Introduction to Digital Audio Compression

B. Cavagnolo and J. Bier
Berkeley Design Technology, Inc.
2107 Dwight Way, Second Floor
Berkeley, CA 94704
(510) 665-1600
info@bdti.com http://www.bdti.com

INTRODUCTION

Advances in digital audio technology are fueled by two sources: hardware developments and new signal processing techniques. When processors dissipated tens of watts of power and memory densities were on the order of kilobits per square inch, portable playback devices like an MP3 player were not possible. Now, however, power dissipation, memory densities, and processor speeds have improved by several orders of magnitude. Advancements in signal processing are exemplified by Internet broadcast applications: if the desired sound quality for an internet broadcast used 16-bit PCM encoding at 44.1 kHz, such an application would require a 1.4 Mbps channel for a stereo signal! Fortunately, new bit-rate-reduction techniques in signal processing for audio of this quality are constantly being released.

Increasing hardware efficiency and an expanding array of digital audio representation formats are giving rise to a wide variety of new digital audio applications. These applications include portable music playback devices, digital surround sound for cinema, high-quality digital radio and television broadcast, Digital Versatile Disc (DVD), and many others.

This paper introduces digital audio signal compression, a technique essential to the implementation of many digital audio applications. Digital audio signal compression is the removal of redundant or otherwise irrelevant information from a digital audio signal—a process that is useful for conserving both transmission bandwidth and storage space. We begin by defining some useful terminology. We then present a typical "encoder" (as compression algorithms are often called), and explain how it functions. Finally we consider some standards that employ digital audio signal compression, and discuss the future of the field.

TERMINOLOGY

Audio Compression vs. Speech Compression

This paper focuses on audio compression techniques, which differ from those used in speech compression. Speech compression uses a model of the human vocal tract to express particular signals in a compressed format. This technique is not usually applied in the field of audio compression due to the vast array of sounds that can be generated—

models that represent audio generation would be too complex to implement. So instead of modeling the source of sounds, modern audio compression models the receiver, i.e., the human ear.

Lossless vs. Lossy

When we speak of compression, we must distinguish between two different types: lossless and lossy. Lossless compression retains all the information in a given signal, i.e., a decoder can perfectly reconstruct a compressed signal. In contrast, lossy compression eliminates information from the original signal. As a result, a reconstructed signal may differ from the original. With audio signals, the differences between the original and reconstructed signals only matter if they are detectable by the human ear. As we will explore shortly, audio compression employs both lossy and lossless techniques.

BASIC BUILDING BLOCKS

Figure 1 shows a generic encoder or "compressor" that takes blocks of a sampled audio signal as its input. These blocks typically consist of between 500 and 1500 samples per channel, depending on the encoder specification. For example, the MPEG-1 layer III (MP3) specification takes 576 samples per channel per input block. The output is a compressed representation of the input block (a "frame") that can be transmitted or stored for subsequent decoding.

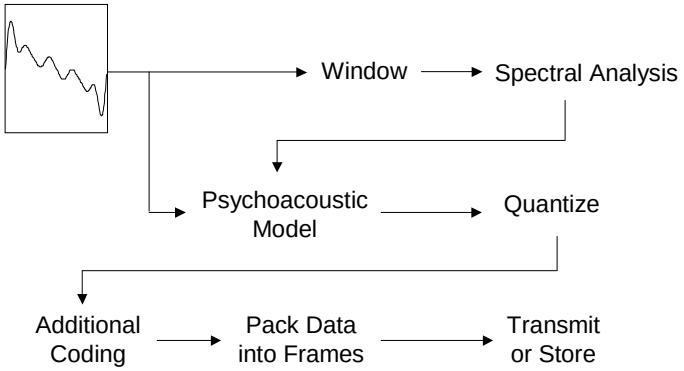


Figure 1. Generic Encoder.

Psychoacoustics

How do we reduce the size of the input data? The basic idea is to eliminate information that is inaudible to the ear. This type of compression is often referred to as perceptual encoding. To help determine what can and cannot be heard, compression algorithms rely on the field of psychoacoustics, i.e., the study of human sound perception. Specifically, audio compression algorithms exploit the conditions under which signal characteristics obscure or mask each other. This phenomenon occurs in three different

ways: threshold cut-off, frequency masking, and temporal masking. The remainder of this section explains the nature of these concepts; subsequent sections explain how they are typically applied to audio signal compression.

Threshold Cut-off

The human ear detects sounds as a local variation in air pressure measured as the Sound Pressure Level (SPL). If variations in the SPL are below a certain threshold in amplitude, the ear cannot detect them. This threshold, shown in Figure 2, is a function of the sound's frequency. Notice in Figure 2 that because the lowest-frequency component is below the threshold, it will not be heard.

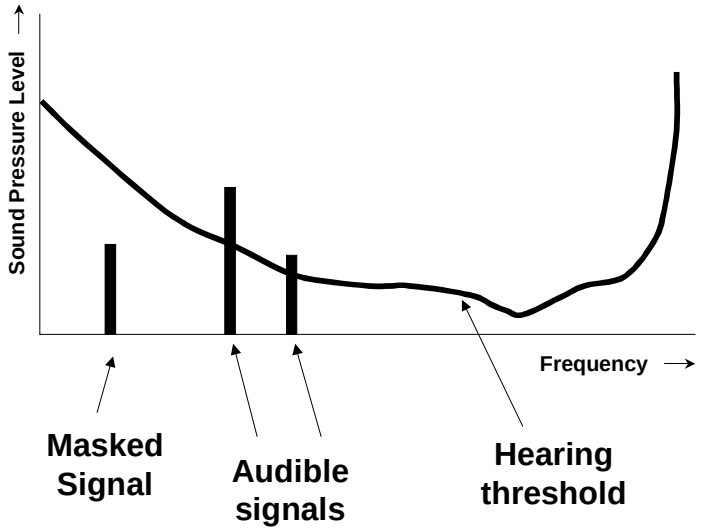


Figure 2. Hearing threshold up to about 20 kHz.

Frequency Masking

Even if a signal component exceeds the hearing threshold, it may still be masked by louder components that are near it in frequency. This phenomenon is known as frequency masking or simultaneous masking. Each component in a signal can cast a "shadow" over neighboring components. If the neighboring components are covered by this shadow, they will not be heard. The effective result is that one component, the masker, shifts the hearing threshold. Figure 3 shows a situation in which this occurs.

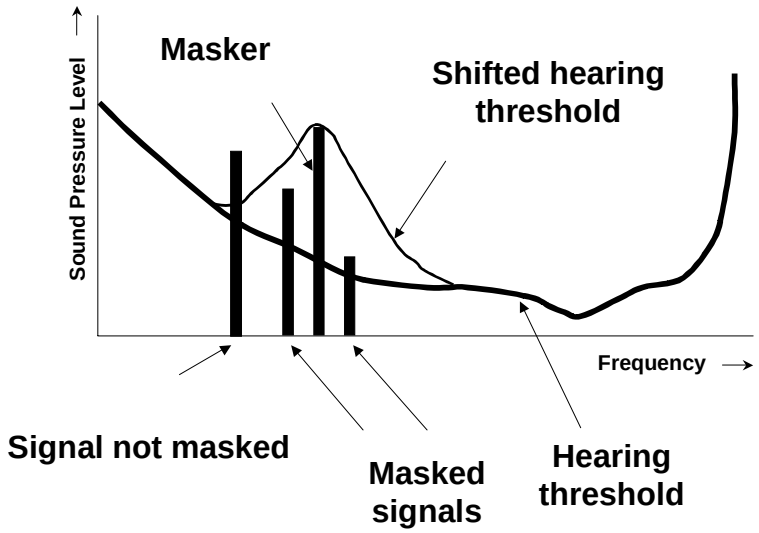
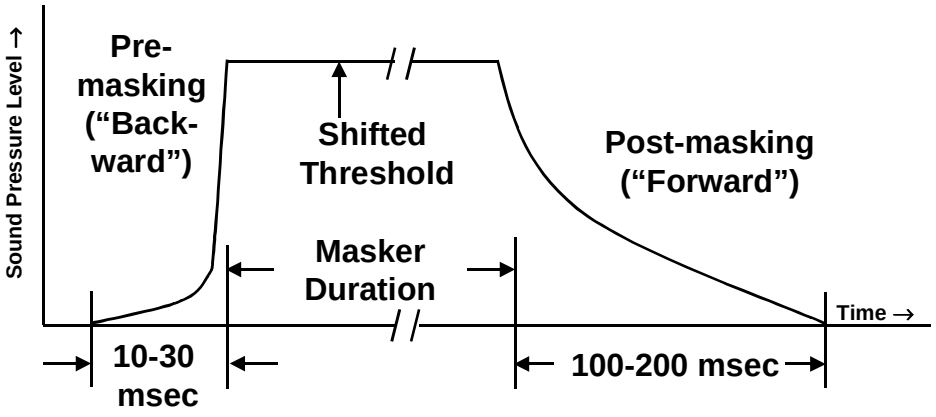


Figure 3. Frequency Masking.

Temporal Masking

Just as tones cast shadows on their neighbors in the frequency domain, a sudden increase in volume can mask quieter sounds that are temporally close. This phenomenon is known as temporal masking. Interestingly, sounds that occur both after *and* before the volume increase can be masked! Figure 4 illustrates a typical temporal masking scenario: events below the indicated threshold will not be heard. Note that the premasking interval is much shorter than the post-masking interval.



After Zwicker/Fastl p. 78, Buser/Imbert p. 47

Figure 4. Temporal Masking.

Spectral Analysis

Of the three masking phenomena explained above, two are best described in the frequency domain. Thus, a frequency domain representation, also called the "spectrum" of a signal, is a useful tool for analyzing the signal's frequency characteristics and determining thresholds. There are several different techniques for converting a finite time sequence into its spectral representation, and these typically fall into one of two categories: transforms and filter banks. Transforms calculate the spectrum of their inputs in terms of a set of basis sequences; e.g., the Fourier Transform uses basis sequences that are complex exponentials. Filter banks apply several different bandpass filters to the input. Typically the result is several time sequences, each of which correspond to a particular frequency band. Before we discuss these structures, note that taking the spectrum of a signal has two purposes: to derive the masking thresholds in order to determine which portion of the signal can be dropped, and to generate a representation of the signal to which the masking threshold can be applied. Some compression schemes use different techniques for these two tasks.

The most popular transform in signal processing is the Fast Fourier Transform (FFT). Given a finite time sequence, the FFT produces a complex-valued frequency domain representation. Encoders often use FFTs as a first step toward determining masking thresholds. Another popular transform is the Discrete Cosine Transform (DCT), which outputs a real-valued frequency domain representation. Both the FFT and the DCT suffer from distortion when transforms are taken from contiguous blocks of time data. To solve this problem, inputs and outputs can be overlapped and windowed in such a way that, in the absence of lossy compression techniques, entire time signals can be perfectly reconstructed. For this reason, most transform-based encoding schemes employ an overlapped and windowed DCT known as the *Modified* Discrete Cosine Transform (MDCT). Some compression algorithms that use the MDCT are MPEG-1 layer-III, MPEG-2 AAC, and Dolby AC-3. A more thorough discussion of these compression standards appears in the next section.

Filter banks pass a block of time samples through several bandpass filters to generate different signals corresponding to different subbands in frequency. After filtering, masking thresholds can be applied to each subband. Two popular filter bank structures are the poly-phase filter bank and the wavelet filter bank. The poly-phase filter bank uses parallel band-pass filters of equal width whose outputs are downsampled to create one (shorter) signal per subband. In the absence of lossy compression techniques, a decoder can achieve perfect reconstruction by upsampling, filtering, and adding each subband. This type of structure is used in all of the MPEG-1 audio encoders.

A wavelet filter bank is a filter bank whose subbands are not evenly spaced. As frequency increases, the subbands get narrower and narrower. This feature is

advantageous because it offers better time resolution and is able to preserve brief "attacks" (such as cymbal crashes) in the input more efficiently. However, this increased time resolution comes at the expense of frequency resolution, which makes the wavelet filter bank less efficient at encoding steady signals. One scheme that uses a wavelet filter bank is Lucent's Enhanced Perceptual Audio Coder (EPAC).

All of these analysis tools take finite blocks of long audio streams as their inputs. This introduces some problems, such as block-edge distortion (an audible artifact of multiplying a time signal by a square window). As mentioned above, this problem is resolved by windowing and overlapping the input blocks. Another problem is the appearance of pre-echoes during reconstruction. Pre-echoes appear within a block when a high amplitude attack interrupts a relatively low amplitude signal, as in Figure 5. (The causes of pre-echoes are discussed in the next section.) The most common solution to this problem is to detect attacks in the encoder and switch to a smaller input block before an attack occurs. Using this technique, the duration of the pre-echo can be minimized to the point that the pre-masking effect of the attack will obscure it.

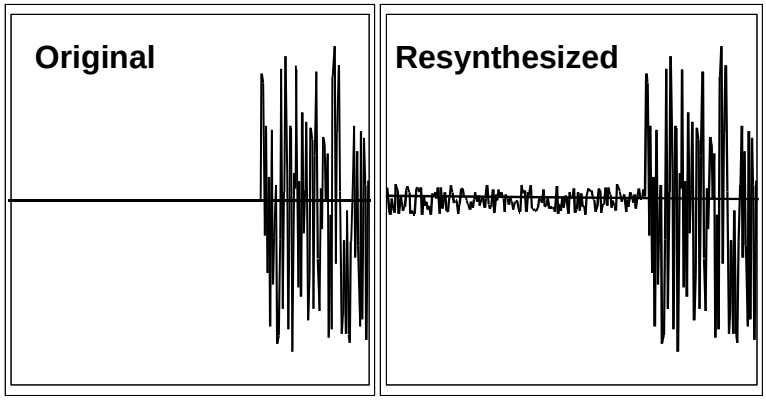


Figure 5. Attack in original signal causes a pre-echo in reconstructed signal.

Noise Allocation

Let us return to our generic block diagram before we continue. By using spectral analysis, the time-domain signal has been represented using a frequency domain representation. The goal now is to compress the input signal by eliminating irrelevant information. This is done by deriving and applying a shifted hearing threshold based on the content of the signal.

Hearing thresholds indicate how strong a certain frequency component must be in order to be heard, based on the characteristics of neighboring components and properties of the ear. This has two useful implications. First, components that fall beneath this shifted threshold need not be transmitted. Second, any noise beneath the threshold is

irrelevant. Application of this second point is discussed in the next paragraph.

To encode a signal's representation, an encoder must *quantize* (assign a binary representation to) each frequency component. If the encoder allocates more bits to each frequency, less error (noise) is introduced, but more space is required to store the result. Conversely, fewer bits allocated to each frequency results in more noise, but less space is required to store the result. However, if noise is kept beneath the shifted hearing threshold, it will not be audible. In order to preserve storage space (transmission bandwidth), perceptual encoders use just enough bits to encode each frequency without introducing audible quantization noise. This technique, known as noise allocation, can be problematic when the input signal contains a brief attack preceded by silence. In this scenario, noise introduced in the frequency domain will be evenly spread throughout the reconstructed time block. Because the shifted hearing threshold is determined by both the loud and soft parts of the input, the silent segment of the reconstructed signal will often be audibly corrupted. This is called a pre-echo; the most common solution to this problem is to use short enough time blocks that the temporal masking phenomenon obscures the noise.

Other Tricks

Although most compression is achieved through the noise allocation technique mentioned above, several other techniques are available for further compression. These include prediction, coupling, and information-theory-based coding.

Prediction is a technique that is commonly used to encode speech signals and images. The basic idea is that one can estimate the value of a future sample based on examination of previous samples. Then, instead of transmitting the next sample, the encoder transmits the difference between the estimate and the actual sample, which can be transmitted using fewer bits than the sample itself. The decoder then generates an estimate based on the same prediction model, and adjusts its estimate by the difference received from the encoder. If the method of prediction is good, then the value to be transmitted will often be zero, resulting in substantial compression. MPEG-2 AAC uses prediction to achieve additional compression.

Coupling is used when several audio signals must be encoded or decoded in parallel, as with stereo or surround audio consisting of two or more channels. Differences among these channels are typically small. By combining or coupling these channels, an encoder can exploit their similarities. For example, an encoder could send the sum and difference of two channels; in the event that the signals are very similar, the difference will most often be near zero. Most compression algorithms make use of inter-channel similarities through some sort of lossless coding similar to the one described here.

Information-theory-based techniques like "Huffman" coding allocate fewer bits to

encode the most frequent signal values. For example, if 25% of an encoder's outputs are a single value, the encoder would benefit by using a very short codeword to represent this value. This technique is typically implemented with a look-up table in the encoder and a look-up table, binary tree, or combination of these in the decoder. Although this technique provides substantial (lossless) compression, it is not common because of the high computational requirement. MPEG-1 and -2 Layer-III and AAC apply a Huffman code as a final compression stage.

EXAMPLES AND APPLICATIONS

The purpose of this section is to discuss some existing standards in digital audio compression, in particular the MPEG-1 layer III, MPEG-2 AAC, and Dolby AC-3. Features of interest for each standard include which compression techniques are used, special details or unique characteristics, and target applications.

Before discussing MPEG algorithms, let us briefly discuss the MPEG naming convention. The Moving Pictures Experts Group (MPEG) has established many audio compression standards. When we speak of MPEG audio encoding schemes we always refer to both a *phase* and a *layer*. The phase is an Arabic numeral that indicates the MPEG release, e.g., MPEG-1 is the standard that was released in 1992. The layer is a Roman numeral between I and III; this refers to a particular decoder. Higher layers indicate a more complex decoder, e.g., MPEG-1 layer-III, known commonly as MP3, is the most powerful decoder specified by MPEG-1. (Note that MP3 does not stand for MPEG-3.)

Unless otherwise specified, MPEG standards are backward compatible, i.e., the decoders used by older standards can interpret data compressed by newer standards. This backward compatibility applies to both a decoder's phase and layer. Thus an MPEG-1 layer III can decode a backward compatible MPEG-2 signal, and any layer III decoder can read data from the simpler layer II. An exception is MPEG-2 Advanced Audio Coding (AAC): this is an extension to the MPEG-2 standard that was added in 1997 to improve compression for surround-sound signals.

MPEG-1 layer-III, with a minimum output rate of 112 kbps, has become the most popular format for Internet music. The first encoding stage combines a 32-subband poly-phase filter bank with an MDCT that transforms the filter bank's 32 output signals into the frequency domain. This combination is known as a hybrid filter bank because it employs both a filter bank and a transform. In the event of a time domain attack, the input block size is cut from 576 samples to 192, which helps avoid pre-echo artifacts. After the MDCT, masking thresholds are calculated and applied based on the spectral properties of the input signal. Stereo redundancies are then eliminated, and the resulting frequency components are Huffman encoded, formatted into a frame, and stored or transmitted. Because the decoder does not have to calculate masking thresholds, it is less complex

than the encoder. All the decoder must do is unpack the frame, decode the Huffman code, rescale the frequency lines, and apply the inverse filter bank. This simplified decoding process is desirable for applications such as portable music devices because it greatly reduces the cost and computing demands of real-time decoding.

MPEG-AAC compresses a 5.1-channel audio signal into a minimum of 320 kbps. "5.1 channel" refers to 5 full-bandwidth audio signals for surround and 1 reduced bandwidth low-frequency enhancement channel. At higher bit rates, AAC can support up to 48 channels for multi-lingual presentations or commentary. This standard employs three different *profiles* for different applications: "main," "low-complexity," and "sampling-rate-scaleable." All three profiles use an MDCT on input blocks of either 2048 or 256 samples. Prior to the transform, the sampling-rate-scaleable profile adds a preprocessor to control the input sampling rate. After the transform, the main and sampling-rate-scalable profiles apply a predictor (as discussed above), which is followed by noise allocation, channel coupling, Huffman encoding, and frame formatting.

Digital broadcasting is a major target application for AAC, which has been chosen as the encoding scheme for the DRM system, a worldwide initiative to standardize digital radio transmission, and also for several digital audio services in Japan. AAC is also a candidate for use in the In-Band On-Channel (IBOC) radio system in the United States. The IBOC system will transmit digital audio signals in the same band as analog radio signals.

Dolby AC-3 (Dolby Digital) is the audio compression format used in many movie theaters, home theaters, and in High-Definition Television (HDTV) in the United States. The AC-3 encodes a 5.1 channel audio signal into a 384 kbps bitstream. The first stage of an AC-3 encoder takes 512 input samples and applies an MDCT. In order to preserve dynamic range, it then splits the output frequency components into mantissas and exponents. These values are then quantized, coupled with other channels, and packed into frames for transmission.

CONCLUSIONS

By eliminating audio information that the human ear cannot detect, modern audio coding standards are able to compress a typical 1.4 Mbps signal by a factor of about twelve. This is done by employing several different methodologies, including noise allocation techniques based on psychoacoustic models.

Future goals for the field of audio compression are quite broad. Several initiatives are focused on establishing a format for digital encryption (watermarking) to protect copyrighted audio content. Improvements in psychoacoustic models are expected to drive bit rates lower. Finally, entirely new avenues are being explored in an effort to compress audio based on how it is produced rather than how it is perceived. This last approach was

integral in the development of the MPEG-4 standard.

FOR MORE INFORMATION

Madisetti, Vijay K., and Williams, Douglass B., eds. The Digital Signal Processing Handbook. Section IX. CRC Press LLC, 1998.

Massie, Dana, and Strawn, John, et. al., Digital Audio: Applications, Algorithms, and Implementation. Seminar presented at Embedded Processor Forum by Berkeley Design Technology, Inc., 12 June 2000.