This article was published when the author was with Motorola, Inc. As of October 7, 1996, Davis Pan will be working at the Cambridge Research Laboratory of Digital Equipment Corporation in Cambridge, Massachusetts.

A Tutorial on MPEG/Audio Compression

Davis Pan

Motorola Inc.
1301 East Algonquin Road,
Schaumburg, IL 60196

ABSTRACT

This tutorial covers the theory behind MPEG/audio compression. This algorithm was developed by the Motion Picture Experts Group (MPEG), as an International Organization for Standardization (ISO) standard for the high fidelity compression of digital audio. The MPEG/audio compression standard is one part of a multiple part standard that addresses the compression of video (11172-2), the compression of audio (11172-3), and the synchronization of the audio, video, and related data streams (11172-1) to an aggregate bit rate of about 1.5 Mbits/sec. The MPEG/audio standard also can be used for audio-only applications to compress high fidelity audio data at much lower bit rates.

While the MPEG/audio compression algorithm is lossy, often it can provide "transparent", perceptually lossless, compression even with compression factors of 6-to-1 or more. The algorithm works by exploiting the perceptual properties of the human auditory system. This paper also will cover the basics of psychoacoustic modeling and the methods used by the MPEG/audio algorithm to compress audio data with least perceptible degradation.

1. INTRODUCTION

This tutorial covers the theory behind MPEG/audio compression. It is written for people with a modest background in digital signal processing and does not assume prior experience in audio compression or psychoacoustics. Here the goal is to give a broad, preliminary understanding of MPEG/audio compression; many details have been omitted. Wherever possible, this tutorial uses figures and illustrative examples to present the intricacies of the algorithm.

The MPEG/audio compression algorithm is the first international standard[1,2] for the digital compression of high-fidelity audio. Other audio compression algorithms address speech-only applications or provide only medium-fidelity audio compression performance. For example, Code Excited Linear Prediction (CELP)[3] is a speech coding algorithm, while μ-law and Adaptive Differential Pulse Code Modulation (ADPCM) are relatively simple compression algorithms that can provide medium fidelity audio compression. To contrast the complexity of the MPEG/audio algorithm with that of some simpler, generic audio compression algorithms, the annex of this paper presents the details of μ-law and the ADPCM algorithm adopted by the Interactive Multimedia Association.

The MPEG/audio standard is the result of over 3 years of collaborative work by an international committee of high-fidelity audio compression experts known as the Motion Picture Experts Group (MPEG/audio). The International Organization for Standards and the International Electrotechnical Commission (ISO/IEC) adopted this standard at the end of 1992.

Although MPEG/audio compression is perfectly suitable for audio-only applications, it is actually one part of a three part compression standard. Combined with the other two parts, video and systems, the MPEG standard addresses the compression of synchronized video and audio at a total bit rate of about 1.5 Megabits/sec.

The MPEG standard is rigid only where necessary to ensure inter-operability. It mandates the syntax of the coded bitstream, defines the decoding process, and provides compliance tests for assessing the accuracy of the decoder[4]. This guarantees that, regardless of origin, any fully compliant MPEG/audio decoder will be able to decode any MPEG/audio bitstream with a predictable result. A wide acceptance of this standard will permit manufacturers to produce and sell, at reasonable cost, large numbers of MPEG/audio codecs.

Where possible, the standard is open to future innovative improvements. Designers are free to try new and different implementations of the encoder or decoder within the bounds of the standard. There is especially good potential for diversity in the encoder.

1.1 MPEG/audio Features and Applications

MPEG/audio is a generic audio compression standard. Unlike vocal-tract-model coders specially tuned for speech signals, the MPEG/audio coder gets its compression without making assumptions about the nature of the audio source. Instead, the coder exploits the perceptual limitations of the human auditory system. Much of the compression results from the removal of perceptually irrelevant parts of the audio signal. Removal of such parts results in inaudible distortions, thus MPEG/audio can compress any signal meant to be heard by the human ear. In keeping with its generic nature, MPEG/audio offers a diverse assortment of compression modes:

- The audio sampling rate can be 32, 44.1, or 48 kHz.

- The compressed bitstream can support one or two audio channels in one of 4 possible modes:

  1. a monophonic mode for a single audio channel,

  2. a dual-monophonic mode for two independent audio channels (this is functionally identical to the stereo mode),

  3. a stereo mode for stereo channels with a sharing of bits between the channels, but no joint-stereo coding, and

  4. a joint-stereo mode that either takes advantage of the correlations between the stereo channels or the irrelevancy of the phase difference between channels, or both.

- The compressed bitstream can have one of several predefined fixed bit rates ranging from 32 to 224 kbits/sec per channel. Depending on the audio sampling rate, this translates to compression factors ranging from 2.7 to 24. In addition, the standard provides a "free" bit rate mode to support fixed bit rates other than the predefined rates.

- MPEG/audio offers a choice of three independent layers of compression. This provides a wide range of tradeoffs between codec complexity and compressed audio quality:

  Layer I is the simplest and is best suited for bit rates above 128 kbits/sec per channel. For example, Philips' Digital Compact Cassette (DCC)[5] uses Layer I compression at 192 kbits/s per channel.

  Layer II has an intermediate complexity and is targeted for bit rates around 128 kbits/s per channel. Possible applications for this layer include the coding of audio for Digital Audio Broadcasting (DAB®)[6] , for the storage of synchronized video-and-audio sequences on CD-ROM, and the full motion extension of CD-interactive, Video CD.

  Layer III is the most complex but offers the best audio quality, particularly for bit rates around 64 kbits/s per channel. This layer is well suited for audio transmission over ISDN.

  All three layers are simple enough to allow single-chip, real-time decoder implementations.

- The coded bitstream supports an optional Cyclic Redundancy Check (CRC) error detection code.

- MPEG/audio provides a means of including ancillary data within the bitstream.

In addition, the MPEG/audio bitstream makes features such as random access, audio fast forwarding, and audio reverse possible.

2. OVERVIEW

The key to MPEG/audio compression is quantization. Although quantization is lossy, this algorithm can give "transparent", perceptually lossless, compression. The MPEG/audio committee conducted extensive subjective listening tests during the development of the standard. The tests showed that even with a 6-to-1 compression ratio (stereo, 16 bits/sample, audio sampled at 48 kHz compressed to 256 kbits/sec) and under optimal listening conditions, expert listeners were unable to distinguish between coded and original audio clips with statistical significance. Furthermore, these clips were specially chosen because they are difficult to compress. Reference 7 gives the details of the set up, procedures and results of these tests.

Figure 1 shows block diagrams of the MPEG/audio encoder and decoder. The input audio stream passes through a filter bank that divides the input into multiple subbands of frequency. The input audio stream simultaneously passes through a psychoacoustic model that determines the ratio of the signal energy to the masking threshold for each subband. The bit or noise allocation block uses the signal-to-mask ratios to decide how to apportion the total number of code bits available for the quantization of the subband signals to minimize the audibility of the quantization noise. Finally, the last block takes the representation of the quantized subband samples and formats this data and side information into a coded bitstream. Ancillary data not necessarily related to the audio stream can be inserted within the coded bitstream. The decoder deciphers this bitstream, restores the quantized subband values, and reconstructs the audio signal from the subband values.

The following sections explore various aspects of MPEG/audio compression in more detail. The first section covers the time to frequency mapping of the polyphase filter bank. The next section covers implementations of the psychoacoustic model followed by a more detailed descriptions of the 3 Layers of MPEG/audio compression. This gives enough background to cover a brief summary of the different bit (or noise) allocation processes used by the three layers and the joint stereo coding methods. The paper finishes with a short description of current MPEG/audio standards work.

2.1 The Polyphase Filter Bank

This section will give some insight into the behavior of the MPEG/audio polyphase filter bank by presenting a detailed examination of the encoder's analysis filter bank. A similar analysis applies to the decoder's synthesis filter bank.

The polyphase filter bank is the key component common to all layers of MPEG/audio compression. This filter bank divides the audio signal into 32 equal-width frequency subbands. The filters are relatively simple and provide good time resolution with reasonable frequency resolution. The design is a good compromise with three notable concessions. First, the equal widths of the subbands do not accurately reflect the human auditory system's frequency dependent behavior. The width of a "critical band" as a function of frequency is a good indicator of this behavior. Many psychoacoustic effects are consistent with a critical band frequency scaling. For example, both the perceived loudness of a signal and its audibility in the presence of a masking signal is different for signals within one critical band than for signals that extend over more than one critical band. Figure 2 compares the polyphase filter bandwidths with the width of these critical bands. At lower frequencies a single subband covers several critical bands. In this circumstance the number of quantizer bits cannot be specifically tuned for the noise masking available for the individual critical bands. Instead, the critical band with the least noise masking dictates the number of quantization bits needed for the entire subband. Second, the filter bank and its inverse are not lossless transformations. Even without quantization, the inverse transformation cannot perfectly recover the original signal. However, by design the error introduced by the filter bank is small and inaudible. Finally, adjacent filter bands have a major frequency overlap. A signal at a single frequency can affect two adjacent filter bank outputs. Other parts of this paper will cover these issues in more detail.

To understand the polyphase filter bank it is useful to examine its origin. The ISO MPEG/audio standard describes a procedure for computing the analysis polyphase filter outputs that is very similar to a method described by Rothweiler[8]. Figure 3 shows a structure for a MPEG-encoder-filter bank based on Rothweiler's proposal. For comparison, figure 4 shows the flow diagram from the ISO MPEG/audio standard for the same filter bank.

By combining the equations and steps shown by flow diagram, one can derive the following equation for the filter bank outputs:

$$s_t[i] = \sum_{k=0}^{63} \sum_{j=0}^{7} M[i][k] * (C[k+64j] * x[k+64j]) \quad (1)$$

where:

i is the subband index and ranges from 0 to 31,
$s_t[i]$ is the filter output sample for subband i at time t, where t is an integer multiple of 32 audio sample intervals,
C[n] is one of 512 coefficients of the analysis window defined in the standard,
x[n] is an audio input sample read from a 512 sample buffer, and
$M[i][k] = \cos[\frac{(2*i+1)*(k-16)*\pi}{64}]$ are the analysis matrix coefficients.

The above equation is partially optimized to reduce the number of computations. Because the function within the parenthesis is independent of the value of i, and M[i][k] is independent of j, the 32 filter outputs need only 512 + 32*64 = 2,560 multiplies and 64*7+32*63 = 2,464 additions, or roughly 80 multiplies and additions per output. Substantially further reductions in multiplies and adds are possible with, for example, a fast Discrete Cosine Transform[9,10] or a fast Fourier Transform implementation[11].

Note this filter bank implementation is critically sampled: for every 32 input samples, the filter bank produces 32 output samples. In effect, each of the 32 subband filters subsamples its output by 32 to produce only one output sample for every 32 new audio samples.

One can manipulate equation (1) into a familiar filter convolution equation:

$$s_t[i] = \sum_{n=0}^{511} x[t-n]*H_i[n] \quad (2)$$

where:

$x[\tau]$ is an audio sample at time τ, and
$H_i[n] = h[n]*\cos[\frac{(2*i+1)*(n-16)*\pi}{64}]$ with
h[n]= -C[n], if the integer part of (n/64) is odd,
= C[n] otherwise, for n = 0 to 511.

In this form, each subband of the filter bank has its own band-pass filter response, $H_i[n]$. Although this form is more convenient for analysis, it is clearly not an efficient solution: A direct implementation of this equation requires 32*512 = 16,384 multiplies and 32*511 = 16,352 additions to compute the 32 filter outputs.

The coefficients, h[n], correspond to the prototype low-pass filter response for the polyphase filter bank. Figure 5 compares a plot of h[n] with C[n]. The C[n] used in the partially optimized equation (1) has every odd numbered group of 64 coefficients of h[n] negated to compensate for M[i][k]. The cosine term of M[i][k] only ranges from k = 0 to 63 and covers an odd number of half cycles whereas the cosine terms of $H_i[n]$ range from n=0 to 511 and cover 8 times the number of half cycles.

The equation for $H_i[n]$ clearly shows that each is a modulation of the prototype response with a cosine term to shift the low pass response to the appropriate frequency band, hence these are called polyphase filters. These filters have center frequencies at odd multiples of $\pi/(64T)$ where T is the audio sampling period and each has a nominal bandwidth of $\pi/(32T)$. As figure 6 shows, the prototype filter response does not have a sharp cutoff at its nominal bandwidth. So when the filter outputs are subsampled by 32, there is a considerable amount of aliasing. The design of the prototype filter, and the inclusion of appropriate phase shifts in the cosine terms, results in a complete alias cancellation at the output of the decoder's synthesis filter bank[8,12]. Another consequence of using a filter with a wider-than-nominal bandwidth is an overlap in the frequency coverage of adjacent polyphase filters. This effect can be detrimental to efficient audio compression because signal energy near nominal subband edges will appear in two adjacent polyphase filter outputs. Figure 7 shows how a pure sinusoid tone, which has energy at only one frequency, appears at the output of two polyphase filters.

Although the polyphase filter bank is not lossless, any consequent errors are small. Figures 8 and 9 show the composite frequency response combining response of the encoder's analysis filter bank with that of the decoder's synthesis filter bank. Without quantization of the subband samples, the composite response has a ripple of less than .07 dB.

2.2 Psychoacoustics

The MPEG/audio algorithm compresses the audio data in large part by removing the acoustically irrelevant parts of the audio signal. That is, it takes advantage of the human auditory system's inability to hear quantization noise under conditions of auditory masking. This masking is a perceptual property of the human auditory system that occurs whenever the presence of a strong audio signal makes a temporal or spectral neighborhood of weaker audio signals imperceptible. A variety of psychoacoustic experiments corroborate this masking phenomenon[13].

Empirical results also show that the human auditory system has a limited, frequency dependent, resolution. This frequency dependency can be expressed in terms of critical band widths which are less than 100 Hz for the lowest audible frequencies and more than 4 kHz at the highest. The human auditory system blurs the various signal components within a critical band although this system's frequency selectivity is much finer than a critical band.

Because of the human auditory system's frequency-dependent resolving power, the noise masking threshold at any given frequency is solely dependent on the signal energy within a limited bandwidth neighborhood of that frequency. Figure 10 illustrates this property. MPEG/audio works by dividing the audio signal into frequency subbands that approximate critical bands, then quantizing each subband according to the audibility of quantization noise within that band. For the most efficient compression, each band should be quantized with no more levels than necessary to make the quantization noise inaudible.

2.2.1 The Psychoacoustic Model

The psychoacoustic model analyzes the audio signal and computes the amount of noise masking available as a function of frequency [,14,15,16,17]. The masking ability of a given signal component depends on its frequency position and its loudness. The encoder uses this information to decide how best to represent the input audio signal with its limited number of code bits. The MPEG/audio standard provides two example implementations of the psychoacoustic model. Psychoacoustic model 1 is less complex than psychoacoustic model 2 and has more compromises to simplify the calculations. Either model works for any of the layers of compression. However, only model 2 includes specific modifications to accommodate Layer III.

There is considerable freedom in the implementation of the psychoacoustic model. The required accuracy of the model is dependent on the target compression factor and the intended application. For low levels of compression, where there is a generous supply of code bits, a complete bypass of the psychoacoustic model may be adequate for consumer use. In this case, the bit allocation process can iteratively assign bits to the subband with the lowest signal-to-noise ratio. For the archiving of music, the psychoacoustic model can be made much more stringent[18].

Below is a general outline of the basic steps involved in the psychoacoustic calculations for either model. Differences between the two models will be highlighted.

- *Time align audio data*. There is one psychoacoustic evaluation per frame. The audio data sent to the psychoacoustic model must be concurrent with the audio data to be coded. The psychoacoustic model must account for both the delay of the audio data through the filter bank and a data offset so that the relevant data is centered within the psychoacoustic analysis window. For example, when using psychoacoustic model 1 for Layer I, the delay through the filter bank is 256 samples and the offset required to center the 384 samples of a Layer I frame in the 512 point analysis window is (512-384)/2 = 64 points. The net offset is 320 points to time align the psychoacoustic model data with the filter bank outputs.

- *Convert audio to a frequency domain representation*. The psychoacoustic model should use a separate, independent, time-to-frequency mapping instead of the polyphase filter bank because it needs finer frequency resolution for an accurate calculation of the masking thresholds. Both psychoacoustic models use a Fourier transform for this mapping. A standard Hann weighting, applied to the audio data before Fourier transformation, conditions the data to reduce the edge effects of the transform window.

  Psychoacoustic model 1 uses a 512 sample analysis window for Layer I and a 1024 sample window for Layers II and III. Because there are only 384 samples in a Layer I frame, a 512 sample window provides adequate coverage. Here the smaller window size reduces the computational load. Layer II and III use a 1,152 sample frame size so the 1,024

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.