and a larger set of exercises. Tracking and fixing bugs is an essential part of code maintenance and support.

It is imperative to test the algorithm over a wider range of audio material and listeners. Since the enhancement model developed here is applied only for long blocks, the performance is highly dependent on the signal characteristics. It would be interesting to apply the model to all types of MDCT blocks and compare the results. Also, introducing memory in the model can help track the signal better.

A more advanced algorithm can be designed to model the low-end spectrum by transform coding (MDCT) and the rest by a parametric model. The point of conjunction of the two can be adaptively determined based on a perceptual metric and the bit-rate.

REFERENCES

[1] B. Scharf, "Critical Bands", *Foundations of Modern Auditory Theory*, New York, Academic Press, 1970.

[2] B. C. J. Moore, *Introduction to the Psychology of Hearing*, University Park Press, 1977.

[3] J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression", *IEEE Trans. On Information Th.*, v. IT-23, n. 3, pp. 337-343, May 1977.

[4] Zelinski, R. and P. Noll. "Adaptive Transform Coding of Speech Signals." *IEEE Trans. on Acoust., Speech, and Signal Processing ASSP-25*, No. 4 , pp. 299 – 309, August 1977:

[5] J. M. Tribolet, R. E. Crochiere, "An Analysis/Synthesis Framework for Transform Coding of Speech", *IEEE Proc. on ASSP*, 1979

[6] M. Schroeder, *et. al.*, "Optimizing Digital Speech Coders by Exploiting Masking Properties of the Human Ear", *J. Acoust. Soc. Am.*, pp. 1647-1652, Dec. 1979.

[7] R. E. Crochiere and l. R. Rabiner, *Multirate Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ., 1983.

[8] J. H. Rothweiler, "Polyphase quadrature filters: A new subband coding technique", *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP-83)*, May 1983, pp. 1280-1283.

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

[10] R. McAulay and T. Quatieri, "Speech Analysis Synthesis Based on a Sinusoidal Representation", *IEEE Trans. ASSP*, pp. 744-754, Aug. 1986.

[11] J. Princen and A. Bradley, "Analysis/synthesis filterbank design based on time domain aliasing cancellation", *IEEE Trans. Acoustic., Speech and Signal Processing*, Vol. ASSP-34, pp. 1153-1161, Oct. 1986.

[12] International Electrotechnical Commission/American National Standards Institute (IEC/ANSI) CEI-IEC 908, "Compact Disc Digital Audio System", ("red book"), 1987.

[13] K. Brandenburg, "OCF – A new coding algorithm for high quality sound signals", in *Proc. ICASSP-87*, May 1987, pp. 5.1.1-5.1.4.

[14] I. Witten, "Arithmetic Coding for Data Compression", *Comm. ACM*, v. 30, n. 6, pp. 520-540, Jun. 1987.

[15] P. P. Vaidyanathan, "Quadrature Mirror Filter Banks, M-Band Extensions and Perfect Reconstruction Techniques", *IEEE ASSP Mag.*, pp. 4-20, Jul. 1987.

[16] J. Johnston, "Transform coding of audio signals using perceptual noise criteria", *IEEE J. Select. Areas. Commn.*, Vol. 6, pp. 314-323, Feb 1988.

[17] J. Johnston, "Estimation of perceptual entropy using noise masking criteria", *Proc. ICASSP-88*, May 1988, pp. 2524-2527.

[18] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ., 1989.

[19] J. Johnston, "Perceptual transform coding of wideband stereo signals", *Proc. ICASSP-89*, May 1989, pp. 1993-96.

[20] T. Welch, "A Technique for High Performance Data Compression", *IEEE Comp.*, v. 17, n. 6, pp. 1993-1996, May 1989.

[21] E. Zwiker and H. Fastl, *Psychoacoustics Facts and Models*, Springer-Verlag, 1990.

[22] K. Rao and P. Yip, *The Discrete Cosine Transform: Algorithm, Advantages and Applications*, Academic Press, 1990.

[23] P. P. Vaidyanathan, "Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial", *Proc. IEEE*, v. 78, no. 1, pp. 56-93, Jan 1990.

[24] K. Brandenburg and J. D. Johnston, "Second generation perceptual audio coding: The hybrid coder", *Proc. 88th Conv. Aud. Eng. Soc.*, Mar. 1990, preprint 2937.

[25] H.S. Malvar, *Signal Processing with Lapped Transforms*, Norwood, MA: Artech House, 1991.

[26] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley and Sone, Inc, New York, 1991.

[27] K. Brandenburg, J. Herre, J.D. Johnston, Y. Mahieux, and E. Schroeder, "ASPEC: Adaptive spectral entropy coding of high quality music signals", Proc. 90th Conv. Aud. Eng. Soc., Feb. 1991, preprint 3011.

[28] E. Zwiker and U. Zwiker, "Audio Engineering and Psychoacoustics: Matching Signals to the Final Receiver, the Human Auditory System", *J. Audio Eng. Soc.*, pp. 115-126, Mar. 1991.

[29] Y.F. Dehery, M. Lever and P. Urcun, "A MUSICAM source codec for digital audio broadcasting and storage", in *Proc. ICASSP-91*, May 91, pp. 3605-3608.

[30] "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)", ISO/IEC, JTC1/SC29, 1992.

[31] B. Paillard, *et. al*, "PERCEVAL: Perceptual Evaluation of the Quality of Audio Signals", *J. Aud. Eng. Soc.*, v. 40, n. ½, pp. 21-31, Jan/Feb, 1992.

[32] M. Vetterli and C. Herely, "Wavelets and filter banks", *IEEE Trans. Signal Proc.*, Vol. 40, pp. 2207-2232, Sept. 1992.

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

[34] G. Stoll, *et al*., Extension of ISO/MPEG-Audio Layer II to Multi-Channel Coding: The Future Standard for Broadcasting, Telecommunication, and Multimedia Applications", *Proc.94th Conv. Aud. Eng. Soc.*, preprint #3550, Mar. 1993.

[35] D. Sinha and A. Tewfik, "Low Bit Rate Transparent Audio Compression Using Adapted Wavelets", *IEEE Trans. Sig. Proc.*, pp. 3463-3479, Dec. 1993.

[36] N. Jayant, J. D. Johnston and R. Safranek, "Signal Compression based on models of human perception", *Proc. IEEE*, Vol. 81, pp. 1385-1422, Oct. 1993.

[37] ISO/IEC, JTC1/SC29/WG11 MPEG, "Information Technology - Generic coding of moving pictures and associated audio – Part 3: Audio", IS 13818-3, 1994 ("MPEG-2").

[38] ISO/IEC, JTC1/SC29/WG11 MPEG, "Information Technology - Generic coding of moving pictures and associated audio – Part 3: Audio", IS 13818-3, 1994 ("MPEG-2 BC-LSF").

[39] K. Konstantinides, "Fast Subband Filtering in MPEG Audio Coding", *IEEE Sig. Proc. Letters*, v. 1, pp. 26-28, Feb. 1994.

[40] K. Brandenburg and G. Stoll, "ISO-MPEG-1 audio: A generic standard for coding of high-quality digital audio", *J. Audio Eng. Soc.*, pp. 780-792, Oct. 1994.

[41] Seymour Shlien, "Guide to the MPEG-1 audio standard", *IEEE Transactions on Broadcasting*, Vol. 40, No. 4, December 1994.

[42] http://www.midi-world.net/English/index.htm

[43] Naoki Iwakami, Takehiro Moriya and Satoshi Miki, "High-quality audio coding at less than 64 kbits/s using Transform-Domain Weighted Interleave Vector Quantization (TWIN-VQ)", *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP-95)*, Vol. 5, pp. 3095-3098, 1995.

[44] D. Pan, "A tutorial on MPEG/audio compression", *IEEE MultiMedia*, pp. 60-74, Summer 1995.

[45] S. Cheung and J. Lim, "Incorporation of Biorthogonality into Lapped Transforms for Audio Compression", *Proc., Int. Conf. Acous., Speech and Sig. Proc.*(ICASSP-95), pp. 3079-3082, May 1995.

[46] F. Baumgarte, C. Feredkidis, and H. Fuchs, " A nonlinear psychoacoustic model applied to the ISO MPEG layer 3 coder", *Proc. 99th Conv. Aud. Eng. Soc.*, New York, Oct. 1995, preprint 4087.

[47] F. Baumgarte, C. Ferekidis and H. Fuchs, "A Nonlinear Psychoacoustic Model Applied to the ISO MPEG Layer 3 Coder", *Proc. 99th Conv. Aud. Eng. Soc.*, preprint #4087, New York, Oct. 1995.

[48] http://sound.media.mit.edu/~mkc/icmc96/icmc96.html

[49] ISO/IEC JTC1/SC29/WG11 MPEG, Committee Draft 13818-7 "Generic Coding of Moving Pictures and Associated Audio: Audio (non backwards compatible coding, NBC)", 1996 ("MPEG-2 NBC/AAC).

[50] J. Herre and J. Johnston, "Enhancing the performance of perceptual audio coders by using temporal noise shaping (TNS)", *Proc. 101st Conv. Aud. Eng. Soc.*, 1996, preprint 4384.

[51] J. Johnston, *et. al.*, "AT&T Perceptual Audio Coding (PAC)", *Collected Papers on Digital Audio Bit-Rate Reduction*, N. Gilchrist and C. Grewin, Eds., Aud. Eng. Soc., pp. 73-81, 1996.

[52] J. Princen and J. Johnston, "Audio Coding with Signal Adaptive Filterbanks", *Proc. Int. Conf. Acous. Speech and Sig. Proc.* (ICASSP-95), pp. 287-307, 1996.

[53] L. Fielder, *et. al.*, "AC-2 and AC-3: Low-Complexity Transform-Based Audio Coding", *Collected Papers on Digital Audio Bit-Rate Reduction*, N. Gilchrist and C. Grewin, Eds., *Aud. Eng. Soc.*, pp.54-72, 1996.

[54] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, and M. Dietz, "MPEG-2 advanced audio coding", *Proc. 101st Conv. Aud. Eng. Soc.*, 1996, preprint.

[55] Takehiro Moriya, Naoki Iwakami, Kazunaga Ikeda and Satoshi Miki, "Extension and Complexity Reduction of the TWINVQ audio coder", *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP-96)*, Vol. 2, pp. 1029-1032, 1996.

[56] Hossein Najafzadeh-Azghandi and Peter Kabal, "Perceptual coding of narrowband audio signals at 8 kbit/s", *Proc. IEEE Workshop on Speech Coding for Telecom.*, pp. 109-110, 1997.

[57] Masayuki Nishiguchi, Kazuyuki Iijima, and Jun Matsumoto, "Harmonic vector excitation coding of speech at 2.0 – 4.0 Kbps", *Dig. of Papers, Intl. Conf. on Consumer Elec.*, pp. 208-209, 1998.

[58] Seymour Shlien, "The Modulated lapped transform, its time-varying forms, and its applications to audio coding standards", *IEEE Transactions on Speech and Audio Processing*, Vol. 5, No. 4, July 1997.

[59] *Applications of Digital Signal Processing to Audio and Acoustics*, M. Kahrs and K. Brandenburg, Eds. Boston, MA: Kluwer Academic, 1998.

[60] D. Sinha, *et. al.*, "The Perceptual Audio Coder (PAC)", *The Digital Signal Processing Handbook*, V. Madisetti and D. Williams, Eds., CRC Press, pp. 42.1-42.18, 1998.

[61] K. Brandenburg, "Perceptual coding of high quality digital audio", *Applications of Digital Signal Processing to Audio and Acoustics*, M. Kahrs and K. Brandenburg, Eds. Boston, MA: Kluwer Academic, 1998.

[62] K. Tsutsui, "ATRAC (Adaptive Transform Acoustic Coding) and ATRAC 2", *The Digital Signal Processing Handbook*, V. K. Madisetti, and D. Williams, Eds., CRC Press, Boca Raton, FL, 1998, pp 43.16- 43.20.

[63] *The Digital Signal Processing Handbook*, V. K. Madisetti, and D. Williams, Eds., CRC Press, Boca Raton, FL, 1998, pp 38.1- 44.8.

[64] Barry Vercoe, William G. Gardner, and Eric D. Scheirer, "Structured Audio: Creation, Transmission, and Rendering of Parametric Sound Representations", *Proceedings of the IEEE*, Vol. 86, No. 5, May 1998.

[65] Eric D. Scheirer, "The MPEG-4 Structured Audio Standard", *Proc. Int. Conf. Acous. Speech, and Sig. Proc.*(ICASSP-98), May1998.

[66] H. Purnhagen, *et. al.*, "Object-Based Analysis/Synthesis Audio Coder for Very Low Bit Rates", *Proc. 104th Conv. Aud. Eng. Soc.*, preprint #4747, May, 1998.

[67] Yuan-Pei Lin and P.P. Vaidyanathan, "A Kaiser window approach for the design of prototype filters of cosine modulated filterbanks", *IEEE Signal Processing Letters*, Vol. 5, No. 6, June 1998.

[68] G. Schuller, "Time-Varying Filter Banks with Low delay for Audio Coding", *Proc. 105th Conv. Aud. Eng. Soc.*, preprint #4809, Sep. 1998.

[69] S. Levine, *Audio Representations for Data Compression and Compressed Domain Processing*, Ph. D. Thesis, Stanford University, Dec. 1998.

[70] Hossein Najafzadeh-Azghandi and Peter Kabal, "Improving perceptual coding of narrowband audio signals at low rates", *Proc. Int. Conf. Acoustics, Speech, and Signal Processing.*, Vol. 2, pp. 913-916, 1999.

[71] Masayuki Nishiguchi, Akira Inoue, Yuji Maeda and Jun Matsumoto, "Parametric speech coding – HVXC at 2.0 kbps", *Proc. IEEE Workshop on Speech Coding*, pp. 84-86, 1999.

[72] T. Verma, *A Perceptually Based Audio Signal Model with Applications to Scalable Audio Compression*, Ph. D. Dissertation, Stanford University, 1999.

[73] "Eric D. Scheirer, Riitta Vaananen, and Jyri Huopaniemi, Audio BIFS: describing audio scenes with the MPEG-4 multimedia standard", *IEEE Transactions on Multimedia*, Vol. 1, No. 3, Sept. 1999.

[74] Bernd Edler and Heiko Purnhagen, "Parametric audio coding", *Proceeding of ICSP 2000*, 2000.

[75] Jurgen Herre and Bernhard Grill, "Overview of MPEG-4 audio and its applications in mobile communications", *Proceeding of ICSP 2000*, 2000.

[76] *The CSound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing and Programming,* Richard Boulanger , Ed., MIT Press, Mar. 2000.

[77] Hossein Najafzadeh-Azghandi, *Perceptual Coding of Narrowband Audio Signals,* Ph. D. Thesis, Mc Gill University, Montreal, Canada, Apr. 2000.

[78] T. Painter and A. Spanias, "Perceptually Coding of Digital Audio", *Proc. IEEE*, v. 88, pp. 451-513, Apr. 2000.

[79] Heiko Purnhagen and Nikolaus Meine, "HILN – the MPEG-4 audio coding tools", ISCAS 2000, *IEEE International Symposium on Circuits and Systems*, May 28-31, 2000.

[80] http://www.codingtechnologies.com/technology/mp3pro.htm

[81] http://www.mp3-tech.org/sbr.html

[82] ISO/IEC JTC1 SC29/WG11, ISO/IEC FDIS 14496-3 Subparts 1, 2, 3, "Coding of Audio-Visual Objects|Part 3: Audio" , ISO/IEC JTC1 SC29/WG11 N2503, Oct. 1998.

[83] Karlheinz Brandenburg, Oliver Kunz, Akihiko Sugiyama "MPEG-4 Natural Audio Coding", http://leonardo.telecomitalialab.com/icjfiles/mpeg-4_si/9-natural_audio_paper/index.html

[84] H. Purnhagen, "An Overview of MPEG-4 Audio Ver. 2 ", invited paper, *AES 17th International Conference on High-Quality Audio Coding*, Florence, Italy, September 1999

APPENDIX

A: TUTORIAL EXERCISES

A.1 Psychoacoustics-based compression is lossy

Exercise: This exercise aims to look at the waveform matching properties of the MPEG-1 Layer III algorithm. This is achieved by comparing the time and frequency domain representations of the original and decoded signal at various bit-rates: 320, 256, 192, 128, 64 and 32 kb/s. Ex1.wav is the test signal used.

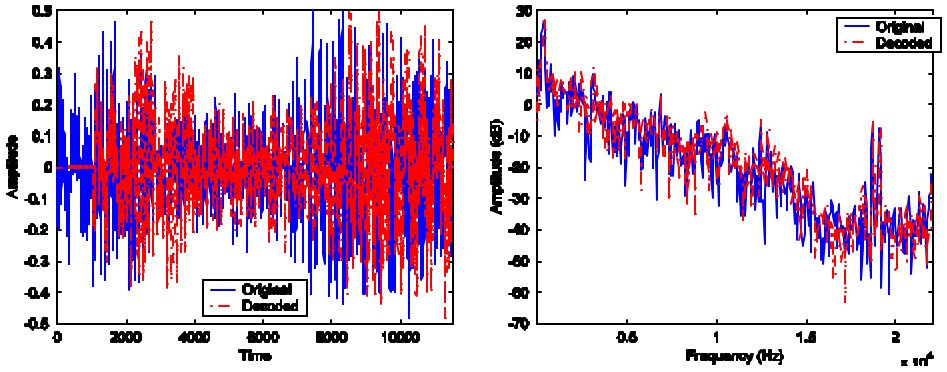
Procedure: Invoke the MP3 encoder by running the MP3Encoder.m script. When asked for visualization, press the 'no' button; this disables the GUI. In the encoder configuration menu, choose the source file to be Ex1.wav. If you do not choose a target file-name, the name will be similar to the source but with a .mp3 extension. In the advanced options, choose the target bit-rate from the drop-down list-box. For this experiment, leave the other options at their default values.

Encode[3] the signal under consideration at each of the target bit-rates. The encoded files are decompressed and the time domain waveforms are superposed. Similarly, for a particular segment of data, the spectra are also superposed.

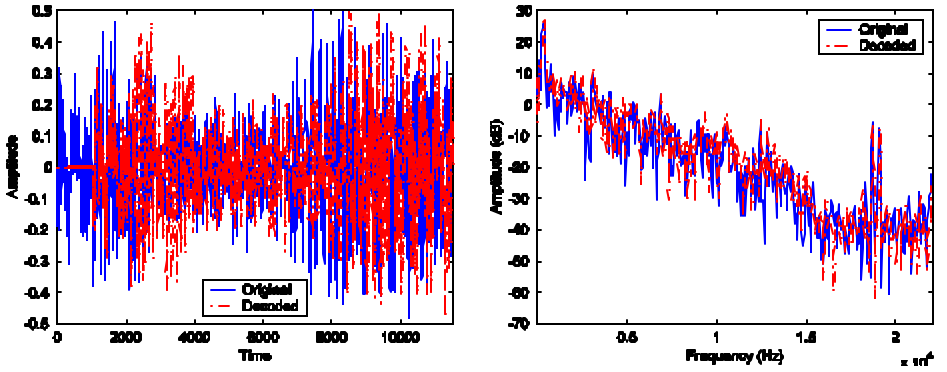
Observation: Fig. A.1 represents the results of the experiment. Compressing and reconstructing a frame of data at various bit-rates reveals that psychoacoustics-based compression is *lossy*. It does not aim to faithfully capture the time-domain signature of the audio signal. It tries to capture the *perceptually* relevant characteristics of the signal,

[3] This MATLAB software has neither been designed for speed nor performance but for strictly educational purposes. In general, do not use it for encoding large files. See Ch. 6.2 for a profile of the code.

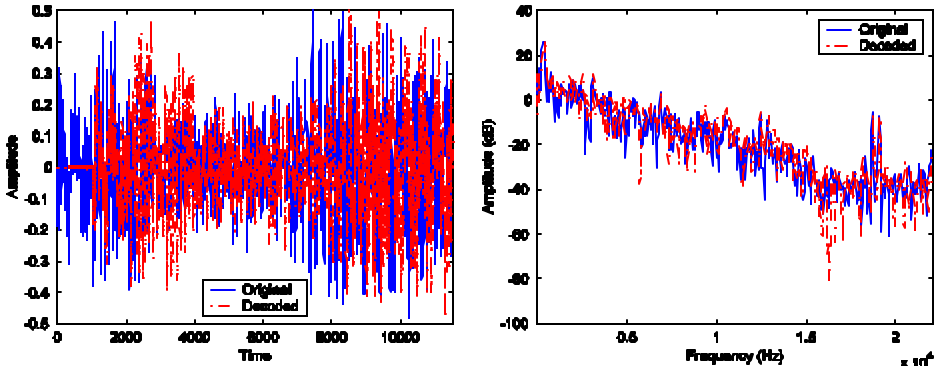
as seen from the spectral plots. At high compression ratios, part of the high frequency spectrum is compromised, resulting in a degradation of signal quality.



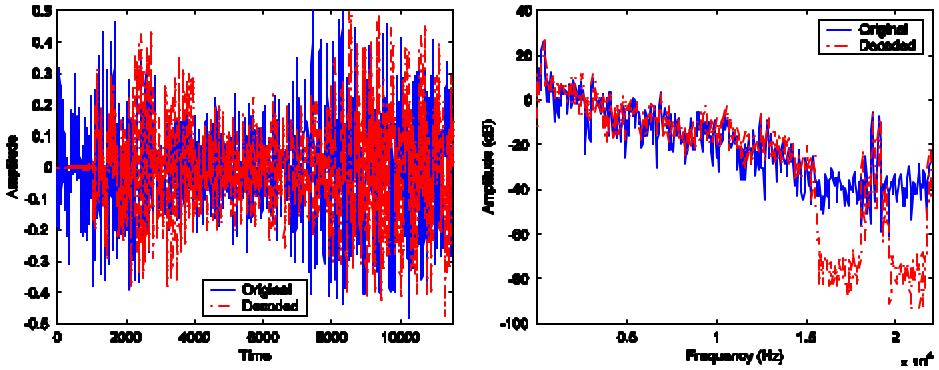
(a) 320 Kb/s



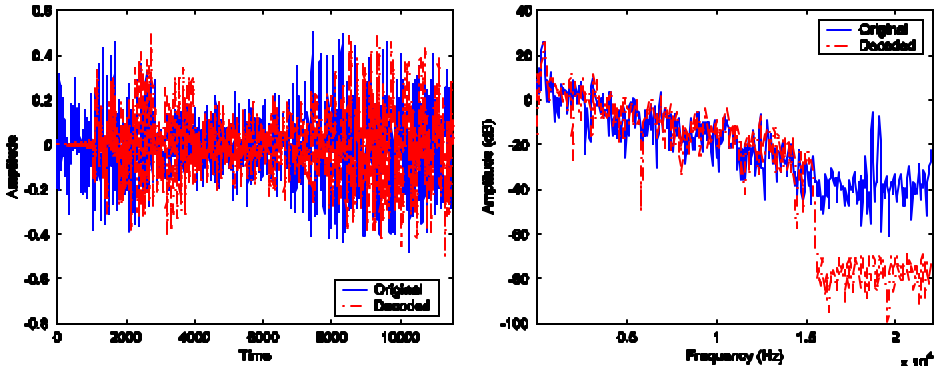
(b) 256 Kb/s



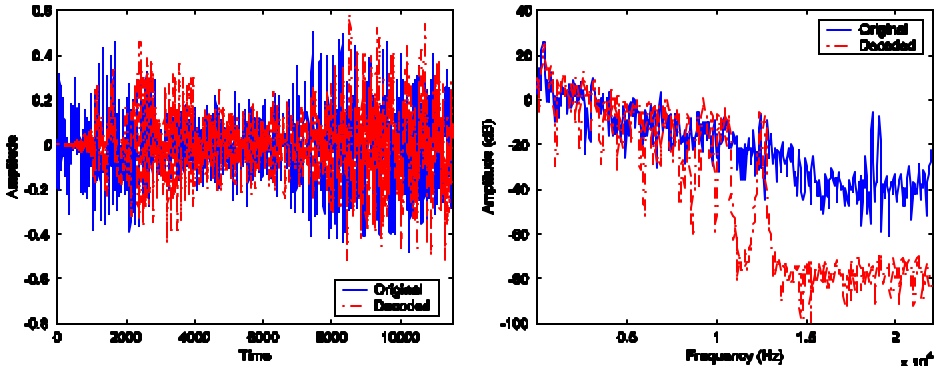
(c) 192 Kb/s



(d) 128 Kb/s



(e) 64 Kb/s



(f) 32 Kb/s

Fig. A.1 Time and Frequency-domain representations of the signal at various bit-rates.

A.2 The Analysis Filterbank

In the context of subband coding, the primary function of an analysis filterbank is to localize energy in the subbands and provide a first step gain. That is, the quantizers in each subband can be optimized independently to achieve compression.

The analysis filterbank used in the MP3 algorithm splits the incoming signal into 32 equal subbands. This *critically sampled* filterbank divides the block of audio into 32 bands, each of a nominal bandwidth $p/(32T)$, where T is the sampling interval. The 512 coefficients of the lowpass prototype filter are plotted in Fig. 5.2.

The corresponding impulse response plotted in Fig. 5.3, attenuates the side-lobes by more than 96 dB. This lowpass filter is cosine modulated to obtain a bank of filters with center frequencies at odd multiples of $p/(64T)$, depicted in Fig. 5.4. At a sampling rate of 44.1 kHz, the nominal BW of the prototype filter is given by

Nominal BW = PI/(number of subbands) = Nyquist/32

= (sampling rate/2)/32

= 22050/32 = 689.0625 Hz

Exercise: Create a signal comprising of 4 sinusoids centered at the subbands[4] 0, 8, 15, 26.Use a sampling rate of 32 kHz. For display reasons, it is recommended to add white noise at a low energy. Save it as a Windows .wav file. The number of samples should be at least 10 frames long. In the rest of the manual, unless stated explicitly, all data to be generated and used are mono. Similarly, encode at least 3 frames before

[4] The convention used is that 0 is the first and 31 the last subband.

observing outputs.

Procedure: Invoke the MP3 encoder by enabling the GUI. Select the generated synthetic signal as the input file. Observe the outputs of the filterbank and the corresponding spectrum in the psychoacoustics block.

Observation: At a sampling rate of 32 kHz, the nominal BW of the prototype filter is 500 Hz. The sinusoidal frequencies (250, 4250, 7750 and 13250 Hz) are centered at the middle of the corresponding subbands, we expect the gain to be high in those subbands. The spectral estimate in the psychoacoustics block also identifies these frequencies, but at a higher resolution. In essence, we can consider the filterbank as a low-resolution spectral estimate; or alternately, the spectrum as a high-resolution filterbank. Fig. A.2 and Fig. A.3 clarify this conclusion.

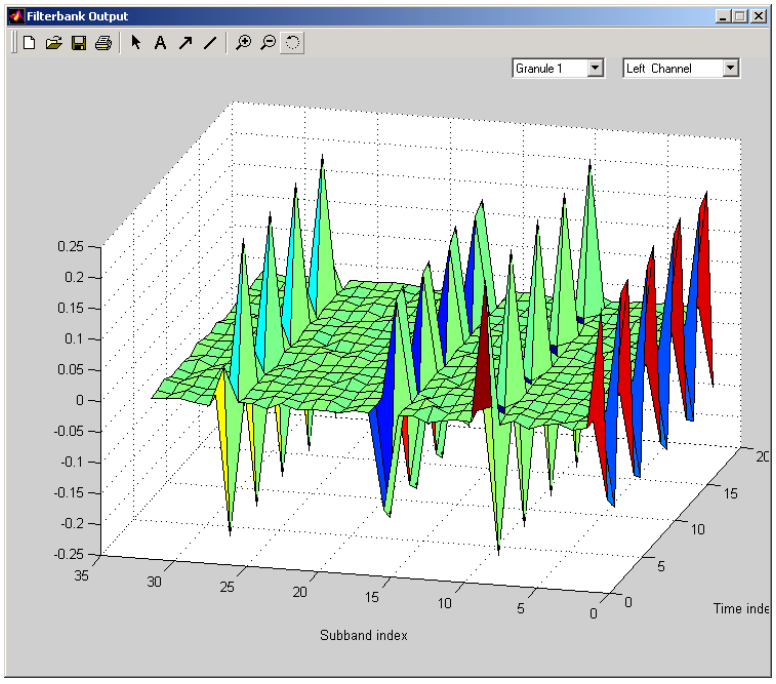


Fig.A.2 The time-domain output of the Analysis Filterbank.

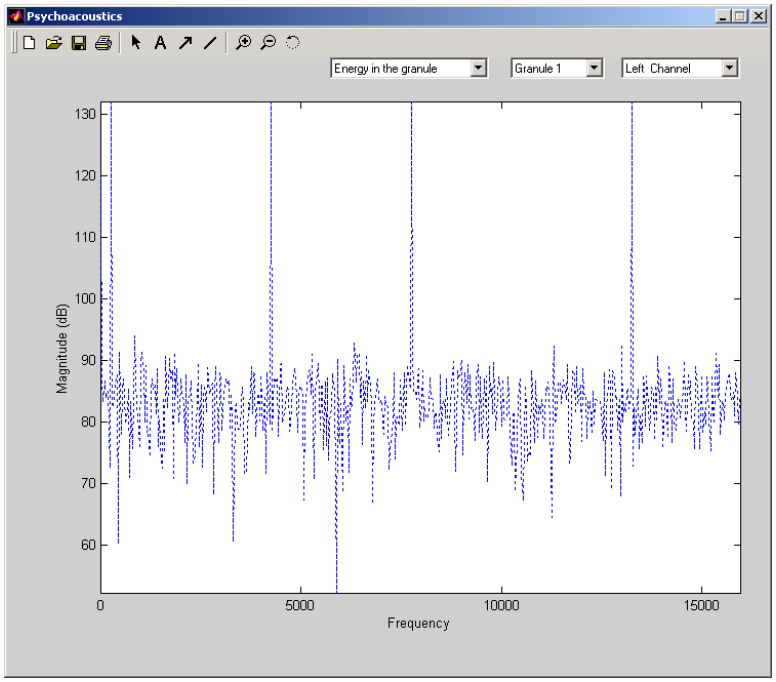


Fig.A.3 The corresponding spectrum, as computed by the Psychoacoustics model.

A.3 Aliasing at the Analysis Filterbank and its (partial) cancellation in the MDCT domain

In an ideal situation, the analysis filterbank is a set of brick-wall filters. But, as seen from Fig. 5.4, there is significant overlap between the adjacent subbands of the MP3 filterbank. This results in leakage of energy and as a consequence, a reduction in subband gain. To complicate matters, critical-sampling in the filterbank introduces significant aliasing.

The outputs of the subbands are transformed into the frequency domain via a signal-adaptive MDCT. Part of the aliasing can be cancelled out in the frequency domain by the application of anti-aliasing butterflies.

Exercise: Create a signal comprising of two sinusoids at 675 Hz and 11000 Hz. Use a sampling rate of 44.1 kHz. For display purposes, it is recommended to add additive white noise at a low energy.

If the analysis filterbank were an ideal set of brick-wall filters, what would the subband boundaries be? In which of the subbands would you expect maximum gain?

Observe and explain what happens in the actual case. Also, observe the MDCT outputs before and after aliasing cancellation butterflies. Comment on the same.

Procedure: Invoke the MP3 encoder by enabling the GUI. Observe the outputs of the filterbank and the MDCT block before and after alias-cancellation.

Observation: We know that at a sampling rate of 44.1 kHz, the nominal BW of the prototype filter is 689 Hz. If the analysis filterbank were composed of a set of brick-

wall filters, their boundaries would be as indicated in Table A-1. For the given signal, it can be seen that the maximum gains would be in subbands 0 and 15. From Fig. 5.4, it can be seen that there is significant overlap b/w adjacent subbands. The actual subband outputs are shown Fig. A.4. The tone at 675 Hz produces outputs at subbands 0 and 1. Similarly, the tone at 11000 Hz produces outputs at subbands 15 and 16.

Subband Number	The upper boundary (Hz)	Subband Number	The upper boundary (Hz)
0	689	16	11714
1	1378	17	12403
2	2067	18	13092
3	2756	19	13781
4	3445	20	14470
5	4134	21	15159
6	4823	22	15848
7	5512	23	16537
8	6201	24	17226
9	6890	25	17915
10	7579	26	18604
11	8268	27	19293
12	8957	28	19982
13	9646	29	20671
14	10335	30	21360
15	11025	31	22050

Table A-1 The ideal brick-wall filterbank.

The MDCT coefficients of the second granule of a frame are as shown in Fig. A.5. It can be seen that the application of alias-reduction butterflies reduces the strength of the signal in the aliased subbands.

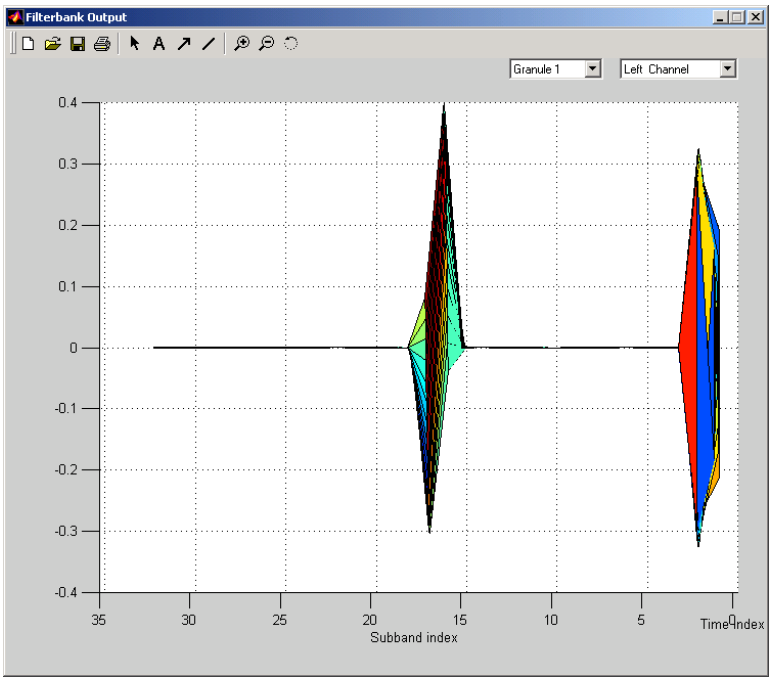


Fig.A.4 Time-domain output of the Analysis Filterbank.

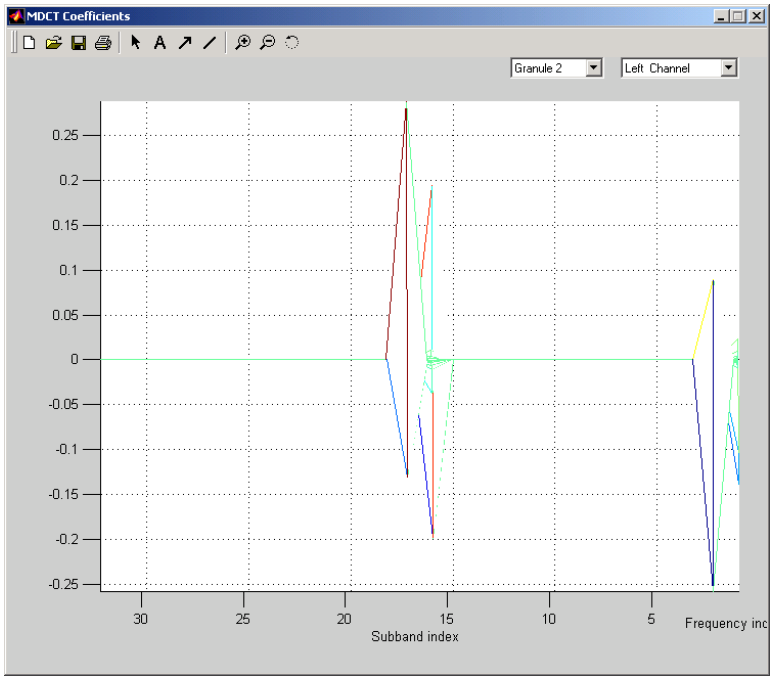


Fig.A.5 The MDCT output before alias-reduction.

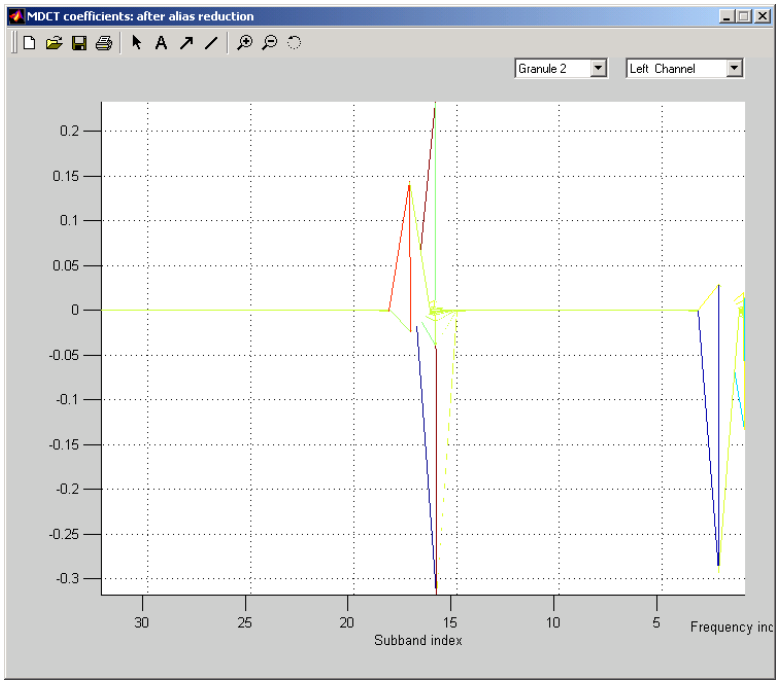


Fig.A.6 The MDCT output after alias-reduction.

A.4 The notion of Perceptual Entropy

The ear perceives only a part of the information present in the stimulus. This is called as *Perceptual Entropy* (PE). From a compression standpoint, this is the 'critical mass' of the signal, the minimum number of bits required to represent the perceptually relevant information in the signal. Any extra information can be safely discarded without affecting the perceptual quality of the signal reconstructed from a compact representation of this critical mass. Decidedly, the scheme is *lossy*, but perceptually *transparent*.

A model for computing the perceptual entropy mimics the working of the auditory system and computes a Just Noticeable Distortion (JND) profile for a given frame of audio data - a measure of the maximum quantization noise that can be injected for perceptually lossless signal recovery. The JND profile can then be used to shape the spectrum of the quantization noise to make it inaudible.

Exercise: Play the audio record (Ex4.wav) and feel the variations in the signal strength/energy. Plot its time-domain waveform. Modify the MP3 source code to successively store the PE value of every frame into a file and encode the signal at 128 kb/s. Plot the resulting PE values. Correlate the audio and visual cues and state your conclusions.

Note: To speed up the computation by orders of magnitude, comment out calls to the bitstream-formatter[5]. You may also comment out calls to the analysis filterbank , the MDCT and the rate-control loop.

[5] See Ch. 6.2 for a profile of the code.

Procedure: The code in the PsychoAcoustics.m function is modified by adding the following code starting at line 72:

Line 71: Psycho.PE(ch, gr, 10) = P.pe;

f_pe = fopen(‘pe.txt’, ‘at’);

fprintf(f_pe, P.pe);

fclose(f_pe);

Invoke the MP3 encoder by disabling the GUI. Encode the file and plot the data in pe.txt.

Observation: The signal waveform and its PE are depicted in Fig. A.7 and Fig. A.8 respectively. Perceptual entropy, a measure of ‘audible’ entropy/information in the signal, is based on a model of human perception. It is computed from a measure of local energy. Using the audio cues and the two figures below, it is clear that sudden increase in local signal power is a clear indicator of an increase in PE.

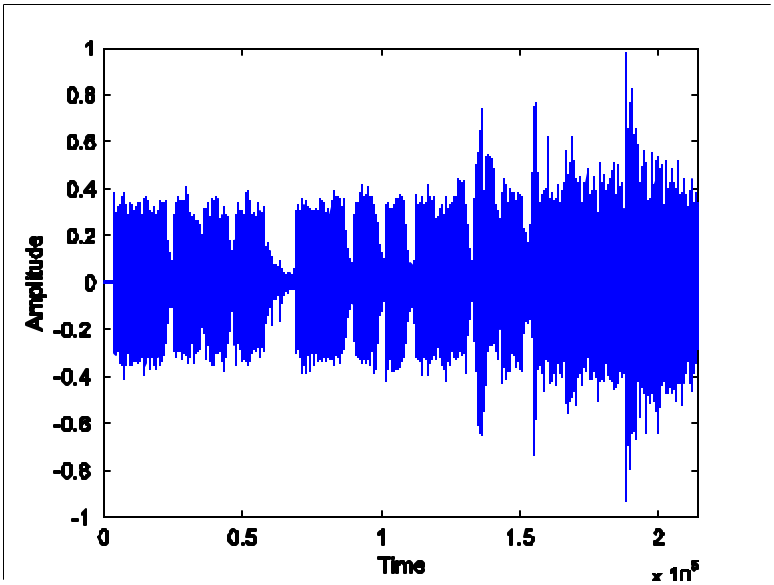


Fig.A.7 The time-domain waveform of the signal.

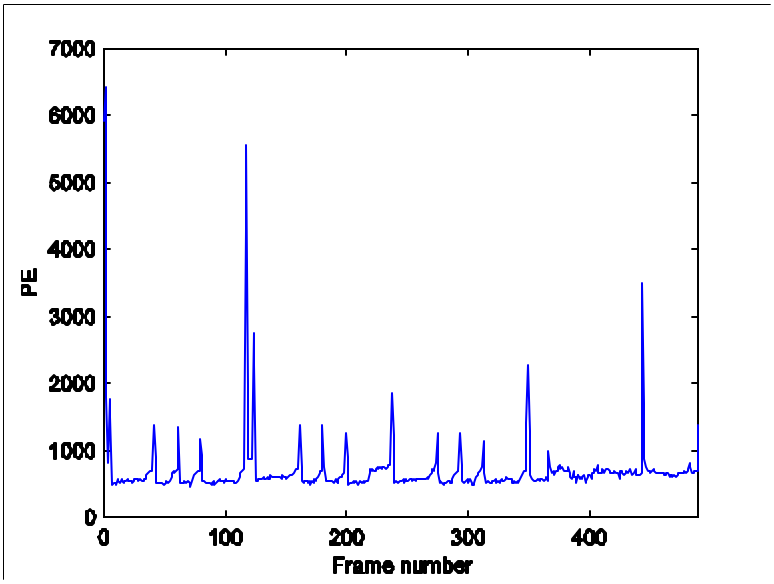


Fig.A.8 The PE of the signal on a per-frame basis.

A.5 Pre-echo and its control

Audio coding algorithms transform blocks of data and code them efficiently using the energy compaction properties of the transformation, supplemented by psychoacoustic analysis to extract perceptual redundancies. The Layer III algorithm uses the MDCT to transform the subband data.

The longer the block length, the better is the frequency resolution of the transform but the poorer is its time resolution. For relatively stationary signals, long blocks provide better compression (coding gain). On the other hand, the characteristics of transients are best captured with short time windows. For best results, the size of the block has to be adapted to the statistics of the signal. See Ch. 5.2.9.

Exercise: Take the test signal (attack.wav) and encode it. Decode the compressed bitstream and observe at the original and reconstructed time-domain waveforms.

Study the source code for the MPEG Psychoacoustics Model 2 carefully. Alter the code to disable the pre-echo control mechanism and the MDCT window-switching state-machine by forcing all blocks to be flagged as long (NORM_TYPE). With these modifications in effect, repeat the experiment. Listen to the original and decoded signals (with and without pre-echo control). Examine at the original and reconstructed time-domain waveforms. Comment on the same.

Procedure: Conduct the first part of the experiment as explained before. You may decode the compressed bitstream using the MATLAB MP3 decoder or a commercial application like Goldwave.

In the file PsychoAcoustics.m, replace lines 396 – 401 with

```
% pre-echo control, bug of IS
for b = 1:CBANDS
        thr(b) = max(qthr_l(b), nb(b));
end;
```

Similarly, replace lines 415 – 505 with the following:

```
% calculate perceptual entropy
pe(gr, chn) = 0.0;
for b = 1:CBANDS
  tp = min(0.0, log((thr(b)+1.0)/(eb(b)+1.0)) ); % not log
  pe(gr, chn) = pe(gr, chn) - numlines(b) * tp ;
end;                        % thr[b] -> thr[b]+1.0 : for non sound portition

Psycho.pe = pe(gr, chn);
switch_pe = 1800;
blocktype = NORM_TYPE;

% all blocks are forced to be long
% threshold calculation (part 2)
for sb = 1:SBMAX_l
  en(sb) = w1_l(sb) * eb(bu_l(sb)+1) + w2_l(sb) * eb(bo_l(sb)+1);
  thm(sb) = w1_l(sb) *thr(bu_l(sb)+1) + w2_l(sb) * thr(bo_l(sb)+1);

  b = bu_l(sb)+1:bo_l(sb);
  en(sb) = en(sb) + sum(eb(b));
  thm(sb) = thm(sb) + sum(thr(b));

  if en(sb) ~= 0.0
```

```
      ratio(chn, sb) = thm(sb)/en(sb);
    else
      ratio(chn, sb) = 0.0;
    end;
  end;
end;
% get a copy of the struct ...
cod_info = get_gr_info_from_l3_side(l3_side, gr, chn);
cod_info.block_type = blocktype_old(chn);
blocktype_old(chn) = blocktype;
cod_info.window_switching_flag = 0;
cod_info.mixed_block_flag = 0;
% now copy the change ...
l3_side = put_gr_info_to_l3_side(cod_info, l3_side, gr, chn);
```

Observation: We see that if a sharp attack occurs at the end of a long block, as for the signal under consideration (Fig. A.9), the psychoacoustic model would be misled to derive a higher masking threshold for that entire block. As a result, the quantization noise is spread over the entire block as shown in Fig. A.10. Switching to shorter windows controls the spread of noise in the time domain, as shown in Fig. A.11.

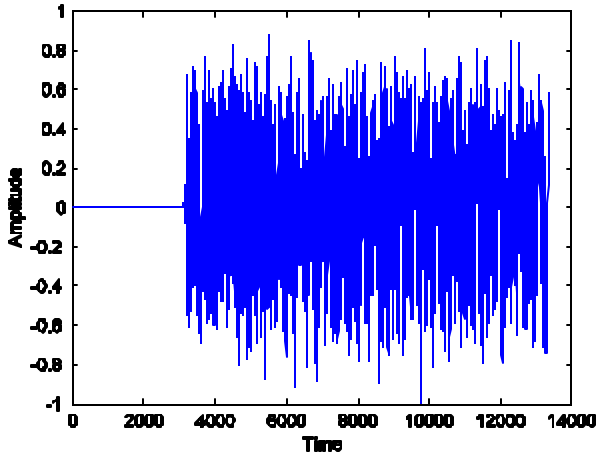


Fig.A.9 The signal under consideration.

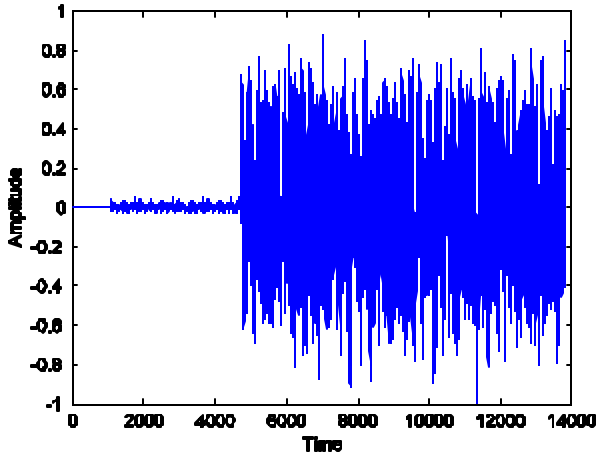


Fig.A.10 Pre-echo distortion when using only long blocks to code the signal.

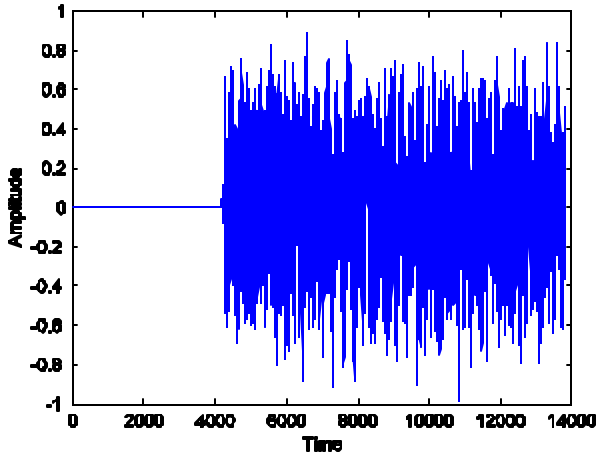


Fig.A.11 Pre-echo distortion is mitigated by the use of signal-adaptive block sizes.

A.6 The effect of Rate Control

Exercise: Take the test signal (Ex6.wav) and observe the quantized MDCT outputs for target bit-rates of 320, 192, 128, 64 and 32 kb/s for a particular frame,. Comment on the same.

Procedure: Start the encoder with the GUI enabled. At each target bit-rate, encode about 3 frames before settling on a granule for comparison. Capture the results of the rate-control loop.

Observation: All high-fidelity audio coders rely upon a model of human auditory masking for shaping quantization noise. The MP3 algorithm meets the target bit-rate by iteratively changing quantizers till the (quantization) noise is below the JND for all scalefactor bands. So, as the bit rate control loop iteratively increases quantizer step-sizes, part of the quantized high-frequency spectrum will contribute to the string of run-length zeros. The bits thus saved are distributed among the stronger signal components. This process is non-linear and aims to maximize SMR at the expense of pruning higher frequencies. It can be seen from Fig. A.12 – Fig. A.16 that as the target bit-rate gradually increases, the quantized spectrum has a marked low-pass filtered effect.

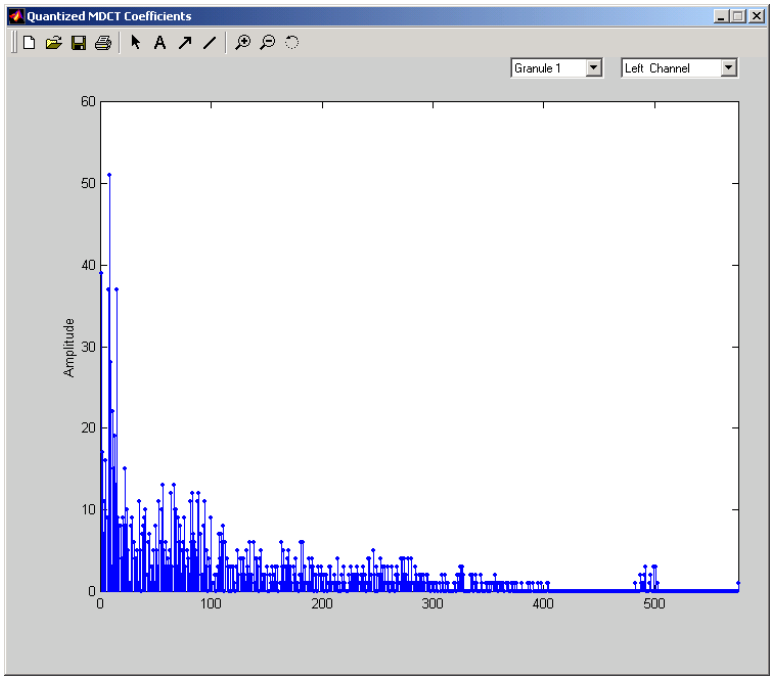


Fig.A.12 Quantized MDCT coefficients for a target bit-rate of 320 kb/s.

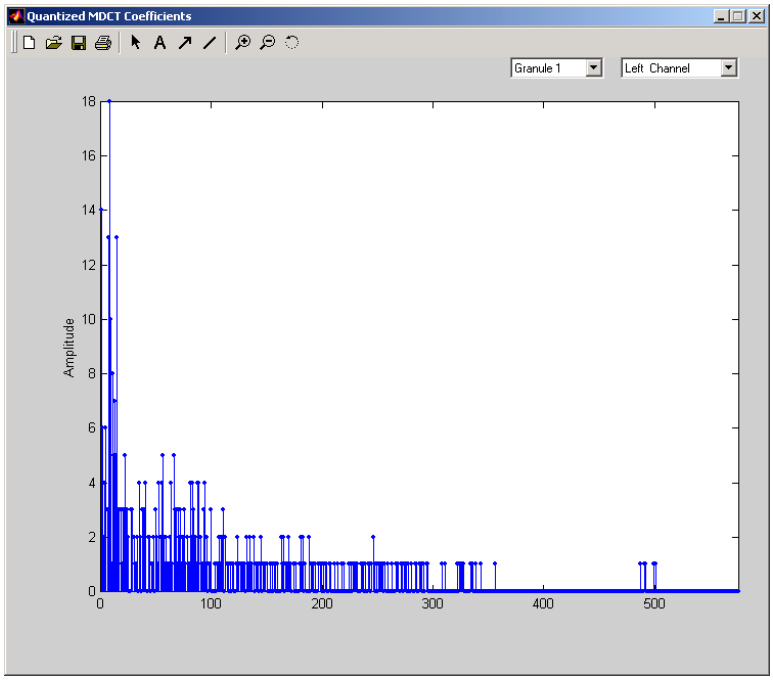


Fig. A.13 Quantized MDCT coefficients for a target bit-rate of 192 kb/s.

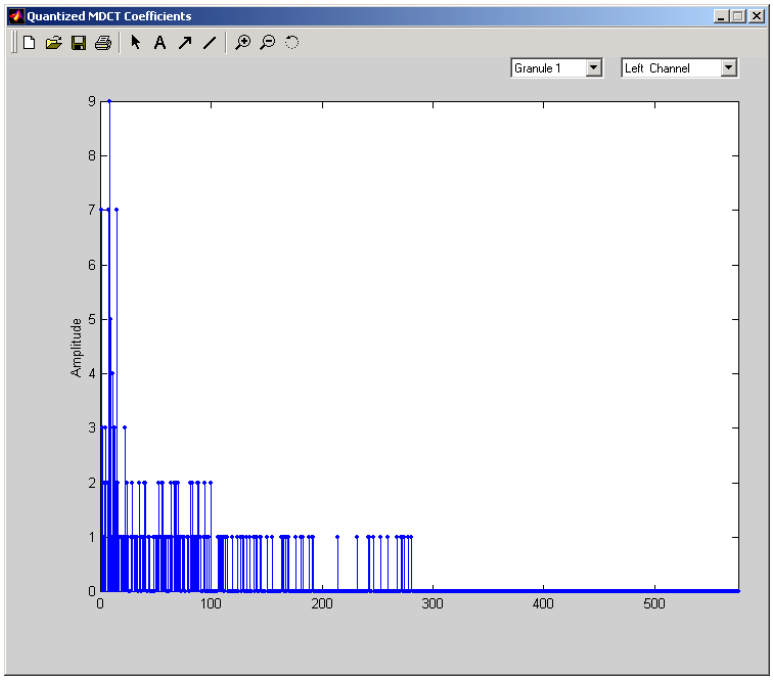


Fig. A.14 Quantized MDCT coefficients for a target bit-rate of 128 kb/s.

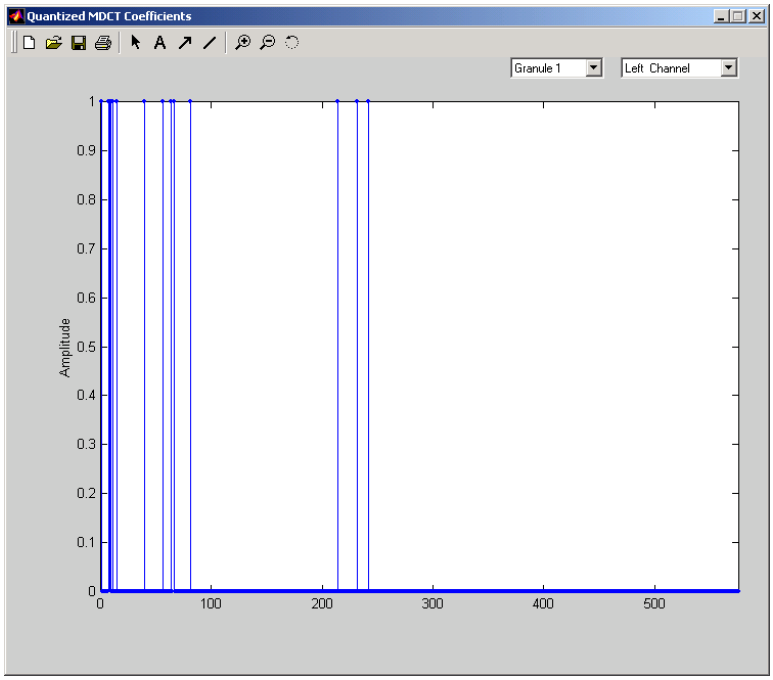


Fig.A.15 Quantized MDCT coefficients for a target bit-rate of 64 kb/s.

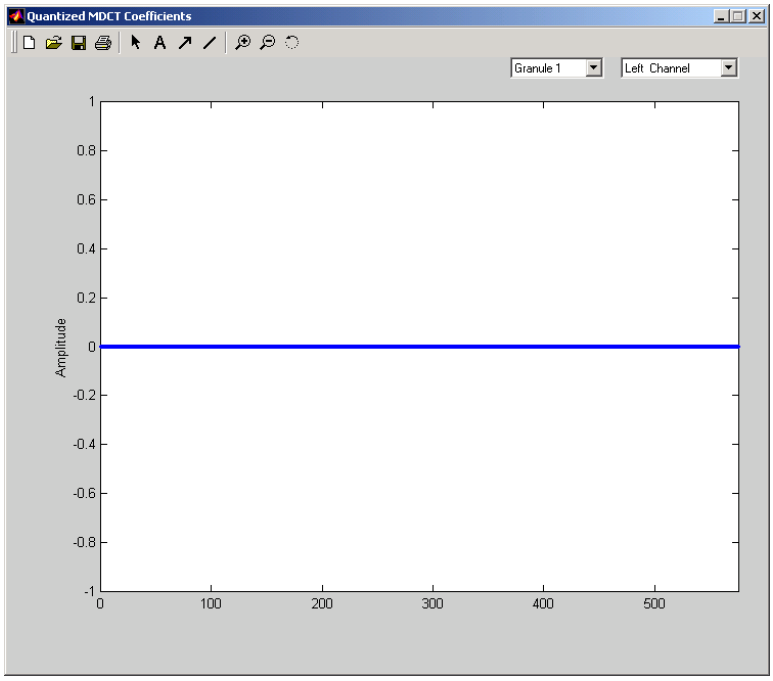


Fig.A.16 Quantized MDCT coefficients for a target bit-rate of 32 kb/s.