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

sample window does not provide complete coverage. While ideally the analysis window should completely cover the samples to be coded, a 1,024 sample window is a reasonable compromise. Samples falling outside the analysis window generally will not have a major impact on the psychoacoustic evaluation.

Psychoacoustic model 2 uses a 1,024 sample window for all layers. For Layer I, the model centers a frame's 384 audio samples in the psychoacoustic window as previously discussed. For Layers II and III, the model computes two 1,024 point psychoacoustic calculations for each frame. The first calculation centers the first half of the 1,152 samples in the analysis window and the second calculation centers the second half. The model combines the results of the two calculations by using the higher of the two signal-to-mask ratios for each subband. This in effect selects the lower of the two noise masking thresholds for each subband.

- *Process spectral values in groupings related to critical band widths.* To simplify the psychoacoustic calculations, both models process the frequency values in perceptual quanta.

- *Separate spectral values into tonal and non-tonal components*. Both models identify and separate the tonal and noise-like components of the audio signal because the masking abilities of the two types of signal are different.

  Psychoacoustic model 1 identifies tonal components based on the local peaks of the audio power spectrum. After processing the tonal components, model 1 sums the remaining spectral values into a single non-tonal component per critical band. The frequency index of each of these concentrated non-tonal components is the value closest to the geometric mean of the enclosing critical band.

  Psychoacoustic model 2 never actually separates tonal and non-tonal components. Instead, it computes a tonality index as a function of frequency. This index gives a measure of whether the component is more tone-like or noise-like. Model 2 uses this index to interpolate between pure tone-masking-noise and noise-masking-tone values. The tonality index is based on a measure of predictability. Model 2 uses data from the previous two analysis windows to predict, via linear extrapolation, the component values for the current window. Tonal components are more predictable and thus will have higher tonality indices. Because this process relies on more data, it is more likely to better discriminate between tonal and non-tonal components than the model 1 method.

- *Apply a spreading function*. The masking ability of a given signal spreads across its surrounding critical band. The model determines the noise masking thresholds by first applying an empirically determined masking (model 1) or spreading function (model 2) to the signal components.

- *Set a lower bound for the threshold values.* Both models include an empirically determined absolute masking threshold, the threshold in quiet. This threshold is the lower bound on the audibility of sound.

- *Find the masking threshold for each subband.* Both psychoacoustic models calculate the masking thresholds with a higher frequency resolution than that provided by the polyphase filter bank. Both models must derive a subband threshold value from possibly a multitude of masking thresholds computed for frequencies within that subband.

  Model 1 selects the minimum masking threshold within each subband. While this approach is good for the lower frequency subbands where the subband is narrow relative to a critical band, it may be inaccurate for the higher frequency subbands because critical bands for that frequency range span several subbands. These inaccuracies arise because model 1 concentrates all the non-tonal components within each critical band into a single value at a single frequency. In effect model 1 converts non-tonal components into a form of tonal component. A subband within a wide critical band but far from the concentrated non-tonal component will not get an accurate non-tonal masking assessment. This approach is a compromise to reduce the computational loads.

  Model 2 selects the minimum of the masking thresholds covered by the subband only where the band is wide relative to the critical band in that frequency region. It uses the average of the masking thresholds covered by the subband where the band is narrow relative to the critical band. Model 2 is not less accurate for the higher frequency subbands because it does not concentrate the non-tonal components.

- *Calculate the signal-to-mask ratio.* The psychoacoustic model computes the signal-to-mask ratio as the ratio of the signal energy within the subband (or, for Layer III , a group of bands) to the minimum masking threshold for that subband. The model passes this value to the bit (or noise) allocation section of the encoder.

2.2.1.1 An Example of Psychoacoustic Model Analysis

This section gives an illustrative example of the analysis used by psychoacoustic model 1 and model 2. Figure 11 is a spectral plot of the example audio signal to be psychoacoustically analyzed and compressed. This signal consists of a combination of a strong, 11,250 Hz, sinusoidal tone with lowpass noise.

2.2.1.1.1 Example for Psychoacoustic Model 2

The processes used by psychoacoustic model 2 are somewhat easier to visualize, so this model will be covered first. Figure 12a shows the result, according to psychoacoustic model 2, of transforming the audio signal to the perceptual domain (63, one-third critical band, partitions) and then applying the spreading function. Note the shift of the sinusoid peak and the expansion of the lowpass noise distribution. The perceptual transformation expands the low frequency region and compresses the higher frequency region. Because the spreading function is applied in a perceptual domain, the shape of the spreading function is relatively uniform as a function of partition. Figure 13 shows a plot of the spreading functions. Figure 12b shows the tonality index for the audio signal as computed by psychoacoustic model 2. Figure 14a shows a plot of the masking threshold as computed by the model based on the spread energy and the tonality index. This figure has plots of the masking threshold both before and after the incorporation of the threshold in quiet to illustrate its impact. Note the threshold in quiet significantly increases the noise masking threshold for the higher frequencies. The human auditory system is much less sensitive in this region. Also note how the sinusoid signal increases the masking threshold for the neighboring frequencies. The masking threshold is computed in the uniform frequency domain instead of the perceptual domain in preparation for the final step of the psychoacoustic model, the calculation of the signal-to-mask ratios (SMR) for each subband. Figure 14b is a plot of these results and figure 14c is a frequency plot of a processed audio signal using these SMR's. In this example the audio compression was severe (768 to 64 kbits/sec) so the coder may not necessarily be able to mask all the quantization noise.

2.2.1.1.2 Example for Psychoacoustic Model 1

This example uses the same example audio signal as above. Figure 15a shows the how psychoacoustic model 1 identifies the local spectral peaks as tonal and non-tonal components. Figure 15b shows the remaining tonal and non-tonal components after the decimation process. This process both removes components that would be below the threshold in quiet and removes the weaker tonal components within roughly half a critical band width (0.5 Bark) of a stronger tonal component. Psychoacoustic model 1 uses the decimated tonal and non-tonal components to determine the global masking threshold in a subsampled frequency domain. This subsampled domain corresponds approximately to a perceptual domain. Figure 15c shows the global masking threshold calculated for the example audio signal. Psychoacoustic model 1 selects the minimum global masking threshold within each subband to compute the SMR's. Figure 16a shows the resulting signal-to-mask ratio and figure 16b is a frequency plot of the processed audio signal using these SMR's.

2.3 Layer Coding Options

The MPEG/audio standard has 3 distinct layers for compression. Layer I forms the most basic algorithm while Layer II and Layer III are enhancements that use some elements found in Layer I. Each successive layer improves the compression performance but at the cost of greater encoder and decoder complexity. Every MPEG/audio bitstream contains periodically spaced frame headers to identify the bitstream. Figure 17 gives a pictoral representation of the header syntax. A 2 bit field in the MPEG header identifies the layer in use.

2.3.1 Layer I

The Layer I algorithm codes audio in frames of 384 audio samples. It does so by grouping together 12 samples from each of the 32 subbands, as shown in figure 18. Besides the code for audio data, each frame contains a header, an optional Cyclic Redundancy Code (CRC) error check word, and possibly ancillary data. Figure 19a shows the arrangement of this data in a Layer I bitstream. The numbers within parentheses give the possible number of bits that can be used to encode each field. Each group of 12 samples gets a bit allocation and, if the bit allocation is not zero, a scale factor. The bit allocation tells the decoder the number of bits used to represent each sample. For Layer I this allocation can be 0 to 15 bits per subband. The scale factor is

a multiplier that sizes the samples to make full use of the range of the quantizer. Each scale factor has a 6 bit representation. The decoder multiplies the decoded quantizer output with the scale factor to recover the quantized subband value. The dynamic range of the scalefactors alone is over 120 dB. The combination of the bit allocation and the scale factor provide the potential for representing the samples with a dynamic range well over 120 dB. Joint stereo coding slightly alters the representation of left and right channel audio samples and will be covered later.

2.3.2 Layer II

The Layer II algorithm is a straightforward enhancement of Layer I. It codes the audio data in larger groups and imposes some restrictions on the possible bit allocations for values from the middle and higher subbands. It also represents the bit allocation, the scale factor values, and the quantized samples with a more compact code. Layer II gets better audio quality by saving bits in these areas so more code bits are available to represent the quantized subband values.

The Layer II encoder forms frames of 1152 samples per audio channel. Whereas Layer I codes data in single groups of 12 samples for each subband, Layer II codes data in 3 groups of 12 samples for each subband. Figure 18 shows this grouping as well. Again discounting stereo redundancy coding, there is one bit allocation and up to three scale factors for each trio of 12 samples. The encoder uses a different scale factor for each group of 12 samples only if necessary to avoid audible distortion. The encoder shares scale factor values among two or all three groups in two other cases. One, when the values of the scale factors are sufficiently close. Two, when the encoder anticipates that temporal noise masking by the human auditory system will hide any distortion caused by using one only scale factor instead of two or three. The scale factor selection information (SCFSI) field in the Layer II bitstream informs the decoder if and how to share the scale factor values. Figure 19b shows the arrangement of the various data fields in a Layer I I bitstream.

Another enhancement is for the occasion when the Layer II encoder allocates 3, 5, or 9 levels for subband quantization. In these circumstances, the Layer II encoder represents 3 consecutive quantized values with a single, more compact codeword.

2.3.3 Layer III

The Layer III algorithm is a much more refined approach derived from ASPEC and OCF algorithms[15,19,20]. Although based on the same filter bank found in Layer I and Layer II, Layer III compensates for some of filter bank deficiencies by processing the filter outputs with a Modified Discrete Cosine Transform (MDCT)[21]. Figure 20 shows a block diagram of this processing for the encoder. Unlike the polyphase filter bank, without quantization the MDCT transformation is lossless. The MDCT's further subdivide the subband outputs in frequency to provide better spectral resolution. Furthermore, once the subband components are subdivided in frequency, the Layer III encoder can partially cancel some aliasing caused by the polyphase filter bank. Of course, the Layer III decoder has to undo the alias cancellation so the inverse MDCT can reconstruct subband samples in their original, aliased, form for the synthesis filter bank.

Layer III specifies two different MDCT block lengths: a long block of 18 samples or a short block of 6. There is a 50 percent overlap between successive transform windows so the window size is 36 and 12, respectively. The long block length allows greater frequency resolution for audio signals with stationary characteristics while the short block length provides better time resolution for transients[22]. Note the short block length is one third that of a long block. In the short block mode, three short blocks replace a long block so that the number of MDCT samples for a frame of audio samples is unchanged regardless of the block size selection. For a given frame of audio samples, the MDCT's can all have same block length (long or short) or have a mixed-block mode. In the mixed block mode the MDCT's for the 2 lower frequency subbands have long blocks and the MDCT's for the 30 upper subbands have short blocks. This mode provides better frequency resolution for the lower frequencies, where it is needed the most, without sacrificing time resolution for the higher frequencies.

The switch between long and short blocks is not instantaneous. A long block with a specialized long-to-short or short-to-long data window serves to transition between long and short block types. Figure 21 shows how the MDCT windows transition between long and short block modes.

Because MDCT processing of a subband signal provides better frequency resolution, it consequently has poorer time resolution. The MDCT operates on 12 or 36 polyphase filter samples so the effective time window of audio samples involved in this processing is 12 or 36 times larger. The quantization of MDCT values will cause errors that are spread over this larger time window so it is more likely that this quantization will produce audible distortions. Such distortions usually manifest themselves as pre-echo because the temporal masking of noise occurring before a given signal is weaker than the masking of noise after. Layer III incorporates several measures to reduce pre-echo. First, the Layer III psychoacoustic model has modifications to detect

the conditions for pre-echo. Second, Layer III can borrow code bits from the bit reservoir to reduce quantization noise when pre-echo conditions exist. Finally, the encoder can switch to a smaller MDCT block size to reduce the effective time window.

Besides the MDCT processing, other enhancements over the Layer I and Layer II algorithms include:

- *Alias reduction*. Layer III specifies a method of processing the MDCT values to remove some artifacts caused by the overlapping bands of the polyphase filter bank.

- *Non uniform quantization.* The Layer III quantizer raises its input to the 3/4 power before quantization to provide a more consistent signal-to-noise ratio over the range of quantizer values. The requantizer in MPEG/audio decoder relinearizes the values by raising its output to the 4/3 power.

- S*calefactor bands*. Unlike Layer I and II, where there can be a different scalefactor for each subband, Layer III uses scalefactor bands. These bands cover several MDCT coefficients and have approximately critical band widths. In Layer III scalefactors serve to color the quantization noise to fit the varying frequency contours of the masking threshold. Values for these scalefactors are adjusted as part of the noise allocation process.

- *Entropy coding of data values*. Layer III uses variable-length Huffman codes to encode the quantized samples to get better data compression. After quantization, the encoder orders the 576 (32 subbands * 18 MDCT coeficients/subband) quantized MDCT coefficients in a predetermined order. The order is by increasing frequency except for the short MDCT block mode. For short blocks there are 3 sets of window values for a given frequency so the ordering is by frequency, then by window, within each scalefactor band. Ordering is advantageous because large values tend to be at the lower frequencies and long runs of zero or near-zero values tend to be at the higher frequencies.

  The encoder delimits the ordered coefficients into 3 distinct regions. This enables the encoder to code each region with a different set of Huffman tables specifically tuned for the statistics of that region. Starting at the highest frequency, the encoder identifies the continuous run of all-zero values as one region. This region does not have to be coded because its size can be deduced from the size of the other two regions. However, it must contain an even number of zeroes because the other regions code their values in even numbered groupings. A second region, the "count1" region, consists of a continuous run of values consisting only of -1, 0, or 1. The Huffman table for this region codes 4 values at a time so the number of values in this region must be a multiple of 4. Finally a third region covers all the remaining values, the "big values" region. The Huffman tables for this region code the values in pairs. The "big values" region is further subdivided into three subregions that each have its own specific Huffman table. Besides improving coding efficiency, partitioning the MDCT coefficients into regions and subregions helps control error propagation. Within the bitstream, the Huffman codes for the values are ordered from low to high frequency.

- *Use of a "bit reservoir".* The design of the Layer III bitstream better fits the encoder's time-varying demand on code bits. As with Layer II, Layer III processes the audio data in frames of 1152 samples. Figure 19c shows the arrangement of the various bit fields in a Layer III bitstream. Unlike Layer II, the coded data representing these samples do not necessarily fit into a fixed length frame in the code bitstream. The encoder can donate bits to a reservoir when it needs less than the average number of bits to code a frame. Later, when the encoder needs more than the average number of bits to code a frame, it can borrow bits from the reservoir. The encoder can only borrow bits donated from past frames; it cannot borrow from future frames. The Layer III bitstream includes a 9-bit pointer, "main_data_begin", with each frame's side-information that points to the location of the starting byte of the audio data for that frame. Figure 22 illustrates the implementation of the bit reservoir within a fixed length frame structure via the “main_data_begin” pointer. Although the main_data_begin limits the maximum variation of the audio data to 2^9 bytes (header and side information are not counted because for a given mode they are of fixed length and occur at regular intervals in the bit stream), the actual maximum allowed variation will often be much less. For practical considerations, the standard stipulates that this variation cannot exceed what would be possible for an encoder with a code buffer limited to 7,680 bits. Because compression to 320 kbits/sec with an audio sampling rate of 48 kHz requires an average number of code bits per frame of

  1152 (samples/frame) * 320,000 (bits/sec) / 48,000 (samples/sec) = 7680 bits/frame,

  absolutely no variation is allowed for this coding mode.

Although the enhancements to Layer III are conceptually complex, there is only a modest increase in the computation requirements of a Layer III decoder over a Layer II decoder. For example, even a direct matrix-multiply implementation of the IMDCT requires only about 19 multiplies and additions per subband value. The enhancements mainly increase the complexity of the encoder and the memory requirements of the decoder.

2.4 Bit Allocation

The bit allocation process determines the number of code bits to be allocated to each subband based on information from the psychoacoustic model. For Layer I and II, this process starts by computing the mask-to-noise ratio as given by the following equation:

$$MNR_{dB} = SNR_{dB} - SMR_{dB}$$

where

MNR_{dB} is the mask-to-noise ratio,
SNR_{dB} is the signal-to-noise ratio, and
SMR_{dB} is the signal-to-mask ratio from the psychoacoustic model.

All values are in decibels.

The MPEG/audio standard provides tables that give estimates for the signal-to-noise ratio resulting from quantizing to a given number of quantizer levels. Designers are free to try other methods of getting the signal-to-noise ratios.

Once the bit allocation unit has mask-to-noise ratios for all the subbands, it searches for the subband with the lowest mask-to-noise ratio and allocates code bits to that subband. When a subband gets allocated more code bits, the bit allocation unit looks up the new estimate for the signal-to-noise ratio and recomputes that subband's mask-to-noise ratio. The process repeats until no more code bits can be allocated.

The Layer III encoder uses noise allocation. The encoder iteratively varies the quantizers in an orderly way, quantizes the spectral values, counts the number of Huffman code bits required to code the audio data and actually calculates the resulting noise. If, after quantization, there are still scalefactor bands with more than the allowed distortion, the encoder amplifies the values in those scalefactor bands and effectively decreases the quantizer step size for those bands. After this the process repeats. The process stops if any of these three conditions is true:

1. None of the scalefactor bands have more than the allowed distortion.

2. The next iteration would cause the amplification for any of the bands to exceed the maximum allowed value.

3. The next iteration would require all the scalefactor bands to be amplified.

Real-time encoders also can include a time-limit exit condition for this process.

2.5 Stereo Redundancy Coding

The MPEG/audio compression algorithm supports two types of stereo redundancy coding: intensity stereo coding and Middle/Side (MS) stereo coding. All layers support intensity stereo coding. Layer III also supports MS stereo coding. Both forms of redundancy coding exploit another perceptual property of the human auditory system. Psychoacoustic results show that above about 2 kHz and within each critical band, the human auditory system bases its perception of stereo imaging more on the temporal envelope of the audio signal than its temporal fine structure.

In intensity stereo mode the encoder codes some upper-frequency subband outputs with a single summed signal instead of sending independent left and right channel codes for each of the 32 subband outputs. The intensity stereo decoder reconstructs the left and right channels based only on a single summed signal and independent left and right channel scale factors. With intensity stereo coding, the spectral shape of the left and right channels is the same within each intensity-coded subband but the magnitude is different.

The MS stereo mode encodes the left and right channel signals in certain frequency ranges as middle (sum of left and right) and side (difference of left and right) channels. In this mode, the encoder uses specially tuned threshold values to compress the side channel signal further.

3. FUTURE MPEG/AUDIO STANDARDS: PHASE 2

The second phase of the MPEG audio compression standard, MPEG-2 audio, has just been completed. This new standard became an international standard in November of 1994. This standard further extends the first MPEG standard in the following ways:

- *Multichannel audio support.* The enhanced standard supports up to 5 high fidelity audio channels, plus a low frequency enhancement channel (also known as 5.1 channels), thus it will be applicable for the compression of audio for High Definition Television or digital movies.

- *Multilingual audio support*. The standard supports up to 7 additional commentary channels.

- *Lower compressed audio bit rates*. The standard supports additional lower, compressed bit rates down to 8 kbits/sec.

- *Lower audio sampling rates*. Besides 32, 44.1, and 48 kHz, the new standard accommodates 16, 22.05, and 24 kHz sampling rates as well. The commentary channels can have a sampling rate that is half the high fidelity channel sampling rate.

In many ways this new standard is compatible with the first MPEG/audio standard (MPEG-1). MPEG-2/audio decoders can decode MPEG-1/audio bitstreams. In addition, MPEG-1/audio decoders can decode two main channels of MPEG-2/audio bitstreams. This backward compatibility is achieved by combining suitably weighted versions of each of the up to 5.1 channels into a "down-mixed" left and right channel. These two channels fit into the audio data framework of a MPEG-1/audio bitstream. Information needed to recover the original left, right, and remaining channels fit into the ancillary data portion of a MPEG-1/audio bitstream, or in a separate auxiliary bitstream. Results of subjective tests conducted in 1994 indicate that, in some cases, the backward compatibility requirement compromises the audio compression performance of the multichannel coder. Consequently the ISO MPEG group is currently working on an addendum to the MPEG-2 standard that specifies a non-backward compatible multichannel coding mode that offers better coding performance.

ACKNOWLEDGMENTS

The author wrote most of this paper while employed at Digital Equipment Corporation. He is grateful for the funding and support this company provided for his work in the MPEG standards. The author also appreciates the many helpful editorial comments given to him by the many reviewers of this paper, especially Karlheinz Brandenburg, Bob Dyas, Jim Fiocca, Leon van de Kerkhof, and Peter Noll.

REFERENCES

1. ISO/IEC International Standard IS 11172-3 "Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/s - Part 3: Audio"
2. K.Brandenburg, G.Stoll, et al."The ISO/MPEG-Audio Codec: A Generic Standard for Coding of High Quality Digital Audio," 92nd AES-Convention, preprint 3336, Vienna 1992.
3. National Communications System Technical Information Bulletin 92-1, "Details to Assist in Implementation of Federal Standard 1016 CELP," Arlington, VA, Jan. 1992.
4. ISO/IEC International Standard IS 11172-4 "Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/s - Part 4: Conformance"
5. G.C.Wirtz, "Digital Compact Cassette: Audio Coding Technique," 91st AES-Convention. preprint 3216, New York 1991.
6. G.Plenge, "A Versatile High Quality Radio System for the Future Embracing Audio and Other Applications," 94th AES-Convention, Berlin 1994.
7. C.Grewin, and T.Ryden, "Subjective Assessments on Low Bit-rate Audio Codecs," Proc. of the 10th International AES Conference, pp 91 - 102, London 1991.
8. J.H.Rothweiler,"Polyphase Quadrature Filters - a New Subband Coding Technique," Proc of the Int. Conf. IEEE ASSP, 27.2, pp1280-1283, Boston 1983.

9. D. Pan, "Digital Audio Compression", Digital Technical Journal, Vol. 5, No. 2, 1993.
10. W.-H.Chen, C.H.Smith, and S.C.Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," IEEE Trans. on Comm., Vol. COM-25, No. 9, September 1977.
11. H.J. Nussbaumer and M. Vetterli, "Computationally efficient QMF filter banks,"Proc. of the Int. Conf. IEEE ASSP, pp.11.3.1-11.3.4, 1984.
12. P.Chu,"Quadrature Mirror Filter Design for an Arbitrary Number of Equal Bandwidth Channels", IEEE Trans. on ASSP, vol. ASSP-33, no. 1, pp. 203-218, Feb. 1985.
13. B. Scharf, "Critical Bands," Foundations of Modern Auditory Theory , J. Tobias, pp 159-202, Academic Press, New York and London, 1970.
14. J.D.Johnston, "Estimation of Perceptual Entropy Using Noise Masking Criteria," Proc. of the Int. Conf. IEEE ASSP, pp 2524-2527, 1988.
15. J.D.Johnston, "Transform Coding of Audio Signals Using Perceptual Noise Criteria," IEEE Journal on Selected Areas in Communications, vol. 6, pp 314-323, Feb. 1988.
16. K.Brandenburg, "OCF - A New Coding Algorithm for High Quality Sound Signals," Proc. of the Int. Conf. IEEE ASSP, pp. 141-144, 1987.
17. D.Wiese and G. Stoll, "Bitrate Reduction of High Quality Audio Signals by Modeling the Ears' Masking Thresholds," 89th AES-Convention, preprint 2970, Los Angeles 1990.
18. K.Brandenburg and J.Herre, "Digital Audio Compression for Professional Applications," 92nd AES-Convention, preprint 3330, Vienna 1992.
19. K.Brandenburg and J.D.Johnston, "Second Generation Perceptual Audio Coding: The Hybrid Coder," 88th AES-Convention, preprint 2937, Montreaux 1990.
20. K.Brandenburg, J.Herre, J.D.Johnston, Y.Mahieux, E.F.Schroeder, "ASPEC: Adaptive Spectral Perceptual Entropy Coding of High Quality Music Signals," 90th AES-Convention, preprint 3011, Paris 1991.
21. J.Princen,A.Johnson, A.Bradley, "Subband/Transform Coding Technique Based on Time Domain Aliasing Cancellation," Proc. of the Int. Conf. IEEE ASSP, pp. 2161-2164, 1987.
22. B. Elder, "Coding of Audio Signals with Overlapping Block Transform and Adaptive Window Functions," (in German), Frequenz, vol. 43, pp. 252-256, 1989.
23. CCITT recommendation G.711, "Pulse Code Modulation (PCM) of Voice Frequencies," Geneva, 1972.
24. L.R.Rabiner and R.W.Schafer, Digital Processing of Speech Signals (Englewood Cliffs, NJ, Prentice-Hall, 1978).
25. CCITT recommendation G.721, "32 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)," 1986.
26. CCITT recommendation G.722, "7 kHz Audio-Coding Within 64 Kbits/s," Melbourne, 1988.
27. CCITT recommendation G.723, "Extensions of Recommendation G.721 Adaptive Differential Pulse Code Modulation to 24 and 40 Kbits/s for Digital Circuit Multiplication Equipment Application," Melbourne, 1988.
28. M. Nishiguchi, K, Akagiri, and T. Suzuki, "A New Audio Bit Rate Reduction System for the CD-I Format," 81st AES-Convention, Los Angeles 1986, preprint 2375.
29. Y. Takahashi, H. Yazawa, K, Yamamoto, and T. Anazawa, "Study and Evaluation of a New Method of ADPCM Encoding," 86th AES-Convention, Hamburg 1989, preprint 2813.