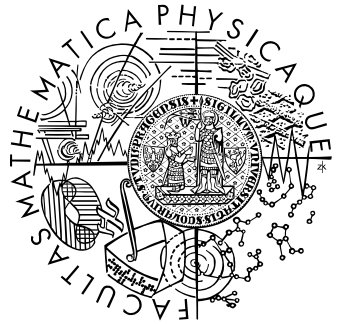
Charles University in Prague
Faculty of Mathematics and Physics

BACHELOR THESIS



Ján Dupej

Komprese zvuku
Audio Coding

Department of Software Engineering

Consultant: Mgr. Jan Lánský

Specialization: Programming

2009

I would like to thank my consultant, Mgr. Jan Lánský for his invaluable assistance with writing this thesis.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 28.05.2009 Ján Dupej

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AAC	Advanced Audio Coding
ABR	Average Bitrate
ASF	Advanced Systems Format
BMLD	Binaural Masking Level Difference
CBR	Constant Bitrate
CD	Compact Disc
dB	Decibel
dB SPL	Decibel of Sound Pressure Level
DCT	Discrete Cosine Transform
FFT	Fast Fourier Transform
FLAC	Free Lossless Audio Coder
IMDCT	Inverse Modified Discrete Cosine Transform
KBD	Kaiser-Bessel Derived (Window)
LAME	LAME Ain't an MP3 Encoder
LFE	Low Frequency Enhancement (Channel)
LPC	Linear Predictive Coding
MDCT	Modified Discrete Cosine Transform
MP3	MPEG1 – Layer 3
MPEG	Motion Picture Expert Group
NMN	Noise Masking Noise
NMT	Noise Masking Tone
PCM	Pulse-Code Modulation
PE	Perceptual Entropy
PF	Polyphase Filterbank
PNS	Perceptual Noise Substitution
PQMS	Pseudo-Quadrature Mirror Filterbank
QMF	Quadrature Mirror Filterbank
SBR	Spectral Band Replication
SFM	Spectral Flatness Measure
SMR	Signal-to-Mask Ratio
SNR	Signal-to-Noise Ration
TDAC	Time-Domain Aliasing Cancellation
TMN	Tone Masking Noise
TNS	Temporal Noise Shaping
VBR	Variable Bitrate
WMA	Windows Media Audio

Title: Audio Coding
Author: Ján Dupej
Department: Department of Software Engineering
Supervisor: Mgr. Jan Lánský
Supervisor's e-mail address: zizelevak@gmail.com

Abstract:
In the last two decades multimedia have become an integral part of our lives. However, we often face the two clashing requirements – limited storage space or internet connection capacity and the demand for reasonable quality of the media. Compression makes these two requirements more compatible by reducing the amount of data neccessary to store the media.

This thesis concentrates on sound, particularly lossy or perceptual compression of audio. As opposed to lossless compression schemes, perceptual coders introduce some noise to the signal to make it better compressible by lossless methods. The tradeoff is an impressive coding efficiency provided by most of these coders. The point of interest in designig a lossy audio coder is to make that damage as imperceptible as possible. This is achieved with knowledge of psychoacoustics (exploiting the imperfections of human auditory system), specifically masking thresholds, perceptual entropy, quiet thresholds and many more. This thesis explains some of these phenomena and their practical implementations in modern audio coders.

Finally an overview of select modern audio coders is given, including some technical details about their operation and capabilities.

Key words: audio, compression, psychoacoustics

Název práce: Komprese zvuku
Autor: Ján Dupej
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí bakalářské práce: Mgr. Jan Lánský
e-mail vedoucího: zizelevak@gmail.com

Abstrakt:
V posledních dvou desetiletích se multimédia staly součástí života každého z nás. Často ale musíme čelit dvěma sporným požadavkům - omezené kapacitě fyzického úložiště nebo kapacitě připojení k síti a požadavku na rozumnou kvalitu našich médií. Komprese zvyšuje kompatibilitu těchto dvou požadavků tím, že zmenšuje objem dat, který je potřebný na reprezentaci originálu.

Tato práce se zaměřuje na kompresi zvuku, specificky na ztrátovou kompresi.
Na rozdíl od bezztrátových kompresních algoritmů, ztrátové zavádějí do originálu šum. Výhodou bývá vysoký kompresní poměr, který mnohé tyto kodéry poskytují. Součástí navrhování ztrátového kodeku je snaha učinit ztráty vzniklé při kompresi méně slyšitelnými. Toho se dosahuje pomocí psychoakustiky (využívaní nedostatků lidského sluchu), konkrétně prostřednictvím maskování, vněmové entropie, prahů slyšitelnosti a mnoha dalších jevů. Tato práce vysvětluje některé z těchto jevů a popisuje jejich praktickou implementaci v moderních ztrátových kodérech.

Na konec se práce zabývá porovnáním některých zvukových kodeků, jejich principů a schopností.

Klíčová slova: zvuk, komprese, psychoakustika

Introduction

In the last two decades, internet and multimedia have gained great importance in our lives. It is safe to say that watching our favorite TV show on the internet or listening to music from an MP3 player in the subway is commonplace. Despite the network connection speeds and local storage being limited, the users demand the best possible quality of playback. Apparently, the most effective response to that problem is compression. In this thesis we will focus on compression of digitally represented sound. Not unlike still images or video – sound can be compressed with or without losses. Lossy compression schemes degrade the quality of stored signal, but as opposed to lossless algorithms, they provide very impressive compression ratios (as high as 1:16) while inflicting only barely (if at all) noticeable damage to the signal. These coders are tailor-made for the human auditory system – they exploit many of its imperfections and peculiarities, they are generally more difficult to implement and to understand than lossless algorithms. This is why we will concentrate on lossy (perceptual) audio coding systems.

We will start with the most basic physical concepts of sound as a mechanical wave and some of the simplest ways of representing audio digitally in temporal domain (sampling, pulse-code modulation, companding algorithms, linear predictive coding). We will then describe some basic psychoacoustic principles like audibility threshold, masking and perceptual entropy that constitute the theoretical basis of the current perceptual coding schemes. After that a breakdown of a typical perceptual audio coder will be given describing some of the most essential components like a time-to frequency domain conversion filterbank and quantization loop. The function of these components will be illustrated with practical examples from current state-of-the-art audio coders. We will conclude this thesis with a comprehensive comparison of some selected codecs. Although they generally share a similar structure, they do have some differences in their capabilities and effectiveness. We will try to expose those differences in a comprehensive way.

The first chapter explains some of the basic physical concepts of sound and some simple methods of representing sound digitally. The second chapter lists some of the psychoacoustic principles and explains why they are useful in coding digital audio. The third chapter enumerates the elementary components of a typical audio coder and describes their respective functions and relations. The fourth and final chapter compares five of the popular modern codecs (MPEG-1 Layer 3, MPEG-2 AAC, Vorbis, Windows Media Audio 9, Dolby Digital). The criteria for the comparison include the

structure of the coders, their capabilities (maximum available channel count, supported sampling rates etc.) and their efficiency.

1. Basic Concepts

This chapter describes or clarifies some of the basic concepts associated with sound. These include its nature as a mechanical wave, sampling and some basic techniques used to represent sound in time domain.

1.1. Physical Concept of Sound

What we perceive as sound is in essence a mechanical wave that is propagated by air, water or other media. Sound is propagated by molecule collisions. When an object vibrates, it pushes molecules of air in its vicinity away. This locally increases the pressure of the medium. The wave created by the local pressure increase is then propagated away from the object by molecule collisions. A receptor placed in the wave's way will then move according to the pressure applied to it. Furthermore, the denser the medium, the better is the wave propagation. This is why sound moves faster in water than in air. (1)

The pressure changes may be periodic or aperiodic. In case of a periodic pressure change, according to the Fourier theorem (2) it makes sense to express it as a sum of sine waves of specific frequencies and amplitudes (and phase shifts). The local pressure of a sine wave in time t can be expressed as follows:

$$p(t) = p_0 \cdot sin(2\pi f t + \varphi_0) \quad [1]$$

Where p_0 is the wave's amplitude, f is its frequency and φ_0 is its initial phase shift. (3) Amplitude of a sine wave is the difference between its maximum pressure level and the equilibrium pressure of the atmosphere. *Frequency* is the number of times the pressure in the wave rises from minimum to maximum.

When studying perception (and measurements) of sound, it may be useful to know the *dynamic range* of the receiver. A dynamic range is the difference (in pressure levels) between the softest and loudest perceivable sound. Human auditory system has a dynamic range spanning a factor of millions. (1)

A logarithmic scale (decibel, dB) for intensity was developed to accommodate such range. The conversion from power to intensity level is simple:

$$Intensity\ level = 10 \cdot log_{10} \frac{P_1}{P_0} dB\ SPL \quad [2]$$

Where P_1 is the power we want to convert and P_0 is the reference level. Typically, it is set to threshold of hearing (10^{-12} Wm^{-2}). When ratios of currents, voltages or sound pressures are used (quantities whose square is proportional to power), the above decibel formula must be multiplied by a factor of 2. (1)

1.2. Digital Representation and Sampling

Digital audio is represented as a list of measurements of the wave amplitude taken a finite number of times per a unit of time. This is called *sampling*. How often the wave is measured is determined by the *sampling rate*. Typical values range from 8 kHz for low-quality voice, through 44.1 kHz for Audio CD, to 96 or 192 kHz for high definition audio.

Nyquist sampling theorem: *A sampled waveform contains all the information without any distortion, when the sampling rate exceeds twice the highest frequency contained by the sampled waveform.* (4)

Proving the Nyquist theorem is beyond the scope of this thesis, however it is very useful to see how fast a sampling rate needs to be. Considering the human ear can seldom perceive any frequencies beyond 20 kHz, (3) a sampling rate of 44.1 kHz seems optimal. On the other hand, there are several reasons, why sampling rates as high as 96 kHz might improve perception of the sound. These include narrower impulse response or better analog-like behavior of the sampled wave. However, how much these factors contribute to improving the perceptual quality remains questionable.

Furthermore, the precision of the sampled values varies greatly. It is determined in *bits per sample*. Typical value for low-quality voice is 8 bits per sample (bps). This bit depth can represent possible values in range -128 to 127. Audio CD uses a bit depth of 16 bits per sample. This way, each sample of Audio CD can have values ranging from -32768 to 32767. This gives sufficient precision and dynamic range to be almost indistinguishable from reality. High definition audio has a bit depth of 24 or even 32 bits per sample.

Bit depth	Possible values	Dynamic range
8	-128 ... 127	42.0 dB
16	-32768 ... 32767	90.3 dB
24	-8388608 ... 8388607	138.5 dB
32	-2147483648 ... 2147483647	186.6 dB
32 (float)	-1.0 ... 1.0	758.6 dB

Table 1 Dynamic Ranges at Various Bitrates

Such representation of digital audio is commonly referred to as *pulse-code modulation* (PCM).

Let us calculate how large one second of audio stored in PCM can be. We want our sound to be of Audio CD quality. This means 44100 samples per second, each with 16-bit precision and two channels for stereo (5). The total bitrate is as high as:

$$44100 \cdot 16 \cdot 2 = 1411200\ bits\ per\ second = 1378\ kbits\ per\ second$$

This is horrible indeed. At such bitrate, we need a rather large medium to store a reasonably long recording. Clearly, this is unsuitable for transferring audio over computer networks because of high bandwidth demands. To make matters even worse, it is not possible to effectively compress PCM audio with entropy or dictionary coders as a result of rapidly changing values across a rather wide range.

1.3. A-law and μ-law

Some reduction in bit rate may be achieved by using *bit-companding* algorithms such as A-law or μ-law. The sample values represent the sound pressure logarithmically, rather than linearly (PCM). This reduces the number of bits needed to represent a wide enough dynamic range and maintains the signal-to noise ratio at a constant level (in dB).

Each sample in μ-law is represented as follows:

$$F(x) = sgn(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)}\ ;\quad 0 \le |x| \le 1 \qquad [3]$$

Where x is the sample value normalized to $(-1, 1)$, μ is the compression constant and is normally set to 255. $F(x)$ is stored as an 8-bit fixed point number. (6)

This function is reversed at the receiver to get pure PCM using the following formula:

$$F^{-1}(y) = sgn(y)(1/\mu)\big((1+\mu)^{|y|} - 1\big)\,; \quad -1 \le y \le 1 \qquad [4]$$

Where y is the normalized encoded value and μ is the compression constant - same as in encoder.

This algorithm is used for coding of voice for transmissions over mobile networks in North America and Japan. A-law is a similar algorithm used primarily in Europe. (6)

These algorithms are acceptable for voice representation especially because of simplicity of their implementations and their low computing power demands (log and power functions are efficiently implemented using lookup-tables). It is therefore easy and cost-effective to use such coders in cell phones or voice recorders. However, for high-fidelity audio, these algorithms are insufficient.

1.4. Linear Predictive Coding

Linear Predictive Coding (LPC) encodes a sample using the knowledge of previous samples. A *predictor* function is used to calculate the *residual* value. Generally a predictor of order p is the function (3):

$$P^p\big(x_{i-1}, x_{i-2}, \dots, x_{i-p}\big) = \sum_{j=1}^{p} a_j x_{i-j} \qquad [5]$$

Where x_i is the i-th sample and a_i are the predictor's coefficients. A zero-order predictor makes no changes to the encoded vector. The residual r_i is then calculated easily:

$$r_i = x_i - P_i \qquad [6]$$

A vector of samples is then coded as the first few values followed by residuals. Typical predictors (used in lossless Shorten algorithm) include (3)

$$p^0 = 0 \qquad [7]$$

$$p^1(x_{i-1}) = x_{i-1} \qquad [8]$$

$$p^2(x_{i-2}, x_{i-1}) = 2x_{i-1} - x_{i-2} \qquad [9]$$

$$p^3(x_{i-3,}x_{i-2}, x_{i-1}) = 3x_{i-1} - 3x_{i-2} + x_{i-3} \quad [10]$$

Statistically, the residuals (if the right predictor is chosen) are densely distributed around zero. This makes an encoded vector more suitable for consecutive entropic coding. (3) This method is primarily used in lossless audio coding (7) and thus further discussion about it would go beyond the scope of this thesis.

2. Psychoacoustics

Let us focus on encoding high fidelity audio. It is safe to assume that the audio will be listened to by humans. We may then as well take advantage of imperfections of the human auditory system. This is the field of *psychoacoustics*. The understanding of human perception of sound has experienced significant progress. Most lossy audio coders exploit the fact that some information in the sound cannot be perceived by humans due to specific psychoacoustic effects. This information is removed by lossy audio coders to improve their compression efficiency.

Modern audio coders generally take advantage of absolute *hearing thresholds*, *simultaneous and temporal masking*, *noise substitution* and other principles. These effects combined with the properties of quantization have led to the concept of *perceptual entropy*, a quantitative estimate of the limit of transparent audio signal compression. (8)

2.1. Absolute Threshold of Hearing

The absolute threshold of hearing characterizes the amount of energy needed in a pure tone to be barely detectible by a listener in a noiseless environment. It is typically expressed in dB SPL. The quiet threshold is well approximated by the non-linear function:

$$T_q(f) = 3.64\left(\frac{f}{1000}\right)^{-0.8} - 6.5e^{-0.6\left(\frac{f}{1000}-3.3\right)^2} + 10^{-3}\left(\frac{f}{1000}\right)^4 \; dB\; SPL \qquad [11]$$

Where f is the frequency in Hz. The equation above applies for a young listener with acute hearing – in other words the worst case. (8)

Applied to audio compression, the threshold of hearing is the maximum allowable noise created by the coder's quantizer. There are, however two caveats with this notion. First, the absolute threshold of hearing determines the maximum unnoticeable energy in a pure tone. Quantization noise is typically non-tonal and is thus much more noticeable than pure tones. Second, it is important to note that the coder designer has no prior knowledge of actual playback level. It is therefore assumed that the softest possible sound will be played back at (or close to) 0 dB SPL. (8)

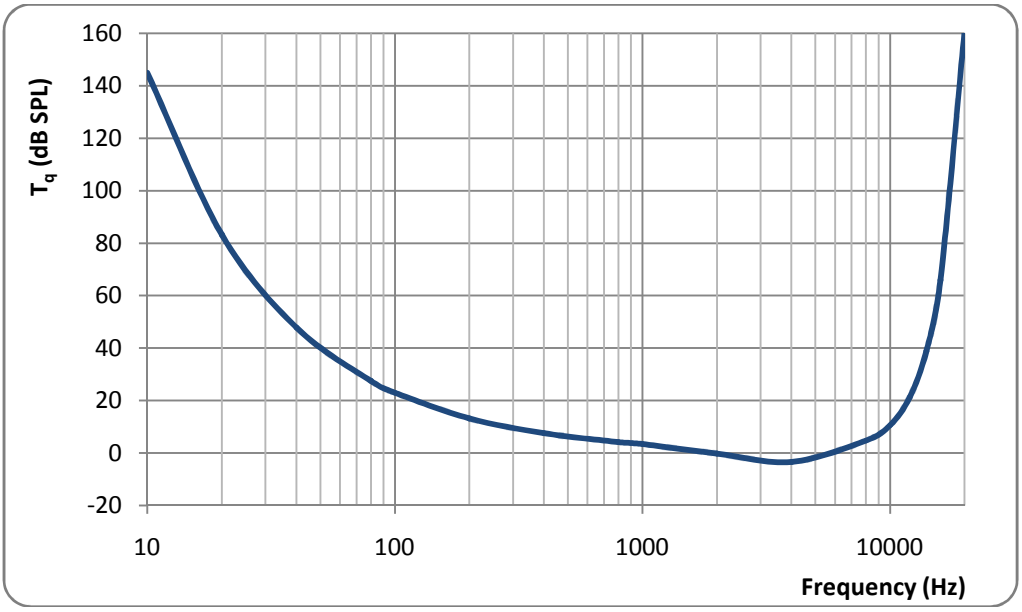


Figure 1 Audibility Threshold

2.2. Critical Bands

The ear transforms the acoustic energy to mechanical energy and at the final stage to electrical impulses that can be interpreted in the brain. The actual transformation to electrical impulses is performed on the basilar membrane (located in cochlea). There are about 30 000 hair cells arranged in multiple rows along the basilar membrane. These cells convert vibrations in the surrounding fluid into neural information sent into the brain. (1)

Analysis of the signals created by the groups of hair cells shows that different groups react to stimuli in different bands of the perceptible spectrum. Frequency discrimination dictates that at low frequencies, the ear is able to distinguish pure tones only a few Hz apart. However, at higher frequencies the tones must differ even by several hundred Hz. In any case, the hair cells respond to the strongest stimulation in their local region. This is called a *critical band* – a concept introduced by Harvey Fletcher in 1940. (1)

The frequency-to-location transformation on the basilar membrane can be viewed as a bank of highly overlapping bandpass filters. Their magnitude responses are asymmetric and non-linear. Moreover, the filters are of non-uniform bandwidths. The term critical bandwidth is a function of frequency that quantifies the width of the basilar membrane's passbands. (4) The critical bandwidth is well approximated (9) by:

$$BW_c = 25 + 75 \cdot \left(1 + 1.4 \cdot \left(\frac{f}{1000}\right)^2\right)^{0.69} Hz \quad [12]$$

This function is continuous however it is useful to consider the ear as a discrete set of bandpass filters, whose passband widths conform to BW_c. As it is clear that the ear does not perceive frequency linearly, a non-linear scale (Bark) has been created that matches human perception of frequency. Conversion (9) is simple:

$$z(f) = 13 \cdot arctan(0.00076 \cdot f) + 3.5 \cdot arctan\left(\left(\frac{f}{7500}\right)^2\right) Bark \quad [13]$$

The critical bands are of uniform width on the Bark scale.

Critical band #	Center freq. (Hz)	Passband (Hz)	Critical band #	Center freq. (Hz)	Passband (Hz)
1	50	– 100	14	2 150	2 000 – 2 320
2	150	100 – 200	15	2 500	2 320 – 2 700
3	250	200 – 300	16	1 900	2 700 – 3 150
4	350	300 – 400	17	3 400	3 150 – 3 700
5	450	400 – 510	18	4 000	3 700 – 4 400
6	570	510 – 630	19	4 800	4 400 – 5 300
7	700	630 – 770	20	5 800	5 300 – 6 400
8	840	770 – 920	21	6 400	6 400 – 7 700
9	1 000	920 – 1 080	22	8 500	7 700 – 9 500
10	1 175	1 080 – 1 270	23	10 500	9 500 – 12 000
11	1 370	1 270 – 1 480	24	13 500	12 000 – 15 500
12	1 600	1 480 – 1 720	25	19 500	15 500 -

Table 2 Idealized Critical Band Filterbank

2.3. Simultaneous Masking

The process where one sound is rendered inaudible by presence of another sound is called *masking*. When two or more stimuli are presented to the auditory system, simultaneous masking may happen. The amount of masking is determined by the spectral shapes of both the masker and maskee. Even phase relationships between masker and maskee may affect masking. (4) This model basically involves a strong tonal or narrow-band noise signal creating sufficient excitation of the

basilar membrane to effectively mask the perception of a weaker signal. This can happen when center frequencies of both masker and maskee are close enough. (4) It is known that noise is much stronger a masker than pure tones. This leads to 3 possible scenarios: noise masking tone (NMT), tone masking noise (TMN) and noise masking noise (NMN)

2.3.1. Noise Masking Tone

In this type of masking there is a narrow-band noise that masks a tone within the same critical band, provided that the intensity of the masked tone is below the threshold that is directly related to the intensity and, to a lesser extent, center frequency of the masker noise. (4)

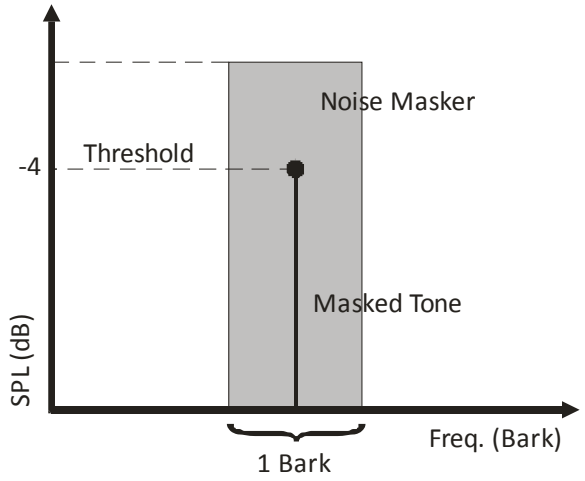


Figure 2 Noise Masking Tone (8)

Minimum signal-to mask ratio (SMR) (difference between SPL of masked signal and masking threshold) occurs when the masked tone frequency is close to center frequency of the masker noise. Studies have shown minimum SMR values in range +5 to -5 dB. (8)

2.3.2. Tone Masking Noise

Tone masking noise (TMN) occurs when a pure tone at the center of a critical band masks any noise of subcritical bandwidth provided that the noise is below the threshold that is directly related to the masker amplitude and, to a lesser extent, its center frequency. Studies have shown that the masking is the most effective when the masker frequency is close to maskee's center frequency. Minimum SMR for TMN lies between 21 and 28 dB. (8)

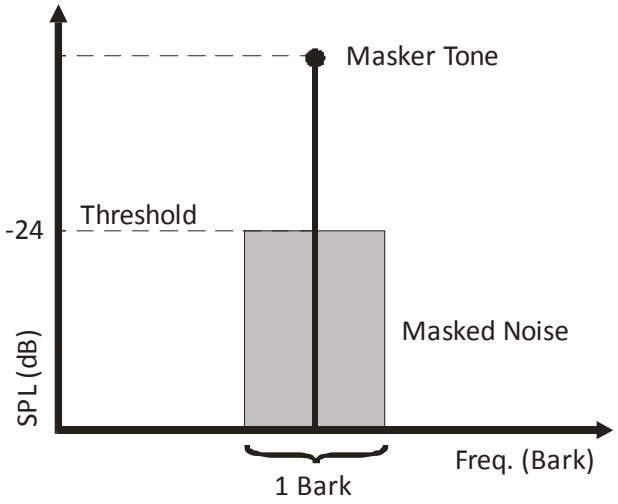


Figure 3 Tone Masking Noise (8)

2.3.3. Noise Masking Noise

The NMN type of masking, where one narrow-band noise masks another narrow-band noise is difficult to characterize because of the confounding influence of phase relationships between the components can lead to different threshold SMRs. (8)

2.3.4. Spread of Masking

The masking effects described above are not restricted to one critical band. Inter-band masking also occurs. In other words, a masker within one critical band can have some predictable effect on neighboring critical bands. In many psychoacoustic models the spread of masking is represented by a linear function (in dB) that slopes +25dB and -10dB per Bark. (8)

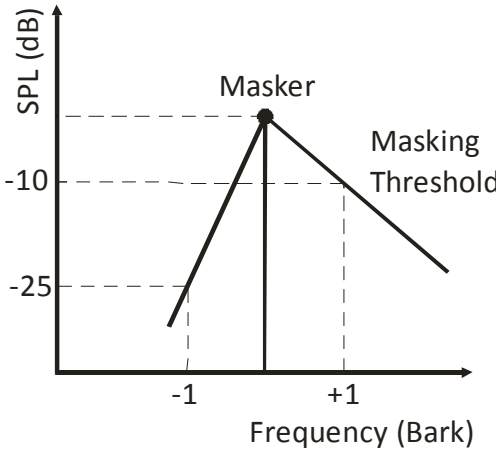


Figure 4 Spread of Masking

2.4. Temporal Masking

Masking phenomena are hardly limited to simultaneous masking. Significant amount of masking occurs shortly after cessation of the masker. A little masking even happens before the start of a strong stimulus. These effects are called *postmasking* and *premasking* respectively. In essence, masking thresholds for masked sounds before and after masking signals are modified. (4)

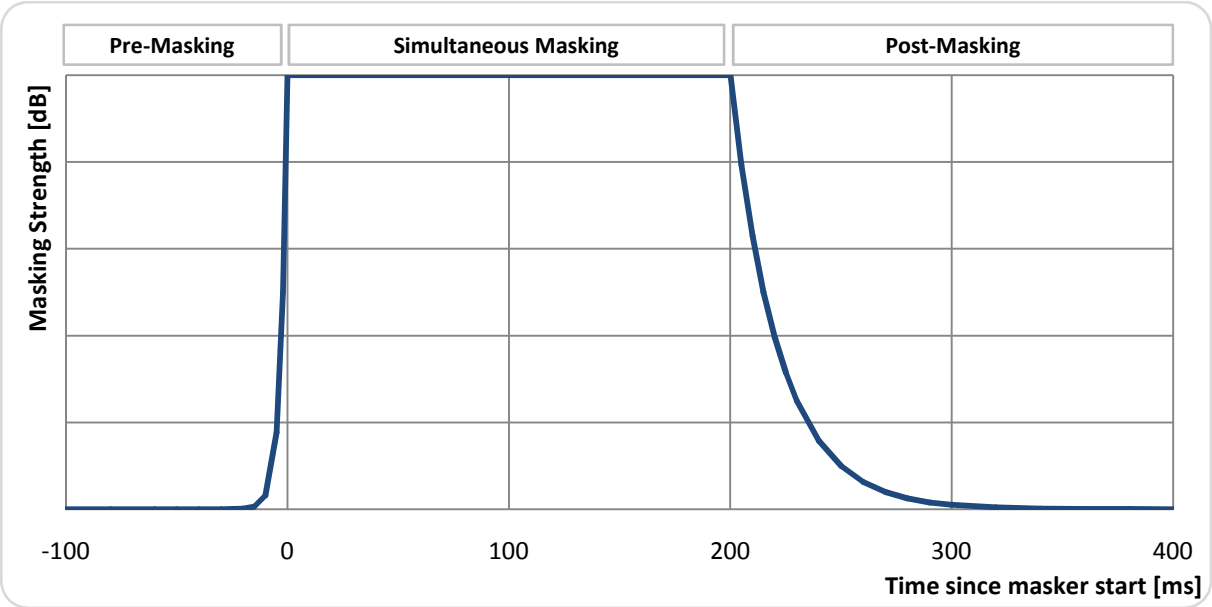


Figure 5 Temporal Masking (8)

Significant premasking lasts only about 1 or 2 milliseconds before the masker. On the other hand, postmasking can have a significant effect on masking thresholds for as long as 300 milliseconds after cessation of the masker signal. (4)

2.5. Perceptual Entropy

The knowledge of critical bands and masking leads to the concept of *perceptual entropy*. Psychoacoustic models assume that the noise introduced by quantization in a critical band should be inaudible as long as $SNR > SMR$. This allows us to calculate an estimate of average minimum number of bits needed to code a spectral line while introducing no perceptible noise to the original signal. The formula (5) is

$$PE = \frac{1}{N}\sum_{i=0}^{N-1} max\left\{0, log_2\left(\sqrt{\frac{I_i}{TM_i}}\right)\right\} \approx \frac{1}{N}\sum_{i=0}^{\mathrm{N-1}} \log_2\left(1+\sqrt{\frac{\mathrm{I_i}}{\mathrm{TM_i}}}\right) \quad [14]$$

Where N is the number of spectral lines in the frequency representation of the signal, I_i is the intensity of i-th spectral line and TM_i is the masking threshold at the i-th spectral line. The resulting perceptual entropy (PE) measure quantifies the lower bound for perceptual coding of audio signals based on spectral analysis and masking threshold.

After calculating the perceptual entropy, the audio coder has an estimate of how large the next frame of compressed audio will be. This can be used to further adjust the coder to better match the bitrate the user selected.

3. Inside a Perceptual Encoder

A typical perceptual audio coder can be broken down into several components. This chapter will provide descriptions of their respective functions and some ideas on their implementations.

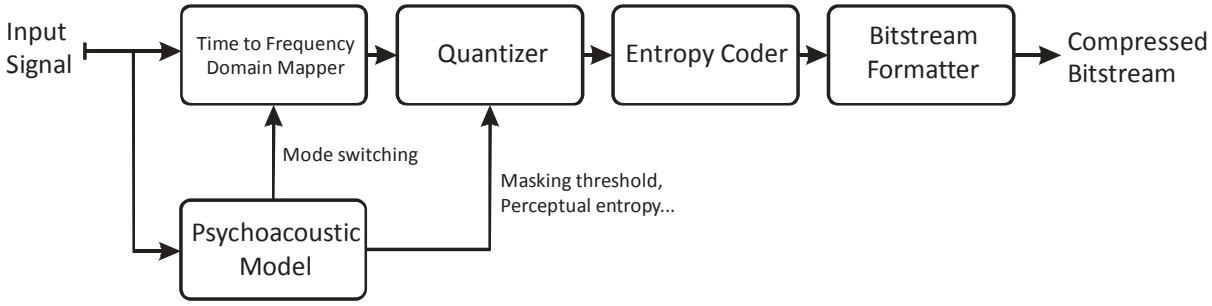


Figure 6 Simplified Block Diagram of a Perceptual Audio Coder for Single-Channel Audio (8)

3.1. Time-to-Frequency Domain Mapper

An audio coder receives its input represented in *temporal domain* (typically PCM). As shown in Chapter 1, there are some schemes to compress audio in temporal domain; however there is little possibility of applying any of the psychoacoustic principles described in Chapter 2. We need to represent our audio differently. It has proven effective to use the *frequency-domain* representation of the input signal.

This is typically achieved using *polyphase filterbanks*, *quadrature mirror filters* or *discrete cosine transforms* (DCT) or a combination of two of above-mentioned methods. Let us describe some of them.

3.1.1. Polyphase Filterbanks

A polyphase filterbank splits an input signal into a given number N (typically a power of 2) of equidistant (by frequency) spectral bands. Polyphase filterbanks are usually implemented as Polyphase Quadrature Filters, Quadrature Mirror Filters or Modified Discrete Cosine Transform. These filterbanks generally provide good time resolution and poor frequency resolution.

Polyphase Filterbanks are used in MPEG-1 Audio (Layers 1 and 2).

3.1.2. Modified Discrete Cosine Transform

Modified Discrete Cosine Transform (MDCT), sometimes referred to as (DCT-IV) is a function that maps a vector x_k of $2N$ real elements into a vector X_k of N real elements defined as follows (10):

$$X_k = \sum_{n=0}^{2N-1} x_n \cdot cos\left(\frac{\pi}{N}\left(n + \frac{1}{2} + \frac{N}{2}\right)\left(k + \frac{1}{2}\right)\right) \quad [15]$$

This transformation is easily reversed by the Inverse MDCT (IMDCT), which is defined as (10):

$$y_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k \cdot cos\left(\frac{\pi}{N}\left(n + \frac{1}{2} + \frac{N}{2}\right)\left(k + \frac{1}{2}\right)\right) \quad [16]$$

At first glance, it may seem like the MDCT should not be invertible. However, the perfect reconstruction is achieved by adding overlapped transforms of subsequent overlapping blocks with a length of N. This causes the errors to conveniently cancel out. This is known as *Time-Domain Aliasing Cancellation* (TDAC). To further improve the properties of the transform at the borders of a block, the vectors x_i and y_i are multiplied by a window function. So, after multiplication by window function, [15] will become:

$$X_k = \sum_{n=0}^{2N-1} w_n \cdot x_n \cdot cos\left(\frac{\pi}{N}\left(n + \frac{1}{2} + \frac{N}{2}\right)\left(k + \frac{1}{2}\right)\right) \quad [17]$$

And IMDCT [16] will turn into:

$$y_n = \frac{1}{N} \cdot w_n \cdot \sum_{k=0}^{N-1} X_k cos\left(\frac{\pi}{N}\left(n + \frac{1}{2} + \frac{N}{2}\right)\left(k + \frac{1}{2}\right)\right) \quad [18]$$

Commonly used functions include:

Sine window – used in MPEG-1 Layer 3 (8) and MPEG-2 AAC (12), defined by the function

$$w_n = sin\left(\frac{\pi}{2N}\left(n + \frac{1}{2}\right)\right) \quad [19]$$

Vorbis window – used in Vorbis (10), defined by the function

$$w_n = sin\left(\frac{\pi}{2} sin^2\left(\frac{\pi}{2N}\left(n + \frac{1}{2}\right)\right)\right) \quad [20]$$

Kaiser-Bessel Derived (KBD) window – used by AC-3 and MPEG-2 AAC (12), defined as

$$w_n = \begin{cases} \sqrt{\frac{\sum_{j=0}^{n} \widetilde{w}_j}{\sum_{j=0}^{N} \widetilde{w}_j}}; 0 \le n < N \\ \sqrt{\frac{\sum_{j=0}^{2N-1-n} \widetilde{w}_j}{\sum_{j=0}^{N} \widetilde{w}_j}}; N \le n < 2N \\ 0 \; otherwise \end{cases} \quad [21]$$

Where $\widetilde{w}_n$ is the value of Kaiser window function at n.

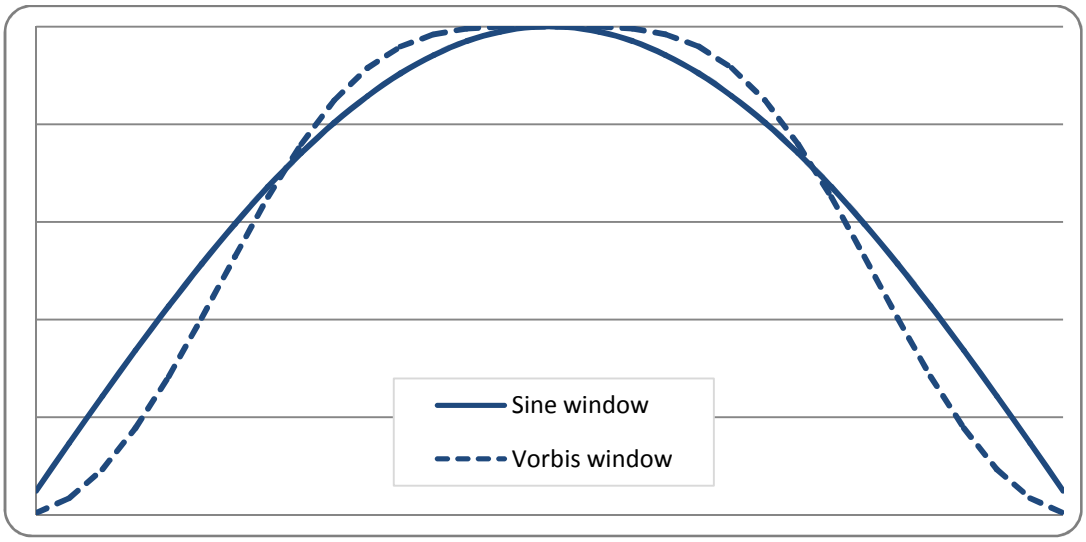


Figure 7 Window Function Shapes

The audio coder must be able to represent a continuous range of frequencies. The MDCT, however provides only finite number of spectral lines (has a finite basis). Those frequencies which do not belong to the basis are not periodic in the transform window. Therefore the transformation will include frequencies from all over the basis (in non-zero magnitudes). This effect is called *spectral leakage* (14).

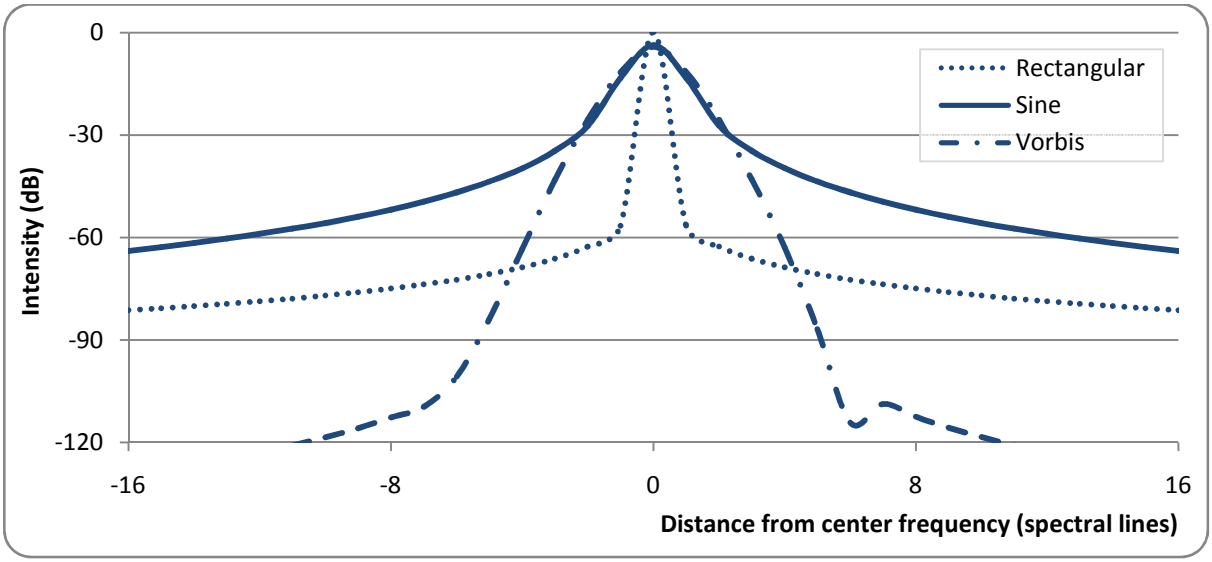


Figure 8 Spectral Leakages for Different Window Functions

The wisdom of choosing different window shapes is that they all have different spectral leakages and thus affect frequency resolution and consequently coding efficiency. The important thing about the windows shapes is that in order to maintain perfect reconstruction of the original signal, the window function must satisfy the Princen-Bradley (9) condition:

$$w_n^2 + w_{n+N}^2 = 1 \qquad [22]$$

Typical transform block lengths (2N) range from 128 to 2048 samples.

3.1.3. Hybrid Filterbanks

Hybrid filterbanks are methods which use a combination of a polyphase filterbank and MDCT. Such approach is used in MPEG1 Layer 3. First, the audio is sent through a filterbank that splits the signal into 32 sub-bands. Then, each subband is transformed with MDCT (36 or 12 samples long using a sine window). (4) This improves the filterbank's resolution to 1152 samples while maintaining low computational complexity (PQF is low-complexity and so is MDCT of such short blocks).

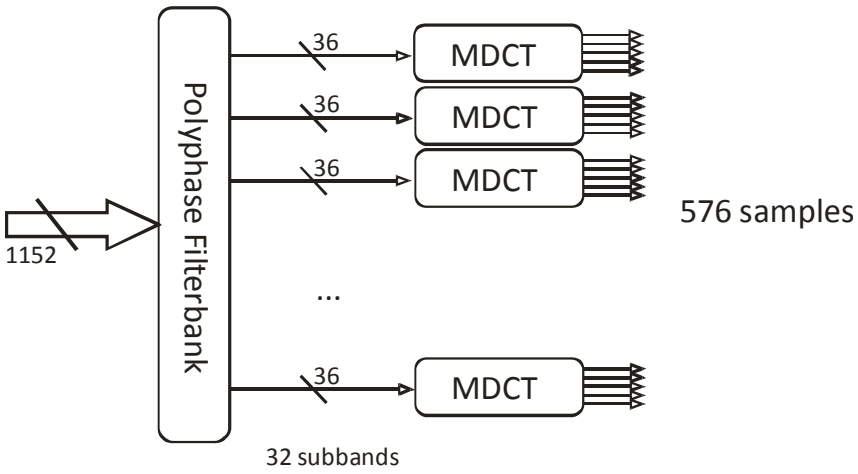


Figure 9 MPEG1 Layer 3 Hybrid Filterbank (11)

3.2. Mode Switching

It is important to decide the proper length of the transformed vector of samples. Long blocks improve coding efficiency, but have poor temporal resolution which may lead to pre-echo distortion typical to lossy audio coders. On the other hand, shorter blocks alleviate the pre-echo problem at the price of reduced frequency resolution and coding efficiency. Furthermore, short blocks may reduce the effectiveness of techniques like perceptual noise substitution. (8)

This dilemma is typically solved by means of mode switching. Simply put, with each block the coder can choose from two (or more) block sizes to favor either coding efficiency or pre-echo resilience.

3.2.1. Pre-echo Distortion

Pre-echo typically occurs within blocks whose energy is not (at least roughly) equally distributed across the block. After MDCT, quantization and IMDCT, the quantization error will be distributed equally across the block, making a surge partially appear in the reconstructed waveform a few milliseconds before it should. This phenomenon is referred to as pre-echo.

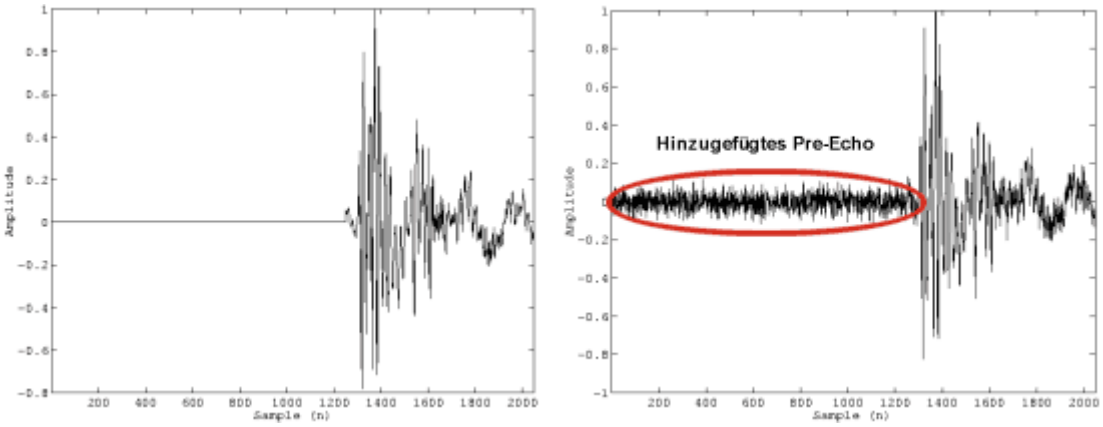


Figure 10 Pre echo distortion on long blocks (15)
Left: original signal
Right: original signal after MDCT, quantization, IMDCT

Dividing the block into several shorter blocks will help isolate the quantization noise into smaller blocks, reducing the pre-echo both in delay and magnitude – thus making it less audible. (15)

3.2.2. Practical Mode Switching

One problem with mode switching is that we cannot put a short block directly after a long block or vice versa. This would violate [22] (Princen-Bradley condition) and thus the MDCT would not provide perfect reconstruction. This issue can be solved by using *transitional windows* or *boundary filters*. Most audio coders utilize transitional windows. (8)

When using transitional windows, we are trying to satisfy the Princen-Bradley [22] at all times. This is accomplished simply by starting the transitional window as if it were of previous block type and finishing it as the next block type.

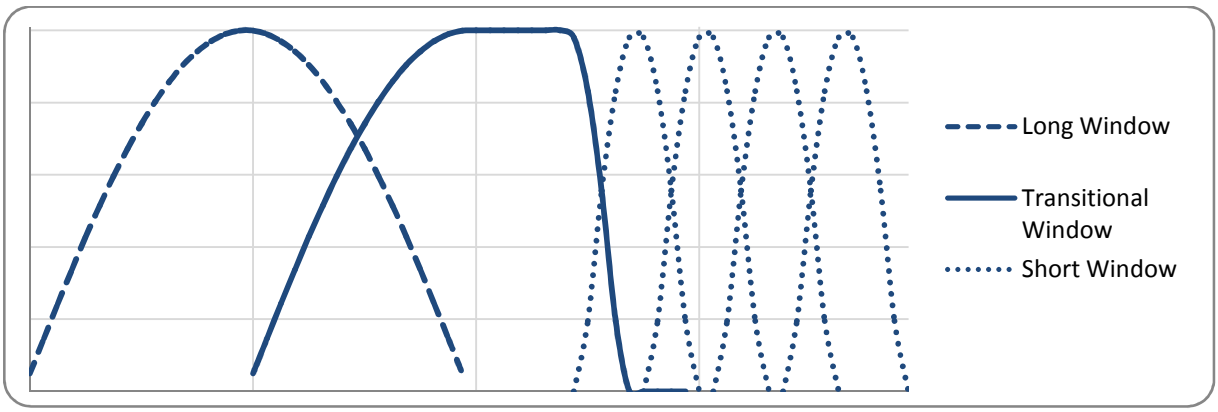


Figure 11 Long Block Followed By Short Blocks (Vorbis I) (13)

For transform coders, mode switching is typically implemented as switching the transform length and window shape. For the most part, these coders have 2 window lengths.

MPEG2 AAC has 4 modes (12):

- One long window (2048 samples)
- One start transitional window (2048 samples)
- A sequence of eight short windows (8x 256 samples)
- One end transitional window (2048 samples)

Each of those modes can use either sine window or Kaiser-Bessel Derived window.

Some coders implement more transform lengths e.g. Windows Media Audio 9 can choose from 5 different window lengths (128, 256, 512, 1024, 2048 samples). (16)

MPEG1 Layer 3 does not change the resolution of its 32-band polyphase filterbank, rather the block length of the following MDCTs can be switched between 12 and 36 samples, which would correspond to transform lengths of 1152 and 384 samples (11).

3.3. Stereo Representation

Many coders support multichannel (at least stereo) coding. Stereo coding presents more options for data reduction, however requires additional care. Studies of the human auditory system indicate that at frequencies above 2 kHz we localize sounds based on the spatial envelope, rather than specific temporal details like phase shifts. (1)

Coding stereo may have an effect on masking. Two channels may interfere with each other, creating unwanted stereo unmasking effects. Coding two channels independently may cause the quantization noise to be audibly unmasked because it does not match the spatial location of masker signal. Furthermore, the masking threshold may be lower when listening with two ears, rather than one (which is usually the case). This is called Binaural Masking Level Difference (BMLD). To make matters more complicated, phase differences between the masker and maskee at two ears can be audible at low frequencies (below 500 Hz). Audio coders must be aware of such effects. (1)

Stereo and multichannel signals usually contain a great deal of redundancies. This can be exploited to increase the compression ratio. The data rate of a dual-channel signal is twice the rate of a single-

channel signal. Joint-stereo coding schemes remove redundancies and thus are more efficient. A joint-stereo coder at 128 kbps will outperform two independent 64 kbps single channel coders. There are various multichannel coding schemes; however Mid-Side stereo and Intensity stereo are the most common. (1)

3.3.1. Mid-Side Stereo

This scheme represents dual channel signal as sum and difference channels, rather than left and right. This scheme has the advantage of concentrating information common to both channels in one channel only. (15) This representation also decreases the possibility of stereo unmasking effects (1).

Standard Left/Right representation is easily converted to Mid/Side (17):

$$\begin{pmatrix} M_i \\ S_i \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} L_i \\ R_i \end{pmatrix} \quad [23]$$

Where L_i, R_i, M_i, S_i are i-th samples of left, right, mid and side channels. Original left and right channels are easily calculated (17):

$$\begin{pmatrix} L_i \\ R_i \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} M_i \\ S_i \end{pmatrix} \quad [24]$$

3.3.2. Intensity Stereo

Intensity stereo is a little more complicated. It codes the energy envelope of the signal, rather than the whole waveform. This is effective for coding spatial information of the signal, which may degrade some of the phase relationships that could be represented by mid-side stereo. In effect, the stereo image may be slightly distorted, however the heavy increase in coding efficiency resulted by the use of Intensity Stereo makes it very useful for coding multichannel audio at lower bitrates. (18)

MPEG-4 HE-AAC v2 uses a modification of Intensity stereo (called Parametric stereo) to store spatial information of the audio signal. (19)

3.3.3. Multichannel Audio

To convey multichannel information, the high frequencies in each channel can be divided into bands and combined band by band into a composite channel; the bands of the common channel are reproduced from each speaker, panned between the speakers according to the spatial envelope for

each band. Use of the composite channel improves data compaction and enables us to use some psychoacoustic principles on the composite channel. (1) This is similar to intensity stereo.

Very generally, the number of bits required to code a multichannel signal is proportional to the square root of the number of channels. Theoretically, a 7.1 codec would require 3 times the bit rate needed to code a single channel. (1)

3.4. Psychoacoustic Model

The psychoacoustic model (or Psy-model) is the part that determines how the signal can be deprived of some detail so that it can be better compressed with an entropy coder while maintaining its perceptual quality. This is achieved by using masking thresholds, perceptual entropy, block mode decision logic and some other methods.

It is safe to say, that the common part of every psychoacoustic analyzer is the Fast Fourier Transform (FFT). It is used to calculate masking thresholds, perceptual entropy, tonality of critical bands and for numerous other purposes. It is important to note, that the FFT should have at least the resolution of the time-to-frequency domain filterbank and is usually windowed (Hann window for MPEG1 Model 1 and 2 (8)).

3.4.1. Masking Threshold

Typically, the first thing a perceptual model does after calculating the FFT of a block of input signal is calculating the masking threshold. This is done in a few simple steps. First, the result of the FFT is divided into critical bands (bandwidth of a critical band may be lower than 1 Bark – 1/3 Bark for LAME). In each critical band, we identify the masker by selecting the spectral line with highest energy.

Then, we calculate the tonality index for each of the found maskers. This is done by calculating its spectral flatness measure (SFM). (8) The equation is simple

$$SFM = \frac{\mu_g}{\mu_a} \qquad [25]$$

Where μ_a and μ_g are the arithmetic and geometric means of energies of the analyzed critical band. The SFM is bounded between zero and unity. Higher values will occur if the spectrum in that particular band is flat (noise-like). On the other hand, values close to zero occur if the spectrum is peaky (tonal).

Now, we have identified a rather large number of maskers. In the next step we will eliminate some of them. We accept only the maskers whose power satisfies (11)

$$P(k) \geq T_q(k) \quad [26]$$

In other words, we discard the maskers that are below the audibility threshold themselves. Next we use a sliding window with a width of 0.5 Bark to eliminate the weaker of two maskers that occur within that window.

After that, we calculate masking thresholds within critical bands. We apply here the knowledge of Noise-Masking-Tone and Tone-Masking-Noise. The power of tonal maskers is reduced by about 24 dB (varies with every psy-model). Noise-like maskers are reduced only about 4 dB. (8)

Finally, a spreading function is used to calculate the inter-band masking threshold.

3.5. Quantizer

Quantizer is the component that performs bit and noise allocation. By allocating more bits to a spectral line, we are actually allocating less noise to it. It is the job of the quantizer to allocate the maximum possible unnoticeable noise to every spectral line (or subband). Noise allocation is regulated by the output of the psychoacoustic model – the masking threshold. (9) There are many strategies regulate the bitrate, the most common are (20):

- Constant Bitrate (CBR) – the bitrate will be the same for every frame, this is useful for transferring audio over networks, where it is useful to know the bitrate exactly
- Variable Bitrate (VBR) – this mode chooses higher bitrate for complex parts and lower bitrate for simpler parts. The user selects target quality rather than bitrate.
- Average Bitrate (ABR) – this is a combination of the last two modes. The bitrate varies according to the signal's complexity, however the final file size remains predictable. This mode generally gives better results than CBR.

The algorithm for noise allocation is very dependent on the bitrate regulation strategy the user has selected. These algorithms generally differ not only among different coders but also among their different implementations.

To enhance the perceptual quality of the quantized signal, a method called Temporal Noise Shaping (TNS) may be used. As opposed to simple quantization, it not only minimizes the quantization error, but also shapes the quantization noise to match the original signal in temporal domain. This allows the coder to use its long blocks more often without introducing perceptible quantization artifacts. TNS is used in MPEG-2 AAC (1).

3.6. Entropy Coder

Entropy coder is one of the lossless components of a perceptual coder. It reduces the redundancy of the quantized spectrum other data contained in the data stream. No additional noise is introduced to the signal; this is why it is also called noiseless coding.

Most coders like MP3 (11), MPEG-2 AAC (12), Vorbis (13) and many others use Huffman coding as their entropy coder. Huffman coding uses a table of frequencies of occurrences for every symbol in the coded vector and assigns short binary codes to the symbols with high frequencies. On the other hand, long codes represent rarely occurring symbols. These codes are unique and distinguishable from one another when concatenated (without separators). These codes are generated using Huffman trees. Depending on the entropy of the coded vector, significant savings can be achieved using Huffman coding (21).

The function of a quantizer is to selectively reduce precision of the coded vector. In effect, some values occur more often, while others do not occur at all. This increases the effectiveness of the entropy coder.

4. Comparing Modern Codecs

This section lists and compares the technical details of some of the most popular modern audio coding systems.

4.1. MPEG-1 Layer 3 (MP3)

This format is probably the most popular and very widely used, even though its efficiency is often more than matched by other coders. This particular coder has been standardized in November 1992, which makes it one of the oldest perceptual coders described in this chapter. (11) MPEG-1 audio uses a layered approach in compressing audio. There are up to 3 layers of compression present in each MPEG-1 coder. With each layer, compression efficiency, complexity and latency are increased. (8)

Layer I transforms the signal using a PQMF into 32 equidistant (by frequency) bands. So, if the sampling rate is 48 kHz, each of the bands has a width of 750 Hz. After the transformation, the subbands are decimated by a factor of 32 to maintain critical sampling. Blocks of 12 samples in each subband are then block-companded (multiplied by a scale factor so that the maximum sample value in a block is 1.0). A simple psychoacoustic analysis is performed on a block of 512 samples (Psy-model I). Then, for each subband a proper quantizer is selected (using the output of the psy-model), so that bit-rate is matched and masking is exploited to the maximum possible extent. (8)

Layer II improves some of the parts of layer I to enhance compression efficiency at the price of increased complexity. The first enhancement is that its perceptual model uses a FFT with a resolution of 1024 samples (instead of 512). The other enhancements are made in the quantizer loop (reusing the same scale factor for adjacent blocks of 12 samples). (8)

Finally MPEG-1 Layer III is the most complex of all 3 layers; however it provides the best compression. The most notable enhancement is the introduction of a hybrid filterbank to enhance the frequency resolution of the original PQMF. The hybrid filterbank is comprised of the original PQMF followed by a MDCT. The resolution of the MDCT can be dynamically switched between 18 and 6 points. This improves the filterbank resolution from 32 to 576 spectral lines (750 to 41.6 Hz per spectral line). Mode switching allows pre-echo control. Short blocks provide better temporal masking of pre-echo, while longer blocks improve coding efficiency on steady signals. To maintain

Princen-Bradley condition, transitional windows are used. (8) Bit allocation is performed inside a quantization loop. To further enhance compression ratio, a Huffman coder is added to the bitstream formatter. There are a total of 32 pre-defined Huffman tables. The 576 spectral lines are considered as 3 groups and each of those groups can be coded using a different Huffman table (1). Moreover, layer III can use the more sophisticated psy-model II. (8)

MPEG-1 Layer III is capable of coding mono and stereo signals sampled at rates of 32, 44.1 and 48 kHz (11). This coder generally achieves acceptable quality (CD-transparent) at bitrates of 128 kbps. (8)

4.2. MPEG-4 Advanced Audio Coding (AAC)

In contrast to MP3, MPEG-4 AAC is one of the most recent coders available. It also provides superior compression ratios, sampling rates and more channel configurations. In an effort to satisfy the needs of various applications, the creators of AAC decided to use the coder as a basis and create additional modules or extensions to it that adjust the coder to match the needs of a particular application. Currently, the following flavors of MPEG-4 AAC are available (22):

- Low Complexity AAC (LC-AAC)
- High Efficiency AAC (HE-AAC)
- Low Delay AAC (LD-AAC)

LC-AAC reduces the complexity of the coder by omitting specific features of standard AAC, which makes it useful for applications, where computing power comes at a premium – such as cell phones or portable music players. The most interesting extension however, is the High Efficiency AAC (HE-AAC), which provides the most features and a very impressive coding efficiency (22).

MPEG-4 AAC is an extension of MPEG-2 AAC. So, like its predecessor, it is a transform coder. As a filterbank, the MDCT is used. Block sizes of 2048 and 256 samples are available, however as opposed to other coders, this one can switch its windows shapes between sine window and KBD window. Transitional windows are used to implement mode switching seamlessly (12). Furthermore, multichannel configurations of up to 48 channels are available at sampling rates beginning as low as 8 kHz and ending at 96 kHz. The technologies of MPEG-4 AAC include (22):

- Intensity and Mid/Side stereo representations

- Temporal Noise Shaping – alleviates the artifacts introduced by relatively poor temporal resolution of the coder's filterbank
- Spectral Band Replication (SBR) – available only with HE-AAC or HE-AAC v2 – based on the observation that higher frequencies have only a minor contribution to the "perceptual information", the coder transmits only the lower part of the spectrum plus a small information about the frequency envelope of the upper band (2-3 kbps). The decoder than partially reconstructs the upper band. This greatly increases the coding efficiency (22).
- Parametric Stereo (PS) – available only with HE-AAC v2 – is an extension of the intensity stereo. While intensity stereo can only reproduce intensity levels of both channels, PS can also reproduce certain phase relationships. This improves the quality of perceived stereo image, while transmitting only little data – typically only few kbps (22).

It must be noted that there are constraints on the number of channels stored in a stream and the maximum sampling rate when SBR is used. MPEG-4 HE-AAC can transmit 5.1-channel audio in CD-transparent quality at bitrates of 160 kbps (22). HE-AAC v2 has the potential to be used in applications, where extreme bandwidth constraints exist and a reasonable perceptual quality is required – like digital radio. In Digital Radio Mondiale (DRM), MPEG-4 HE-AAC v2 is used to deliver near-CD quality at bitrates near 30 kbps. (23)

4.3. Dolby Digital (AC-3)

AC-3 is most often used to store audio tracks for video. It is typically packed into a MPEG-2 stream together with video. It is its ability to represent audio in configurations up to 5.1 which makes this coder suitable for storing audio tracks for video.

The coder can represent audio at sampling rates of 32, 44.1 and 48 kHz. Constant bitrates from 32 to 640 kbps are possible. Its bitstream can store a variety of information associated with the audio stream including a CRC16 for data integrity control. The following channel configurations are available (24):

- Mono
- Stereo (Left/Right and Mid/Side configurations)
- Left + Right + Center
- Left + Right + Surround channel
- Left + Right + Center + Surround

- Left + Right + Rear left + Rear right
- Left + Right + Center + Rear left + Rear right

With all these channel configurations, there is the option of using the low-frequency enhancement (LFE) channel. The LFE channel is normally routed to the subwoofer (if available), or it can be mixed into all other channels. (24)

The coder employs mode switching to reduce pre-echo artifacts; however it does not use transitional windows to maintain perfect reconstruction as most other coders do. It is also peculiar that the coder uses such short blocks (only 512 and 256 samples) considering that this does not reduce the coder's latency (6 such blocks are packed inside a frame) (24) and other coders are able to effectively cope with pre-echo despite the fact that they have much larger blocks.

4.4. Vorbis I

Vorbis I is an open-source coder. In terms of coding efficiency it is competing with MPEG2-AAC or WMA 9. A wide range of sampling rates (8 – 192 kHz). Many channel configurations including mono, stereo, 5.1 are supported. In fact, up to 255 distinct channels can be stored in one stream. (13)

Vorbis is a transform coder. MDCT is used as the time-to-frequency domain conversion filterbank. Block sizes of 64 up to 8192 samples are supported. The window shape is not sine, as it is with most other coders. Vorbis rather uses its own window function. Another peculiarity of Vorbis is that the transformed samples are represented as a combination of a coarse curve (floor) and a residue. Other coders generally use exponents with groups of mantissas or spectral pairs. Vorbis represents stereo in polar coordinates, this is similar to intensity stereo, however it is able to convey phase relationships between the channels. Codebooks are used to store Huffman tables in the stream – these tables are not pre-defined as with other coders. (13)

Vorbis uses a different approach to many coding methods than other coders (window shape, spectrum representation, stereo representation …). This creativity has paid off as the coder can deliver CD transparent quality at bitrates near 96 kbps.

4.5. Windows Media Audio 9 (WMA9)

Many of the details on the internals of WMA9 remain obscured as this standard is closely guarded by Microsoft corp. As there are many applications which have different demands on the audio coder, there are several different flavors of WMA (although they all share the name WMA, they use different algorithms).

4.5.1. WMA9

This is the basic version of the Windows Media Audio 9 audio coder. It can sample the audio at rates of 44.1 and 48 kHz with a precision of 16 bits per sample. It is claimed by (25) that this codec can achieve CD transparent quality at bitrates from 64 kbps – about half of that of MP3. It can represent mono and stereo signals at low bitrates (below 32 kbps).

4.5.2. WMA10 Professional

This is the high-performance, high-fidelity version of the coder. It is superior to WMA9 in terms of coding efficiency, features and scalability. It can encode audio at sampling rates up to 96 kHz (24 bits per sample) with channel configurations up to 7.1. Supported features include dynamic range compression and frequency interpolation mode (a sort of spectral band replication) (26). It is claimed that this coder can be twice as strong as AC-3 at 48 kHz sampling rate in coding 6-channel audio (25).

The older version of this coder (WMA9 Professional) does not include frequency interpolation mode.

4.5.3. WMA9 Lossless

Some applications like archiving require that the stored audio be compressed while leaving the audio intact (bit-precise). WMA9 Lossless was designed to do just that. Sampling rates up to 96 kHz are available with precisions of up to 24 bits per sample. Supported channel configurations range to 5.1. (25)

4.5.4. WMA9 Voice

This codec is intended for speech content. It generally performs better with voice than other coders. It exploits the lower complexity of human voice (compared to signals like music) and its narrow frequency range. The coder can decide whether the signal is voice-like or not. If the signal is too complex to be considered voice, it works just like normal WMA9. (25)

WMA9 Voice can store voice at bitrates as low as 4 kbps (at a sampling rate of 8 kHz). (25)

4.6. Summary

The following table provides an overview of the coders described in this chapter. Specific technical details about WMA are unavailable due to licensing and specifications confidentiality.

	MP3	MPEG-4 HE-AAC	AC-3	Vorbis I	WMA 9
Typical container	MPEG	MPEG	MPEG-2	Ogg	ASF
Sampling rates available (kHz)	32; 44.1; 48	8 – 96	32; 44.1; 48	8 - 192	44.1; 48
Maximum channels	2	48	5 + LFE	255	8
Channel configurations	Left/Right; Mid/Side	many	Up to 5.1	many	
CD transparency bitrate (approx.)	128 kbps	64 kbps	160 kbps	96 kbps	64 – 96 kbps
Maximum bitrate (constant)	320 kbps		640 kbps	500 kbps	384 kbps
Filterbank type	Hybrid (PF+MDCT)	MDCT	MDCT	MDCT	MDCT
Block sizes	384; 1152	256; 2048	512; 256	64 - 8192	128; 256; 512; 1024; 2048
Window shapes	Sine	Sine / KBD	KBD	Vorbis	
Mode switching implementation	transitional windows	transitional windows	boundary filters	transitional windows	
PNS	No	Yes	No	No	Yes (?)
SBR	No	Yes	No	no	Yes

Table 3 Features of Selected Current Audio Coders

References

1. **Pohlmann, Ken C.** *Principles of Digital Audio.* s.l. : McGraw-Hill Professional, 2005. p. 842. ISBN 0071441565.

2. Fourier theorem. *Wikipedia, the free wncyclopedia.* [Online] [Cited: March 11, 2009.] http://en.wikipedia.org/wiki/Fourier_Theorem.

3. **Adam, Pavol.** Princípy bezstratovej kompresie zvuku. *Úvod do metód spracovania zvuku v súčasnom multimediálnom prostredí.* [Online] http://zvuk.atrip.sk/index.php?site=3_4.

4. **Lavry, Dan.** Sampling Theory. [Online] http://www.lavryengineering.com/documents/Sampling_Theory.pdf.

5. **Marina Bosi, Richard E. Goldberg.** *Introduction to Digital Audio Coding and Standards.* s.l. : Springer, 2003. p. 434. ISBN 1402073577.

6. **Cisco Systems.** Waveform Coding. [Online] http://www.cisco.com/application/pdf/paws/8123/waveform_coding.pdf.

7. **Coalson, Josh.** Format. *FLAC.* [Online] [Cited: March 11, 2009.] http://flac.sourceforge.net/format.html.

8. **Ted Painter, Andreas Spanias.** Perceptual Coding of Digital Audio. *Proceedings of the IEEE.* 2000, Vol. 80, 4.

9. **Andreas Spanias, Ted Painter, Venkatraman Atti.** *Audio Signal Processing and Coding.* s.l. : Wiley-Interscience, 2007. ISBN 047004196X.

10. **Chi-Min Liu, Wen-Chieh Lee.** A Unified fast Algorithm for Cosine Modulated Filter Banks in Current Audio Coding Standards. [Online] [Cited: March 25, 2009.] http://wenchiehlee1020.googlepages.com/AES-paper-vol.47.PDF.

11. **Jacaba, Joebert S.** *Audio Compression Using Modified Discrete Cosine Transform: The MP3 Coding Standard.* Diliman, Quezon City : s.n., 2001.

12. Information technology - Generic coding of moving pictures and associated audio information - Part 7: Advanced Audio Coding (AAC). October 15, 2004. ISO/IEC 13818-7.

13. **Xiph.org Foundation.** *Vorbis I Specification.* 2004.

14. **Harris, Frederic J.** On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. *Proceedings of the IEEE.* 1978, Vol. 66, 1.

15. **Kappes, André.** Die Audiokodierung mp3. *Proseminar Redundanz, Fehlertoleranz und Kompression.* [Online] [Cited: May 15, 2009.] http://goethe.ira.uka.de/seminare/rftk/mp3/.

16. Windows Media Audio. *Wikipedia.* [Online] [Cited: May 15, 2009.] http://en.wikipedia.org/wiki/Windows_Media_Audio.

17. Joint stereo. *Hydrogenaudio Knowledgebase.* [Online] [Cited: May 16, 2009.] http://wiki.hydrogenaudio.org/index.php?title=Intensity_stereo.

18. **Coding Technologies.** Parametric Stereo. [Online] Coding Technologies, 2008. http://www.codingtechnologies.com/products/paraSter.htm.

19. **Fraunhofer IIS.** The MPEG AAC Family. [Online] http://www.iis.fraunhofer.de/fhg/images/mpeg_aac_family_v0808_02092008_en_tcm278-67331.pdf.

20. Encoding modes. *LAME Documentation.* [Online] [Cited: May 17, 2009.] http://lame.cvs.sourceforge.net/*checkout*/lame/lame/doc/html/modes.html.

21. **Thomas H. Cormen, Charles E. Leierson, Ronald L. Rivest, Clifford Stein.** *Introduction to Algorithms.* s.l. : MIT Press, 2001. ISBN 9780262032933.

22. **Jürgen Herre, Martin Dietz.** MPEG-4 High Efficiency AAC Coding. *IEEE Signal Processing Magazine.* 2008, May.

23. **EBU-UER.** Technical Bases for DRM Services Coverage Planning. [Online] [Cited: May 16, 2009.] http://www.drm.org/fileadmin/media/downloads/June_2008_EBU_Technical_Bases_for_DRM_services_coverage_planning.pdf.

24. **Advanced Television Systems Committee, Inc.** Digital Audio Compression Standard (AC-3, E-AC-3) Revision B. *ATSC STANDARDS.* [Online] June 14, 2005. [Cited: April 24, 2009.] http://www.atsc.org/standards/a_52b.pdf.

25. **Microsoft Corp.** Windows Media Audio Codecs. [Online] Microsoft Corp. [Cited: April 24, 2009.] http://www.microsoft.com/windows/windowsmedia/forpros/codecs/audio.aspx.

26. **Waggoner, Ben.** Best Practices for Windows Media Encoding. *Streamingmedia.com.* [Online] [Cited: April 24, 2009.] http://www.streamingmedia.com/article.asp?id=9510&page=2&c=4.