14 MPEG Audio Compression

Have you ever attended a dance and found that for quite some time afterward you couldn't hear much? You were dealing with a type of *temporal masking*!

Have you ever noticed that the person on the sound board at a dance basically cannot hear high frequencies anymore? Since many technicians have such hearing damage, some compensate by increasing the volume levels of the high frequencies, so they can hear them. If your hearing is not damaged, you experience this music mix as too piercing.

Moreover, if a very loud tone is produced, you also notice it is impossible to hear any sound nearby in the frequency spectrum—the band's singing may be drowned out by the lead guitar. If you've noticed this, you have experienced *frequency masking*!

MPEG audio uses this kind of perception phenomenon by simply giving up on the tones that cannot be heard anyway. Using a curve of human hearing perceptual sensitivity, an MPEG audio codec makes decisions on when and to what degree frequency masking and temporal masking make some components of the music inaudible. It then controls the quantization process so that these components do not influence the output.

So far, in the previous chapter, we have concentrated on telephony applications—usually, LPC and CELP are tuned to speech parameters. In contrast, in this chapter, we consider compression methods applicable to general audio, such as music or perhaps broadcast digital TV. Instead of modeling speech, the method used is a *waveform* coding approach—one that attempts to make the decompressed amplitude-versus-time waveform as much as possible like the input signal.

A main technique used in evaluating audio content for possible compression makes use of a *psychoacoustic model* of hearing. The kind of coding carried out, then, is generally referred to as *perceptual coding*.

In this chapter, we look at how such considerations impact MPEG audio compression standards and examine in some detail at the following topics:

- Psychoacoustics
- MPEG-1 Audio Compression
- Later MPEG audio developments: MPEG-2, 4, 7, and beyond.

Z.-N. Li et al., *Fundamentals of Multimedia*,
Texts in Computer Science, DOI: 10.1007/978-3-319-05290-8_14,

14.1 Psychoacoustics

Recall that the range of human hearing is about 20 Hz to about 20 kHz. Sounds at higher frequencies are *ultrasonic*. However, the frequency range of the voice is typically only from about 500 Hz to 4 kHz. The dynamic range, the ratio of the maximum sound amplitude to the quietest sound humans can hear, is on the order of about 120 dB.

Recall that the decibel unit represents ratios of intensity on a logarithmic scale. The reference point for 0 dB is the threshold of human hearing—the quietest sound we can hear, measured at 1 kHz. Technically, this is a sound that creates a barely audible sound intensity of 10^{-12} Watt per square meter. Our range of magnitude perception is thus incredibly wide: the level at which the sensation of sound begins to give way to the sensation of pain is about 1 Watt/m^2, so we can perceive a ratio of 10^{12}!

The range of hearing actually depends on frequency. At a frequency of 2 kHz, the ear can readily respond to sound that is about 96 dB more powerful than the smallest perceivable sound at that frequency, or in other words a power ratio of 2^{32}. Table 6.1 lists some of the common sound levels in decibels.

14.1.1 Equal-Loudness Relations

Suppose we play two pure tones, sinusoidal sound waves, with the same amplitude but different frequencies. Typically, one may sound louder than the other. The reason is that the ear does not hear low or high frequencies as well as it does frequencies in the middle range. In particular, at normal sound volume levels, the ear is most sensitive to frequencies between 1 and 5 kHz.

Fletcher-Munson Curves

The Fletcher-Munson equal-loudness curves display the relationship between perceived loudness (*in phons*) for a given stimulus sound volume (*Sound Pressure Level*, in dB), as a function of frequency. Figure 14.1 shows the ear's perception of equal loudness. The abscissa—the x axis—(shown in a semi-log plot) is frequency, in kHz. The ordinate axis is sound pressure level—the *actual* intensity (i.e., loudness) of the tone generated in an experiment. The curves show the loudness with which such tones are *perceived* by humans. The bottom curve shows what level of pure tone stimulus is required to produce the perception of a 10 dB sound.

All the curves are arranged so that the perceived loudness level gives the same loudness as for that loudness level of a pure tone at 1 kHz. Thus, the loudness level at the 1 kHz point is always equal to the dB level on the ordinate axis. The bottom curve, for example, is for 10 phons. All the tones on this curve will be perceived as loud as a 10 dB, 1,000 Hz tone. The figure shows more accurate curves, developed by Robinson and Dadson [1], than the Fletcher and Munson originals [2].

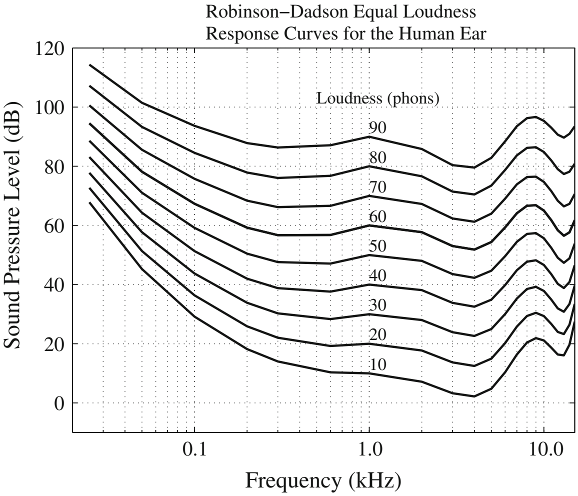

Fig. 14.1 Fletcher–Munson equal loudness response curves for the human ear (remeasured by Robinson and Dadson)

The idea is that a tone is produced at a certain frequency and *measured* loudness level, then a human rates the loudness as it is perceived. On the lowest curve shown, each pure tone between 20 Hz and 15 kHz would have to be produced at the volume level given by the ordinate for it to be perceived at a 10 dB loudness level [1]. The next curve shows what the magnitude would have to be for pure tones to each be perceived as being at 20 dB, and so on. The top curve is for perception at 90 dB.

For example, at 5,000 Hz, we perceive a tone to have a loudness level of 10 phons when the source is actually only 5 dB. Notice that at the dip at 4 kHz, we perceive the sound as being about 10 dB, when in fact the stimulation is only about 2 dB. To perceive the same effective 10 dB at 10 kHz, we would have to produce an absolute magnitude of 20 dB. The ear is clearly more sensitive in the range 2–5 kHz and not nearly as sensitive in the range 6 kHz and above.

At the lower frequencies, if the source is at level 10 dB, a 1 kHz tone would also sound at 10 dB; however, a lower, 100 Hz tone must be at a level 30 dB—20 dB higher than the 1 kHz tone! So we are not very sensitive to the lower frequencies. The explanation of this phenomenon is that the ear canal amplifies frequencies from 2.5 to 4 kHz.

Note that as the overall loudness increases, the curves flatten somewhat. We are approximately equally sensitive to low frequencies of a few hundred Hz if the sound level is loud enough. And we perceive most low frequencies better than high ones at high volume levels. Hence, at the dance, loud music sounds better than quiet music, because then we can actually hear low frequencies and not just high ones. (A "loudness" switch on some sound systems simply boosts the low frequencies as well as some high ones.) However, above 90 dB, people begin to become uncomfortable. A typical city subway operates at about 100 dB.

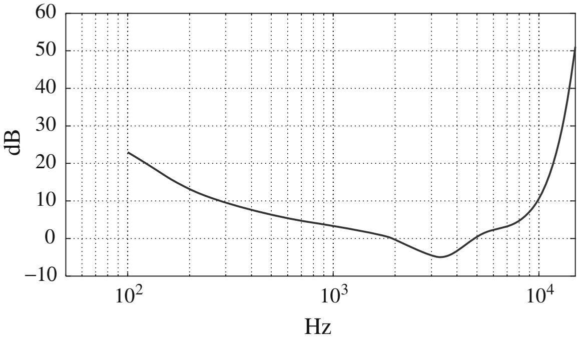

Fig. 14.2 Threshold of human hearing, for pure tones

14.1.2 Frequency Masking

How does one tone interfere with another? At what level does one frequency drown out another? This question is answered by masking curves. Also, masking answers the question of how much noise we can tolerate before we cannot hear the actual music. Lossy audio data compression methods, such as MPEG Audio or Dolby Digital (AC-3) encoding, which is popular in movies, remove some sounds that are masked anyway, thus reducing the total amount of information.

The general situation in regard to masking is as follows:

- A lower tone can effectively mask (make us unable to hear) a higher tone.
- The reverse is not true. A higher tone does not mask a lower tone well. Tones can in fact mask lower frequency sounds, but not as effectively as they mask higher frequency ones.
- The greater the power in the masking tone, the wider its influence—the broader the range of frequencies it can mask.
- As a consequence, if two tones are widely separated in frequency, little masking occurs.

Threshold of Hearing

Figure 14.2 shows a plot of the threshold of human hearing, for pure tones. To determine such a plot, a particular frequency tone is generated, say 1 kHz. Its volume is reduced to zero in a quiet room or using headphones, then turned up until the sound is just barely audible. Data points are generated for all audible frequencies in the same way.

The point of the threshold of hearing curve is that if a sound is above the dB level shown—say it is above 2 dB for a 6 kHz tone—then the sound is audible. Otherwise, we cannot hear it. Turning up the 6 kHz tone so that it equals or surpasses the curve means we can then distinguish the sound.

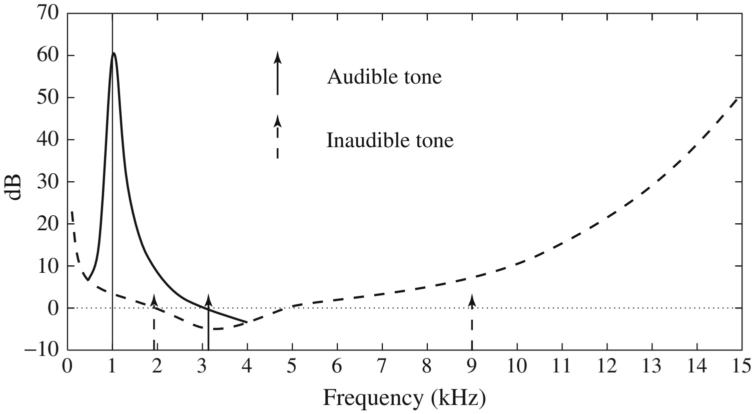

Fig. 14.3 Effect on threshold of human hearing for a 1 kHz maskingtone

An approximate formula exists for this curve, as follows [3]:

$$\text{Threshold}(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4 \tag{14.1}$$

The threshold units are dB. Since the dB unit is a ratio, we do have to choose which frequency will be pinned to the origin, (0, 0). In Eq. (14.1), this frequency is 2,000 Hz: Threshold$(f) = 0$ at $f = 2$ kHz.

Frequency Masking Curves

Frequency masking is studied by playing a particular pure tone, say 1 kHz again, at a loud volume and determining how this tone affects our ability to hear tones at nearby frequencies. To do so, we would generate a 1 kHz *masking tone* at a fixed sound level of 60 dB, then raise the level of a nearby tone, say 1.1 kHz, until it is just audible. The threshold in Fig. 14.3 plots this audible level.

It is important to realize that this masking diagram holds only for a single masking tone: the plot changes if other masking tones are used. Figure 14.4 shows how this looks: the higher the frequency of the masking tone, the broader a range of influence it has.

If, for example, we play a 6 kHz tone in the presence of a 4 kHz masking tone, the masking tone has raised the threshold curve much higher. Therefore, at its neighbor frequency of 6 kHz, we must now surpass 30 dB to distinguish the 6 kHz tone.

The practical point is that if a signal can be decomposed into frequencies, then for frequencies that will be partially masked, only the audible part will be used to set quantization noise thresholds.

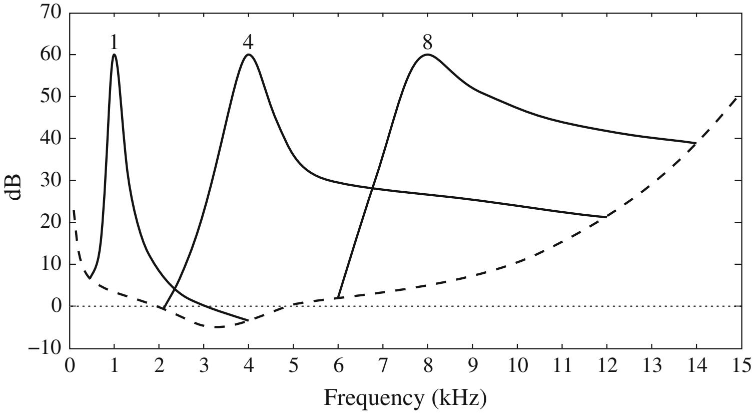

Fig. 14.4 Effect of masking tones at three different frequencies

Critical Bands

The human hearing range naturally divides into *critical bands*, with the property that the human auditory system cannot resolve sounds better than within about one critical band when other sounds are present. Hearing has a limited, frequency-dependent resolution. According to [4], "In a complex tone, the critical bandwidth corresponds to the smallest frequency difference between two partials such that each can still be heard separately. …the critical bandwidth represents the ear's resolving power for simultaneous tones or partials."

At the low-frequency end, a critical band is less than 100 Hz wide, while for high frequencies, the width can be greater than 4 kHz. This indeed is yet another kind of *perceptual nonuniformity*.

Experiments indicate that the critical bandwidth remains approximately constant in width for masking frequencies below about 500 Hz—this width is about 100 Hz. However, for frequencies above 500 Hz, the critical bandwidth increases approximately linearly with frequency.

Generally, the audio frequency range for hearing can be partitioned into about 24 critical bands (25 are typically used for coding applications), as Table 14.1 shows.

Notwithstanding the *general* definition of a critical band, it turns out that our hearing apparatus actually is somewhat tuned to *certain* critical bands. Since hearing depends on physical structures in the inner ear, the frequencies at which these structures best resonate is important. Frequency masking is a result of the ear structures becoming "saturated" at the masking frequency and nearby frequencies.

Hence, the ear operates something like a set of bandpass filters, where each allows a limited range of frequencies through and blocks all others. Experiments that indicate this property are based on the observation that a constant-volume sound will seem louder if it spans the boundary between two critical bands than it would were it contained entirely within one critical band [5]. In effect, the ear is not very discriminating *within* a critical band, because of masking.

Table 14.1 Critical bands and their bandwidths

Band #	Lower bound (Hz)	Center (Hz)	Upper bound (Hz)	Bandwidth (Hz)
1	–	50	100	–
2	100	150	200	100
3	200	250	300	100
4	300	350	400	100
5	400	450	510	110
6	510	570	630	120
7	630	700	770	140
8	770	840	920	150
9	920	1000	1080	160
10	1080	1170	1270	190
11	1270	1370	1480	210
12	1480	1600	1720	240
13	1720	1850	2000	280
14	2000	2150	2320	320
15	2320	2500	2700	380
16	2700	2900	3150	450
17	3150	3400	3700	550
18	3700	4000	4400	700
19	4400	4800	5300	900
20	5300	5800	6400	1100
21	6400	7000	7700	1300
22	7700	8500	9500	1800
23	9500	10500	12000	2500
24	12000	13500	15500	3500
25	15500	18775	22050	6550

Bark Unit

Since the range of frequencies affected by masking is broader for higher frequencies, it is useful to define a new frequency unit such that, in terms of this new unit, each of the masking curves (the parts of Fig. 14.4 above the threshold in quiet) have about the same width.

The new unit defined is called the *Bark*, named after Heinrich Barkhausen (1881–1956), an early sound scientist. One Bark unit corresponds to the width of one critical band, for any masking frequency [6,7]. Figure 14.5 displays critical bands, with the frequency (the abscissa or x-axis) given in Bark units.

The conversion between a frequency f and its corresponding critical-band number b, expressed in Bark units, is as follows:

$$\text{Critical band number(Bark)} = \begin{cases} f/100, & \text{for} f < 500 \\ 9 + 4\log_2(f/1000), & \text{for} f \geq 500 \end{cases} \tag{14.2}$$

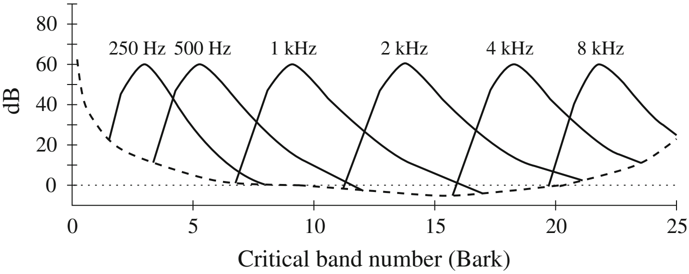

Fig. 14.5 Effect of masking tones, expressed in Bark units

In terms of this new frequency measure, the critical-band number b equals 5 when $f = 500$ Hz. At double that frequency, for a masking frequency of 1 kHz, the Bark value goes up to 9. Another formula used for the Bark scale is as follows:

$$b = 13.0 \arctan(0.76 f) + 3.5 \arctan(f^2/56.25) \tag{14.3}$$

where f is in kHz and b is in Barks. The inverse equation gives the frequency (in kHz) corresponding to a particular Bark value b:

$$f = [(\exp(0.219 \times b)/352) + 0.1] \times b - 0.032 \times \exp[-0.15 \times (b - 5)^2] \tag{14.4}$$

Frequencies forming the boundaries between two critical bands are given by integer Bark values. The critical bandwidth (df) for a given center frequency f can also be approximated by the following [8]:

$$df = 25 + 75 \times [1 + 1.4(f^2)]^{0.69} \tag{14.5}$$

where f is in kHz and df is in Hz.

The idea of the Bark unit is to define a more perceptually uniform unit of frequency, in that every critical band's width is roughly equal in terms of Barks.

14.1.3 Temporal Masking

Recall that after the dance it takes quite a while for our hearing to return to normal. Generally, any loud tone causes the hearing receptors in the inner ear (little hairlike structures called *cilia*) to become *saturated*, and they require time to recover. (Many other perceptual systems behave in this temporally slow fashion—for example, the receptors in the eye have this same kind of "capacitance" effect.)

To quantify this type of behavior, we can measure the time sensitivity of hearing by another masking experiment. Suppose we again play a masking tone at 1 kHz with a volume level of 60 dB, and a nearby tone at, say, 1.1 kHz with a volume level of 40 dB. Since the nearby test tone is masked, it cannot be heard. However, once the masking tone is turned off, we can again hear the 1.1 kHz tone, but only after a small amount of time. The experiment proceeds by stopping the test tone slightly after the masking tone is turned off, say 10 ms later.

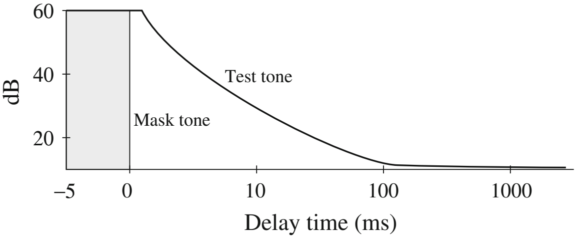

Fig. 14.6 The louder the test tone, the shorter the amount of time required before the test tone is audible once the masking tone is removed

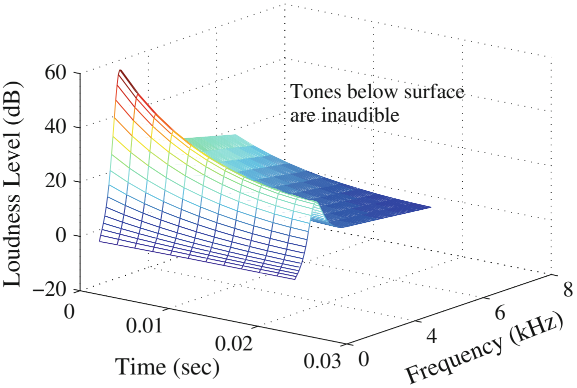

Fig. 14.7 Effect of temporal masking depends on both time and closeness in frequency

The delay time is adjusted to the minimum amount of time such that the test tone can just be distinguished. In general, the louder the test tone, the less time it takes for our hearing to get over hearing the masking tone. Figure 14.6 shows this effect: it may take up to as much as 500 ms for us to discern a quiet test tone after a 60 dB masking tone has been played. Of course, this plot would change for different masking tone frequencies.

Test tones with frequencies near the masking tone are, of course, the most masked. Therefore, for a given masking tone, we have a two-dimensional temporal masking situation, as in Fig. 14.7. The closer the frequency to the masking tone and the closer in time to when the masking tone is stopped, the greater likelihood that a test tone cannot be heard. The figure shows the combined effect of both frequency and temporal masking.

The phenomenon of saturation also depends on just how long the masking tone has been applied. Figure 14.8 shows that for a masking tone played longer (200 ms) than another (100 ms), it takes longer before a test tone can be heard.

As well as being able to mask other signals that occur just after it sounds (*post-masking*), a particular signal can even mask sounds played just before the stronger signal (*pre-masking*). Premasking has a much shorter effective interval (2–5 ms) in which it is operative than does postmasking (usually 50–200 ms).

MPEG audio compression takes advantage of these considerations in basically constructing a large, multidimensional lookup table. It uses this to transmit frequency components that are masked by frequency masking or temporal masking or both, using fewer bits.

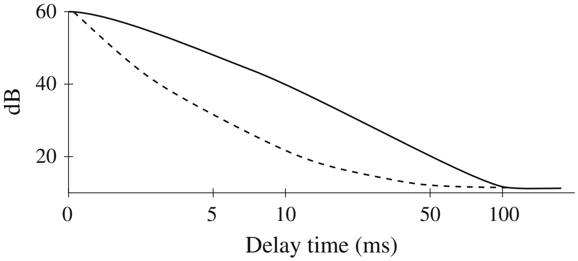

Fig. 14.8 Effect of temporal masking also depends on the length of time the masking tone is applied. *Solid curve* masking tone played for 200 ms; *dashed curve* masking tone played for 100 ms

14.2 MPEG Audio

MPEG Audio [9,10] proceeds by first applying a filter bank to the input, to break the input into its frequency components. In parallel, it applies a psychoacoustic model to the data, and this model is used in a bit allocation block. Then the number of bits allocated is used to quantize the information from the filter bank. The overall result is that quantization provides the compression, and bits are allocated where they are most needed to lower the quantization noise below an audible level.

14.2.1 MPEG Layers

MP3 is a popular audio compression standard. The ‘3’ stands for Layer 3, and “MP” stands for the MPEG standard. Recall that we looked at MPEG video compression in Chap. 11. However, the MPEG standard actually delineates three different aspects of multimedia: audio, video, and systems. MP3 forms part of the audio component of this first phase of MPEG. It was released in 1992 and resulted in the international standard ISO/IEC 11172-3, published in 1993.

MPEG audio sets out three downward-compatible *layers* of audio compression, each able to understand the lower layers. Each offers more complexity in the psychoacoustic model applied and correspondingly better compression for a given level of audio quality. However, an increase in complexity, and concomitantly in compression effectiveness, is accompanied by extra delay.

Layers 1 to 3 in MPEG Audio are compatible, because all layers include the same file header information.

Layer 1 quality can be quite good, provided a comparatively high bitrate is available. While outdated by Layers 2 and 3, Layer 1 formed the basis for MPEG Audio. It is still largely supported, e.g., audio in packages in Ubuntu linux. Layer 2 has more complexity and was proposed for use in digital audio broadcasting. Layer 3 (MP3) is most complex and was originally aimed at audio transmission over ISDN lines. Each of the layers also uses a different frequency transform.

Most of the complexity increase is at the encoder rather than at the decoder side, and this accounts for the popularity of MP3 players. Layer 1 incorporates the simplest psychoacoustic model, and Layer 3 uses the most complex. The objective is a good tradeoff between quality and bitrate. “Quality” is defined in terms of listening test scores (the psychologists hold sway here), where a quality measure is defined by:

- 5.0 = “Transparent”—undetectable difference from original signal; equivalent to CD-quality audio at 14- to 16-bit PCM
- 4.0 = Perceptible difference, but not annoying
- 3.0 = Slightly annoying
- 2.0 = Annoying
- 1.0 = Very annoying.

(Now that’s scientific!) At 64 kbps (kilobits per second) per channel, Layer 2 scores between 2.1 and 2.6, and Layer 3 scores between 3.6 and 3.8. So Layer 3 provides a substantial improvement but is still not perfect by any means.

14.2.2 MPEG Audio Strategy

Compression is certainly called for, since even audio can take fairly substantial bandwidth: CD audio is sampled at 44.1 kHz and 16 bits/channel, so for two channels needs a bitrate of about 1.4 Mbps. MPEG-1 aims at about 1.5 Mbps overall, with 1.2 Mbps for video and 256 kbps for audio. For the audio part, this amounts to a target reduction in size of about 5.5 to 1 (6 to 1 for sampling at 48 kHz).

The MPEG approach to compression relies on quantization, of course, but also recognizes that the human auditory system is not accurate within the width of a critical band, both in terms of perceived loudness and audibility of a test frequency. The encoder employs a bank of filters that act to first analyze the frequency (*spectral*) components of the audio signal by calculating a frequency transform of a window of signal values. The bank of filters decomposes the signal into subbands. Layer 1 and Layer 2 codecs use a *quadrature-mirror filter* bank, while the Layer 3 codec adds a DCT. For the psychoacoustic model, a Fourier transform is used.

Then frequency masking can be brought to bear by using a psychoacoustic model to estimate the just noticeable noise level. In its quantization and coding stage, the encoder balances the masking behavior and the available number of bits by discarding inaudible frequencies and scaling quantization according to the sound level left over, above masking levels.

A sophisticated model would take into account the actual width of the critical bands centered at different frequencies. Within a critical band, our auditory system cannot finely resolve neighboring frequencies and instead tends to blur them. As mentioned earlier, audible frequencies are usually divided into 25 main critical bands, inspired by the auditory critical bands.

However, in keeping with design simplicity, the model adopts a *uniform width* for all frequency analysis filters, using 32 overlapping subbands [9,11]. This means that at lower frequencies, each of the frequency analysis “subbands” covers the width of several critical bands of the auditory system, whereas at higher frequencies this is not so, since a critical band’s width is less than 100 Hz at the low end and more than 4 kHz at the high end. For each frequency band, the sound level above the masking level dictates how many bits must be assigned to code signal values so that quantization noise is kept below the masking level and hence cannot be heard.

In Layer 1, the psychoacoustic model uses only frequency masking. Bitrates range from 32 kbps (mono) to 448 kbps (stereo). Near-CD stereo quality is possible with a bitrate of 256–384 kbps. Layer 2 uses some temporal masking by accumulating more samples and examining temporal masking between the current block of samples and the ones just before and just after. Bitrates can be 32–192 kbps (mono) and 64–384 kbps (stereo). Stereo CD-audio quality requires a bitrate of about 192–256 kbps.

However, temporal masking is less important for compression than frequency masking, which is why it is sometimes disregarded entirely in lower complexity coders. Layer 3 is directed toward lower bitrate applications and uses a more sophisticated subband analysis, with nonuniform subband widths. It also adds nonuniform quantization and entropy coding. Bitrates are standardized at 32–320 kbps.

14.2.3 MPEG Audio Compression Algorithm

Basic Algorithm

Figure 14.9 shows the basic MPEG audio compression algorithm. It proceeds by dividing the input into 32 frequency subbands, via a filter bank. This is a linear operation that takes as its input a set of 32 PCM samples, sampled in time, and produces as its output 32 frequency coefficients. If the sampling rate is f_s, say $f_s = 48$ ksps (kilosamples per second; i.e., 48 kHz), then by the Nyquist theorem, the maximum frequency mapped will be $f_s/2$. Thus, the mapped bandwidth is divided into 32 equal-width segments, each of width $f_s/64$ (these segments overlap somewhat).

In the Layer 1 encoder, the sets of 32 PCM values are first assembled into a set of 12 groups of 32 s. Hence, the coder has an inherent time lag, equal to the time to accumulate 384 (i.e., 12×32) samples. For example, if sampling proceeds at 32 kbps, then a time duration of 12 ms is required since each set of 32 samples is transmitted each millisecond. These sets of 12 samples, each of size 32, are called *segments*. The point of assembling them is to examine 12 sets of values at once in each of the 32 subbands, after frequency analysis has been carried out, and then base quantization on just a summary figure for all 12 values.

The delay is actually somewhat longer than that required to accumulate 384 samples, since header information is also required. As well, *ancillary data*, such as multilingual data and surround-sound data, is allowed. Higher layers also allow more than 384 samples to be analyzed, so the format of the subband-samples (SBS) is also added, with a resulting *frame* of data, as in Fig. 14.10. The header contains a synchronization code (twelve 1s — 111111111111), the sampling rate used, the bitrate, and stereo information. And as mentioned the frame format also contains room for so-called "ancillary" (extra) information. (In fact, an MPEG-1 audio decoder can at least partially decode an MPEG-2 audio bitstream, since the file header begins with an MPEG-1 header and places the MPEG-2 datastream into the MPEG-1 Ancillary Data location.)

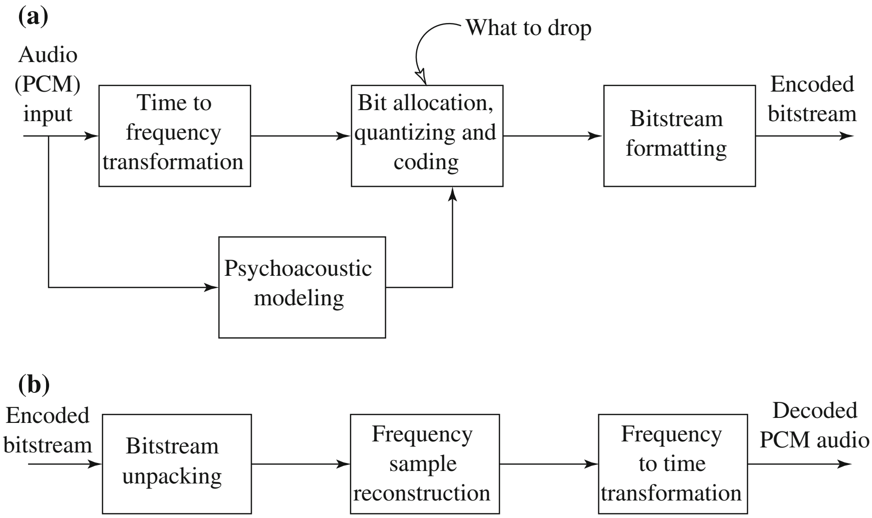

Fig. 14.9 **a** Basic MPEG Audio encoder; and **b** decoder

Fig. 14.10 Example MPEG Audio frame

Header	SBS format	SBS	Ancillary data

MPEG Audio is set up to be able to handle stereo or mono channels, of course. A special *joint-stereo* mode produces a single stream by taking into account the redundancy between the two channels in stereo. This is the audio version of a composite video signal. It can also deal with *dual-monophonic*—two channels coded independently. This is useful for parallel treatment of audio—for example, two speech streams, one in English and one in Spanish.

Consider the 32 × 12 segment as a 32 × 12 matrix. The next stage of the algorithm is concerned with scale, so that proper quantization levels can be set. For each of the 32 subbands, the maximum amplitude of the 12 samples in that row of the array is found, which is the *scaling factor* for that subband. This maximum is then passed to the bit allocation block of the algorithm, along with the SBS (subband samples). The key point of the bit allocation block is to determine how to apportion the total number of code bits available for the quantization of subband signals to minimize the audibility of the quantization noise.

As we know, the psychoacoustic model is fairly complex—more than just a set of simple lookup tables (and in fact this model is not standardized in the specification—it forms part of the "art" content of an audio encoder and is one major reason all encoders are not the same). In Layer 1, a decision step is included to decide whether each frequency band is basically like a tone or like noise. From that decision and the scaling factor, a masking threshold is calculated for each band and compared with the threshold of hearing.

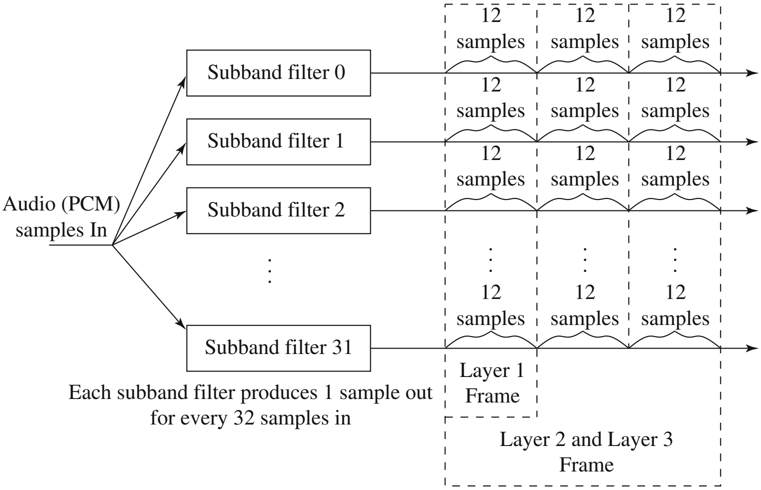

Fig. 14.11 MPEG Audio frame sizes

The model's output consists of a set of what are known as *signal-to-mask ratios* (*SMR*s) that flag frequency components with amplitude below the masking level. The SMR is the ratio of the short-term signal power within each frequency band to the minimum masking threshold for the subband. The SMR gives the amplitude resolution needed and therefore also controls the bit allocations that should be given to the subband. After determination of the SMR, the scaling factors discussed above are used to set quantization levels such that quantization error itself falls below the masking level. This ensures that more bits are used in regions where hearing is most sensitive. In sum, the coder uses fewer bits in critical bands when fewer can be used without making quantization noise audible.

The scaling factor is first quantized, using 6 bits. The 12 values in each subband are then quantized. Using 4 bits, the bit allocations for each subband are transmitted, after an iterative bit allocation scheme is used. Finally, the data is transmitted, with appropriate bit depths for each subband. Altogether, the data consisting of the quantized scaling factor and the 12 codewords are grouped into a collection known as the Subband-Sample format.

On the decoder side, the values are de-quantized, and magnitudes of the 32 samples are re-established. These are passed to a bank of *synthesis filters*, which reconstitute a set of 32 PCM samples. Note that the psychoacoustic model is not needed in the decoder.

Figure 14.11 shows how samples are organized. A Layer 2 or Layer 3 frame actually accumulates more than 12 samples for each subband: instead of 384 samples, a frame includes 1,152 samples.

Bit Allocation

The bit allocation algorithm (for Layer 1 and Layer 2) works in the following way. To reiterate, the aim is to ensure that all quantization noise values are below the masking thresholds. The psychoacoustic model is brought into play for such cases, to allocate more bits, from the number available, to the subbands where increased resolution will be most beneficial.

Algorithm 14.1 (Bit Allocation in MPEG Audio Compression (Layers 1 and 2))

1. From the psychoacoustic model, calculate the *Signal-to-Mask Ratio (SMR)* in decibels (dBs) for each subband:

$$\text{SMR} = 20 \log_{10} \frac{\text{Signal}}{\text{Minimum\_masking\_threshold}}$$

   - This determines the quantization, i.e., the minimum number of bits that is needed, if available. The amount of a signal above the threshold, i.e., SMR, is the amount that needs to be coded. Signals that are below the threshold do not.

2. Calculate *Signal-to-(quantization)-Noise Ratio (SNR)* for all signals.
   - A lookup table provides an estimate of SNR assuming a given number of quantizer levels.

3. *Mask-to-(quantization)-Noise Ratio (MNR)* is defined as the difference, in dB (See Fig. 14.12).

$$\text{MNR} = \text{SNR} - \text{SMR}$$

4. Iterate until no bits left to allocate:
   - Allocate bits to the subband with the lowest MNR
   - Look up new estimate of SNR for the subband allocated more bits, and recalculate MNR

Note:

- The masking effect means we can raise the quantization noise floor around a strong sound because the noise will be masked off anyway. As indicated in Fig. 14.12, adjusting the number of bits m allocated to a subband can move this floor up and down.
- To ensure that all the quantization noise values are inaudible, i.e., below the masking thresholds, so that all MNRs are ≥ 0, a minimum number of bits is needed. Otherwise, SNR could be too small, causing MNR to be < 0, and the quality of the compressed audio could be significantly affected.
- If more bits than the minimum are allowed from the budget, allocate them anyway so as to further increase SNR. For each additional bit, we get 6 dB better SNR.

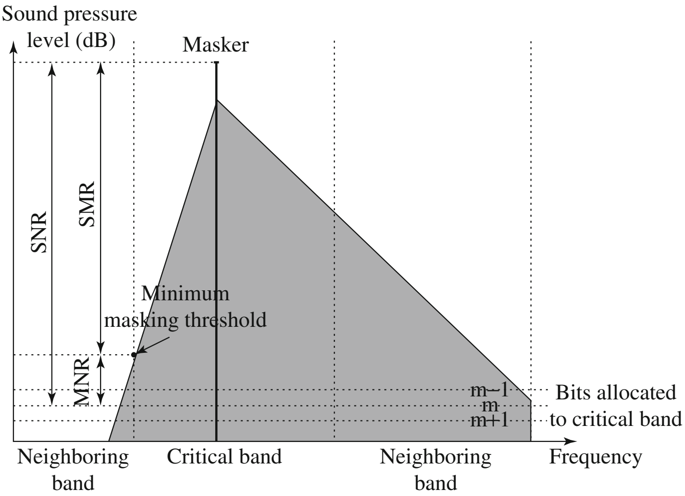
Sound pressure level (dB)
Masker
SNR
SMR
Minimum masking threshold
MNR
m−1
m
m+1
Bits allocated to critical band
Neighboring band
Critical band
Neighboring band
Frequency

Fig. 14.12 Mask-to-noise ratio and signal-to-mask ratio. A qualitative view of SNR, SMR, and MNR, with one dominant masker and *m* bits allocated to a particular critical band

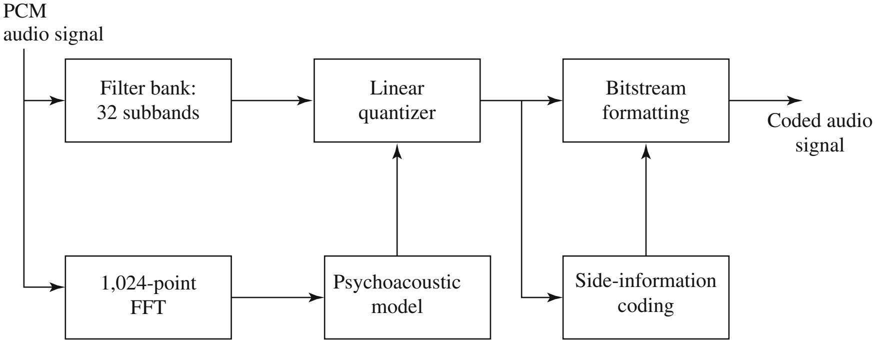

Fig. 14.13 MPEG-1 Audio Layers 1 and 2

Mask calculations are performed in parallel with subband filtering, as in Fig. 14.13. The masking curve calculation requires an accurate frequency decomposition of the input signal, using a Discrete Fourier Transform (DFT). The frequency spectrum is usually calculated with a 1,024-point Fast Fourier Transform (FFT).

In Layer 1, 16 uniform quantizers are precalculated, and for each subband the quantizer giving the lowest distortion is chosen. The index of the quantizer is sent as 4 bits of side information for each subband. The maximum resolution of each quantizer is 15 bits.

Layer 2

Layer 2 of the MPEG-1 Audio codec includes small changes to effect bitrate reduction and quality improvement, at the price of an increase in complexity. The main difference in Layer 2 is that three groups of 12 samples are encoded in each frame, and temporal masking is brought into play, as well as frequency masking. One advantage is that if the scaling factor is similar for each of the three groups, a single scaling factor can be used for all three. But using three frames in the filter (before, current, and next), for a total of 1,152 samples per channel, approximates taking temporal masking into account.

As well, the psychoacoustic model does better at modeling slowly changing sound if the time window used is longer. Bit allocation is applied to window lengths of 36 samples instead of 12, and resolution of the quantizers is increased from 15 bits to 16. To ensure that this greater accuracy does not mean poorer compression, the number of quantizers to choose from decreases for higher subbands.

Layer 3

Layer 3, or MP3, uses a bitrate similar to Layers 1 and 2 but produces substantially better audio quality, again at the price of increased complexity.

A filter bank similar to that used in Layer 2 is employed, except that now perceptual critical bands are more closely adhered to by using a set of filters with nonequal frequencies. This layer also takes into account stereo redundancy. It also uses a refinement of the Fourier transform: the *Modified Discrete Cosine Transform* (*MDCT*) addresses problems the DCT has at boundaries of the window used. The Discrete Fourier Transform can produce block edge effects. When such data is quantized and then transformed back to the time domain, the beginning and ending samples of a block may not be coordinated with the preceding and subsequent blocks, causing audible periodic noise.

The MDCT shown in Eq. (14.6), removes such effects by overlapping frames by 50 %.

$$F(u) = \sum_{i=0}^{N-1} f(i) \cos\left[\frac{2\pi}{N}\left(i + \frac{N/2+1}{2}\right)(u + 1/2)\right], u = 0, \ldots, N/2 - 1 \tag{14.6}$$

Here the window length is $M = N/2$ and M is the number of transform coefficients.

The MDCT also gives better frequency resolution for the masking and bit allocation operations. Optionally, the window size can be reduced back to 12 samples from 36. Even so, since the window is 50 % overlapped, a 12-sample window still includes an extra six samples. A size-36 window includes an extra 18 points. Since lower frequencies are more often tonelike rather than noiselike, they need not be analyzed as carefully, so a mixed mode is also available, with 36-point windows used for the lowest two frequency subbands and 12-point windows used for the rest.

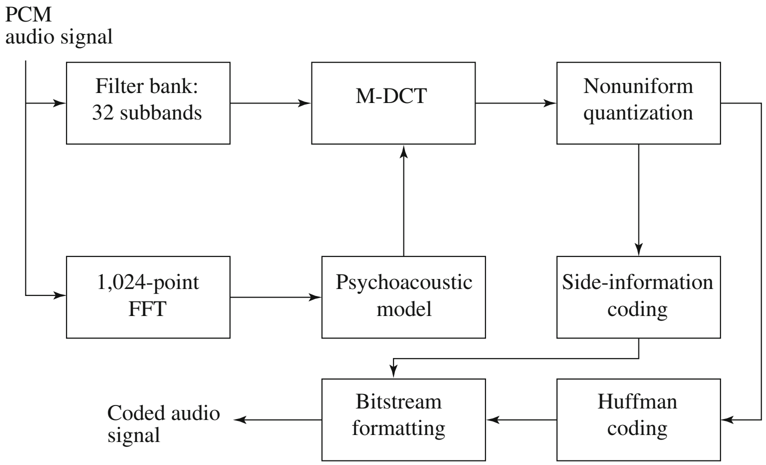

Fig. 14.14 MPEG-1 Audio Layer 3

Table 14.2 MP3 compression performance

Sound quality	Bitrate	Mode	Compression ratio
Telephony	8 kbps	Mono	96:1
Better than shortwave	16 kbps	Mono	48:1
Better than AM radio	32 kbps	Mono	24:1
Similar to FM radio	56–64 kbps	Stereo	26:1–24:1
Near-CD	96 kbps	Stereo	16:1
CD	112–128 kbps	Stereo	14:1–12:1

As well, instead of assigning scaling factors to uniform-width subbands, MDCT coefficients are grouped in terms of the auditory system's actual critical bands, and scaling factors, called *scale factor bands*, are calculated from these.

More bits are saved by carrying out entropy (Huffman) coding and making use of nonuniform quantizers. And, finally, a different bit allocation scheme is used, with two parts. First, a nested loop is used, with an inner loop that adjusts the shape of the quantizer, and an outer loop that then evaluates the distortion from that bit configuration. If the error ("distortion") is too high, the scale factor band is amplified. Second, a *bit reservoir* banks bits from frames that don't need them and allocates them to frames that do. Figure 14.14 shows a summary of MPEG Audio Layer 3 coding.

Table 14.2 shows various achievable MP3 compression ratios. In particular, CD-quality audio is achieved with compression ratios in the range of 12:1 (i.e., bitrate of 112 kbps), assuming 16-bit samples at 44.1 kHz, times two for stereo. Table 14.2 shows typical performance data using MP3 compression.

14.2.4 MPEG-2 AAC (Advanced Audio Coding)

The MPEG-2 standard is widely employed, since it is the standard vehicle for DVDs, and it, too, has an audio component. The *MPEG-2 Advanced Audio Coding* (*AAC*) standard [12] was originally aimed at transparent sound reproduction for theaters. It can deliver this at 320 kbps for five channels, so that sound can be played from five directions: left, right, center, left-surround, and right-surround. So-called 5.1 channel systems also include a *low-frequency enhancement* (*LFE*) channel (a "woofer"). On the other hand, MPEG-2 AAC is also capable of delivering high-quality stereo sound at bitrates below 128 kbps. It is the audio coding technology for the *DVD-Audio Recordable* (*DVD-AR*) format and is also adopted by XM Radio, one of the two main satellite radio services in North America. AAC was developed as a further compression and encoding scheme for digital audio to succeed MP3, and delivers better sound quality than MP3 for the same bitrate [13]. AAC is currently the default audio format for YouTube, iPhone and other Apple products plus iTunes, Nintendo, and PlayStation. It is also supported on Android mobile phones.

MPEG-2 audio can support up to 48 channels, sampling rates between 8 and 96 kHz, and bitrates up to 576 kbps per channel. Like MPEG-1, MPEG-2 supports three different "profiles," but with a different purpose. These are the *Main*, *Low Complexity* (*LC*), and the *Scalable Sampling Rate* (*SSR*). The LC profile requires less computation than the Main profile, but the SSR profile breaks up the signal so that different bitrates and sampling rates can be used by different decoders.

The three profiles follow mostly the same scheme, with a few modifications. First, an MDCT transform is carried out, either on a "long" window with 2,048 samples or a "short" window with 256 samples. The MDCT coefficients are then filtered by a *Temporal Noise Shaping* (*TNS*) tool, with the objective of reducing premasking effects and better encoding signals with stable pitch. The MDCT coefficients are then grouped into 49 scale factor bands, approximately equivalent to a good-resolution version of the human acoustic system's critical bands. In parallel with the frequency transform, a psychoacoustic model similar to the one in MPEG-1 is carried out, to find masking thresholds.

The Low complexity profile is the most widely used AAC profile and is more efficient than MP3 with a 30 % increase in efficiency in terms of quality versus bitrate. It offers near-CD quality at very low bitrates such as 80 kbps for mono and 128 kbps for stereo audio input (44.1 kHz sampling frequency). It is mostly used for music development, vocal recordings, and the like.

The Main profile uses a predictor. Based on the previous two frames, and only for frequency coefficients up to 16 kHz, MPEG-2 subtracts a prediction from the frequency coefficients, provided this step will indeed reduce distortion. Quantization for the Main profile is governed by two rules: keep distortion below the masking threshold, and keep the average number of bits used per frame controlled, using a bit reservoir. Quantization also uses scaling factors, used to amplify some of the scale factor bands and nonuniform quantization. MPEG-2 AAC also uses entropy coding for both scale factors and frequency coefficients.

For implementation, a nested loop is used for bit allocation. The inner loop adapts the nonlinear quantizer, then applies entropy coding to the quantized data. If the bit limit is reached for the current frame, the quantizer step size is increased to use fewer bits. The outer loop decides whether for each scale factor band the distortion is below the masking threshold. If a band is too distorted, it is amplified to increase the SNR of that band, at the price of using more bits.

In the SSR profile, a *Polyphase Quadrature Filter* (*PQF*) bank is used. The meaning of this phrase is that the signal is first split into four frequency bands of equal width, then an MDCT is applied. The point of the first step is that the decoder can decide to ignore one of the four frequency parts if the bitrate must be reduced.

14.2.5 MPEG-4 Audio

MPEG-4 AAC is another audio compression standard under ISO/IEC 14496. MPEG-4 audio integrates several different audio components into one standard: speech compression, perceptually based coders, text-to-speech, 3D localization of sound, and MIDI. MPEG-4 can be classified into MPEG-4 Scalable Lossless Coding (HD AAC) [14] and MPEG-4 (HE AAC) [14]. While MPEG-4 HD (High Definition) AAC is used for lossless high quality audio compression for High Definition videos etc., MPEG-4 HE (High Efficiency) AAC is an extension of the Low complexity MPEG-2 AAC profile used for low bit rate applications such as streaming audio. MPEG-4 HE AAC has two versions: HE AAC v1, which uses only Spectral Band Replication (SBR, enhancing audio at low bitrates) and HE AAC v2, which uses SBR and Parametric Stereo (PS, enhancing efficiency of low bandwidth input). MPEG-4 HE AAC is alo used for the digital radio standards DAB+, developed by the standards group WorldDMB (Digital Multimedia Broadcasting) in 2006, and in Digital Radio Mondiale, a consortium of national radio stations aimed at making better use of the bands currently used for AM broadcasting, including shortwave.

Perceptual Coders

One change in AAC in MPEG-4 is to incorporate a *Perceptual Noise Substitution* module, which looks at scale factor bands above 4 kHz and includes a decision as to whether they are noiselike or tonelike. A noiselike scale factor band itself is not transmitted; instead, just its energy is transmitted, and the frequency coefficient is set to zero. The decoder then inserts noise with that energy.

Another modification is to include a *Bit-Sliced Arithmetic Coding* (*BSAC*) module. This is an algorithm for increasing bitrate scalability, by allowing the decoder side to be able to decode a 64 kbps stream using only a 16 kbps baseline output (and steps of 1 kbps from that minimum). MPEG-4 audio also includes a second perceptual audio coder, a vector-quantization method entitled *Transform-domain Weighted Interleave Vector Quantization* (*TwinVQ*). This is aimed at low bitrates and allows the decoder to discard portions of the bitstream to implement both adjustable bitrate and sampling rate. The basic strategy of MPEG-4 audio is to allow decoders to apply as many or as few audio tools as bandwidth allows.

Structured Coders

To have a low bitrate delivery option, MPEG-4 takes what is termed a *Synthetic/Natural Hybrid Coding* (*SNHC*) approach. The objective is to integrate both "natural" multimedia sequences, both video and audio, with those arising synthetically. In audio, the latter are termed *structured* audio. The idea is that for low bitrate operation, we can simply send a pointer to the audio model we are working with and then send audio model parameters.

In video, such a *model-based* approach might involve sending face-animation data rather than natural video frames of faces. In audio, we could send the information that English is being modeled, then send codes for the basesounds (phonemes) of English, along with other assembler-like codes specifying duration and pitch.

MPEG-4 takes a *toolbox* approach and allows specification of many such models. For example, *Text-To-Speech* (*TTS*) is an ultra-low bitrate method and actually works, provided we need not care what the speaker actually sounds like. Assuming we went on to derive Face-Animation Parameters from such low bitrate information, we arrive directly at a very low bitrate videoconferencing system. Another "tool" in structured audio is called *Structured Audio Orchestra Language* (*SAOL*, pronounced "sail"), which allows simple specification of sound synthesis, including special effects such as reverberation.

Overall, structured audio takes advantage of redundancies in music to greatly compress sound descriptions.

14.3 Other Audio Codecs

14.3.1 Ogg Vorbis

Ogg Vorbis [15] is an open-source audio compression format, part of the Vorbis project headed by Chris Montgomery of the Xiph.org foundation, which started in 1993. It was designed to replace existing patented audio compression formats by incorporating a variable bit rate (VBR) codec similar to MP3 with file sizes smaller compared to those of MP3 for the same bitrate and quality. It is targeted primarily at the MP3 standard, being more efficient even at low bit rates and with better quality audio at higher bitrates. Ogg Vorbis also uses a form of MDCT, specifically a forward adaptive codec. One of the major advantages of the Ogg Vorbis standard is its ability to be wrapped in other media containers, the most popular being Matroska and WebM. Ogg Vorbis is supported by many media players such as VLC, Mplayer, Audacity audio editing software, and most Linux distributions as well. It has limited native support in Windows and Mac OS but the Vorbis team have decoders available

Table 14.3 Comparison of MP3, MPEG-4 AAC, and Ogg vorbis

	MP3	MPEG-4 AAC	Ogg vorbis
File extension	.mp3	.aac, .mp4, .3gp	.ogg
Original name	MPEG-1 Audio Layer 3	Advanced Audio Coding	Ogg
Developer	CCETT, IRT, Fraunhofer Society	Fraunhofer IIS, AT&T Bell Labs, Dolby, Sony Corp., and Nokia	Xiph.org Foundation
Released	1994	1997	v1.0 frozen May 2000
Algorithm	lossy compression	lossy compression	lossy comrpession
Quality	Lower quality than AAC and Ogg	Better quality at same bit rate as MP3	Better quality and smaller file size than MP3 at same bit rates
Used in	Default standard for audio files	iTunes raised its popularity	Open-source platform

for various applications. Ogg Vorbis is gaining popularity with the gaming industry: Ubisoft uses the Ogg Vorbis format for its most recent game releases. Many popular browsers such as Firefox, Chrome, and Opera have native support for Ogg Vorbis. Table 14.3 compares the MP3, AAC and Ogg vorbis standards.

Table 14.4 summarizes the target bitrate range and main features of other modern general audio codecs. They include many similarities to MPEG-2 audio codecs. Dolby Digital (AC-3) dates from 1992; it was devised to code multichannel digital audio for 35 mm movie film, placed alongside the optical analog audio channel. It is also used in HDTV audio and DVD-Video. AC-3 is a perceptual coder with 256 sample block length. The maximum bitrate for compressed 5.1 channel surround sound audio for 35 mm film is 320 kbps (5.1 is one front left channel, one right front, one center channel, two surround channels, and a subwoofer). AC-3's predecessor, Dolby AC-2, was a transform-based codec.

Dolby Digital Plus (E-AC-3, or "Enhanced" AC-3) supports 13.1 channels. It is based on AC-3, with a low-loss and low-complexity conversion from E-AC-3 to AC-3. DTS (or Coherent Acoustics) is a digital surround system aimed at theaters; it forms part of the Blue Ray audio standard. WMA is a proprietary audio coder developed by Microsoft. MPEG SAOC [16], published in 2010, stands for "Spatial Audio Object Coding." It extends "MPEG Surround," which allows the addition of additional multichannel side-information to core stereo data. MPEG SAOC processes "object signals" instead of channel signals, with not a great deal of extra bandwidth for the side-information. SAOC is aimed at such innovative usages as Interactive Remix, Karaoke, gaming, and mobile conferencing over headphones,

Table 14.4 Comparison of audio coding systems

Codec	Bitrate kbps/channel	Complexity	Main application
Dolby AC-2	128–192	Low (encoder/decoder)	Point-to-point, cable
Dolby AC-3	32–640	Low (decoder)	HDTV, cable, DVD
Dolby Digital (Enhanced AC-3)	32–6144	Low (decoder)	HDTV, cable, DVD
DTS: Digital Surround	8–512	Low (for lossless audio extension)	DVD, entertainment, professional
WMA: Windows Media Audio	128–768	Low (low-bit-rate streaming)	Many applications
MPEG SAOC	As low as 48	Low (decoder/rendering)	Many applications

14.4 MPEG-7 Audio and Beyond

Recall that MPEG-4 is aimed at compression using objects. MPEG-4 audio has several interesting features, such as 3D localization of sound, integration of MIDI, text-to-speech, different codecs for different bitrates, and use of the sophisticated MPEG-4 AAC codec. However, newer MPEG standards are also aimed at "search": how can we find objects, assuming that multimedia is indeed coded in terms of objects?

MPEG-7 aims to describe a structured model of audio [17], so as to promote ease of search for audio objects. Officially called a method for *Multimedia Content Description Interface*, MPEG-7 provides a means of standardizing metadata for audiovisual multimedia sequences. MPEG-7 is meant to represent information about multimedia information.

The objective, in terms of audio, is to facilitate the representation and search for sound content, perhaps through the tune or other descriptors. Therefore, researchers are laboring to develop descriptors that efficiently describe, and can help find, specific audio in files. These might require human or automatic content analysis and might be aimed not just at low-level structures, such as melody, but at actually grasping information regarding structural and semantic content [18].

An example application supported by MPEG-7 is *automatic speech recognition* (*ASR*). Language understanding is also an objective for MPEG-7 "content". In theory, MPEG-7 would allow searching on spoken and visual events: "Find me the part where Hamlet says, 'To be or not to be.'" However, the objective of delineating a complete, structured audio model for MPEG-7 is by no means complete.

Nevertheless, low-level features are important. Useful summaries of such work [19,20] describe sets of such descriptors.

Further standards in the MPEG sequence are mostly not aimed at further audio compression standardization. For example, MPEG-DASH (Dynamic Adaptive Streaming over HTTP) is aimed at streaming of multimedia using existing HTTP resources such as servers and content distribution networks, but is meant to be independent of specific video or audio codecs. We will examine it in more details in Chap. 16.

14.5 Further Exploration

Good reviews of MPEG Audio are contained in the articles [9,10]. A comprehensive explication of natural audio coding in MPEG-4 appears in [21]. Structured audio is introduced in [22], and exhaustive articles on natural, synthetic, and SNHC audio in MPEG-4 appear in [23] and [24].

14.6 Exercises

1. (a) What is the threshold of quiet, according to Eq. (14.1), at 1,000 Hz? (Recall that this equation uses 2 kHz as the reference for the 0 dB level.)
   (b) Take the derivative of Eq. (14.1) and set it equal to zero, to determine the frequency at which the curve is minimum. What frequency are we most sensitive to? Hint: One has to solve this numerically.
2. Loudness versus amplitude. Which is louder: a 1,000 Hz sound at 60 dB or a 100 Hz sound at 60 dB?
3. For the (newer versions of the) Fletcher-Munson curves, in Fig. 14.1, the way this data is actually observed is by setting the y-axis value, the sound pressure level, and measuring a human's estimation of the effective perceived loudness. Given the set of observations, what must we do to turn these into the set of perceived loudness curves shown in the figure?
4. Two tones are played together. Suppose tone 1 is fixed, but tone 2 has a frequency that can vary. The *critical bandwidth* for tone 1 is the frequency range for tone 2 over which we hear *beats*, and a roughness in the sound. Beats are overtones at a lower frequency than the two close tones; they arise from the difference in frequencies of the two tones. The critical bandwidth is bounded by frequencies beyond which the two tones sound with two distinct pitches.

   (a) What would be a rough estimate of the critical bandwidth at 220 Hz?
   (b) Explain in words how you would set up an experiment to measure the critical bandwidth.

5. Search the web to discover what is meant by the following psychoacoustic phenomena:
   (a) Virtual pitch
   (b) Auditory scene analysis
   (c) Octave-related complex tones
   (d) Tri-tone paradox
   (e) Inharmonic complex tones.
6. What is the compression ratio of MPEG audio if stereo audio sampled with 16 bits per sample at 48 kHz is reduced to a bitstream of 256 kbps?
7. In MPEG's polyphase filter bank, if 24 kHz is divided into 32 equal-width frequency subbands,
   (a) What is the size of each subband?
   (b) How many critical bands, at worst, does a subband overlap?
8. If the sampling rate f_s is 32 ksps, in MPEG Audio Layer 1, what is the width in frequency of each of the 32 subbands?
9. Given that the level of a *masking tone* at the 8th band is 60 dB, and 10 ms after it stops, the masking effect to the 9th band is 25 dB.
   (a) What would MP3 do if the original signal at the ninth band is at 40 dB?
   (b) What if the original signal is at 20 dB?
   (c) How many bits should be allocated to the ninth band in (a) and (b) above?
10. What does MPEG Layer 3 (MP3) audio do differently from Layer 1 to incorporate temporal masking?
11. Explain MP3 in a few paragraphs, for an audience of consumer-audio-equipment salespeople.
12. Implement MDCT, just for a single 36-sample signal, and compare the frequency results to those from DCT. For low-frequency sound, which does better at concentrating the energy in the first few coefficients?
13. Convert a CD-audio cut to MP3. Compare the audio quality of the original and the compressed version—can you hear the difference? (Many people cannot.)
14. For two stereo channels, we would like to be able to use the fact that the second channel behaves, usually, in a parallel fashion to the first, and apply information gleaned from the first channel to compression of the second. Discuss how you think this might proceed.

References

1. D.W. Robinson, R.S. Dadson, A re-determination of the equal-loudness relations for pure tones. British Journal of Applied Physics **7**, 166–181 (1956)
2. H. Fletcher, W.A. Munson, Loudness, its definition, measurement and calculation. Journal of the Acoustical Society of America **5**, 82–107 (1933)
3. T. Painter, A. Spanias, Perceptual coding of digital audio. Proceedings of the IEEE **88**(4), 451–513 (2000)

4. B. Truax, *Handbook for Acoustic Ecology*, 2nd edn. (Street Publishing, Cambridge, 1999)
5. D. O'Shaughnessy, *Speech Communications: Human and Machine*. (IEEE Press, New York, 1999)
6. A.J.M. Houtsma, Psychophysics and modern digital audio technology. Philips J. Res. **47**, 3–14 (1992)
7. E. Zwicker, U. Tilmann, Psychoacoustics: matching signals to the final receiver. J. Audio Eng. Soc. **39**, 115–126 (1991)
8. D. Lubman, Objective metrics for characterizing automotive interior sound quality. in *Inter-Noise* '92, pp. 1067–1072, 1992
9. D. Pan, A tutorial on MPEG/Audio compression. IEEE Multimedia **2**(2), 60–74 (1995)
10. S. Shlien, Guide to MPEG-1 audio standard. IEEE Trans. Broadcast. **40**, 206–218 (1994)
11. P. Noll, Mpeg digital audio coding. IEEE Signal Process. Mag. **14**(5), 59–81 (1997)
12. International Standard: ISO/IEC 13818-7. Information technology—Generic coding of moving pictures and associted audio information. in *Part 7: Advanced Audio Coding (AAC)*, 1997
13. K. Brandenburg, MP3 and AAC explained. in *17th International Conference on High Quality Audio Coding*, pp. 1–12 (1999)
14. International Standard: ISO/IEC 14496-3. Information technology—Coding of audio-visual objects. in *Part 3: Audio*, 1998
15. Vorbis audio compression, (2013), http://xiph.org/vorbis/
16. J. Engdegård, B. Resch, C. Falch, O. Hellmuth, J. Hilpert, A. Hoelzer, L. Terentiev, J. Breebaart, J. Koppens, E. Schuijers, W. Oomen, Spatial Audio Object Coding (SAOC)—The Upcoming MPEG Standard on Parametric Object Based Audi Coding. In *Audio Engineering Society 124th Convention*, 2008
17. Information technology—Multimedia content description interface, Part 4: Audio. International Standard: ISO/IEC 15938-4, 2001
18. A.T. Lindsay, S. Srinivasan, J.P.A. Charlesworth, P.N. Garner, W. Kriechbaum, Representation and linking mechanisms for audio in MPEG-7. Signal Processing: Image Commun. **16**, 193–209 (2000)
19. P. Philippe, Low-level musical descriptors for MPEG-7. Signal Processing: Image Commun. **16**, 181–191 (2000)
20. M.I. Mandel, D.P.W. Ellis, Song-level features and support vector machines for music classification. In: *The 6th International Conference on Music Information Retrieval*
21. K. Brandenburg, O. Kunz, A. Sugiyama, MPEG-4 natural audio coding. Signal Processing: Image Commun. **15**, 423–444 (2000)
22. E.D. Scheirer, Structured audio and effects processing in the MPEG-4 multimedia standard. Multimedia Syst. **7**, 11–22 (1999)
23. J.D. Johnston, S.R. Quackenbush, J. Herre, B. Grill, in *Multimedia Systems, Standards, and Networks*, eds. by A. Puri and T. Chen. Review of MPEG-4 general audio coding, (Marcel Dekker Inc, New York, 2000), pp. 131–155
24. E.D. Scheirer, Y. Lee, J.-W. Yang, in *Multimedia Systems, Standards, and Networks* eds. by A. Puri & T. Chen. Synthetic audio and SNHC audio in MPEG-4, (Marcel Dekker Inc, New York, 2000), pp. 157–177