A Tutorial on MPEG/Audio Compression

Davis Pan
Motorola

This tutorial covers the theory behind MPEG/audio compression. While lossy, the algorithm often can provide "transparent," perceptually lossless compression even with factors of 6-to-1 or more. It exploits the perceptual properties of the human auditory system. The article also covers the basics of psychoacoustic modeling and the methods the algorithm uses to compress audio data with the least perceptible degradation.

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. The goal is to give a broad, preliminary understanding of MPEG/audio compression, so I omitted many of the details. Wherever possible, figures and illustrative examples 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.[3]

The MPEG/audio standard results from more than three years of collaborative work by an international committee of high-fidelity audio compression experts within the Moving 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 perfectly suitable for audio-only applications, MPEG/audio is actually one part of a three-part compression standard that also includes video and systems. The MPEG standard addresses the compression of synchronized video and audio at a total bit rate of about 1.5 megabits per second (Mbps).

The MPEG standard is rigid only where necessary to ensure interoperability. 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. Designers are free to try new and different implementations of the encoder or decoder within the bounds of the standard. The encoder especially offers good potential for diversity. Wide acceptance of this standard will permit manufacturers to produce and sell, at reasonable cost, large numbers of MPEG/audio codecs.

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. Since removal of such parts results in inaudible distortions, 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, as follows. In addition, the MPEG/audio bitstream makes features such as random access, audio fast-forwarding, and audio reverse possible.

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

Audio channel support. The compressed bitstream can support one or two audio channels in one of four possible modes:

1. a monophonic mode for a single audio channel,
2. a dual-monophonic mode for two independent audio channels (functionally identical to the stereo mode),
3. a stereo mode for stereo channels that share bits but do not use joint-stereo coding, and
4. a joint-stereo mode that takes advantage of either the correlations between the stereo channels or the irrelevancy of the phase difference between channels, or both.

Predefined bit rates. The compressed bitstream can have one of several predefined fixed bit rates ranging from 32 to 224 kilobits per second

1070-986X/95/$4.00 © 1995 IEEE

(Kbps) 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.

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

Layer I, the simplest, best suits bit rates above 128 Kbps per channel. For example, Philips' Digital Compact Cassette (DCC)[5] uses Layer I compression at 192 Kbps per channel.

Layer II has an intermediate complexity and targets bit rates around 128 Kbps per channel. Possible applications for this layer include the coding of audio for digital audio broadcasting (DAB),[6] 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 Kbps per channel. This layer suits audio transmission over ISDN.

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

Error detection. The coded bitstream supports an optional cyclic redundancy check (CRC) error-detection code.

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

Overview

The key to MPEG/audio compression—quantization—is lossy. Nonetheless, this algorithm can give "transparent," perceptually lossless compression. The MPEG/audio committee conducted extensive subjective listening tests during development of the standard. The tests showed that even with a 6-to-1 compression ratio (stereo, 16 bits per sample, audio sampled at 48 kHz compressed to 256 Kbps) and under optimal listening conditions, expert listeners could not distinguish between coded and original audio clips with statistical significance. Furthermore, these clips were specially chosen as difficult to compress. Grewin and Ryden[7] gave the details of the setup, 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.

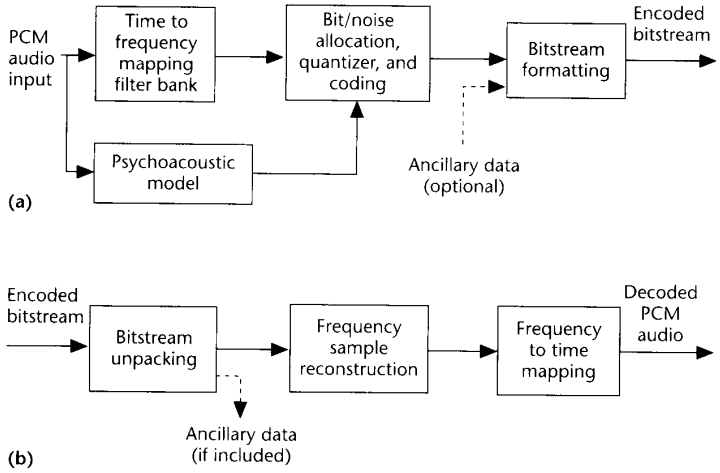

Figure 1. MPEG/audio compression and decompression. (a) MPEG/audio encoder. (b) MPEG/audio decoder.

First we'll look at the time to frequency mapping of the polyphase filter bank, then implementations of the psychoacoustic model and more detailed descriptions of the three layers of MPEG/audio compression. That 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 polyphase filter bank

The polyphase filter bank is common to all layers of MPEG/audio compression. This filter bank divides the audio signal into 32 equal-width frequency subbands. Relatively simple, the filters

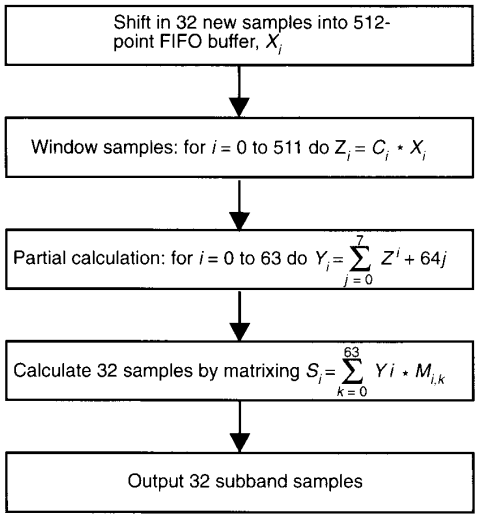

Figure 2. Flow diagram of the MPEG/audio encoder filter bank.

provide good time resolution with reasonable frequency resolution. The design has three notable concessions.

First, the equal widths of the subbands do not accurately reflect the human auditory system's frequency-dependent behavior. Instead, 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 differ for signals within one critical band versus signals that extend over more than one critical band. 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 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.

Third, adjacent filter bands have a major frequency overlap. A signal at a single frequency can affect two adjacent filter-bank outputs.

To understand the polyphase filter bank, it helps to examine its origin. The ISO MPEG/audio standard describes a procedure for computing the analysis polyphase filter outputs that closely resembles a method described by Rothweiler.[8] Figure 2 shows the flow diagram from the ISO MPEG/audio standard for the MPEG-encoder filter bank based on Rothweiler's proposal.

By combining the equations and steps shown by Figure 2, we can derive the following equation for the filter bank outputs:

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

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

$$M[i][k] = \cos\left[\frac{(2\times i+1)\times(k-16)\times\pi}{64}\right]$$

are the analysis matrix coefficients.

Equation 1 is partially optimized to reduce the number of computations. Because the function within the parentheses is independent of the value of i, and $M[i][k]$ is independent of j, the 32 filter outputs need only $512 + 32 \times 64 = 2{,}560$ multiplies and $64 \times 7 + 32 \times 63 = 2{,}464$ 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[3] or a fast Fourier transform implementation.[9]

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.

We can manipulate Equation 1 into a familiar filter convolution equation:

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

where $x[t]$ is an audio sample at time t, and

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

with $h[n] = -C[n]$ if the integer part of $(n/64)$ is odd and $h[n] = C[n]$ 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 $32 \times 512 = 16{,}384$ multiplies and $32 \times 511 = 16{,}352$ 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 3 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 eight times the number of half cycles.

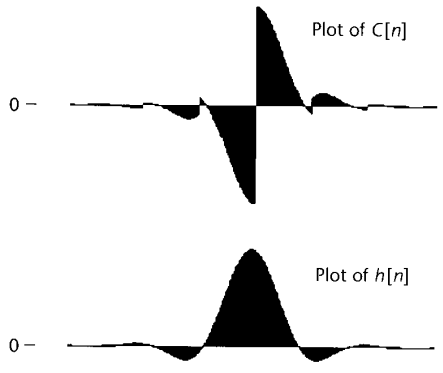

Figure 3. Comparison of **h[n]** *with* **C[n]**.

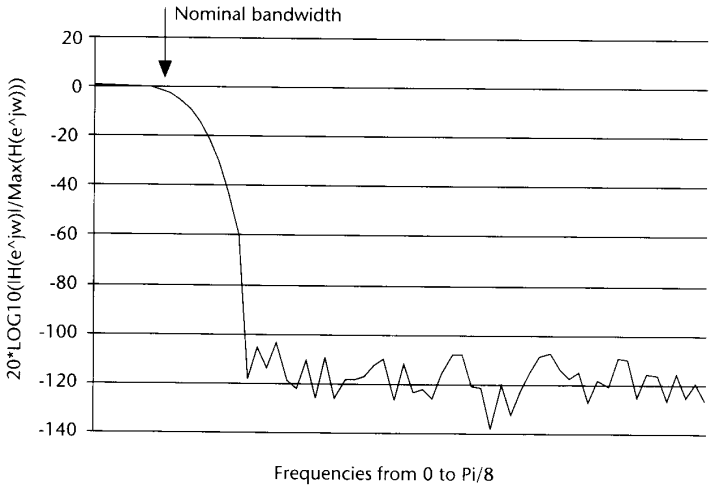

Figure 4. Frequency response of prototype filter, **h[n]**.

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 4 shows, the prototype filter response does not have a sharp cutoff at its nominal bandwidth. So when the filter outputs are subsampled by 32, a considerable amount of aliasing occurs. 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.[8,10]

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 5, next page, 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. Assuming you do not quantize the subband samples, the composite frequency response combining the response of the encoder's analysis filter bank with that of the decoder's synthesis bank has a ripple of less than 0.07 dB.

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 condi-

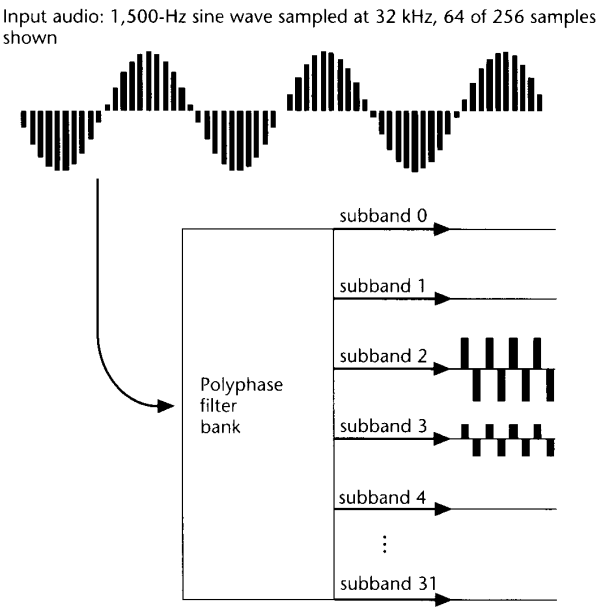

Figure 5. Aliasing: Pure sinusoid input can produce nonzero output for two subbands.

tions 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.[11]

Empirical results also show that the human auditory system has a limited, frequency-dependent resolution. This dependency can be expressed in terms of critical-band widths that 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 depends solely on the signal energy within a limited bandwidth neighborhood of that frequency. Figure 6 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.

The psychoacoustic model

The psychoacoustic model analyzes the audio signal and computes the amount of noise masking available as a function of frequency.[12,13] 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 makes 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 depends on the target compression factor and the intended application. For low levels of compression, where a generous supply of code bits exists, a complete bypass of the psychoacoustic model might 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 archiving music, the psychoacoustic model can be made much more stringent.[14]

Now let's look at 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 1,024-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. Layers 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 into 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 nontonal 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 differ.

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 nontonal component per critical band. The frequency index of each concentrated nontonal component is the value closest to the geometric mean of the enclosing critical band.

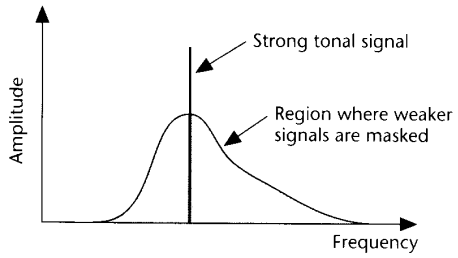

Figure 6. Audio noise masking.

Psychoacoustic model 2 never actually separates tonal and nontonal 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 probably discriminates better between tonal and nontonal 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 function (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 thresh-

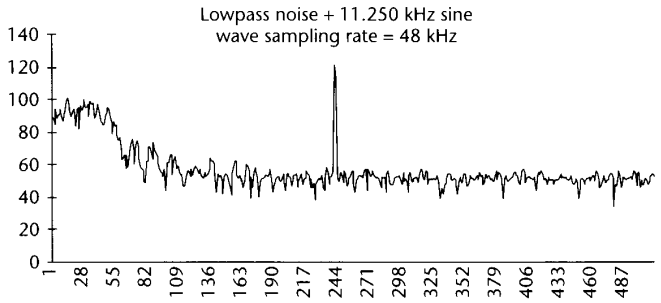

Figure 7. Input audio energy.

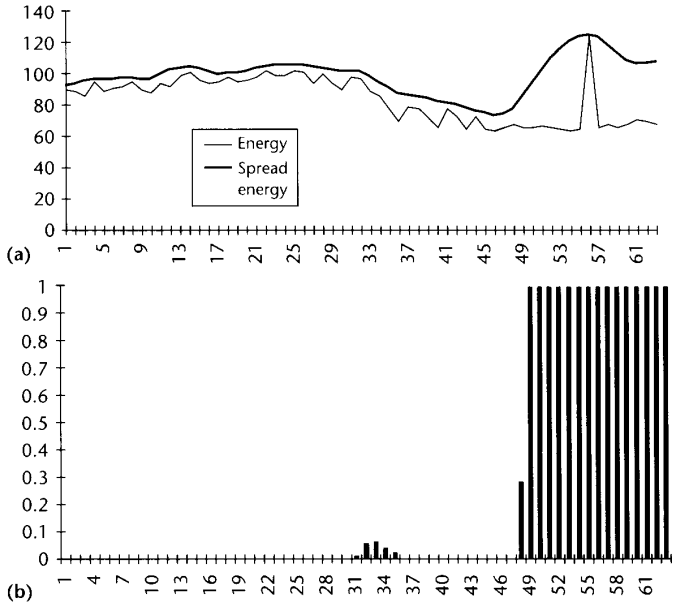

Figure 8. Psychoacoustic model 2 partition-domain processing. (a) Audio energy and spread energy in the perceptual domain. (b) Tonality index.

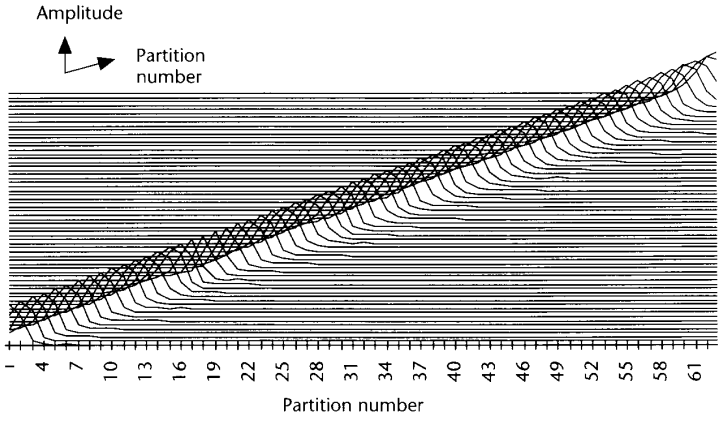

Figure 9. Spreading function for each partition (psychoacoustic model 2).

old within each subband. While this approach works well for the lower frequency subbands where the subband is narrow relative to a critical band, it might 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 nontonal components within each critical band into a single value at a single frequency. In effect, model 1 converts nontonal components into a form of tonal component. A subband within a wide critical band but far from the concentrated nontonal component will not get an accurate nontonal 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 has the same accuracy for the higher frequency subbands as for lower frequency subbands because it does not concentrate the nontonal 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.

Example of psychoacoustic model analysis

Figure 7 shows 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 low-pass noise.

Because the processes used by psychoacoustic model 2 are somewhat easier to visualize, we will cover this model first. Figure 8a 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. Figure 8b shows the tonali-

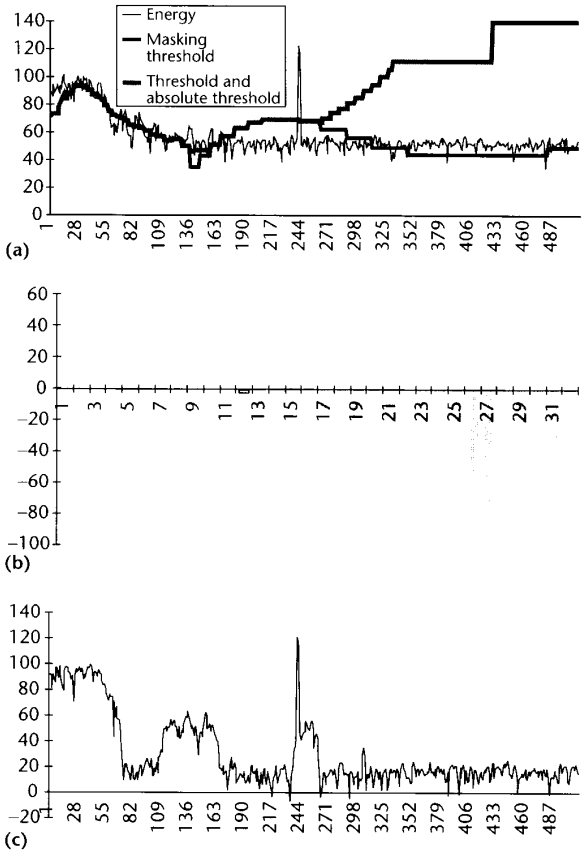

Figure 10. Psychoacoustic model 2 processing. (a) Original signal energy and computed masking thresholds. (b) Signal-to-mask ratios. (c) Coded audio energy (64 Kbps).

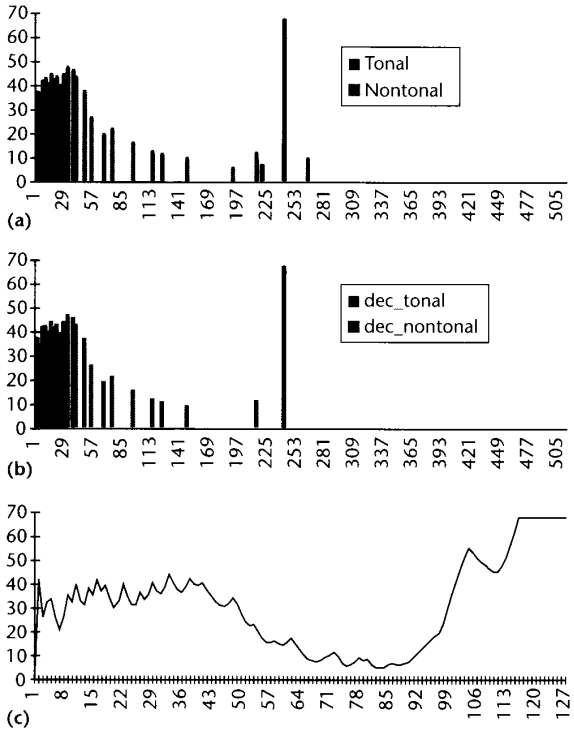

Figure 11. Psychoacoustic model 1 processing. (a) Identified tonal and nontonal components. (b) Decimated tonal and nontonal components. (c) Global masking thresholds.

ty index for the audio signal as computed by psychoacoustic model 2. Note the shift of the sinusoid peak and the expansion of the low-pass 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 9 shows a plot of the spreading functions.

Figure 10a 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 10b is a plot of these results, and Figure 10c is a frequency plot of a processed audio signal using these SMRs. In this example the audio compression was severe (768 to 64 Kbps), so the coder cannot necessarily mask all the quantization noise.

For psychoacoustic model 1, we use the same example audio signal as above. Figure 11a shows how the psychoacoustic model 1 identifies the local spectral peaks as tonal and nontonal components. Figure 11b shows the remaining tonal and nontonal components after the decimation

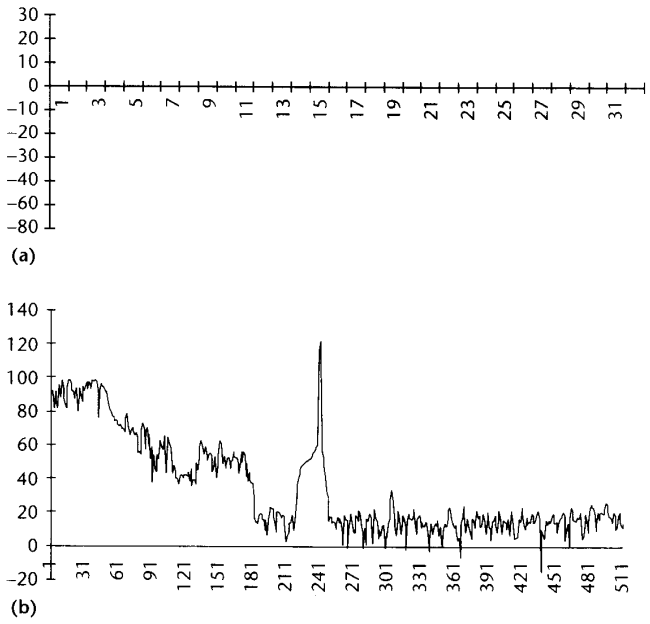

Figure 12. Psychoacoustic model 1 processing results. (a) Signal-to-mask ratios. (b) Coded audio energy (64 Kbps).

process. This process both removes components that would fall 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 nontonal components to determine the global masking threshold in a subsampled frequency domain. This subsampled domain corresponds approximately to a perceptual domain.

Figure 11c 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 SMRs. Figure 12a shows the resulting signal-to-mask ratio, and Figure 12b is a frequency plot of the processed audio signal using these SMRs.

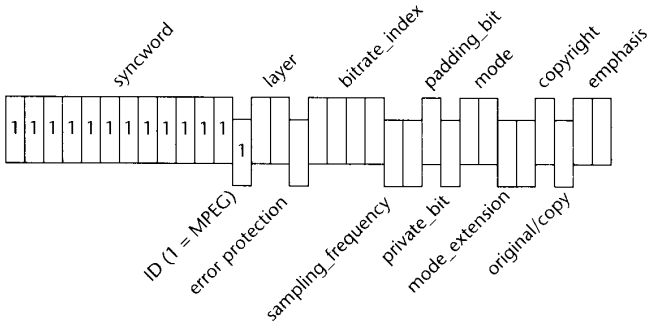

Figure 13. MPEG header syntax.

Layer coding options

The MPEG/audio standard has three 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—at the cost of greater encoder and decoder complexity.

Every MPEG/audio bitstream contains periodically spaced frame headers to identify the bitstream. Figure 13 gives a pictorial representation of the header syntax. A 2-bit field in the MPEG header identifies the layer in use.

Layer I

The Layer I algorithm codes audio in frames of 384 audio samples. It does so by grouping 12 samples from each of the 32 subbands, as shown in Figure 14. 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 15a shows the arrangement of this data in a Layer I bitstream. The numbers within parentheses give the number of bits possible 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 fully use 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 scale factors alone exceeds 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.

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 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.

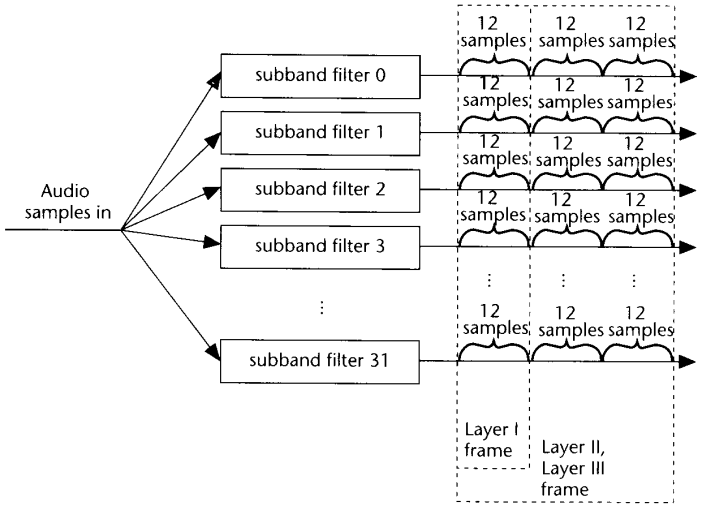

Figure 14. Grouping of subband samples for Layer I and Layer II. Note: Each subband filter produces 1 sample out for every 32 samples in.

The Layer II encoder forms frames of 1,152 samples per audio channel. Whereas Layer I codes data in single groups of 12 samples for each subband, Layer II codes data in three groups of 12 samples for each subband. Figure 14 shows this grouping. Again discounting stereo redundancy coding, each trio of 12 samples has one bit allocation and up to three scale factors.

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 only one 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 15b shows the arrangement of the various data fields in a Layer II bitstream.

Another enhancement covers the occasion when the Layer II encoder allocates three, five, or nine levels for subband quantization. In these circumstances, the Layer II encoder represents three consecutive quantized values with a single, more compact code word.

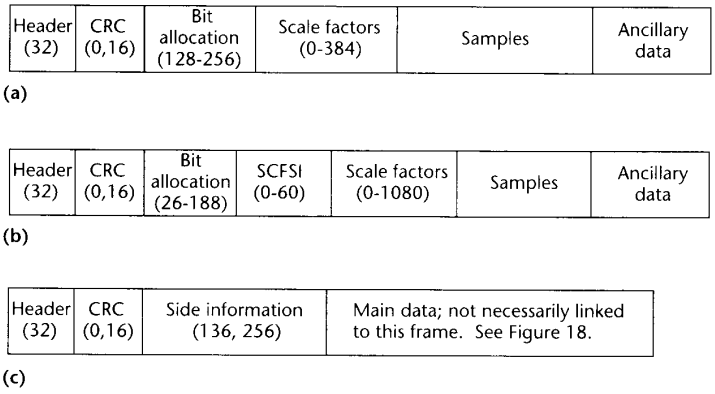

Figure 15. Frame formats of the three MPEG/audio layers' bitstreams: (a) Layer I, (b) Layer II, and (c) Layer III.

Layer III

The Layer III algorithm is a much more refined approach derived from ASPEC (audio spectral perceptual entropy coding) and OCF (optimal coding in the frequency domain) algorithms.[12,15,16] Although based on the same filter bank found in Layer I and Layer II, Layer III compensates for some filter-bank deficiencies by processing the filter outputs with a modified discrete cosine transform (MDCT).[17]

Figure 16 on the next page shows a block diagram of this processing for the encoder. Unlike the polyphase filter bank, without quantization the MDCT transformation is lossless. The MDCTs 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 that the inverse MDCT

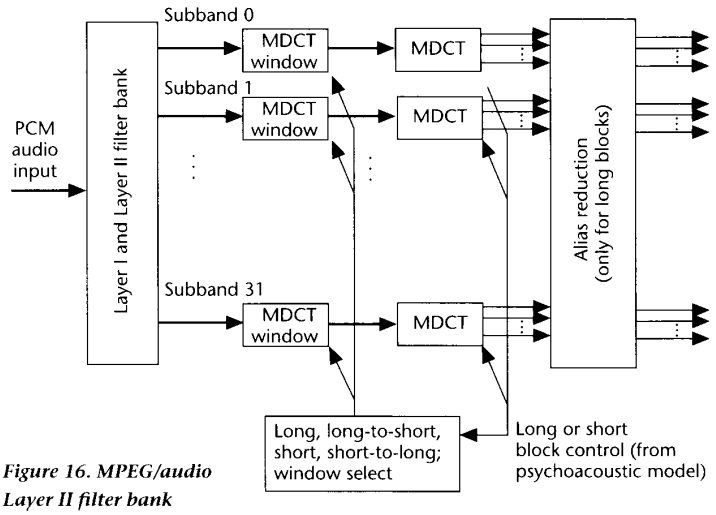

Figure 16. MPEG/audio Layer II filter bank processing (encoder side).

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.[18]

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 remains unchanged regardless of the block size selection. For a given frame of audio samples, the MDCTs can all have the same block length (long or short) or a mixed-block mode. In the mixed-block mode the MDCTs for the two lower frequency subbands have long blocks, and the MDCTs 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 17 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 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 the following.

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.

Nonuniform 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 the MPEG/audio decoder relinearizes the values by raising its output to the 4/3 power.

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

Entropy coding of data values. To get better data compression, Layer III uses variable-length Huffman codes to encode the quantized samples. After quantization, the encoder orders the 576 (32 subbands × 18 MDCT coefficients/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

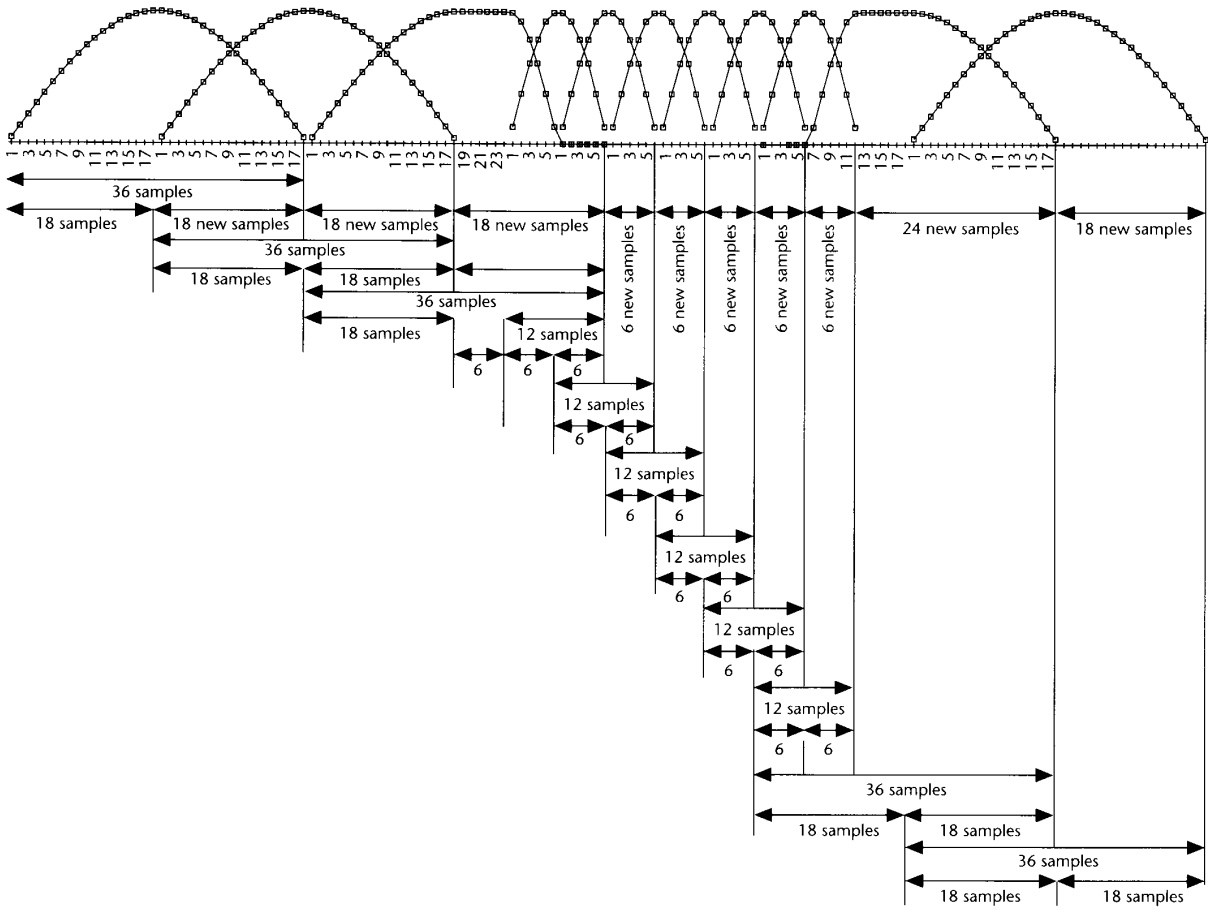

Figure 17. The arrangement of overlapping MDCT windows.

three sets of window values for a given frequency, so the ordering is by frequency, then by window, within each scale-factor band. Ordering is advantageous because large values tend to fall at the lower frequencies and long runs of zero or near-zero values tend to occupy the higher frequencies.

The encoder delimits the ordered coefficients into three 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.

The second region, the "count1" region, consists of a continuous run of values made up only of –1, 0, or 1. The Huffman table for this region codes four values at a time, so the number of values in this region must be a multiple of four.

The third region, the "big values" region, covers all remaining values. The Huffman tables for this region code the values in pairs. The "big values" region is further subdivided into three subregions, each having 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 1,152 samples. Figure 15c shows the arrangement of the various bit fields in a Layer III bitstream.

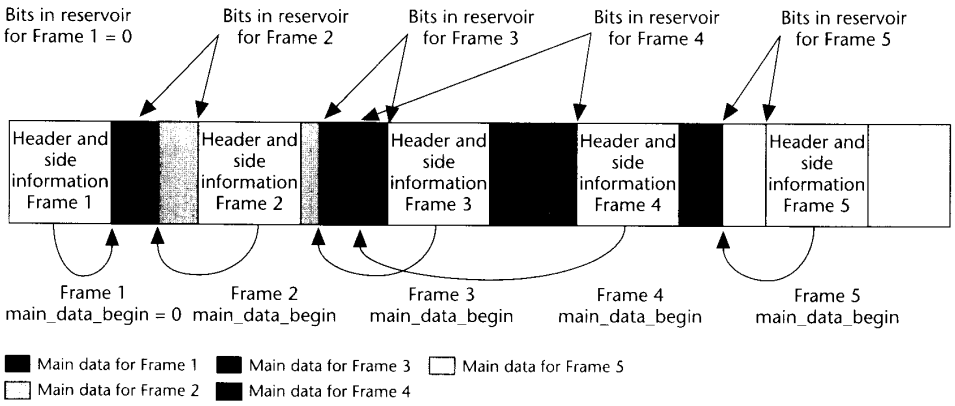

Figure 18. Layer III bitstream diagram.

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 fewer 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 pointing to the location of the starting byte of the audio data for that frame. Figure 18 illustrates the implementation of the bit reservoir within a fixed-length frame structure via the main_data_begin pointer. Although main_data_begin limits the maximum variation of the audio data to 29 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 Kbps with an audio sampling rate of 48 kHz requires an average number of code bits per frame of 1,152 (samples/frame) × 320,000 (bits/second) / 48,000 (samples/second) = 7,680 bits/frame, absolutely no variation is allowed for this coding mode.

Despite the conceptually complex enhancements to Layer III, a Layer III decoder has only a modest increase in computation requirements over a Layer II decoder. For example, even a direct matrix-multiply implementation of the inverse MDCT 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.

Bit allocation

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

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

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.

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. In addition, designers can 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, some scale-factor bands still have more than the allowed distortion, the encoder amplifies the values in those scale-factor bands and effectively decreases the quan-

tizer step size for those bands. Then the process repeats. The process stops if any of the following three conditions is true:

1. None of the scale-factor 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 scale-factor bands to be amplified.

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

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 on 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 differs.

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.

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 1994. It further extends the first MPEG standard in the following ways:

- Multichannel audio support. The enhanced standard supports up to five high-fidelity audio channels, plus a low-frequency enhancement channel (also known as 5.1 channels). Thus, it will handle audio compression for high-definition television (HDTV) or digital movies.
- Multilingual audio support. It supports up to seven additional commentary channels.
- Lower, compressed audio bit rates. The standard supports additional lower, compressed bit rates down to 8 Kbps.
- Lower audio sampling rates. Besides 32, 44.1, and 48 kHz, the new standard accommodates 16-, 22.05-, and 24-kHz sampling rates. 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 an MPEG-1/audio bitstream. Information needed to recover the original left, right, and remaining channels fits into the ancillary data portion of an 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. **MM**

Acknowledgments

I wrote most of this article while employed at Digital Equipment Corporation. I am grateful for the funding and support this company provided for my work in the MPEG standards. I also appre-

ciate the many helpful editorial comments given by the many reviewers, especially Karlheinz Brandenburg, Bob Dyas, Jim Fiocca, Leon van de Kerkhof, and Peter Noll.

References

1. ISO/IEC Int'l 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 et al., "The ISO/MPEG-Audio Codec: A Generic Standard for Coding of High Quality Digital Audio," 92nd AES Convention, preprint 3336, Audio Engineering Society, New York, 1992.
3. D. Pan, "Digital Audio Compression", *Digital Technical J.*, Vol.5, No.2, 1993.
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, Audio Engineering Society, New York, 1991.
6. G. Plenge, "A Versatile High Quality Radio System for the Future Embracing Audio and Other Applications," 94th AES Convention, Audio Engineering Society, New York, 1994.
7. C. Grewin and T. Ryden, "Subjective Assessments on Low Bit-rate Audio Codecs," *Proc. 10th Int'l AES Conf.*, Audio Engineering Society, 1991, pp. 91-102.
8. J.H. Rothweiler, "Polyphase Quadrature Filters—A New Subband Coding Technique," *Proc. Int'l Conf. IEEE ASSP*, 27.2, IEEE Press, Piscataway, N.J., 1983, pp. 1280-1283.
9. H.J. Nussbaumer and M. Vetterli, "Computationally Efficient QMF Filter Banks," *Proc. Int'l Conf. IEEE ASSP*, IEEE Press, Piscataway, N.J., 1984, pp. 11.3.1-11.3.4.
10. 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.
11. B. Scharf, "Critical Bands," in *Foundations of Modern Auditory Theory*, J. Tobias, ed., Academic Press, New York and London, 1970, pp. 159-202.
12. J.D. Johnston, "Transform Coding of Audio Signals Using Perceptual Noise Criteria," *IEEE J. on Selected Areas in Comm.*, Vol. 6, Feb. 1988, pp. 314-323.
13. D. Wiese and G. Stoll, "Bitrate Reduction of High Quality Audio Signals by Modeling the Ears' Masking Thresholds," 89th AES Convention, preprint 2970, Audio Engineering Society, New York, 1990.
14. K. Brandenburg and J. Herre, "Digital Audio Compression for Professional Applications," 92nd AES Convention, preprint 3330, Audio Engineering Society, New York, 1992.
15. K. Brandenburg and J.D. Johnston, "Second Generation Perceptual Audio Coding: The Hybrid Coder," 88th AES Convention, preprint 2937, Audio Engineering Society, New York, 1990.
16. K. Brandenburg et al., "ASPEC: Adaptive Spectral Perceptual Entropy Coding of High Quality Music Signals," 90th AES Convention, preprint 3011, Audio Engineering Society, New York, 1991.
17. J. Princen, A. Johnson, and A. Bradley, "Subband/Transform Coding Technique Based on Time Domain Aliasing Cancellation," *Proc. Int'l Conf. IEEE ASSP*, IEEE Press, Piscataway, N.J., 1987, pp. 2161-2164.
18. B. Elder, "Coding of Audio Signals with Overlapping Block Transform and Adaptive Window Functions," *Frequenz*, Vol. 43, 1989, pp. 252-256 (in German).

Davis Pan is a principal staff engineer in Motorola's Chicago Corporate Research Labs, where he heads a team working on audio compression and audio processing algorithms. He received both a bachelor's and a master's degree in electrical engineering from the Massachusetts Institute of Technology in 1981, and a PhD in electrical engineering from the same institute in 1986.

Pan has worked in the MPEG/audio compression standards since 1990 and led the development of the ISO software simulations for the MPEG/audio algorithm.

Readers may contact the author at Motorola, Inc., 1301 E. Algonquin Rd., Schaumberg, IL 60196, e-mail pan@ukraine.corp.mot.com.