Guide to MPEG-1 Audio Standard

Seymour Shlien
Communications Research Centre
P.O. Box 11490, Station H
Ottawa, Ontario
Canada, K2H 8S2

Abstract

This paper is designed to help the reader to unravel the ISO MPEG-1 standard for compressing digital audio data. In comparison to the standards for video, little additional information is available on this standard.

The standard describes the implementation of three compression schemes called layer 1, layer 2 and layer 3. The layer 1 scheme is the simplest to implement but its efficiency is not as good as the other layers. The layer 3 scheme provides the best performance at low bit rates but it is also the most difficult one to implement.

The compression techniques use psychoacoustic models for predicting the human auditory response to the noise that is introduced by the coding scheme. Using these models, the characteristics of the compression scheme can be changed dynamically in order to minimize the audibility of these noise impairments. The implementation of two such psychoacoustic models are described in the standard.

The MPEG-1 standard is very complicated and this paper is intended to provide further information to help the reader understand this standard.

I. Overview

A. Introduction

Standards are developed to allow different manufacturers to build and sell compatible components (eg. FAX machines, televisions, compact discs, etc.) When there is one standard, consumers are more willing to buy a product, as they have more confidence that it will be supported over its lifetime by one or more companies. The International Organization for Standardization (ISO) has developed a standard for compressing high-quality digital audio [5].

Digital audio broadcast is a new technology which is expected to replace the current mode of FM broadcast within the next twenty years. It should allow radio stations to broadcast compact-disc-quality audio without any increase in power or bandwidth requirements. Development of a compression standard is a prerequisite for the establishment of such a communication system.

In comparison with the compression of digital video, the compression of digital audio is relatively complex. First, the human ear has a sensitivity over a dynamic range exceeding 100 dB. The psychoacoustic models of the human auditory system are nonlinear and complex. The quality of audio reproduction equipment has advanced to the stage where even unsophisticated consumers have developed demanding tastes. In contrast, the dynamic range and the resolution of television broadcasts are still far below the capacities of the human visual system.

The audio coding standard is a complex coding scheme offering three layers of compression. Each of these layers may be viewed as a distinct compression scheme with increasing complexity. The standard contains provisions for coding stereo information, error protection, pre-emphasis, and ancillary data. The standard begins with the layout of the bitstream for all three layers. This is followed with an implementation of the decoder, the encoder for the three layers and the two psychoacoustic models. The standard concludes with appendices discussing diverse topics such as error concealment.

The purpose of this guide is to provide a detailed overview of the standard as well as to provide some clarification to some of the steps in the implementation of the standard. Ideally, this guide should be treated as an accompaniment to the standard, as many of the terms used in the guide are the same ones defined in the standard. The guide does not attempt describe the standard exhaustively and neither does it provide the reader with all the background material in signal processing and psychoacoustics. Instead, the guide attempts to provide some intuitive grasp of the the algorithms and functions by means of the 30 or more figures in this document. Other detailed overviews to the standard are beginning to appear [3], [8], [10] and the reader is urged to refer to these papers.

The basis of the standard is frequency domain coding which is described in the first section. The protocol for quantizing and encoding the subband data in layer 1 and 2 is described in the next two sections. This is followed by a description of the two psychoacoustic models suggested by the standard. The remainder of the guide describes the layer 3 compression scheme.

B. Background

Any lossy compression scheme introduces a certain amount of quantization noise which is dependent upon the bit rate of the coding scheme. Using adaptive bit allocation algorithms, the ISO standard attempts to control the spectral properties of the noise as a function of the signal so that the noise remains imperceptible. The noise spectrum is shaped by decomposing the audio signal into 32 equal-width subbands. (This scheme is different for layer 3 but the same in principle.) The signals in each of the subbands are adaptively quantized and encoded so that the noise introduced is still below the masking threshold. The masking thresholds are dependent upon the audio signal and are determined from the spectral properties of the signal using a psychoacoustic model.

Much of our current model of the human auditory system is based upon the experimental work of Zwicker and Feldtkeller [17] in the late 1950's. The model describes how loud signals conceal fainter signals which are in proximity in both time and frequency. The model partitions the frequency scale into 24 critical bands within which the masking effects apply. The masking effects depend upon whether the source is tone-like or noise-like as well as its frequency and loudness. Since the loudness of the reproduction is user-selectable, it is necessary to design the psychoacoustic model for the worst case.

Presently, we must still rely on subjective testing for the final evaluation of audio compression schemes. Though considerable progress has been made in understanding the human auditory system, we have yet to find an algorithm which can reliably predict the audibility of certain types of noise inserted in the audio signal. Furthermore, listeners differ considerably in their ability to detect such artifacts. The development of a good human auditory model has become an important research topic for this area.

0018-9316/94$04.00 © 1994 IEEE

II. Frequency Domain Coding

The audio signal is divided into 32 subbands by passing it through a filter bank and decimating the outputs by a factor of 32. Assuming ideal filters with a rectangular frequency response, the Nyquist theorem guarantees that the original signal can be reconstructed exactly by interpolating the subband signals to their original sampling frequency and summing the results.

Since it is not possible to construct filters with perfectly flat response in the pass band and zero output elsewhere, aliasing effects can be introduced during the decimation process and this results in some information loss [14]. Two approaches, Frequency Domain Alias Cancellation (FDAC) and Time Domain Alias Cancellation (TDAC) have been developed to reduce these effects [15], [7]. The MPEG audio standard uses a variant of the TDAC technique described elsewhere – [11], [2].

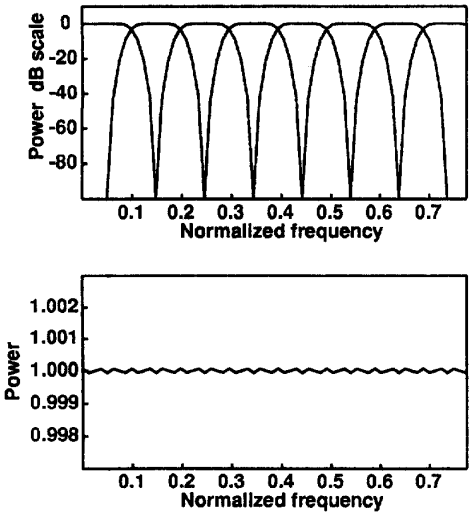

Fig. 1. Top: frequency response of the first 8 subband filters. Bottom: frequency response of the analysis filter bank. The frequency has been normalized to the Nyquist frequency 2π.

MPEG utilizes a set of 512 tap FIR filters during both the encoding (analysis) and decoding (synthesis) operations. These alias cancellation filters were designed to have high side-lobe attenuation as shown in Figure 1. Efficient implementation of the filters using a polyphase structure [12] for reducing the computations is described in the standard. It was recently shown [9] that even more efficient implementations can be achieved using, for example, the Fast Discrete Cosine Transform (FDCT).

Figures 2 and 3 illustrates the MPEG polyphase implementation of the analysis and synthesis filter banks. During the analysis phase (top), the audio signal is shifted into a 512 sample X buffer, 32 samples at a time. The contents of the buffer are multiplied by a C window function (tabulated in the standard) and the results are recorded into the Z buffer. The contents of the Z buffer are divided into 8 64-element vectors which are summed to form a Y vector. The Y vector is transformed using a variant of the Modified Discrete Cosine Transform (MDCT) to yield the desired 32 subband values.

In the synthesis operation, the 32 subband values are transformed back to their 64 value V vector using a variant of the Inverse Modified Discrete Cosine Transform (IMDCT). The V vector is pushed into a fifo which stores the last 16 V vectors. A U vector is created from the alternate 32 component blocks in the fifo as illustrated and a window function D is applied to U to produce the W vector. The reconstructed samples are obtained from the W vector by decomposing it into 16 vectors each 32 values in size and summing these vectors. It should be noted that the C and D window functions tabulated to 10-figure accuracy in the standard differ merely by a factor of 32.00000. The C window function is shown in Figure 4.

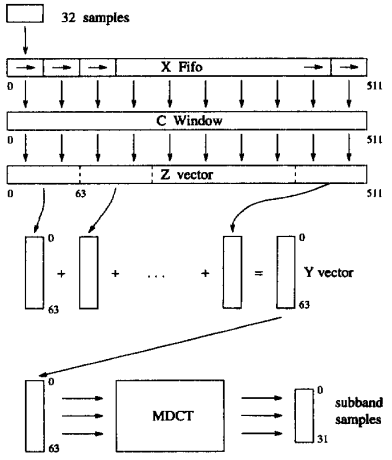

Fig. 2. Polyphase implementation of the analysis filter bank.

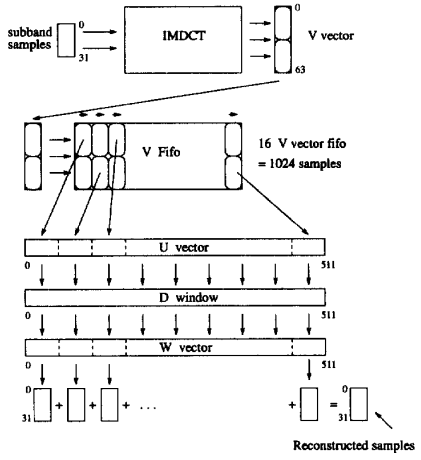

Fig. 3. Polyphase implementation of the synthesis filter bank.

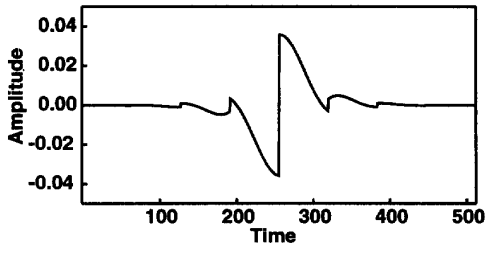

Fig. 4. Analysis window function C for polyphase filter bank.

As evident from the impulse response functions in Figure 5, the subband filter functions do not have linear phase. Nevertheless, the cascade of the analysis and synthesis filters for each subband channel is a linear phase filter with length 1023. Therefore, the whole filter bank is linear phase and the uncancelled aliasing will only result in amplitude distortion [7].

The frequency responses of the subband analysis filters are shown in the top half of Figure 1. Though the aliasing errors of the filters appear substantial, they are cancelled out during reconstruction as the filter responses sum up almost exactly to one (bottom half of Figure 1).

The analysis filter bank splits the input stream of audio samples into 32 output streams where each output stream has been

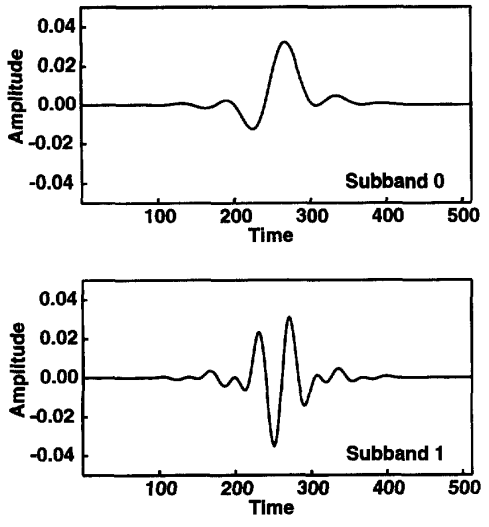

Fig. 5. Impulse response function of the first two subband filters in the polyphase filter bank.

decimated by a factor of 32. As a result the number of samples going out of the filter bank is the same as the number of samples coming into the filter bank. The samples are then packaged into frames containing a fixed number of samples (384 or 1152) depending on the layer of the compression scheme. In each frame, the subband samples are scaled and quantized according to a psychoacoustic model. Since the scaling factor and number of quantization levels vary with time and subband number, both of these must be transmitted along with the quantized samples for each of the subbands in every frame.

III. Format of the MPEG Data

A. Introduction

The MPEG audio compression scheme uses an adaptive bit allocation scheme where the scale factors and number of bits per sample vary from frame to frame as determined by a psychoacoustic model. In layer 1, a frame encodes 384 samples or 12 samples per subband. In layers 2 and 3, a frame contains 1152 samples (ie. 36 samples per subband).

Each frame begins with a 32-byte header block whose format is the same in all layers. The purpose of the header block is to allow receiver to establish synchronization and to determine the basic coding parameters anytime during a broadcast. Some of the key parameters are the sampling rate, the bit rate, the layer number, the channel parameters (monophonic or stereo) and the existence of error protection codes. The sampling rate is restricted to either 32.0, 44.1 or 48.0 kHz. The bit rate is also restricted to certain choices in the range of 32 to 448 kilobits/second and depend upon the sampling rate and channel parameters.

If error protection is specified, then the header block is immediately followed by a 16-bit CRC check word. The audio block which follows next contains a frame of the audio encoded data. This is followed by the the ancillary data block which is user-definable and may contain any application-related information. The format of the audio block depends upon the compression layer as discussed in the other sections.

The synchronization code in the header block contains a 12-bit of ones. Some attempt was made to avoid the accidental appearance of this code in the audio blocks of layer 1 and 2.

Stereo information may be encoded in one of three modes – stereo, dual and joint stereo. In both stereo and dual channel modes, the two channels are transmitted independently in the same format without any attempt to remove the redundancies. The stereo mode is used to transmit the left and right channels of a broadcast while the dual channel mode is used to transmit different material in the two channels as in a bilingual broadcast.

The joint stereo mode provides two methods for reducing the redundancies in a stereo broadcast. The first method called intensity stereo is the only option for the layer 1 and 2 coding schemes. In layers 1 and 2, the scale factors are sent as usual for both channels. In layer 3, only the right channel scale factors are sent; also, the quantized subband samples are treated differently. For all subbands up to a certain bound depending on the mode extension, the left and right channels are sent separately; for subbands including and beyond this bound, only the sum of the left and right audio samples are sent.

In MS stereo, which is only available in layer 3, the redundancies between the two channels is exploited by transforming the left and right channels to sum and difference signals [16] and [4].

Though humans are able to localize the spatial source of sound based on both intensity and time differences of the signals at both ears [1], the recording industry mostly relies on the intensity differences.

B. Layer 1

Each frame contains the last 12 decimated samples from each of the 32 subbands from the filter bank (Figure 6). For each of the subbands in the frame, the 12 samples are scaled so that the sample with the largest magnitude is closest to but does not exceed one. The psychoacoustic model and the required bit rate are used to compute the bit allocation per sample for each subband. The scaled samples are then quantized into the number of levels as determined by the bit allocation. The bit allocation, scale factors and samples are all encoded and placed in the three designated areas of the frame.

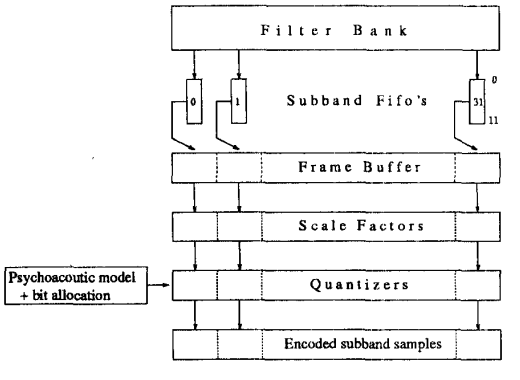

Fig. 6. Flow diagram for encoding a frame of audio data in layer 1.

The bit allocation section indicates (through a lookup table) the word length assigned to each sample in the subband and ultimately determines the number of levels of the linear quantizer. Increasing the number of levels reduces the quantization noise but increases the number of bits required to encode the

samples as well as the overall bit rate. The bit allocation section contains 32 4-bit values which allows a choice of 15 different quantizers for each subband. (To avoid any conflicts with the synchronization code, code '1111' is defined to be illegal.) Except for the case where the bit allocation value is zero (ie. the subband is not transmitted), the number of bits allocated to the subband samples is one larger than the value recorded in the layer 1 frame.

The scale factor section contains 32 6-bit values each of which indexes one of 63 values in the scale factor table. (Again code '111111' is illegal.) The scale factor is used to multiply the requantized sample value. Its value is dependent on the amplitude of the subband filtered signal. The scale factors in the table increase by a factor of $\sqrt[3]{2}$ or approximately 2 dB.

For each of the 12 samples, the 32 (or less) quantized subband values are encoded in the sample section. The standard uses an n level mid-tread quantizer where n is one less than the antilog base 2 of the numbers of bits assigned to the subband. As an example, a 7-level quantizer is shown in Figure 7 for a 3-bit per sample quantizer. The method for performing the quantization and inverse quantization in the standard avoids long strings of 1's which can be confused with the synchronization code, and it is also computationally efficient by using binary shifts.

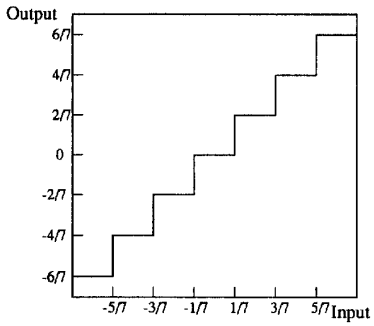

Fig. 7. An example of a 7 level quantizer for layer 1 and 2. The sample codes 000, 001, ...110 are dequantized into the values -6/7, -4/7, ...6/7, respectively.

C. Layer 2

The basic approach is the same; however, significant savings in bit rate result from increasing the frame size to 36 samples per subband, and coding the bit allocation and scale factor data more efficiently. Finer quantization, now up to 65535 levels, is available.

A frame now consisting of 36 samples per subband is divided into three called *part 0*, *part 1* and *part 2*. Each part contains 12 samples per subband like the frames in layer 1. Though the bit allocation section applies to all three parts, separate scale factor data can be sent for each part or one set of scale factor data can apply to two or more parts together. The scale factor section now contains a two bit *scale factor selection information - sfsi* which indicates whether one, two or three scale factors are transmitted for each subband and how they are applied. Thus the layer 2 frames are still able to handle large transients, by the proper use of the *sfsi*.

The bit allocation section has also been reduced by limiting the choices of quantizers for the higher subbands and lower bit rates. Instead of always sending 4 bits per subband in order to specify the bit allocation choice, the number of bits varies from 0 to 4 as a function of subband number as dictated by the table appropriate to the given sampling frequency and bit rate.

The samples are quantized and coded in a method similar to that used for layer 1, however, now there is provision for packing three consecutive samples in a single code word for certain quantizers. This reduces the number of wasted bits when the number of levels in the quantizer is not exactly a power of two.

IV. Encoding Layers I and II

A. Introduction

The MPEG audio standard is able to maintain CD audio quality reproduction up to a compression ratio of 5 to 1 or better. This bit rate corresponds to encoding each audio sample with an average of 3 bits per sample. For a PCM encoded signal, this corresponds to a nominal Signal to Noise Ratio of 18 dB. Considering that the average listener has no difficulty hearing the tape hiss on a good audio system (-60 dB below the audio level), this is a remarkable achievement.

The MPEG coding scheme achieves this compression by placing the quantization noise in the frequency subbands where the ear is least sensitive. The psychoacoustic model determines from the input audio the maximum noise level which would just be perceptible (masking level) for each of the subbands. As the amount of the quantization noise is directly related to the number of bits used by the quantizer, the bit allocation algorithm assigns the available bits in a manner which minimizes the audible distortion.

B. Scale Factor

As described in the standard, the maximum of the absolute value of 12 samples is determined. The next largest value of the scale factor in the scale factor table is chosen.

In layer 2, three scale factors are computed for each subband – one for each part. The differences between the first two and last two scale factors are determined. Based on the two differences a decision is made on whether to encode all three scale factors or to replace them with one or two scale factors.

C. Bit Allocation

Both psychoacoustic models described in the standard return the Signal to Mask Ratio (SMR) for each subband. The bit allocation algorithm computes the the Mask to Noise Ratio (MNR) from the SMR and the Signal to Noise Ratio (SNR) using the following expressions. (The SNR is given as a function of bit allocation in tables provided in the standard.)

$$MNR = SMR - SNR$$

The number of bits available to encode a frame is determined from the desired bit rate and the sampling rate.

$$Bits/frame = \frac{Bits/second}{frames/second}$$

$$frames/second = \frac{samples/second}{samples/frame}$$

Next one subtracts 32 bits for the header block, 16 bits for the CRC check-word if used, the number of bits used by the bit allocation data and any bits used for the ancillary data. The bit allocation algorithm now attempts to maximize the minimum MNR of all subbands by assigning the remaining bits to the scale factors and sample data. No bits are needed for the scale factors if zero bits are assigned to that subband; otherwise, the number of bits depends on layer number and on whether 1, 2 or 3 scale factors are sent for that subband. Initially, the algorithm assumes zero bits are assigned to each subband. The algorithm computes the SNR and MNR for each subband and finds the subband with the lowest MNR whose bit allocation has not reached its maximum limit. The bit allocation for that

subband is increased one level and the number of additional bits required is subtracted from the available bit pool. The process is repeated until all the available bits have been used or all the subbands have reached their maximum limit. Figure 8 shows the MNR function before and after executing the bit allocation algorithm. The rate distortion curve (minimum MNR versus number of bits) computed for the same frame is shown in Figure 9.

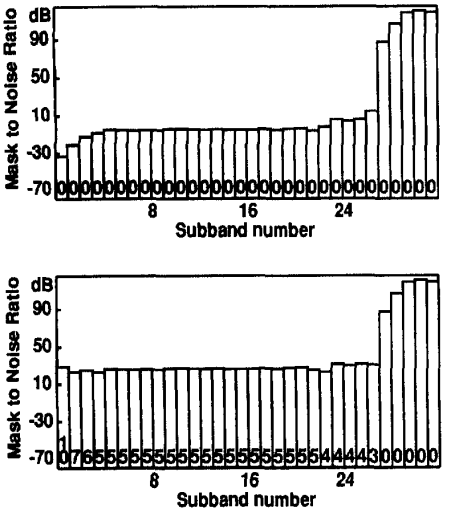

Fig. 8. Mask to noise ratio at the beginning (top) and end (bottom) of the bit allocation algorithm. The number of bits assigned to each subband sample appears at the base of each bar.

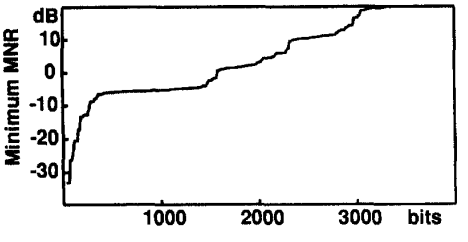

Fig. 9. Rate distortion curve: minimum MNR versus number of bits required to encode a layer 1 frame determined for a particular frame.

V. Psychoacoustic Models

The standard describes the implementation of two sample psychoacoustic models. The first model is designed to be computationally simple and to provide adequate accuracy at high bit rates. The second model is more complex and is recommended at the lower bit rates. Both models require computing the Fourier power spectrum of the signal, mapping it to the critical band domain, distinguishing tonal and noise components, applying spreading functions to these components, computing the masking function and mapping it back to the Fourier spectral domain and the subband domain. Tables of the mapping functions and auditory thresholds and parametric representations for the spreading functions and masking functions are supplied for each psychoacoustic model and for all sampling rates. Furthermore, special modifications to psychoacoustic model 2 specifically for the layer 3 compression mode are given.

A. Psychoacoustic Model I

The auditory spectrum is approximated by a list of tonal and non-tonal components. The masking function is computed by summing the masking effects of these components with the auditory threshold.

The auditory spectrum (Figure 10) is computed using the Fast Fourier Transform (FFT) after applying an appropriate delay and Hanning window function to the auditory signal. The size of the transform is either 512 for layer 1 or 1024 for layer 2. For better clarity, only the first 100 spectral lines are shown in the next four graphs (Figures 11 and 12).

In order to identify the tonal components, a list of all the local maxima in the spectrum is compiled (part of which is shown in the bottom half of Figure 10) and then pruned by applying a set of heuristic rules (Figure 11, top). The rules specify the relative height of the maxima in a neighbourhood whose size varies with frequency in the manner described in the standard.

All the remaining spectral lines are used for calculating the non-tonal components. They are grouped into critical bands and within each critical band, a single non-tonal component which represents the effect of these lines is computed (Figure 11, bottom).

In the next step, the list of tonal and noise components are decimated by eliminating those components which are below the auditory threshold or are less than one half of a critical band width from a neighbouring component. A sliding window with a width of 0.5 barks (or half a critical band unit) is used in this operation and only the component with the highest power is retained (Figure 12).

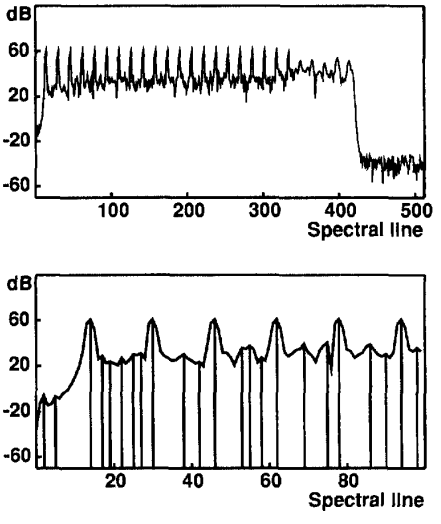

Fig. 10. Top: Fourier power spectrum of an audio signal. Bottom: magnified portion of the Fourier power spectrum; the local maxima are indicated by vertical spikes.

To compute the masking effect of a tonal or non-tonal component on the neighbouring spectral frequencies, the strength of the component (in the dB scale) is summed with two terms called the *masking index* and the *masking function*. The *masking index* is an attenuation term which depends on the critical band rate of the component and whether it is tonal or non-tonal (Figure 13). The *masking function* is another attenuation factor which depends on both the displacement of the component from the neighbouring frequency and the component's signal strength (Figure 14). Since the *masking function* has infinite attenuation

beyond -3 barks and +8 barks, the component has no masking effect on frequencies beyond those ranges.

Thus for a tonal component j, at critical band rate $z(j)$, the masking threshold $LT_{tm}(j,i)$ at critical band rate $z(i)$ is given by

$$LT_{tm}(j,i) = X_{tm}(j) + av_{tm}(z(j)) + vf[z(i) - z(j), X_{tm}(j)]$$

where $z(j)$ is the function mapping spectral frequency to critical band rate, $X_{tm}(j)$ is the strength of the tonal component at frequency j, $av_{tm}(z(j))$ is the tonal *masking index* and $vf[\cdots]$ is the *masking function*. The expression for the non-tonal masking component is identical except that av_{nm}, the non-tonal *masking index* replaces av_{tm}.

$$LT_{nm}(j,i) = X_{nm}(j) + av_{nm}(z(j)) + vf[z(i) - z(j), X_{nm}(j)]$$

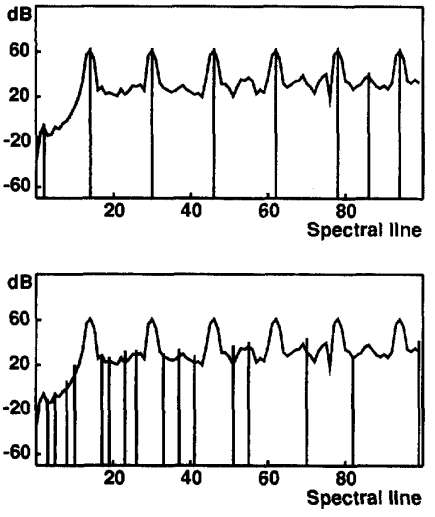

Fig. 11. Top: list of tonal components superimposed on audio Fourier power spectrum. Bottom: list of non-tonal components.

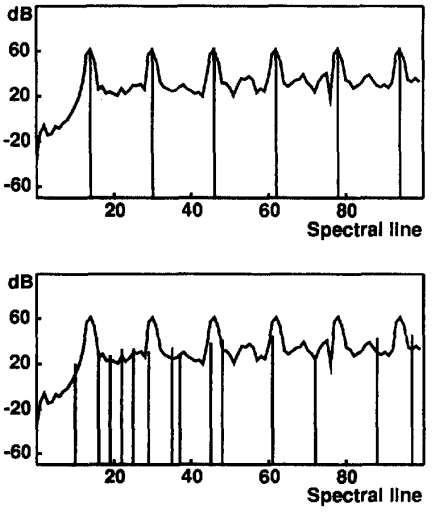

Fig. 12. Decimated list of tonal components (top) and non-tonal components (bottom).

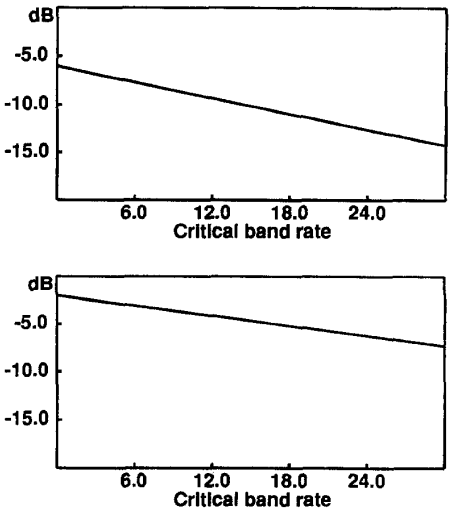

Fig. 13. *Tonal masking index* (top) and *non-tonal masking index* (bottom) for psychoacoustic model 1.

The *global masking thresholds* are computed for all spectral frequencies by adding the masking thresholds computed above for all the neighbouring tonal and non-tonal components with the threshold of hearing (Figure 15). The addition is performed in the normal square amplitude scale of the spectrum rather than the dB scale. It is curious to note that the absolute hearing threshold (called threshold in quiet in the standard) is quite different for the layer 2 model.

In the next step, the minimum masking threshold function is determined for each subband from the minimum of all the *global masking thresholds* contributing to that subband (Figure 16). Finally, the signal to mask ratio is computed for each subband (Figure 17).

B. Psychoacoustic Model II

Unlike the first model, this model does not make a dichotomous distinction between tonal and non-tonal components. In-

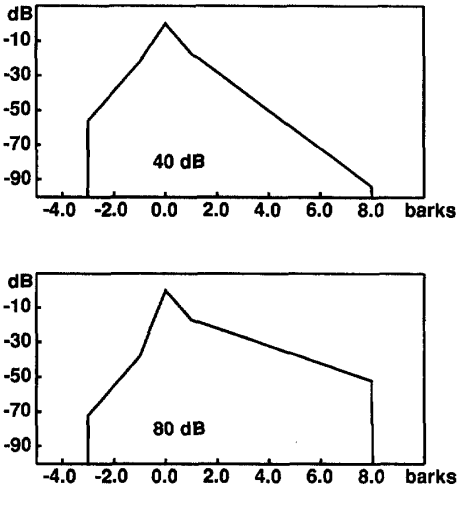

Fig. 14. Masking function for psychoacoustic model 1 for a 40 dB (top) and 80 dB (bottom) masking tone.

stead the spectral data is transformed to a "partition" domain and the fraction of the tonal and non-tonal components is estimated in each partition. This fraction ultimately determines the amount of masking.

The partition domain bears an almost linear relationship to the critical band space (Figure 18, top). The domain was designed to provide a resolution of either one frequency line or 1/3 of a critical band, whichever is wider. The mapping is given in table form in the standard and depends on the sampling rate (see for example bottom half of Figure 18 for 44.1kHz).

The auditory spectrum is computed using the 1024 point DFT after applying appropriate delay and the Hanning window function to the data (Figure 19). The tonality of the spectral lines are determined by measuring the *unpredictability* of the spectrum with time. This is done by linearly extrapolating the phase and amplitude from the two previous spectra, measuring the difference and applying suitable normalization (Figure 20). For a pure tone, one would expect the *unpredictability* to remain zero, while for a non-tonal component the measure should assume any value within the range of zero to one.

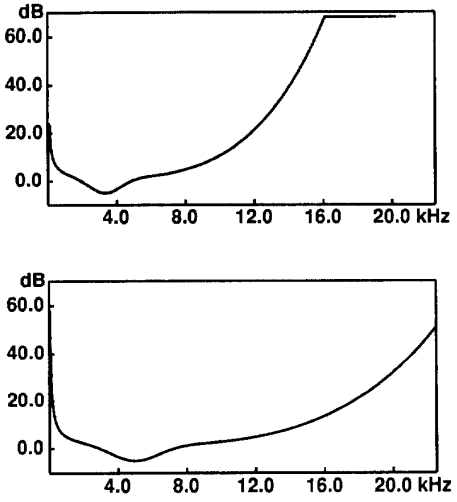

Fig. 15. Absolute threshold of hearing for layer 1 (top) and layer 2 (bottom) for 48 kHz. sampling rate.

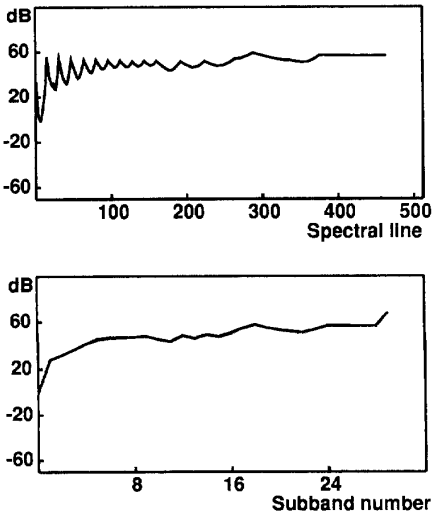

Fig. 16. Top:*global masking thresholds*. Bottom: minimum *global masking thresholds*.

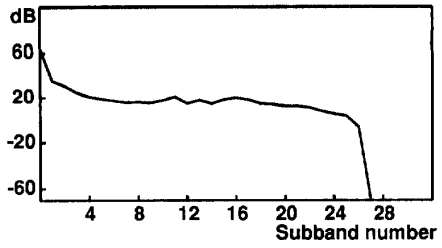

Fig. 17. Signal to mask ratio.

The auditory power spectrum and the *unpredictability weighted auditory power spectrum* are both mapped into the partition domain, yielding the *partitioned energy spectrum* and the *partitioned unpredictability* (upper and lower graphs, respectively, in top half of Figure 21). Both of these spectra are convolved by a frequency dependent spreading function (Figure 25) and then renormalized to correct for the gain introduced by spreading function (bottom half of Figure 21).

For each partition, the ratio of the convolved *partitioned unpredictability* over the convolved *partitioned energy spectrum* is

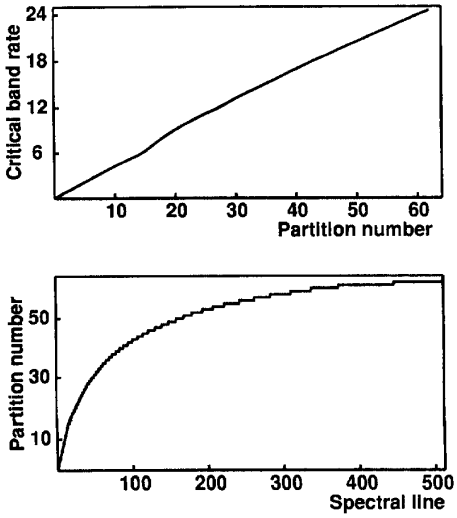

Fig. 18. Top: the relation between partition number and critical band rate. Bottom: conversion from Fourier spectral frequency to partition number at 44.1 kHz.

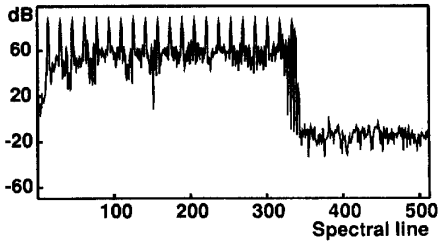

Fig. 19. Left: Fourier power spectrum of an auditory signal.

determined (Figure 22, top). The *tonality* measure is derived from the logarithm of this ratio (base 10) and restricted to the range of zero to one (Figure 22, bottom).

The masking threshold at any specific partition is equal to the *partitioned energy spectrum* multiplied by an attenuation factor. The logarithm of this factor (called *SNR(b)* in the standard where b is the partition number) is computed from the *tonality*, *tbb(b)* which interpolates the attenuation factor between the *Tone Masking Noise TMN(b)* function (Figure 26, top) and the *Noise Masking Tone NMT(b)* function (always 5.5 dB). The attenuation factor is then restricted to a maximum of *minval(b)* which is given in the standard (Figure 26, bottom).

$$SNR(b) = MAX(minval(b), TMN(b) \cdot tbb(b) + NMT(b) \cdot (1 - tbb(b)))$$

$$attenuation = 10^{-SNR(b)}$$

$$masking\_threshold = convolved\_partitioned\_energy\_spectrum \cdot attenuation$$

In the remaining steps, the masking threshold function is transformed back to the Fourier frequency scale by spreading it evenly over all the spectral lines corresponding to the partition. The threshold is raised to the absolute auditory threshold if it is below that value. Pre-echo control is applied here for layer 3 compression scheme. Finally the signal to mask ratio is computed for the subbands (in layer 1 or 2) or the scale factor bands (in layer 3) – see Figure 24.

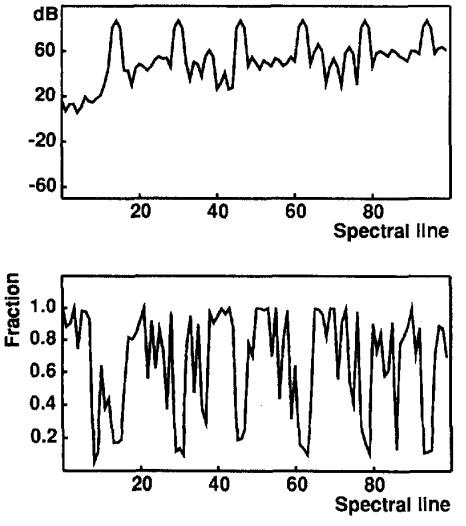

Fig. 20. Top: magnified portion of the Fourier power spectrum of the auditory signal shown in previous figure. Bottom: *unpredictability* measure for the same signal.

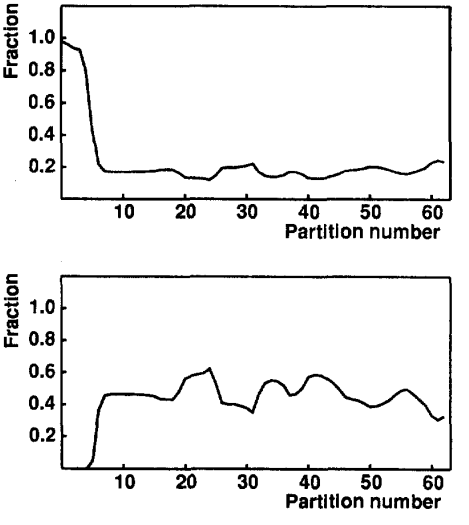

Fig. 22. Top: ratio of the convolved *partitioned unpredictability* over the convolved *partitioned energy spectrum* as a function of partition number. Bottom: *tonality* measure as a function of the partition number.

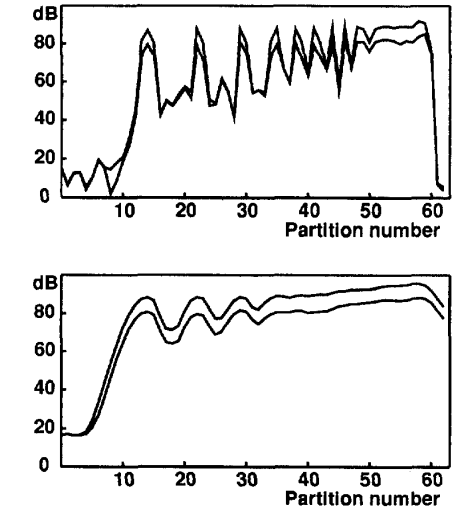

Fig. 21. Top: *partitioned energy spectrum* and *partitioned unpredictability*. Bottom: convolved *partitioned energy spectrum* and convolved *partitioned unpredictability*.

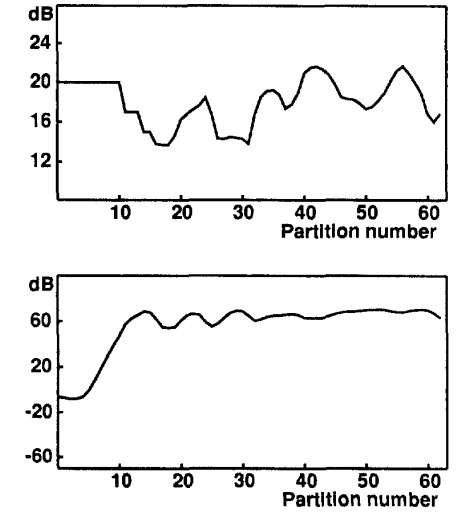

Fig. 23. Top: Attenuation factor *SNR(b)* computed from the *tonality*, masking functions and *minval* functions. Bottom: masking threshold.

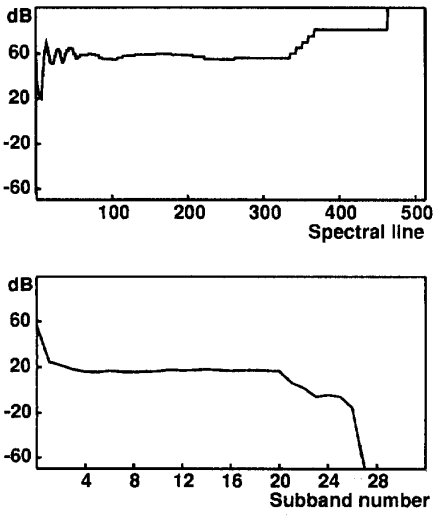

Fig. 24. Top: masking threshold function mapped back to the Fourier spectral domain. Bottom: signal to mask ratio.

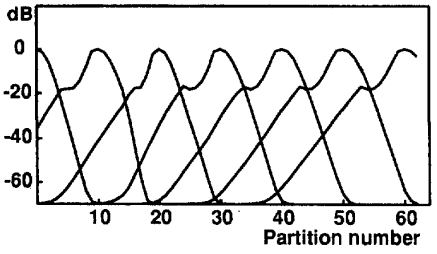

Fig. 25. Spreading function for psychoacoustic model 2 layer 2.

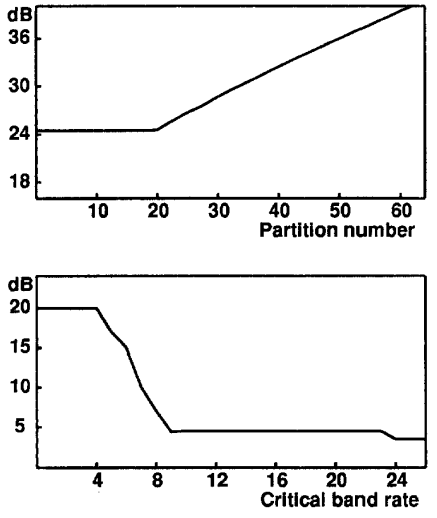

Fig. 26. Left: Tone Masking Noise function. Right: the *minval* function.

VI. Layer 3

A. Introduction

There are several new features in the Layer 3 encoding scheme. (1) Each of the 32 subbands is now split up into 18 spectral lines using the Modified Discrete Cosine Transform (MDCT) with window length 36 followed by a sequence of alias reduction butterflies. (The aliasing reduction butterflies consist of a sequence of reversible planar rotations applied to the MDCT components [7].) (2) Variable frequency/time resolution is used to control transient effects. Finally, (3) variable bit rate coding using a choice of 32 Huffman tables is used to encode the data. A frame of data now consists of two granules where each granule consists of 18 by 32 or 576 frequency components.

Time artifacts are controlled by the use of four window functions which are applied to the subband data prior to computing the MDCT. During steady state conditions, the regular window Type 0 (see Figure 27) is used. This provides the resolution of 18 spectral lines per subband per granule. Under other conditions, a narrower window (Type 2) is applied producing 6 spectral lines. The Type 2 windows are overlapped for a granule

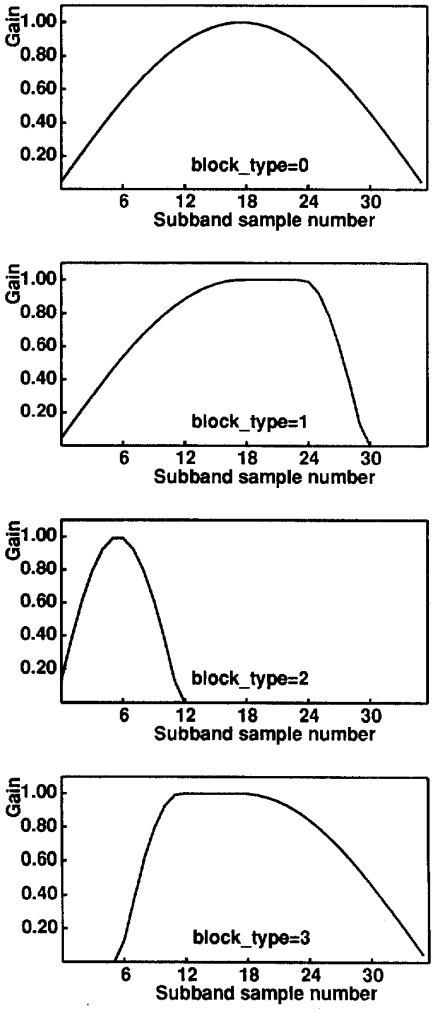

Fig. 27. The four window types applied to the subband data prior to applying the MDCT during the encoding process.

as shown in Figure 28. Two other window functions (Type 1 and Type 3) are used to handle the transitions from high to low resolution and back. The standard also has the provision for using different window types for different subbands in the same granule. More specifically, if the *mixed_block_flag* is set then the spectral frequency lines corresponding to the two lowest subbands are encoded using the regular window while the remaining spectral frequency lines are encoded using the short windows.

The bit rate is controlled by the *global_gain* which ultimately controls the quantizer step size for all the spectral lines in the granule. (As the quantizer follows a 0.75 power law, the actual step sizes vary with amplitude.) Finer control and noise spectrum shaping is achieved by manipulating the 21 or 12 scale factors which amplify specific groups of spectral lines prior to quantization. There is also a provision for pre-emphasis of the high frequency components prior to quantization.

Up to three different Huffman code tables can be specified for the spectral lines in a granule. The 576 spectral lines are split into the three regions where the Huffman code tables apply.

B. Bitstream Format

A frame of Layer 3 is of fixed length and is subdivided into 4 sections:

header	4 bytes	same as Layer I and Layer II
side information	17 or 32 bytes	one or two channels
main data		
ancillary data		

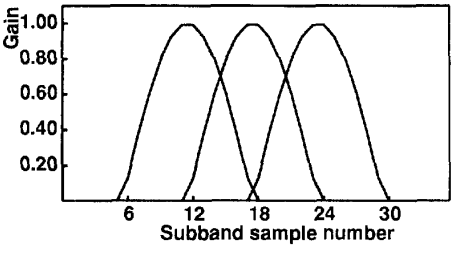

Fig. 28. This figure illustrates how the *block_type=2* windows are applied to a granule of a frame.

The side information includes the window type, the Huffman Table numbers and the regions to which they apply, and scale factor descriptors which are described below. The main data encodes the scale factors and Huffman data.

The size of the main data section can vary from frame to frame within certain limits but it is always placed in a designated area (main data area) of fixed size (determined by the bit rate). Therefore, it is possible for the main data section to extend over two consecutive main data areas; alternatively, the main data area of one frame may contain the main data sections belonging to the current frame and the next frame. (See Figure 29 for examples.) A pointer to the beginning of the main data section is transmitted in the side information section of each frame.

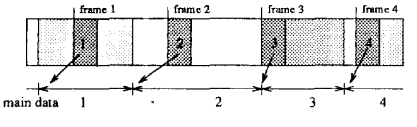

Fig. 29. This figure shows placement of the main data section in the layer 3 bitstream. Since the size of the main data section is variable, it does not always fit into the designated area of the bitstream and an additional pointer is needed to indicate where the main data section begins for a particular frame.

C. Huffman Coding

The coding scheme assumes that the large values occur at the low spectral frequencies and mainly low values and zeros occur at the high spectral frequencies. Therefore, the 576 spectral lines are partitioned into three sections. Starting from the highest frequencies, all the contiguous pairs of zero values are considered to form the *rzero* section. Next, all the contiguous quadruples consisting of values 0, 1 or -1 are assigned to the *count1* section. Finally, the remaining pairs whose absolute values can range between 0 and 8191 (*big_value* pairs) form the last section. The *big_value* pairs and the quadruples are assigned different sets of Huffman tables. There is a choice of two Huffman tables for the quadruples and a choice of 32 (only 30 are used) Huffman tables for the *big_value* pairs.

The spectral lines are transmitted starting from the lowest frequencies (*big_value* pairs first). The number of *big_value* pairs is sent in the side information section. The number of quadruples is inferred by the encoder when the Huffman data runs out. All the remaining spectral lines are assumed to belong to the *rzero* section. No Huffman encoding is perfomed on the pairs in the *rzero* section.

The Huffman tables encode the absolute values of the entries in the *big_value* pairs or quadruples. Extra bits are appended when necessary to the encoded *big_value* pairs or the quadruples to indicate the polarity of the nonzero values.

The choice of Huffman table to be used for encoding the *big_value* pairs depends largely on the dynamic range of the values. Low number tables can handle entries whose absolute value is less than a specific small number. Tables 16 to 31 are treated differently in the sense that a value equal to the maximum entry value of the table constitutes an escape code indicating that more bits, *linbits* will follow. The *linbits* form a number which must be added to the maximum entry value. Tables 16 to 23 are all identical except for the number of *linbits* which must follow the escape code (similarly for tables 24 to 31).

D. Scale Factor Bands

For purposes of transmitting the scale factors, most of the 576 spectral lines for the long MDCT are grouped into 21 *scale factor bands* and most of the 192 spectral lines for the short MDCT are grouped into 12 *scale factor bands*. The exact grouping depends on the sampling rate (either 32, 44.1 or 48 kHz.) and attempt to follow the critical band scale. When the scale factors are not sent for the high spectral lines, they are assumed to be 1.0.

Scale factors are sent either for each granule in a frame or for both granules together, depending upon the contents of the *scale factor selection information* (sfsi) variable. The 21 *scale factor bands* are assigned to four groups (0-5, 6-10, 11-15 and 16-20). For each group, a single bit flag called the *scale factor selector information* is sent. Depending on whether the flag is 0 or 1, the number of scale factors sent for the *scale factor band* is two or one (either one for each granule or one for both granules).

The number of bits used for the transmission of the scale factor is specified by a 4 bit variable called *scalefac_compress*. Now the 21 *scale factor bands* (or 12 in the case of the short transforms) are divided into two groups 0-10 and 11-20 (or 0-5 and 6-11). The *scalefac_compress* variable indexes a table which returns two numbers called *slen1* and *slen2*, the number of bits assigned to the scale factor bands in group 1 and group 2 respectively. The numbers *slen1* and *slen2* vary between 0 and 4.

E. Dequantization

The Huffman decoded values $is(i)$ are converted back to their spectral values $xr(i)$ using the following formulae.

$xr(i) = is(i)^{4/3} * 2^{A/4-B/2}$

$A = global\_gain(gr) - 210 - 8 * subblock\_gain(window, gr)$

$B = C*(scale\_factor(cb, window, gr)+preflag(gr)*pretab(cb))$
where

C is either $\sqrt{2}$ or 2 depending on whether the *scalefac(gr)* in the side information section was set to 0 or 1 respectively.

gr is the granule number (0 or 1) and window is 0 when window type function is 0, 1 or 3; otherwise it assumes a value of 0, 1, or 2 depending upon which short window (type 2) is applicable to the i th spectral sample.

cb is the *scale factor band* number (either 0 to 20 for window type functions 0, 1 or 3, or 0 to 11 for window type function 2).

subblock_gain(window) is a gain offset applied to one of the 3 short windows in the granule when applicable.

preflag is set to 0 or 1 depending on whether pre-emphasis was applied.

pretab(cb) is a table of pre-emphasis factors specified by the standard.

F. Window Functions

The subband samples are multiplied by one of four window functions (called type 0, type 1, type 2 and type 3), prior to the application of the MDCT transform. Except for type 2, all the window functions are applied to 36 subband samples. The type 2 window is only 12 points wide, but it is applied to 3 consecutive overlapping blocks of 12 samples. Plots of the window functions were shown in Figure 27.

The samples in the two granules undergo the specified window types as indicated in the side information section of the frame. Unless the *window_switching_flag* is set in the side information section, the regular window type 0 is applied to all 32 subbands. If the *window_switching_flag* is set and the *mixed_block_flag* is clear, then all 32 subbands are treated with the window block type specified in the side information. If both the *window_switching_flag* and the *mixed_block_flag* are set then the first two subbands are treated with the regular window type and the remaining subbands are treated with the window block type specified in the side information.

VII. Layer 3 Encoding

A. Introduction

The standard describes the adaption of the psychoacoustic model 2 to the layer 3 coder and outlines a possible implementation of the the encoder. Given the output parameters of the psychoacoustic model, the coder must decide whether to use short blocks (window type 2) or long blocks. The coder must ensure an orderly transition between the short and long blocks, using the intermediate window types 1 and 3. When the coder consumes fewer than its nominal number of bits in a particular granule, it accumulates a reserve of bits which can be used to handle peak demand or otherwise improve the quality of the coding. An iteration loop is executed to determine the best set of scale factors and global gain given the masking function and the spectral samples. Due to the complexity of the Huffman encoding scheme, the determination of the number of bits required for a given set of parameters is computationally involved. Not all the aspects of the coder are described; nevertheless, the description in the standard gives the implementer a very good start.

B. Adaption of the Psychoacoustic Model 2

New tables are provided for transforming from the frequency domain to the partition domain. Most of the changes are minor; however, a new spreading function is recommended (Figure 30, top) and its normalization function is given. The *Tone Masking Noise* function is now fixed at 29.0 dB for all partitions and the *Noise Masking Tone* function is now always 6.0 dB (instead of 5.5 dB). Minor changes were made to the *minval* function (Figure 30, bottom) which is now specified as a function of partition number instead of critical band number.

The psychoacoustic model returns a new value called the *perceptual entropy* which is used to select either short blocks or a long block. The *perceptual entropy* is calculated from the weighted average of the logarithm of the ratio of the masking threshold over the energy for all partitions. This ratio is then transformed to the *scale factor bands.*

The *unpredictability* measure is now computed using long FFTs (1024) for the first 6 spectral lines and short FFTs (256) for the next 200 spectral lines. For the remaining spectral lines, the *unpredictability* assumes a value of 0.4.

Pre-echo control is now incorporated by computing a new threshold based on the current threshold and the thresholds calculated for the previous two blocks.

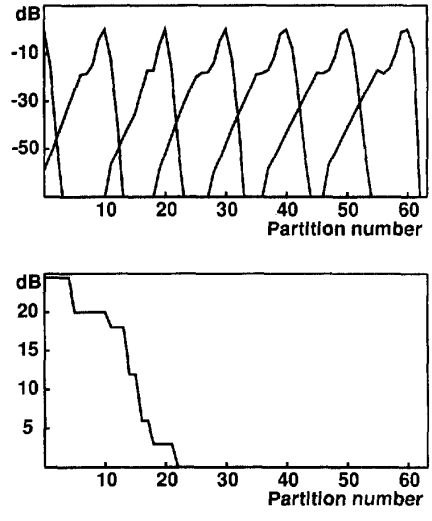

Fig. 30. The new spreading function (top) and *minval* function (bottom) recommended for the psychoacoustic model in layer 3.

C. Reserve Buffer Control

Since it is not possible for the encoder to compress the data exactly to a certain number of bits, there are usually a few unused bits left over. Sometimes the coder does not need to use all the available bits to achieve a certain minimum distortion. The encoder keeps track of the extra bits remaining in case they are needed for certain peak demands. Only a certain number of bits may be accumulated; surpluses above this limit must be discarded as stuffing bits at the end of the main_body section of the granule. The buffer is controlled by a set of rules which

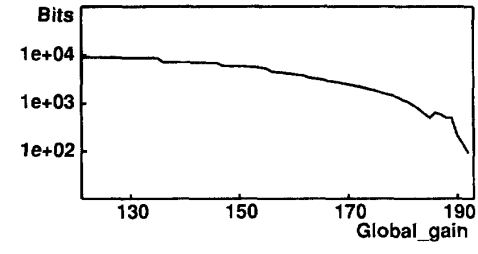

Fig. 31. An example of the relationship between the number of bits required to code a granule and the *global_gain* as determined for a particular frame.

determines how many bits may be depleted from the reserve at any specific time. The depleted bits are added to the nominal number of bits assigned to the granule yielding the quantity called *maxbits*.

D. Scale Factors and Global Gain Determination

The scale factors and *global_gain* are determined by an outer and inner iteration loop. The outer loop adjusts the scale factors to shape the noise spectrum while the inner loop sets the *global_gain* to the value which brings the number of bits to encode the granule to the value closest but not exceeding *maxbits* (Figure 31).

During the initialization stage, the maximum allowable quantization noise, *xmin*, is determined for each *scale factor band* from the ratios returned by the psychoacoustic model (Figure 32 and 33). The maximum number of bits for encoding the granule, *maxbits*, is determined and all the scale factors are initialized to their lowest values. During the iteration loop the scale factors are increased in small increments until the quantization noise is below *xmin* or until the scale factors cannot be increased any more (Figure 33). In Figure 34, the noise to *xmin* ratio is shown at the beginning and end of the iteration loop. Finally, the granule data is encoded into the bitstream and any surplus bits are added to the reserve buffer.

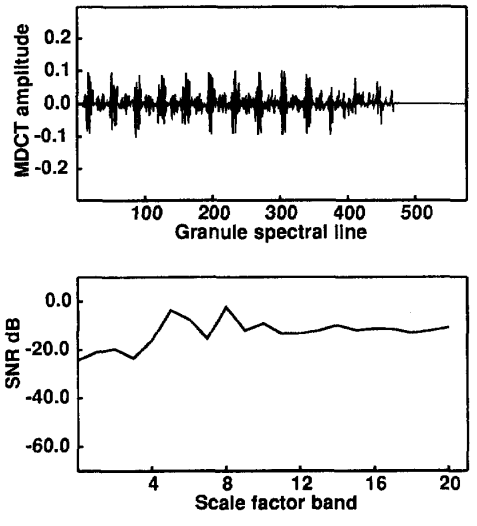

Fig. 32. Top: the 576 spectral components in a granule to be quantized and encoded. Bottom: the ratios of the scale factor threshold to scale factor energy returned by the psychoacoustic model.

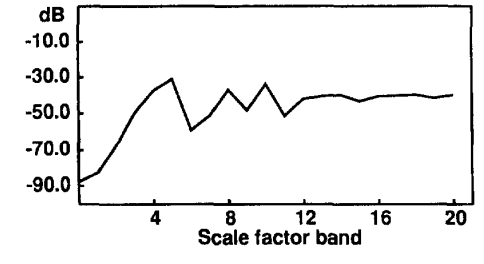

Fig. 33. *xmin*, the allowed distortion of the scale factor bands computed from the scale factor thresholds divided by the scale factor bandwidths.

The main bottleneck to the iteration occurs in the inner loop where the number of bits needed to encode the granule is determined. This involves quantizing the 576 spectral values and counting the number of bits needed to Huffman encode the quadruples and *big_value* pairs.

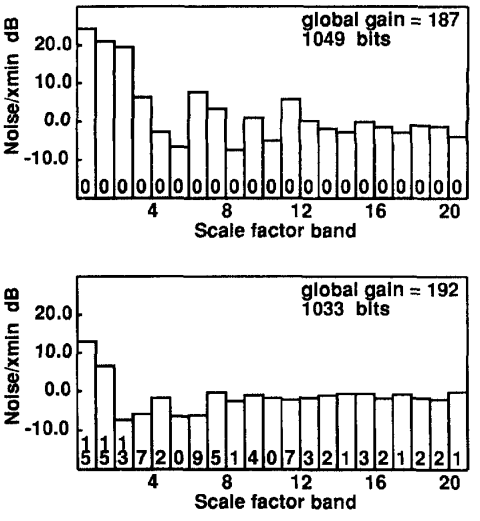

Fig. 34. Ratio of the quantization noise over the allowed distortion *xmin* computed at the start and end of the outer iteration loop. The numbers at the base of the bars are the scale factor values. The iteration attempts to reduce the largest positive deviation from zero while keeping the number of bits to encode the granule below *maxbits*.

VIII. Epilogue

The psychoacoustic models and the description of the encoder are part of the "informative" section of the standard and are not binding on the implementer, leaving room for evolutionary improvement. Current work in multiresolution filter bank design [15] and wavelets [6] may provide finer tools to model the listener's response.

The development of these audio standards in an on-going process. Current activity is now concentrated on multichannel coding (beyond stereo) and lower bit rate coding. Some of the newer techniques such as adapted wavelets [13] may lead to more efficient compression schemes.

IX. Acknowledgements

In developing this guide, I am grateful to the members of the ad-hoc simulation group who have developed and provided me with a partial software simulation of this standard. I am also grateful to Dr. Davis Pan (Digital Equipment Corporation) and Dr. Karlheinz Brandenburg (Fraunhofer-Institut für Integrierte Schaltungen), for their critical comments during various stages of preparation of this guide as well as an anonymous reviewer at McGill University. Also, I would like to thank Daniel Lauzon for help with LaTeX, Bill Treurniet for help with PostScript and Bob Warburton for help with some of the drawing packages. Ann Toth for the layout and preparation of the camera ready copy. Finally, I would like to thank Dr. William Sawchuk for granting me permission to publish this paper.

REFERENCES

[1] J. Blauert. *The Psychophysics of Human Sound Localization.* MIT Press, Cambridge, Massachusetts, 1974. Translated from German by J.S. Allen. Original title is Raumliches Horen (c) 1974 Hirzel Verlag.

[2] P.L. Chu. Quadrature mirror filter design for an arbitrary number of equal bandwidth channels. *IEEE Trans. Acoustics Speech and Signal Proc.*, ASSP-33(1):203–217, 1985.

[3] N.Jayant J.Johnston and R.Safranek. Signal compression based on models of human perception. *Proc. of IEEE*, 81(10):1385–1422, 1993.

[4] J.D. Johnston. Perceptual transform coding of wideband stereo signals. In *International Conference on Acoustics, Speech and Signal Processing*, pages 1993–1996, 1989.

[5] ISO/IEC JTCI/SC29. Information technology – coding of moving pictures and associated audio for digital storage media at up to about 1.5 mpbs – cd 11172 (part 3, audio). Doc. ISO/IEC JTCI/SC29 NO71.

[6] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI 11(7):674–693, 1989.

[7] H.S. Malvar. *Signal Processing with Lapped Transforms.* Artech House, Boston, 1992.

[8] P. Noll. Wideband speech and audio coding. *IEEE Communications Magazine*, 31(11):34–44, 1993.

[9] D.Y. Pan. Digital audio compression. *Digital Technical Journal*, 5(2):28–40, 1993.

[10] D.Y. Pan. Overview of the mpeg/audio compression algorithm. In *Proceedings of the SPIE*, volume 2187, pages 260–273, 1994.

[11] J.P. Princen and A.B. Bradley. Analysis/synthesis filter bank design based on time domain aliasing cancellation. *IEEE Trans. Acoustics, Speech and Signal Proc.*, ASSP-34(5):1153–1160, 1986.

[12] J.H. Rothweiler. Polyphase quadrature filters – a new subband coding technique. In *International Conference on Acoustics, Speech and Signal Processing*, pages 1280–1283, 1983.

[13] D. Sinha and A.H. Tewfik. Low bit rate transparent audio compression using adapted wavelets. *IEEE Trans. on Signal Proc.*, SP 41(12):3463–3479, 1993.

[14] P.P. Vaidyanathan. Quadrature mirror filter banks, m-band extensions and perfect-reconstruction techniques. *IEEE ASSP Magazine*, july 1987.

[15] P.P. Vaidyanathan. *Multirate Systems and Filter Banks.* Prentice Hall Signal Processing Series, Englewood Cliffs, 1993.

[16] R.G. van der Waal and R.N.J. Veldhuis. Subband coding of stereophonic digital audio signals. In *International Conference on Acoustics, Speech and Signal Processing*, pages 3601–04, 1991.

[17] E. Zwicker and R. Feldtkeller. *Das Ohr als Nachrichtenempfanger.* Hirzel Verlag, Stutgart, West Germany, 1967.

Seymour Shlien obtained his B.Sc. Honours in Geology and Physics at McGill University in 1968 and his Ph'D from the Department of Earth and Planetary Sciences at the Massachusetts Institute of Technology. He has worked in applications of image processing and pattern recognition at the Canada Centre for Remote Sensing (1973-1978), and since 1978 has been with Communications Research Centre in the Canadian federal government. His current research interests include signal processing and digital audio processing. He is a member of the IEEE Computer Society.