16

Audio Coding

16.1 Overview

Lossy compression schemes can be based on a source model, as in the case of speech compression, or a user or sink model, as is somewhat the case in image compression. In this chapter we look at audio compression approaches that are explicitly based on the model of the user. We will look at audio compression approaches in the context of audio compression standards. Principally, we will examine the different MPEG standards for audio compression. These include MPEG Layer I, Layer II, Layer III (or *mp3*) and the Advanced Audio Coding Standard. As with other standards described in this book, the goal here is not to provide all the details required for implementation. Rather the goal is to provide the reader with enough familiarity so that they can then find it much easier to understand these standards.

16.2 Introduction

The various speech coding algorithms we studied in the previous chapter rely heavily on the speech production model to identify structures in the speech signal that can be used for compression. Audio compression systems have taken, in some sense, the opposite tack. Unlike speech signals, audio signals can be generated using a large number of different mechanisms. Lacking a unique model for audio production, the audio compression methods have focused on the unique model for audio perception, a psychoacoustic model for hearing. At the heart of the techniques described in this chapter is a psychoacoustic model of human perception. By identifying what can and, more important what cannot be heard, the schemes described in this chapter obtain much of their compression by discarding information that cannot be perceived. The motivation for the development of many of these perceptual coders was their potential application in broadcast multimedia. However, their major impact has been in the distribution of audio over the Internet.

We live in an environment rich in auditory stimuli. Even an environment described as quiet is filled with all kinds of natural and artificial sounds. The sounds are always present and come to us from all directions. Living in this stimulus-rich environment, it is essential that we have mechanisms for ignoring some of the stimuli and focusing on others. Over the course of our evolutionary history we have developed limitations on what we can hear. Some of these limitations are physiological, based on the machinery of hearing. Others are psychological, based on how our brain processes auditory stimuli. The insight of researchers in audio coding has been the understanding that these limitations can be useful in selecting information that needs to be encoded and information that can be discarded. The limitations of human perception are incorporated into the compression process through the use of psychoacoustic models. We briefly describe the auditory model used by the most popular audio compression approaches. Our description is necessarily superficial and we refer readers interested in more detail to [97, 194].

The machinery of hearing is frequency dependent. The variation of what is perceived as equally loud at different frequencies was first measured by Fletcher and Munson at Bell Labs in the mid-1930s [96]. These measurements of perceptual equivalence were later refined by Robinson and Dadson. This dependence is usually displayed as a set of equal loudness curves, where the sound pressure level (SPL) is plotted as a function of frequency for tones perceived to be equally loud. Clearly, what two people think of as equally loud will be different. Therefore, these curves are actually averages and serve as a guide to human auditory perception. The particular curve that is of special interest to us is the threshold-of-hearing curve. This is the SPL curve that delineates the boundary of audible and inaudible sounds at different frequencies. In Figure 16.1 we show a plot of this audibility threshold in quiet. Sounds that lie below the threshold are not perceived by humans. Thus, we can see that a low amplitude sound at a frequency of 3 kHz may be perceptible while the same level of sound at 100 Hz would not be perceived.

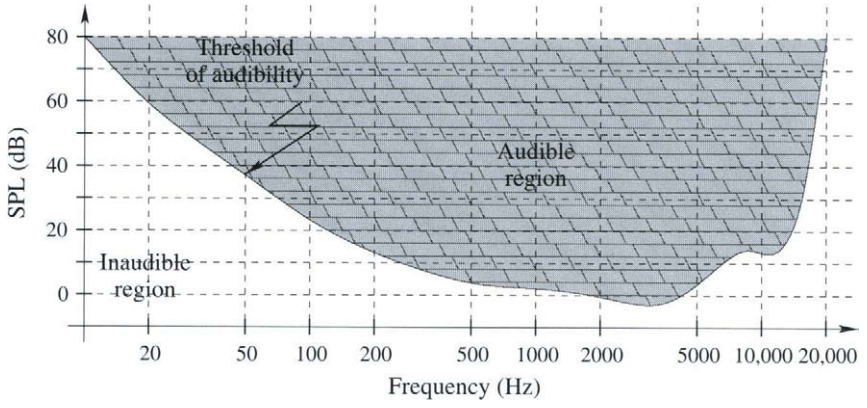

FIGURE 16.1 A typical plot of the audibility threshold.

16.2.1 Spectral Masking

Lossy compression schemes require the use of quantization at some stage. Quantization can be modeled as as an additive noise process in which the output of the quantizer is the input plus the quantization noise. To hide quantization noise, we can make use of the fact that signals below a particular amplitude at a particular frequency are not audible. If we select the quantizer step size such that the quantization noise lies below the audibility threshold, the noise will not be perceived. Furthermore, the threshold of audibility is not absolutely fixed and typically rises when multiple sounds impinge on the human ear. This phenomenon gives rise to *spectral masking*. A tone at a certain frequency will raise the threshold in a *critical band* around that frequency. These critical bands have a constant Q, which is the ratio of frequency to bandwidth. Thus, at low frequencies the critical band can have a bandwidth as low as 100 Hz, while at higher frequencies the bandwidth can be as large as 4 kHz. This increase of the threshold has major implications for compression. Consider the situation in Figure 16.2. Here a tone at 1 kHz has raised the threshold of audibility so that the adjacent tone above it in frequency is no longer audible. At the same time, while the tone at 500 Hz is audible, because of the increase in the threshold the tone can be quantized more crudely. This is because increase of the threshold will allow us to introduce more quantization noise at that frequency. The degree to which the threshold is increased depends on a variety of factors, including whether the signal is sinusoidal or atonal.

16.2.2 Temporal Masking

Along with spectral masking, the psychoacoustic coders also make use of the phenomenon of temporal masking. The temporal masking effect is the masking that occurs when a sound raises the audibility threshold for a brief interval preceding and following the sound. In Figure 16.3 we show the threshold of audibility close to a masking sound. Sounds that occur in an interval around the masking sound (both after and before the masking tone) can be masked. If the masked sound occurs prior to the masking tone, this is called premasking

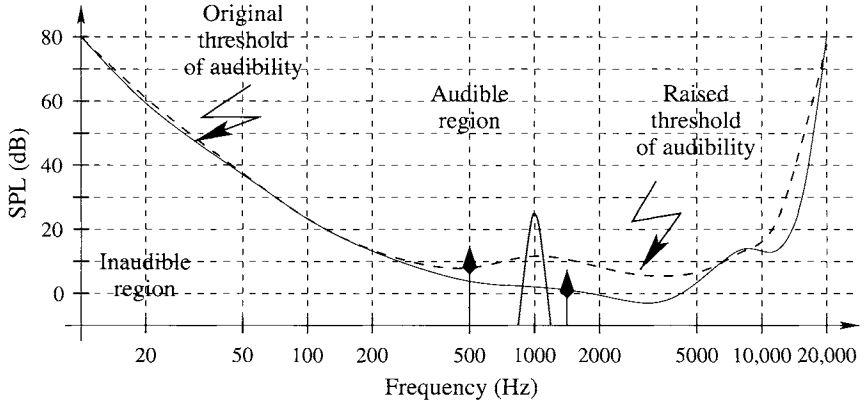

FIGURE 16.2 Change in the audibility threshold.

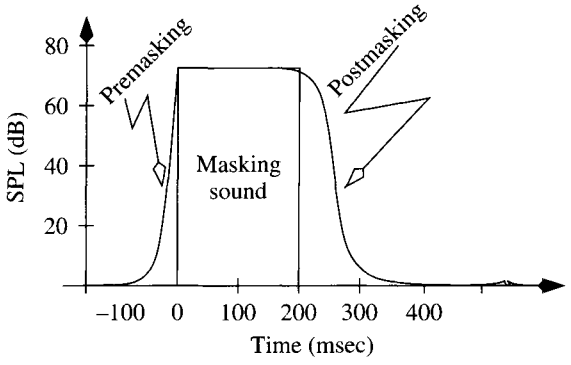

FIGURE 16.3 Change in the audibility threshold in time.

or backward masking, and if the sound being masked occurs after the masking tone this effect is called postmasking or forward masking. The forward masking remains in effect for a much longer time interval than the backward masking.

16.2.3 Psychoacoustic Model

These attributes of the ear are used by all algorithms that use a psychoacoustic model. There are two models used in the MPEG audio coding algorithms. Although they differ in some details, the general approach used in both cases is the same. The first step in the psychoacoustic model is to obtain a spectral profile of the signal being encoded. The audio input is windowed and transformed into the frequency domain using a filter bank or a frequency domain transform. The Sound Pressure Level (SPL) is calculated for each spectral band. If the algorithm uses a subband approach, then the SPL for the band is computed from the SPL for each coefficient X_k. Because tonal and nontonal components have different effects on the masking level, the next step is to determine the presence and location of these components. The presence of any tonal components is determined by first looking for local maxima where a local maximum is declared at location k if $|X_k|^2 > |X_{k-1}|^2$ and $|X_k|^2 \geq |X_{k+1}|^2$. A local maximum is determined to be a tonal component if

$$20 \log_{10} \frac{|X_k|}{|X_{k+j}|} \geq 7$$

where the values j depend on the frequency. The identified tonal maskers are removed from each critical band and the power of the remaining spectral lines in the band is summed to obtain the nontonal masking level. Once all the maskers are identified, those with SPL below the audibility threshold are removed. Furthermore, of those maskers that are very close to each other in frequency, the lower-amplitude masker is removed. The effects of the remaining maskers are obtained using a spreading function that models spectral masking. Finally, the masking due to the audibility level and the maskers is combined to give the final masking thresholds. These thresholds are then used in the coding process.

In the following sections we describe the various audio coding algorithms used in the MPEG standards. Although these algorithms provide audio that is perceptually noiseless, it is important to remember that even if we cannot perceive it, there is quantization noise distorting the original signal. This becomes especially important if the reconstructed audio signal goes through any postprocessing. Postprocessing may change some of the audio components, making the previously masked quantization noise audible. Therefore, if there is any kind of processing to be done, including mixing or equalization, the audio should be compressed only after the processing has taken place. This "hidden noise" problem also prevents multiple stages of encoding and decoding or tandem coding.

16.3 MPEG Audio Coding

We begin with the three separate, stand-alone audio compression strategies that are used in MPEG-1 and MPEG-2 and known as Layer I, Layer II, and Layer III. The Layer III audio compression algorithm is also referred to as *mp3*. Most standards have *normative* sections and *informative* sections. The *normative* actions are those that are required for compliance to the standard. Most current standards, including the MPEG standards, define the bitstream that should be presented to the decoder, leaving the design of the encoder to individual vendors. That is, the bitstream definition is normative, while most guidance about encoding is informative. Thus, two MPEG-compliant bitstreams that encode the same audio material at the same rate but on different encoders may sound very different. On the other hand, a given MPEG bitstream decoded on different decoders will result in essentially the same output.

A simplified block diagram representing the basic strategy used in all three layers is shown in Figure 16.4. The input, consisting of 16-bit PCM words, is first transformed to the frequency domain. The frequency coefficients are quantized, coded, and packed into an MPEG bitstream. Although the overall approach is the same for all layers, the details can vary significantly. Each layer is progressively more complicated than the previous layer and also provides higher compression. The three layers are backward compatible. That is, a decoder for Layer III should be able to decode Layer I– and Layer II–encoded audio. A decoder for Layer II should be able to decode Layer I– encoded audio. Notice the existence of a block labeled *Psychoacoustic model* in Figure 16.4.

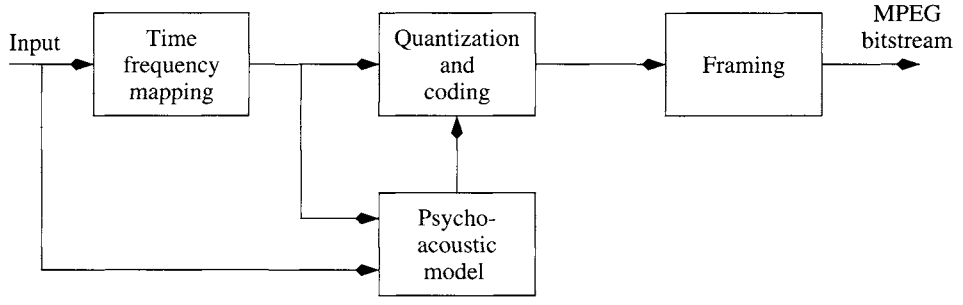

FIGURE 16.4 The MPEG audio coding algorithms.

16.3.1 Layer I Coding

The Layer I coding scheme provides a 4:1 compression. In Layer I coding the time frequency mapping is accomplished using a bank of 32 subband filters. The output of the subband filters is critically sampled. That is, the output of each filter is down-sampled by 32. The samples are divided into groups of 12 samples each. Twelve samples from each of the 32 subband filters, or a total of 384 samples, make up one frame of the Layer I coder. Once the frequency components are obtained the algorithm examines each group of 12 samples to determine a *scalefactor*. The scalefactor is used to make sure that the coefficients make use of the entire range of the quantizer. The subband output is divided by the scalefactor before being linearly quantized. There are a total of 63 scalefactors specified in the MPEG standard. Specification of each scalefactor requires 6 bits.

To determine the number of bits to be used for quantization, the coder makes use of the psychoacoustic model. The inputs to the model include an the Fast Fourier Transform (FFT) of the audio data as well as the signal itself. The model calculates the masking thresholds in each subband, which in turn determine the amount of quantization noise that can be tolerated and hence the quantization step size. As the quantizers all cover the same range, selection of the quantization stepsize is the same as selection of the number of bits to be used for quantizing the output of each subband. In Layer I the encoder has a choice of 14 different quantizers for each band (plus the option of assigning 0 bits). The quantizers are all midtread quantizers ranging from 3 levels to 65,535 levels. Each subband gets assigned a variable number of bits. However, the total number of bits available to represent all the subband samples is fixed. Therefore, the bit allocation can be an iterative process. The objective is to keep the noise-to-mask ratio more or less constant across the subbands.

The output of the quantization and bit allocation steps are combined into a frame as shown in Figure 16.5. Because MPEG audio is a streaming format, each frame carries a header, rather than having a single header for the entire audio sequence. The header is made up of 32 bits. The first 12 bits comprise a sync pattern consisting of all 1s. This is followed by a 1-bit version ID, a 2-bit layer indicator, a 1-bit CRC protection. The CRC protection bit is set to 0 if there is no CRC protection and is set to a 1 if there is CRC protection. If the layer and protection information is known, all 16 bits can be used for providing frame synchronization. The next 4 bits make up the bit rate index, which specifies the bit rate in kbits/sec. There

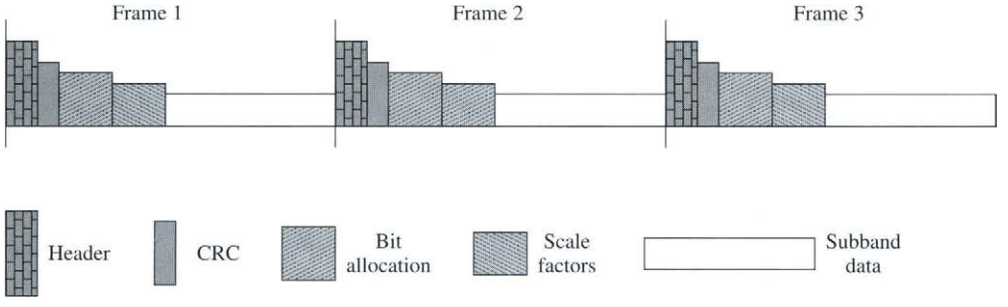

FIGURE 16.5 Frame structure for Layer 1.

TABLE 16.1 Allowable sampling frequencies in MPEG-1 and MPEG-2.

Index	MPEG-1	MPEG-2
00	44.1 kHz	22.05 kHz
01	48 kHz	24 kHz
10	32 kHz	16 kHz
11	Reserved	

are 14 specified bit rates to chose from. This is followed by 2 bits that indicate the sampling frequency. The sampling frequencies for MPEG-1 and MPEG-2 are different (one of the few differences between the audio coding standards for MPEG-1 and MPEG-2) and are shown in Table 16.1 These bits are followed by a single padding bit. If the bit is "1," the frame needs an additional bit to adjust the bit rate to the sampling frequency. The next two bits indicate the mode. The possible modes are "stereo," "joint stereo," "dual channel," and "single channel." The stereo mode consists of two channels that are encoded separately but intended to be played together. The joint stereo mode consists of two channels that are encoded together. The left and right channels are combined to form a *mid* and a *side* signal as follows:

$$M = \frac{L+R}{2}$$

$$S = \frac{L-R}{2}$$

The dual channel mode consists of two channels that are encoded separately and are not intended to be played together, such as a translation channel. These are followed by two mode extension bits that are used in the joint stereo mode. The next bit is a copyright bit ("1" if the material is copy righted, "0" if it is not). The next bit is set to "1" for original media and "0" for copy. The final two bits indicate the type of de-emphasis to be used.

If the CRC bit is set, the header is followed by a 16-bit CRC. This is followed by the bit allocations used by each subband and is in turn followed by the set of 6-bit scalefactors. The scalefactor data is followed by the quantized 384 samples.

16.3.2 Layer II Coding

The Layer II coder provides a higher compression rate by making some relatively minor modifications to the Layer I coding scheme. These modifications include how the samples are grouped together, the representation of the scalefactors, and the quantization strategy. Where the Layer I coder puts 12 samples from each subband into a frame, the Layer II coder groups three sets of 12 samples from each subband into a frame. The total number of samples per frame increases from 384 samples to 1152 samples. This reduces the amount of overhead per sample. In Layer I coding a separate scalefactor is selected for each block of 12 samples. In Layer II coding the encoder tries to share a scale factor among two or all three groups of samples from each subband filter. The only time separate scalefactors are used

for each group of 12 samples is when not doing so would result in a significant increase in distortion. The particular choice used in a frame is signaled through the *scalefactor selection information* field in the bitstream.

The major difference between the Layer I and Layer II coding schemes is in the quantization step. In the Layer I coding scheme the output of each subband is quantized using one of 14 possibilities; the same 14 possibilities for each of the subbands. In Layer II coding the quantizers used for each of the subbands can be selected from a different set of quantizers depending on the sampling rate and the bit rates. For some sampling rate and bit rate combinations, many of the higher subbands are assigned 0 bits. That is, the information from those subbands is simply discarded. Where the quantizer selected has 3, 5, or 9 levels, the Layer II coding scheme uses one more enhancement. Notice that in the case of 3 levels we have to use 2 bits per sample, which would have allowed us to represent 4 levels. The situation is even worse in the case of 5 levels, where we are forced to use 3 bits, wasting three codewords, and in the case of 9 levels where we have to use 4 bits, thus wasting 7 levels. To avoid this situation, the Layer II coder groups 3 samples into a *granule*. If each sample can take on 3 levels, a granule can take on 27 levels. This can be accommodated using 5 bits. If each sample had been encoded separately we would have needed 6 bits. Similarly, if each sample can take on 9 values, a granule can take on 729 values. We can represent 729 values using 10 bits. If each sample in the granule had been encoded separately, we would have needed 12 bits. Using all these savings, the compression ratio in Layer II coding can be increase from 4:1 to 8:1 or 6:1.

The frame structure for the Layer II coder can be seen in Figure 16.6. The only real difference between this frame structure and the frame structure of the Layer I coder is the scalefactor selection information field.

16.3.3 Layer III Coding—*mp3*

Layer III coding, which has become widely popular under the name *mp3*, is considerably more complex than the Layer I and Layer II coding schemes. One of the problems with the Layer I and coding schemes was that with the 32-band decomposition, the bandwidth of the subbands at lower frequencies is significantly larger than the critical bands. This

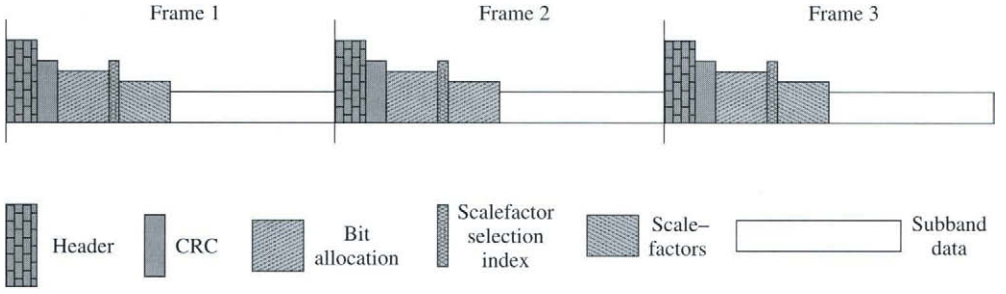

FIGURE 16.6 Frame structure for Layer 2.

makes it difficult to make an accurate judgement of the mask-to-signal ratio. If we get a high amplitude tone within a subband and if the subband was narrow enough, we could assume that it masked other tones in the band. However, if the bandwidth of the subband is significantly higher than the critical bandwidth at that frequency, it becomes more difficult to determine whether other tones in the subband will be be masked.

A simple way to increase the spectral resolution would be to decompose the signal directly into a higher number of bands. However, one of the requirements on the Layer III algorithm is that it be backward compatible with Layer I and Layer II coders. To satisfy this backward compatibility requirement, the spectral decomposition in the Layer III algorithm is performed in two stages. First the 32-band subband decomposition used in Layer I and Layer II is employed. The output of each subband is transformed using a modified discrete cosine transform (MDCT) with a 50% overlap. The Layer III algorithm specifies two sizes for the MDCT, 6 or 18. This means that the output of each subband can be decomposed into 18 frequency coefficients or 6 frequency coefficients.

The reason for having two sizes for the MDCT is that when we transform a sequence into the frequency domain, we lose time resolution even as we gain frequency resolution. The larger the block size the more we lose in terms of time resolution. The problem with this is that any quantization noise introduced into the frequency coefficients will get spread over the entire block size of the transform. Backward temporal masking occurs for only a short duration prior to the masking sound (approximately 20 msec). Therefore, quantization noise will appear as a *pre-echo*. Consider the signal shown in Figure 16.7. The sequence consists of 128 samples, the first 118 of which are 0, followed by a sharp increase in value. The 128-point DCT of this sequence is shown in Figure 16.8. Notice that many of these coefficients are quite large. If we were to send all these coefficients, we would have data expansion instead of data compression. If we keep only the 10 largest coefficients, the

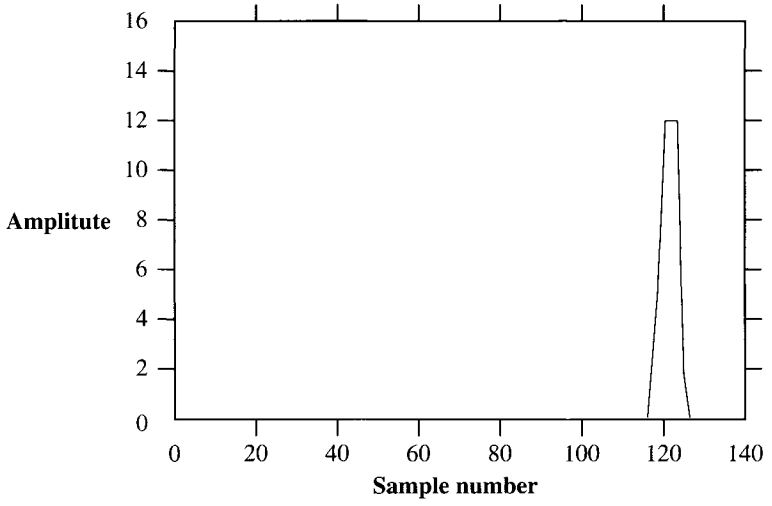

FIGURE 16.7 **Source output sequence.**

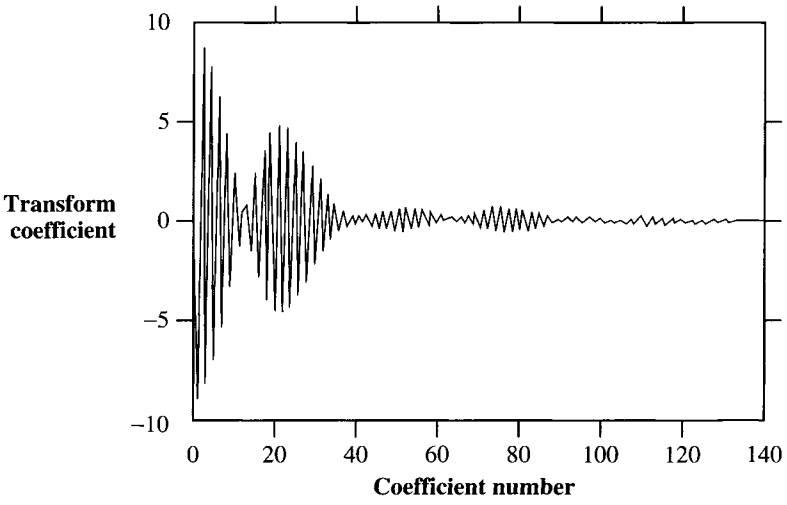

FIGURE 16.8 Transformed sequence.

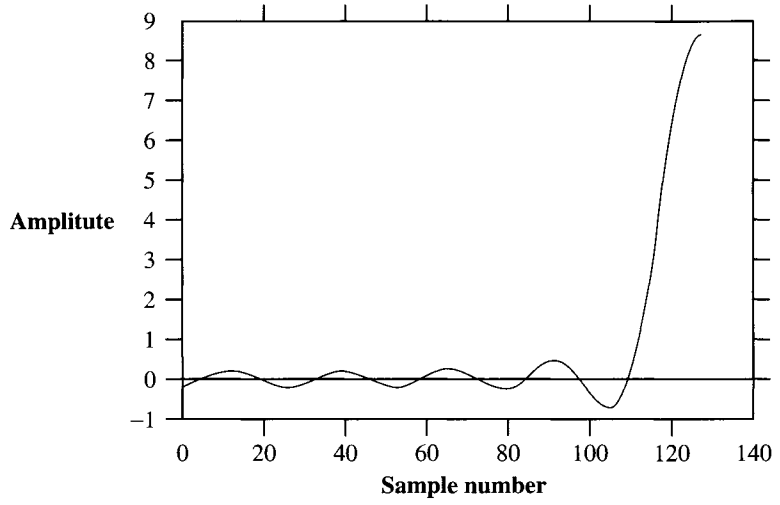

FIGURE 16.9 Reconstructed sequence from 10 DCT coefficients.

reconstructed signal is shown in Figure 16.9. Notice that not only are the nonzero signal values not well represented, there is also error in the samples prior to the change in value of the signal. If this were an audio signal and the large values had occurred at the beginning of the sequence, the forward masking effect would have reduced the perceptibility of the quantization error. In the situation shown in Figure 16.9, backward masking will mask some of the quantization error. However, backward masking occurs for only a short duration prior

to the masking sound. Therefore, if the length of the block in question is longer than the masking interval, the distortion will be evident to the listener.

If we get a sharp sound that is very limited in time (such as the sound of castanets) we would like to keep the block size small enough that it can contain this sharp sound. Then, when we incur quantization noise it will not get spread out of the interval in which the actual sound occurred and will therefore get masked. The Layer III algorithm monitors the input and where necessary substitutes three short transforms for one long transform. What actually happens is that the subband output is multiplied by a window function of length 36 during the stationary periods (that is a blocksize of 18 plus 50% overlap from neighboring blocks). This window is called the *long window*. If a sharp attack is detected, the algorithm shifts to a sequence of three *short windows* of length 12 after a transition window of length 30. This initial transition window is called the *start* window. If the input returns to a more stationary mode, the short windows are followed by another transition window called the *stop* window of length 30 and then the standard sequence of long windows. The process of transitioning between windows is shown in Figure 16.10. A possible set of window transitions is shown in Figure 16.11. For the long windows we end up with 18 frequencies per subband, resulting in a total of 576 frequencies. For the short windows we get 6 coefficients per subband for a total of 192 frequencies. The standard allows for a mixed block mode in which the two lowest subbands use long windows while the remaining subbands use short windows. Notice that while the number of frequencies may change depending on whether we are using long or short windows, the number of samples in a frame stays at 1152. That is 36 samples, or 3 groups of 12, from each of the 32 subband filters.

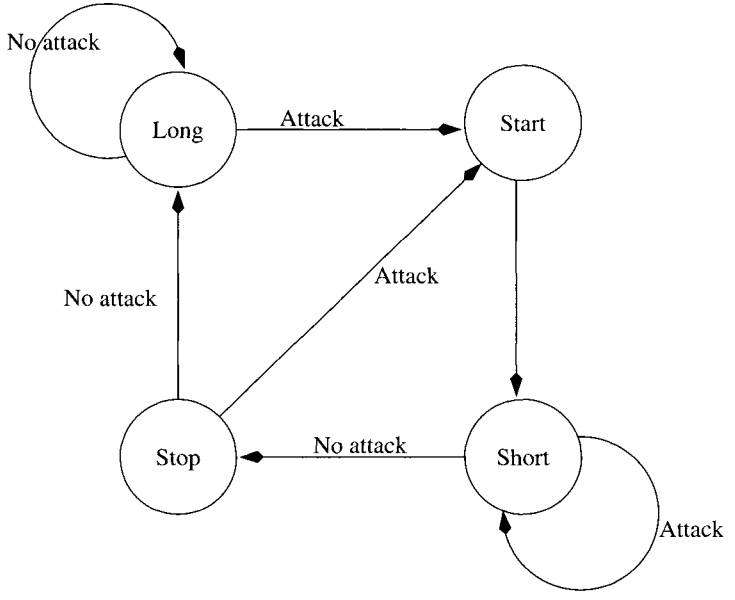

FIGURE 16.10 **State diagram for the window switching process.**

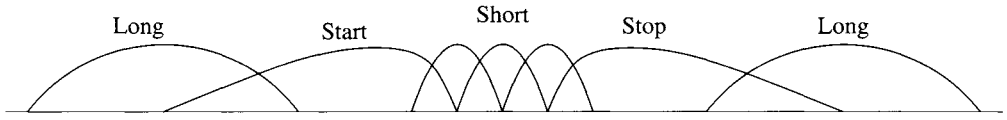

FIGURE 16. 11 Sequence of windows.

The coding and quantization of the output of the MDCT is conducted in an iterative fashion using two nested loops. There is an outer loop called the *distortion control loop* whose purpose is to ensure that the introduced quantization noise lies below the audibility threshold. The scalefactors are used to control the level of quantization noise. In Layer III scalefactors are assigned to groups or "bands" of coefficients in which the bands are approximately the size of critical bands. There are 21 scalefactor bands for long blocks and 12 scalefactor bands for short blocks.

The inner loop is called the *rate control loop*. The goal of this loop is to make sure that a target bit rate is not exceeded. This is done by iterating between different quantizers and Huffman codes. The quantizers used in *mp3* are companded nonuniform quantizers. The scaled MDCT coefficients are first quantized and organized into regions. Coefficients at the higher end of the frequency scale are likely to be quantized to zero. These consecutive zero outputs are treated as a single region and the run-length is Huffman encoded. Below this region of zero coefficients, the encoder identifies the set of coefficients that are quantized to 0 or ± 1. These coefficients are grouped into groups of four. This set of quadruplets is the second region of coefficients. Each quadruplet is encoded using a single Huffman codeword. The remaining coefficients are divided into two or three subregions. Each subregion is assigned a Huffman code based on its statistical characteristics. If the result of using this variable length coding exceeds the bit budget, the quantizer is adjusted to increase the quantization stepsize. The process is repeated until the target rate is satisfied.

Once the target rate is satisfied, control passes back to the outer, distortion control loop. The psychoacoustic model is used to check whether the quantization noise in any band exceeds the allowed distortion. If it does, the scalefactor is adjusted to reduce the quantization noise. Once all scalefactors have been adjusted, control returns to the rate control loop. The iterations terminate either when the distortion and rate conditions are satisfied or the scalefactors cannot be adjusted any further.

There will be frames in which the number of bits used by the Huffman coder is less than the amount allocated. These bits are saved in a conceptual *bit reservoir*. In practice what this means is that the start of a block of data does not necessarily coincide with the header of the frame. Consider the three frames shown in Figure 16.12. In this example, the main data for the first frame (which includes scalefactor information and the Huffman coded data) does not occupy the entire frame. Therefore, the main data for the second frame starts before the second frame actually begins. The same is true for the remaining data. The main data can begin in the *previous frame*. However, the main data for a particular frame cannot spill over into the *following* frame.

All this complexity allows for a very efficient encoding of audio inputs. The typical *mp3* audio file has a compression ratio of about 10:1. In spite of this high level of compression, most people cannot tell the difference between the original and the compressed representation.

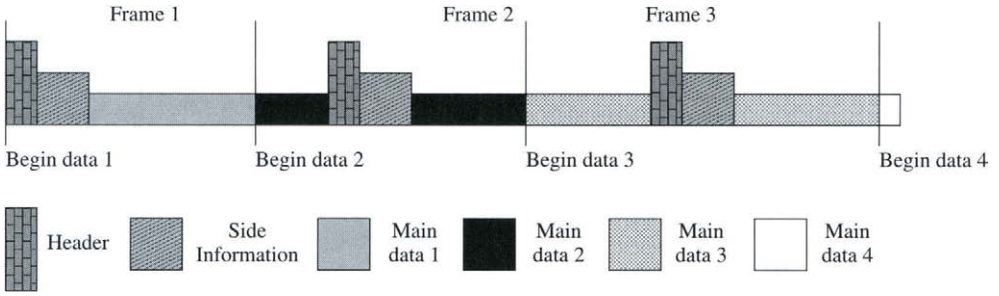

FIGURE 16. 12 Sequence of windows.

We say most because trained professionals can at times tell the difference between the original and compressed versions. People who can identify very minute differences between coded and original signals have played an important role in the development of audio coders. By identifying where distortion may be audible they have helped focus effort onto improving the coding process. This development process has made *mp3* the format of choice for compressed music.

16.4 MPEG Advanced Audio Coding

The MPEG Layer III algorithm has been highly successful. However, it had some built-in drawbacks because of the constraints under which it had been designed. The principal constraint was the requirement that it be backward compatible. This requirement for backward compatibility forced the rather awkward decomposition structure involving a subband decomposition followed by an MDCT decomposition. The period immediately following the release of the MPEG specifications also saw major developments in hardware capability. The Advanced Audio Coding (AAC) standard was approved as a higher quality multichannel alternative to the backward compatible MPEG Layer III in 1997.

The AAC approach is a modular approach based on a set of self-contained tools or modules. Some of these tools are taken from the earlier MPEG audio standard while others are new. As with previous standards, the AAC standard actually specifies the decoder. The decoder tools specified in the AAC standard are listed in Table 16.2. As shown in the table, some of these tools are required for all profiles while others are only required for some profiles. By using some or all of these tools, the standard describes three profiles. These are the *main* profile, the *low complexity* profile, and the *sampling-rate-scalable* profile. The AAC approach used in MPEG-2 was later enhanced and modified to provide an audio coding option in MPEG-4. In the following section we first describe the MPEG-2 AAC algorithm, followed by the MPEG-4 AAC algorithm.

16.4.1 MPEG-2 AAC

A block diagram of an MPEG-2 AAC encoder is shown in Figure 16.13. Each block represents a tool. The psychoacoustic model used in the AAC encoder is the same as the

TABLE 16.2 AAC Decoder Tools [213].

Tool Name	
Bitstream Formatter	Required
Huffman Decoding	Required
Inverse Quantization	Required
Rescaling	Required
M/S	Optional
Interblock Prediction	Optional
Intensity	Optional
Dependently Switched Coupling	Optional
TNS	Optional
Block switching / MDCT	Required
Gain Control	Optional
Independently Switched Coupling	Optional

model used in the MPEG Layer III encoder. As in the Layer III algorithm, the psychoacoustic model is used to trigger switching in the blocklength of the MDCT transform and to produce the threshold values used to determine scalefactors and quantization thresholds. The audio data is fed in parallel to both the acoustic model and to the modified Discrete Cosine Transform.

Block Switching and MDCT

Because the AAC algorithm is not backward compatible it does away with the requirement of the 32-band filterbank. Instead, the frequency decomposition is accomplished by a Modified Discrete Cosine Transform (MDCT). The MDCT is described in Chapter 13. The AAC algorithm allows switching between a window length of 2048 samples and 256 samples. These window lengths include a 50% overlap with neighboring blocks. So 2048 time samples are used to generate 1024 spectral coefficients, and 256 time samples are used to generate 128 frequency coefficients. The k^{th} spectral coefficient of block i, $X_{i,k}$ is given by:

$$X_{i,k} = 2\sum_{n=0}^{N-1} z_{i,n} \cos\left(\frac{2\pi(n+n_o)}{N}(k+\frac{1}{2})\right)$$

where $z_{i,n}$ is the n^{th} time sample of the i^{th} block, N is the window length and

$$n_o = \frac{N/2+1}{2}.$$

The longer block length allows the algorithm to take advantage of stationary portions of the input to get significant improvements in compression. The short block length allows the algorithm to handle sharp attacks without incurring substantial distortion and rate penalties. Short blocks occur in groups of eight in order to avoid framing issues. As in the case of MPEG Layer III, there are four kinds of windows: long, short, start, and stop. The decision about whether to use a group of short blocks is made by the psychoacoustic model. The coefficients

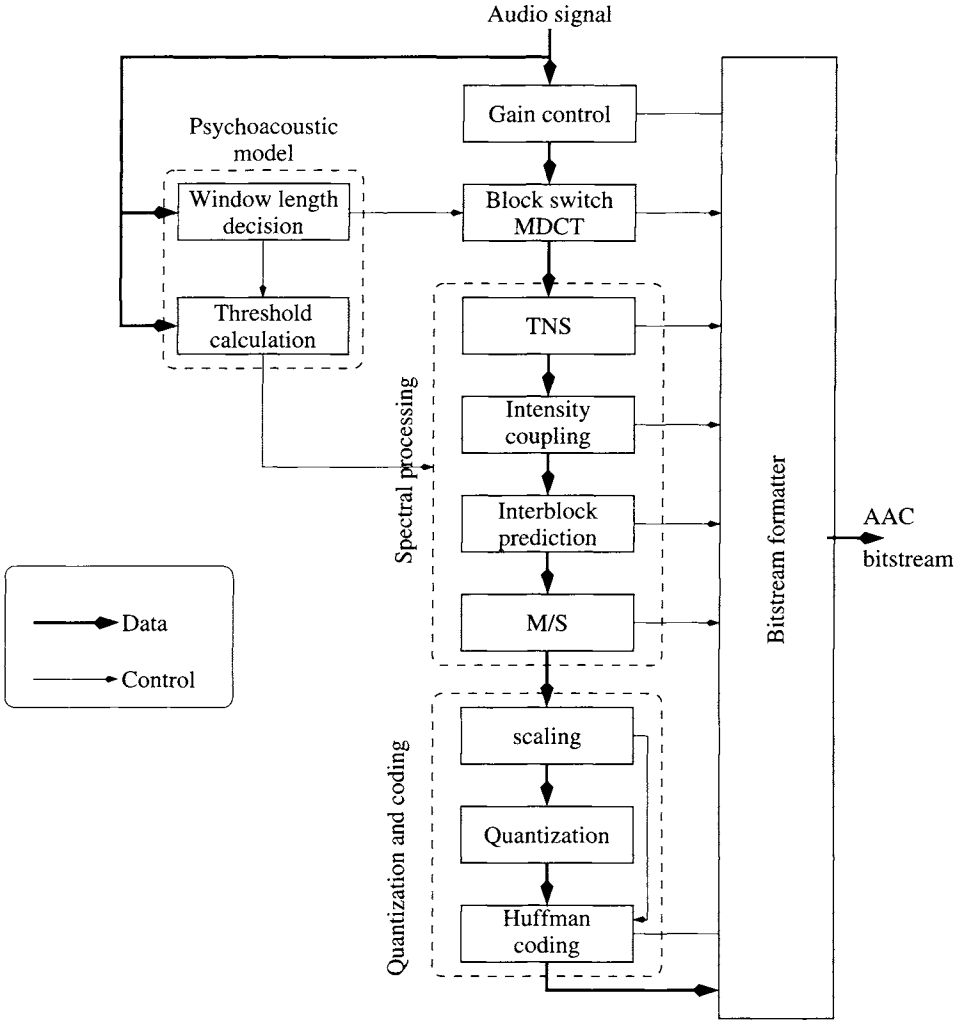

FIGURE 16.13 An MPEG-2 AAC encoder [213].

are divided into scalefactor bands in which the number of coefficients in the bands reflects the critical bandwidth. Each scalefactor band is assigned a single scalefactor. The exact division of the coefficients into scalefactor bands for the different windows and different sampling rates is specified in the standard [213].

Spectral Processing

In MPEG Layer III coding the compression gain is mainly achieved through the unequal distribution of energy in the different frequency bands, the use of the psychoacoustic model,

and Huffman coding. The unequal distribution of energy allows use of fewer bits for spectral bands with less energy. The psychoacoustic model is used to adjust the quantization step size in a way that masks the quantization noise. The Huffman coding allows further reductions in the bit rate. All these approaches are also used in the AAC algorithm. In addition, the algorithm makes use of prediction to reduce the dynamic range of the coefficients and thus allow further reduction in the bit rate.

Recall that prediction is generally useful only in stationary conditions. By their very nature, transients are almost impossible to predict. Therefore, generally speaking, predictive coding would not be considered for signals containing significant amounts of transients. However, music signals have exactly this characteristic. Although they may contain long periods of stationary signals, they also generally contain a significant amount of transient signals. The AAC algorithm makes clever use of the time frequency duality to handle this situation. The standard contains two kinds of predictors, an intrablock predictor, referred to as Temporal Noise Shaping (TNS), and an interblock predictor. The interblock predictor is used during stationary periods. During these periods it is reasonable to assume that the coefficients at a certain frequency do not change their value significantly from block to block. Making use of this characteristic, the AAC standard implements a set of parallel DPCM systems. There is one predictor for each coefficient up to a maximum number of coefficients. The maximum is different for different sampling frequencies. Each predictor is a backward adaptive two-tap predictor. This predictor is really useful only in stationary periods. Therefore, the psychoacoustic model monitors the input and determines when the output of the predictor is to be used. The decision is made on a scalefactor band by scalefactor band basis. Because notification of the decision that the predictors are being used has to be sent to the decoder, this would increase the rate by one bit for each scalefactor band. Therefore, once the preliminary decision to use the predicted value has been made, further calculations are made to check if the savings will be sufficient to offset this increase in rate. If the savings are determined to be sufficient, a *predictor_data_present* bit is set to 1 and one bit for each scalefactor band (called the *prediction_used* bit) is set to 1 or 0 depending on whether prediction was deemed effective for that scalefactor band. If not, the *predictor_data_present* bit is set to 0 and the *prediction_used* bits are not sent. Even when a predictor is disabled, the adaptive algorithm is continued so that the predictor coefficients can track the changing coefficients. However, because this is a streaming audio format it is necessary from time to time to reset the coefficients. Resetting is done periodically in a staged manner and also when a short frame is used.

When the audio input contains transients, the AAC algorithm uses the intraband predictor. Recall that narrow pulses in time correspond to wide bandwidths. The narrower a signal in time, the broader its Fourier transform will be. This means that when transients occur in the audio signal, the resulting MDCT output will contain a large number of correlated coefficients. Thus, unpredictability in time translates to a high level of predictability in terms of the frequency components. The AAC uses neighboring coefficients to perform prediction. A target set of coefficients is selected in the block. The standard suggests a range of 1.5 kHz to the uppermost scalefactor band as specified for different profiles and sampling rates. A set of linear predictive coefficients is obtained using any of the standard approaches, such as the Levinson-Durbin algorithm described in Chapter 15. The maximum order of the filter ranges from 12 to 20 depending on the profile. The process of obtaining the filter

coefficients also provides the expected prediction gain g_p. This expected prediction gain is compared against a threshold to determine if intrablock prediction is going to be used. The standard suggests a value of 1.4 for the threshold. The order of the filter is determined by the first PARCOR coefficient with a magnitude smaller than a threshold (suggested to be 0.1). The PARCOR coefficients corresponding to the predictor are quantized and coded for transfer to the decoder. The reconstructed LPC coefficients are then used for prediction. In the time domain predictive coders, one effect of linear prediction is the spectral shaping of the quantization noise. The effect of prediction in the frequency domain is the *temporal* shaping of the quantization noise, hence the name Temporal Noise Shaping. The shaping of the noise means that the noise will be higher during time periods when the signal amplitude is high and lower when the signal amplitude is low. This is especially useful in audio signals because of the masking properties of human hearing.

Quantization and Coding

The quantization and coding strategy used in AAC is similar to what is used in MPEG Layer III. Scalefactors are used to control the quantization noise as a part of an outer *distortion control loop*. The quantization step size is adjusted to accommodate a target bit rate in an inner *rate control loop*. The quantized coefficients are grouped into *sections*. The section boundaries have to coincide with scalefactor band boundaries. The quantized coefficients in each section are coded using the same Huffman codebook. The partitioning of the coefficients into sections is a dynamic process based on a greedy merge procedure. The procedure starts with the maximum number of sections. Sections are merged if the overall bit rate can be reduced by merging. Merging those sections will result in the maximum reduction in bit rate. This iterative procedure is continued until there is no further reduction in the bit rate.

Stereo Coding

The AAC scheme uses multiple approaches to stereo coding. Apart from independently coding the audio channels, the standard allows Mid/Side (M/S) coding and intensity stereo coding. Both stereo coding techniques can be used at the same time for different frequency ranges. Intensity coding makes use of the fact that at higher frequencies two channels can be represented by a single channel plus some directional information. The AAC standard suggests using this technique for scalefactor bands above 6 kHZ. The M/S approach is used to reduce noise imaging. As described previously in the joint stereo approach, the two channels (L and R) are combined to generate sum and difference channels.

Profiles

The main profile of MPEG-2 AAC uses all the tools except for the gain control tool of Figure 16.13. The low complexity profile in addition to the gain control tool the interblock prediction tool is also dropped. In addition the maximum prediction order for intra-band prediction (TNS) for long windows is 12 for the low complexity profile as opposed to 20 for the main profile.

The Scalable Sampling Rate profile does not use the coupling and interband prediction tools. However this profile does use the gain control tool. In the scalable-sampling profile the MDCT block is preceded by a bank of four equal width 96 tap filters. The filter coefficients are provided in the standard. The use of this filterbank allows for a reduction in rate and decoder complexity. By ignoring one or more of the filterbank outputs the output bandwidth can be reduced. This reduction in bandwidth and sample rate also leads to a reduction in the decoder complexity. The gain control allows for the attenuation and amplification of different bands in order to reduce perceptual distortion.

16.4.2 MPEG-4 AAC

The MPEG-4 AAC adds a perceptual noise substitution (PNS) tool and substitutes a long term prediction (LTP) tool for the interband prediction tool in the spectral coding block. In the quantization and coding section the MPEG-4 AAC adds the options of Transform-Domain Weighted Interleave Vector Quantization (TwinVQ) and Bit Sliced Arithmetic Coding (BSAC).

Perceptual Noise Substitution (PNS)

There are portions of music that sound like noise. Although this may sound like a harsh (or realistic) subjective evaluation, that is not what is meant here. What is meant by noise here is a portion of audio where the MDCT coefficients are stationary without containing tonal components [214]. This kind of noise-like signal is the hardest to compress. However, at the same time it is very difficult to distinguish one noise-like signal from another. The MPEG-4 AAC makes use of this fact by not transmitting such noise-like scalefactor bands. Instead the decoder is alerted to this fact and the power of the noise-like coefficients in this band is sent. The decoder generates a noise-like sequence with the appropriate power and inserts it in place of the unsent coefficients.

Long Term Prediction

The interband prediction in MPEG-2 AAC is one of the more computationally expensive parts of the algorithm. MPEG-4 AAC replaces that with a cheaper long term prediction (LTP) module.

TwinVQ

The Transform-Domain Weighted Interleave Vector Quantization (TwinVQ) [215] option is suggested in the MPEG-4 AAC scheme for low bit rates. Developed at NTT in the early 1990s, the algorithm uses a two-stage process for flattening the MDCT coefficients. In the first stage, a linear predictive coding algorithm is used to obtain the LPC coefficients for the audio data corresponding to the MDCT coefficients. These coefficients are used to obtain the spectral envelope for the audio data. Dividing the MDCT coefficients with this spectral envelope results in some degree of "flattening" of the coefficients. The spectral envelope computed from the LPC coefficients reflects the gross features of the envelope

of the MDCT coefficients. However, it does not reflect any of the fine structure. This fine structure is predicted from the previous frame and provides further flattening of the MDCT coefficients. The flattened coefficients are interleaved and grouped into subvectors and quantized. The flattening process reduces the dynamic range of the coefficients, allowing them to be quantized using a smaller VQ codebook than would otherwise have been possible. The flattening process is reversed in the decoder as the LPC coefficients are transmitted to the decoder.

Bit Sliced Arithmetic Coding (BSAC)

In addition to the Huffman coding scheme of the MPEG-2 AAC scheme, the MPEG-4 AAC scheme also provides the option of using binary arithmetic coding. The binary arithmetic coding is performed on the bitplanes of the magnitudes of the quantized MDCT coefficients. By bitplane we mean the corresponding bit of each coefficient. Consider the sequence of 4-bit coefficients x_n : 5, 11, 8, 10, 3, 1. The most significant bitplane would consist of the MSBs of these numbers, 011100. The next bitplane would be 100000. The next bitplane is 010110. The least significant bitplane is 110011.

The coefficients are divided into *coding bands* of 32 coefficients each. One probability table is used to encode each coding band. Because we are dealing with binary data, the probability table is simply the number of zeros. If a coding band contains only zeros, this is indicated to the decoder by selecting the probability table 0. The sign bits associated with the nonzero coefficients are sent after the arithmetic code when the coefficient has a 1 for the the first time.

The scalefactor information is also arithmetic coded. The maximum scalefactor is coded as an 8-bit integer. The differences between scalefactors are encoded using an arithmetic code. The first scalefactor is encoded using the difference between it and the maximum scalefactor.

16.5 Dolby AC3 (Dolby Digital)

Unlike the MPEG algorithms described in the previous section, the Dolby AC-3 method became a de facto standard. It was developed in response to the standardization activities of the *Grand Alliance*, which was developing a standard for HDTV in the United States. However, even before it was accepted as the recommendation for HDTV audio, Dolby-AC3 had already made its debut in the movie industry. It was first released in a few theaters during the showing of *Star Trek IV* in 1991 and was formally released with the movie *Batman Returns* in 1992. It was accepted by the *Grand Alliance* in October of 1993 and became an Advanced Television Systems Committee (ATSC) standard in 1995. Dolby AC-3 had the multichannel capability required by the movie industry along with the ability to downmix the channels to accommodate the varying capabilities of different applications. The 5.1 channels include right, center, left, left rear, and right rear, and a narrowband low-frequency effects channel (the 0.1 channel). The scheme supports downmixing the 5.1 channels to 4, 3, 2, or 1 channel. It is now the standard used for DVDs as well as for Direct Broadcast Satellites (DBS) and other applications.

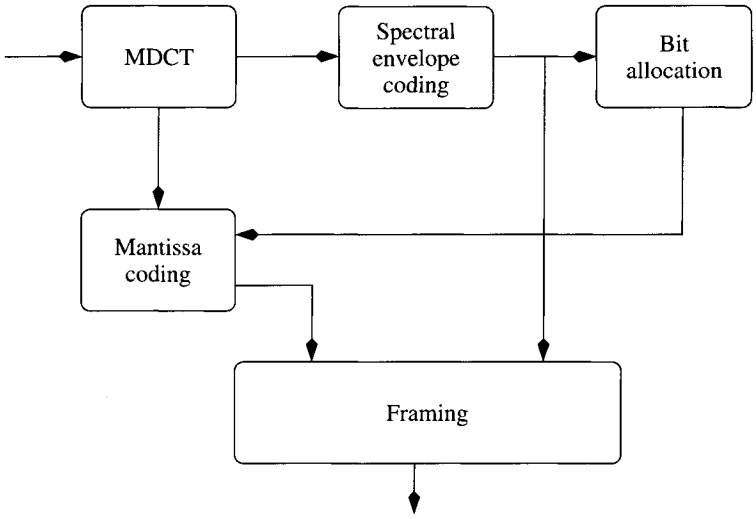

FIGURE 16.14 The Dolby AC3 algorithm.

A block diagram of the Dolby-AC3 algorithm is shown in Figure 16.14. Much of the Dolby-AC3 scheme is similar to what we have already described for the MPEG algorithms. As in the MPEG schemes, the Dolby-AC3 algorithm uses the modified DCT (MDCT) with 50% overlap for frequency decomposition. As in the case of MPEG, there are two different sizes of windows used. For the stationary portions of the audio a window of size 512 is used to get a 256 coefficient. A surge in the power of the high frequency coefficients is used to indicate the presence of a transient and the 512 window is replaced by two windows of size 256. The one place where the Dolby-AC3 algorithm differs significantly from the algorithm described is in the bit allocation.

16.5.1 Bit Allocation

The Dolby-AC3 scheme has a very interesting method for bit allocation. Like the MPEG schemes, it uses a psychoacoustic model that incorporates the hearing thresholds and the presence of noise and tone maskers. However, the input to the model is different. In the MPEG schemes the audio sequence being encoded is provided to the bit allocation procedure and the bit allocation is sent to the decoder as side information. In the Dolby-AC3 scheme the signal itself is not provided to the bit allocation procedure. Instead a crude representation of the spectral envelope is provided to both the decoder and the bit allocation procedure. As the decoder then possesses the information used by the encoder to generate the bit allocation, the allocation itself is not included in the transmitted bitstream.

The representation of the spectral envelope is obtained by representing the MDCT coefficients in binary exponential notation. The binary exponential notation of a number 110.101 is 0.110101×2^3, where 110101 is called the mantissa and 3 is the exponent. Given a sequence of numbers, the exponents of the binary exponential representation provide

an estimate of the relative magnitude of the numbers. The Dolby-AC3 algorithm uses the exponents of the binary exponential representation of the MDCT coefficients as the representation of the spectral envelope. This encoding is sent to the bit allocation algorithm, which uses this information in conjunction with a psychoacoustic model to generate the number of bits to be used to quantize the mantissa of the binary exponential representation of the MDCT coefficients. To reduce the amount of information that needs to be sent to the decoder, the spectral envelope coding is not performed for every audio block. Depending on how stationary the audio is, the algorithm uses one of three strategies [194].

The D15 Method

When the audio is relatively stationary, the spectral envelope is coded once for every six audio blocks. Because a frame in Dolby-AC3 consists of six blocks, during each block we get a new spectral envelope and hence a new bit allocation. The spectral envelope is coded differentially. The first exponent is sent as is. The difference between exponents is encoded using one of five values $\{0, \pm 1, \pm 2\}$. Three differences are encoded using a 7-bit word. Note that three differences can take on 125 different combinations. Therefore, using 7 bits, which can represent 128 different values, is highly efficient.

The D25 and D45 Methods

If the audio is not stationary, the spectral envelope is sent more often. To keep the bit rate down, the Dolby-AC3 algorithm uses one of two strategies. In the D25 strategy, which is used for moderate spectral activity, every other coefficient is encoded. In the D45 strategy, used during transients, every fourth coefficient is encoded. These strategies make use of the fact that during a transient the fine structure of the spectral envelope is not that important, allowing for a more crude representation.

16.6 Other Standards

We have described a number of audio compression approaches that make use of the limitations of human audio perception. These are by no means the only ones. Competitors to Dolby Digital include Digital Theater Systems (DTS) and Sony Dynamic Digital Sound (SDDS). Both of these proprietary schemes use psychoacoustic modeling. The Adaptive TRansform Acoustic Coding (ATRAC) algorithm [216] was developed for the minidisc by Sony in the early 1990s, followed by enhancements in ATRAC3 and ATRAC3plus. As with the other schemes described in this chapter, the ATRAC approach uses MDCT for frequency decomposition, though the audio signal is first decomposed into three bands using a two-stage decomposition. As in the case of the other schemes, the ATRAC algorithm recommends the use of the limitations of human audio perception in order to discard information that is not perceptible.

Another algorithm that also uses MDCT and a psychoacoustic model is the open source encoder Vorbis. The Vorbis algorithm also uses vector quantization and Huffman coding to reduce the bit rate.

16.7 Summary

The audio coding algorithms described in this chapter take, in some sense, the opposite tack from the speech coding algorithms described in the previous chapter. Instead of focusing on the source of information, as is the case with the speech coding algorithm, the focus in the audio coding algorithm is on the sink, or user, of the information. By identifying the components of the source signal that are not perceptible, the algorithms reduce the amount of data that needs to be transmitted.

Further Reading

1. The book *Introduction to Digital Audio Coding and Standards* by M. Bosi and R.E. Goldberg [194] provides a detailed accounting of the standards described here as well as a comprehensive look at the process of constructing a psychoacoustic model.
2. *The MPEG Handbook*, by J. Watkinson [214], is an accessible source of information about aspects of audio coding as well as the MPEG algorithms.
3. An excellent tutorial on the MPEG algorithms is the appropriately named *A Tutorial on MPEG/Audio Compression*, by D. Pan [217].
4. A thorough review of audio coding can be found in *Perceptual Coding of Digital Audio*, by T. Painter and A. Spanias [218].
5. The website *http://www.tnt.uni-hannover.de/project/mpeg/audio/faq/* contains information about all the audio coding schemes described here as well as an overview of MPEG-7 audio.