Sound is different in kind from any of the other digital media types we have considered. All other media are primarily visual, being perceived through our eyes, while sound is perceived through the different sense of hearing.† Our ears detect vibrations in the air in a completely different way from that in which our eyes detect light, and our brains respond differently to the resulting nerve impulses. Sound does have much in common with one other topic we have considered, though. Although sound is, for most of us, a familiar everyday phenomenon, like colour, it is a complex mixture of physical and psychological factors, which is difficult to model accurately.

Another feature that sound has in common with colour is that you may not always need it. Whereas a multimedia encyclopedia of musical instruments will be vastly enriched by the addition of recordings of each instrument, few, if any, Web pages need to play a fanfare every time they are visited. Sounds can be peculiarly irritating; even one's favourite pieces of music can become a jarring and unwelcome intrusion on the ears when inflicted repeatedly by a neighbour's sound system. Almost everyone has at some time been infuriated by the electronic noises of a portable games console, the cuter varieties of ringing tone of a mobile phone, or the rhythmic hiss that leaks out of the headphones of a personal stereo. The thoughtless use of such devices has become a fact of modern life; a similar thoughtlessness in the use of sound in multimedia productions should be avoided. At the very least, it should always be possible for users to turn the sound off.

Some users, though, don't need to be able to turn sounds off, because they can't hear them anyway. Not only are some people unable to hear, many others use computers that are not equipped to reproduce sound. Although new PCs intended for domestic use (and all Macs) have sound cards, older PCs, and those used in the business environment, are rarely fitted with them. It is always considerate to provide

† Text may exceptionally be rendered in other ways, but the graphic representation is the norm.

some alternative to sound, such as captions or transcripts of speech, for the benefit of those who cannot hear. If you know that your multimedia production is destined to be used in an environment where sound hardware is not typically available, then it may be advisable to avoid the use of sound altogether.

There are two types of sound that are special: music and speech. These are also the most commonly used types of sound in multimedia productions. The cultural status of music and the linguistic content of speech mean that these two varieties of sound function in a different way from other sounds and noises, and play special roles in multimedia. Representations specific to music and speech have been developed, to take advantage of their unique characteristics. In particular, compression algorithms tailored to speech are often employed, while music is sometimes represented not as sound, but as instructions for playing virtual instruments.

The Nature of Sound

If a tuning fork is struck sharply on a hard surface, the tines will vibrate at a precise frequency. As they move backwards and forwards, the air is compressed and rarefied in time with the vibrations. Interactions between adjacent air molecules cause this periodic pressure fluctuation to be propagated as a wave. When the sound wave reaches the ear, it causes the eardrum to vibrate at the same frequency. The vibration is then transmitted through the mechanism of the inner ear, and converted into nerve impulses, which we interpret as the sound of the pure tone produced by the tuning fork.

All sounds are produced by the conversion of energy into vibrations in the air or some other elastic medium. Generally, the entire process may involve several steps, in which the energy may be converted into different forms. For example, if one of the strings of an acoustic guitar is picked with a plectrum, the kinetic energy of the musician's hand is converted to a vibration in the string, which is then transmitted via the bridge of the instrument to the resonant cavity of its body, where it is amplified and enriched by the distinctive resonances of the guitar, and then transmitted through the sound hole. If one of the strings of an electric guitar is picked instead, the vibration of the string as it passes through the magnetic fields of the pickups induces fluctuations in the current which is sent through the guitar lead to an amplifier, where it is amplified and used to drive a loudspeaker. Variations in the signal sent to the speaker coil cause magnetic variations, which

are used to drive the speaker cone, which then behaves as a sound source, compressing and rarefying the adjacent air.

While the tines of a good tuning fork vibrate cleanly at a single frequency, most other sound sources vibrate in more complicated ways, giving rise to the rich variety of sounds and noises we are familiar with. As we mentioned in Chapter 2, a single note, such as that produced by a guitar string, is composed of several components, at frequencies that are multiples of the fundamental pitch of the note. Some percussive sounds and most natural sounds do not even have a single identifiable fundamental frequency, but can still be decomposed into a collection – often a very complex one – of frequency components. As in the general case of representing a signal in the frequency domain, which we described in Chapter 2, we refer to a sound's description in terms of the relative amplitudes of its frequency components as its *frequency spectrum*.

The human ear is generally considered to be able to detect frequencies in the range between 20 Hz and 20 kHz, although individuals' frequency responses vary greatly. In particular, the upper limit decreases fairly rapidly with increasing age: few adults can hear sounds as high as 20 kHz, although children can. Frequencies at the top end of the range generally only occur as components of the transient attack of sounds. (The general rule that high frequencies are associated with abrupt transitions applies here.) The highest note on an ordinary piano – which more or less defines the limit of most Western music – has a fundamental frequency of only 4186 Hz when in concert pitch.† However, it is the transient behaviour of notes that contributes most to the distinctive timbre of instruments: if the attack portion is removed from recordings of an oboe, violin, and soprano playing or singing the same note, the steady portions are indistinguishable.

† That is, using even temperament, with the A above middle C equal to 440 Hz.

Interesting sounds change over time. As we just observed, a single musical note has a distinctive attack, and subsequently it will decay, changing its frequency spectrum first as it grows, and then as it dies away. Sounds that extend over longer periods of time, such as speech or music, exhibit a constantly changing frequency spectrum. We can display the *waveform* of any sound by plotting its amplitude against time. Examination of waveforms can help us characterize certain types of sound.

The idea of a sound's frequency spectrum changing might be slightly confusing, if you accept that any complex waveform is built out of a collection of

frequency components. Strictly, Fourier analysis (as introduced in Chapter 2) can only be applied to *periodic* signals (i.e. ones that repeat indefinitely). When analyzing signals with a finite duration, various expedients must be adopted to fit into the analytic framework. One approach is to treat the entirety of a signal as one cycle of a periodic waveform; this is roughly what is done when images are broken down into their frequency components. An alternative is to use a brief section of the signal as if it were a cycle, thus obtaining a snapshot of the frequency make-up at one point. For audio signals, this provides more useful information. A spectrum analysis is typically obtained by sliding a window through the waveform to obtain a sequence of spectra, showing how the signal's frequency components change over time.

Figures 9.1 to 9.7 show waveforms for a range of types of sound. Figure 9.1 is a short example of speech: the main speaker repeats the phrase 'Feisty teenager' twice, then a more distant voice responds. You can clearly identify the syllables, and recognize that the same phrase is repeated, the second time faster and with more emphasis. In between the phrases there is almost silence – the sound was recorded in the open air and there is background noise, which is visible as the thin band running along the axis. You can see that it could be possible to extract individual syllables and recombine them to synthesize new words, and that, if it were necessary to compress speech, a lot could be achieved by removing the silences between phrases. The clearly demarcated syllables also provide a good basis for synchronizing sound with video, as we will see later.

The next four figures show the waveforms of some different types of music. The first three are purely instrumental, and do not exhibit the same character as speech. The first, Figure 9.2, is taken from an Australian aboriginal didgeridoo piece. This is characterized by a continuous drone, which requires the musician to employ a 'circular breathing' technique to maintain it. The waveform shows this drone, as the thick continuous black region, with its rhythmic modulation. Figure 9.3 shows the waveform of a piece of boogie-woogie, played

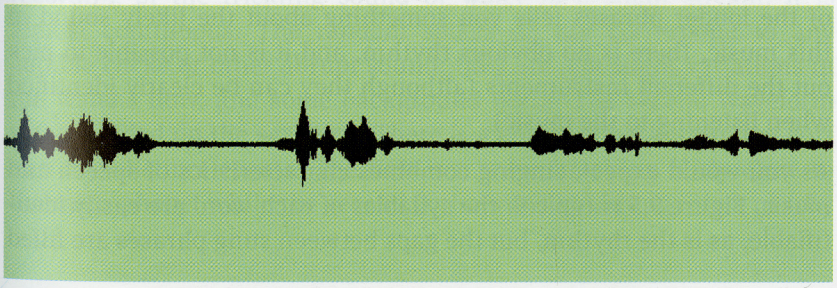


Figure 9.1 'Feisty teenager'

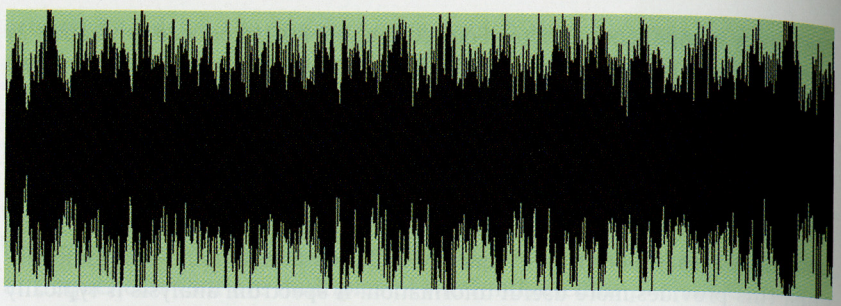


Figure 9.2 Didgeridoo

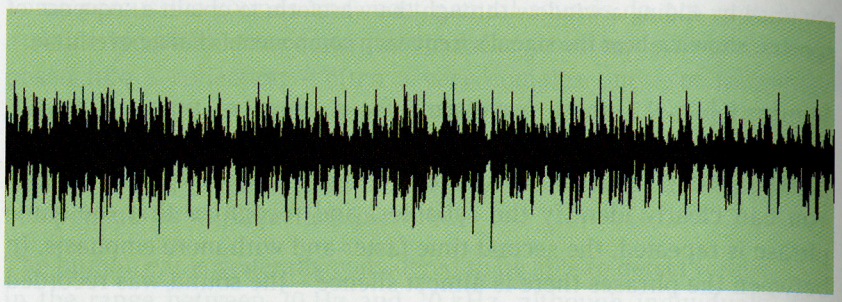


Figure 9.3 Boogie-woogie

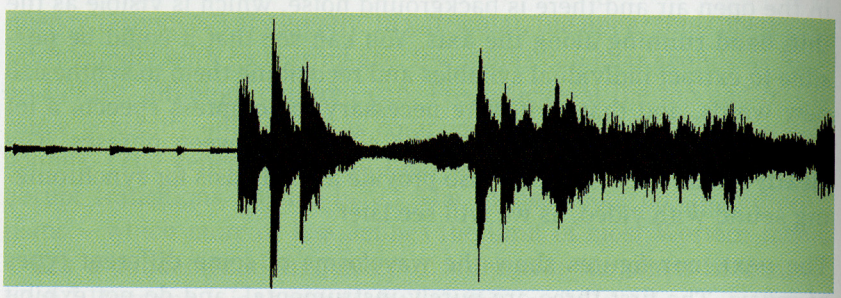


Figure 9.4 Violin, cello and piano

by a pianist accompanied by a small group. The rhythm is clearly visible, but it is not possible to distinguish the melody played by the right hand (unless, perhaps, you are a very experienced audio technician). Figure 9.4 is a completely different waveform, corresponding to a very different piece of music: a contemporary work arranged for violin, cello, and piano. It shows a great dynamic range (difference between the loudest and quietest sounds). Although the steep attack of the louder phrases tells you something about the likely sound of this music, there is no obvious rhythm, and it is not possible to pick out the different instruments (although they can be clearly identified when listening to the music).

As you would expect, singing combines characteristics of speech and music. Figure 9.5 is typical: the syllables of each word are easily identifiable, as is the rhythm, but the gaps between sung phrases are filled

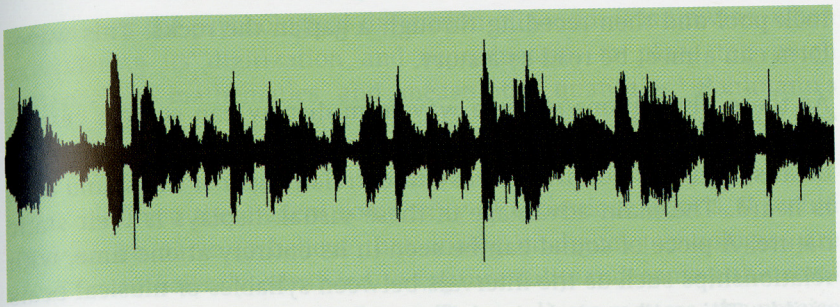


Figure 9.5 'Men grow cold...'

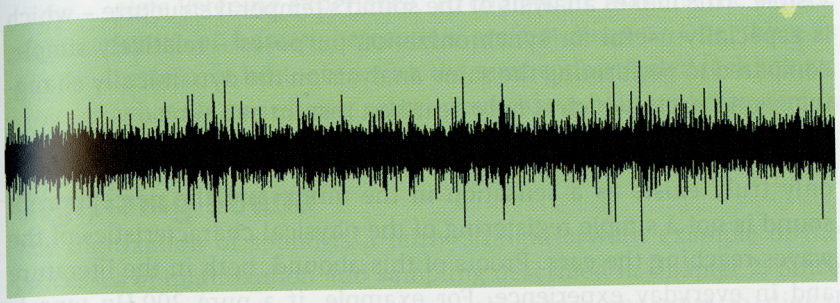


Figure 9.6 A trickling stream

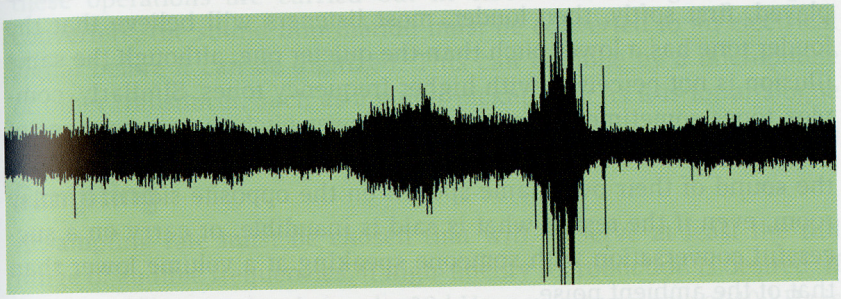


Figure 9.7 The sea

with the musical accompaniment. It is possible to see the singer's phrasing, but quite impossible to deduce the lyrics,† and, although voice prints are unique to each individual, we doubt whether any readers could identify the singer from this waveform, despite her distinctive voice. (It's Marilyn Monroe.)

Figures 9.6 and 9.7 are both natural water sounds. The first is a recording of the trickling sound of water in a small stream; it is almost continuous. The random spikes do not correspond to any audible clicks or other abrupt sound; they are just slight variations in the water's flow, and some background noise. The second waveform was recorded on the seashore. There is a constant background of surf and two distinct events. The first is a wave breaking fairly close to the microphone, while the second is the water splashing into a nearby

† Men grow cold, as girls grow old/And we all lose our charms in the end.

rock pool and then receding through a gap in the rocks. This waveform can almost be read as a story.

As these illustrations show, a waveform display can show a certain amount of the gross character of a sound, but it does not convey the details, and it is not always easy to correlate against the sound as it is heard. The main advantage of these visual displays is their static nature. A piece of sound can be seen in its entirety at one time, with relationships such as the intervals between syllables or musical beats visible. This makes analysis of the sound's temporal structure – which is especially useful for synchronization purposes – relatively simple, compared to performing the same analysis on the dynamically changing sound itself, which is only heard an instant at a time.

Waveforms and the physics of sound are only part of the story. Sound only truly exists as a sensation in the mind, and the perception of sound is not a simple registering of the physical characteristics of the waves reaching the ears. Proofs of this abound, both in the literature and in everyday experience. For example, if a pure 200 Hz tone is played, first softly, then louder, most listeners will believe that the louder tone has a lower pitch than the quieter one, although the same illusion is not perceived with higher frequency tones. Similarly, complex tones sometimes seem to have a lower pitch than pure tones of the same frequency. Most people with good hearing can distinguish the sound of their own name spoken on the opposite side of a noisy room, even if the rest of what is said is inaudible, or carry on a successful conversation with someone speaking at a volume lower than that of the ambient noise.

One of the most useful illusions in sound perception is *stereophony*. The brain identifies the source of a sound on the basis of the differences in intensity and phase between the signals received from the left and right ears. If identical signals are sent to both ears, the brain interprets the sound as coming from a non-existent source that lies straight ahead. By extension, if a sound is recorded using a pair of microphones to produce two monophonic channels, which are then fed to two speakers that are a suitable distance apart, the apparent location of the sound will depend on the relative intensity of the two channels: if they are equal it will appear in the middle, if the left channel is louder (because the original sound source was nearer to the left-hand microphone) it will appear to the left, and so on. In this way, the familiar illusion of a sound stage between the speakers is constructed.

Because of the psychological dimension of sound, it is unwise, when considering its digitization and reproduction, to place too much reliance on mathematics and measurable quantities. Pohlmann's comments† about the nature of sound and its reproduction should be borne in mind:

> "Given the evident complexity of acoustical signals, it would be naive to believe that analog or digital technologies are sufficiently advanced to capture fully and convey the complete listening experience. To complicate matters, the precise limits of human perception are not known. One thing is certain: at best, even with the most sophisticated technology, what we hear being reproduced through an audio system is an approximation of the actual sound."

† Ken C Pohlmann, *Principles of Digital Audio*, p. 5.

Digitizing Sound

The digitization of sound is a fairly straightforward example of the processes of quantization and sampling described in Chapter 2. Since these operations are carried out in electronic analogue to digital converters, the sound information must be converted to an electrical signal before it can be digitized. This can be done by a microphone or other transducer, such as a guitar pickup, just as it is for analogue recording or broadcasting.

Sampling

A sampling rate must be chosen that will preserve at least the full range of audible frequencies, if high-fidelity reproduction is desired. If the limit of hearing is taken to be 20 kHz, a minimum rate of 40 kHz is required by the Sampling Theorem. The sampling rate used for audio CDs is 44.1 kHz – the precise figure being chosen by manufacturers to produce a desired playing time‡ given the size of the medium. (The same rate is used in mini discs.) Because of the ubiquity of the audio CD, the same rate is commonly used by the sound cards fitted to computers, to provide compatibility. Where a lower sound quality is acceptable, or is demanded by limited bandwidth, sub-multiples of 44.1 kHz are used: 22.05 kHz is commonly used for audio destined for delivery over the Internet, while 11.025 kHz is sometimes used for speech. Another important sampling rate is that used by DAT (digital audio tape) recorders, and also supported by the better sound cards. Although these commonly offer a variety of sampling rates, 48 kHz is used when the best quality is desired. DAT is a very suitable medium

‡ According to legend, the time to play von Karajan's recording of Beethoven's 9th symphony.

for live recording and low budget studio work, and is often used for capturing sound for multimedia.

DAT and CD players both have the advantage that they can generate digital output, which can be read in by a suitably equipped computer without the need for extra digitizing hardware. In this respect, they resemble DV cameras. Where a digital signal cannot be produced, or where the computer is not fitted with the appropriate digital audio input, a digitizing sound card must be fitted to the computer, in the same way as a video capture board must be used for analogue video. Digital audio inputs are surprisingly uncommon, so it is often necessary for the (analogue) line output of a DAT or CD player to be redigitized by the sound card. This is clearly unfortunate, since it is preferable to work entirely with digital data and prevent noise and signal degradataion. It does, however, avoid the problem of incompatible sampling rates that can occur if, say, a recording on DAT is to be combined with an extract from a CD. Resampling audio is as undesirable as resampling images.

The necessity to resample data sampled at 48 kHz often occurs if the sound is to be combined with video. Some video applications do not yet support the higher sampling rate, even though DAT is widely used for capturing sound, and sound cards that support 48 kHz are becoming common. For multimedia work it may therefore be preferable to sample sound at 44.1 kHz, which is supported by all the major desktop video editing programs.

Sampling relies on highly accurate clock pulses to determine the intervals between samples. If the clock drifts, so will the intervals. Such timing variations are called *jitter*. The effect of jitter is to introduce noise into the reconstructed signal. At the high sampling frequencies required by sound, there is little tolerance for jitter: it has been estimated that for CD quality sound, the jitter in the ADC must be less than 200 picoseconds (200×10^{-12} seconds).

Even if they are inaudible, frequencies in excess of 20 kHz are present in the spectra of many sounds. If a sampling rate of around 40 kHz is used, these inaudible components will manifest themselves as aliasing when the signal is reconstructed. In order to avoid this, a filter is used to remove any frequencies higher than half the sampling rate before the signal is sampled.

Quantization

We mentioned in Chapter 2 that the number of quantization levels for analogue to digital conversion in any medium is usually chosen to fit into a convenient number of bits. For sound, the most common choice of sample size is 16 bits, as used for CD audio, giving 65,536 quantization levels. This is generally sufficient to eliminate quantization noise, if the signal is *dithered*, as we will describe shortly. As with images, smaller samples sizes (lower bit-depths, as we would say in the context of images) are sometimes needed to maintain small file sizes and bit rates. The minimum acceptable is 8-bit sound, and even this has audible quantization noise, so it can only be used for applications such as voice communication, where the distortion can be tolerated. In the search for higher fidelity reproduction, as many as 24 bits are sometimes used to record audio samples, but this imposes considerable demands on the accuracy of ADC circuitry.

Quantization noise will be worst for signals of small amplitude. In the extreme, when the amplitude is comparable to the difference between quantization levels, an analogue signal will be coarsely approximated

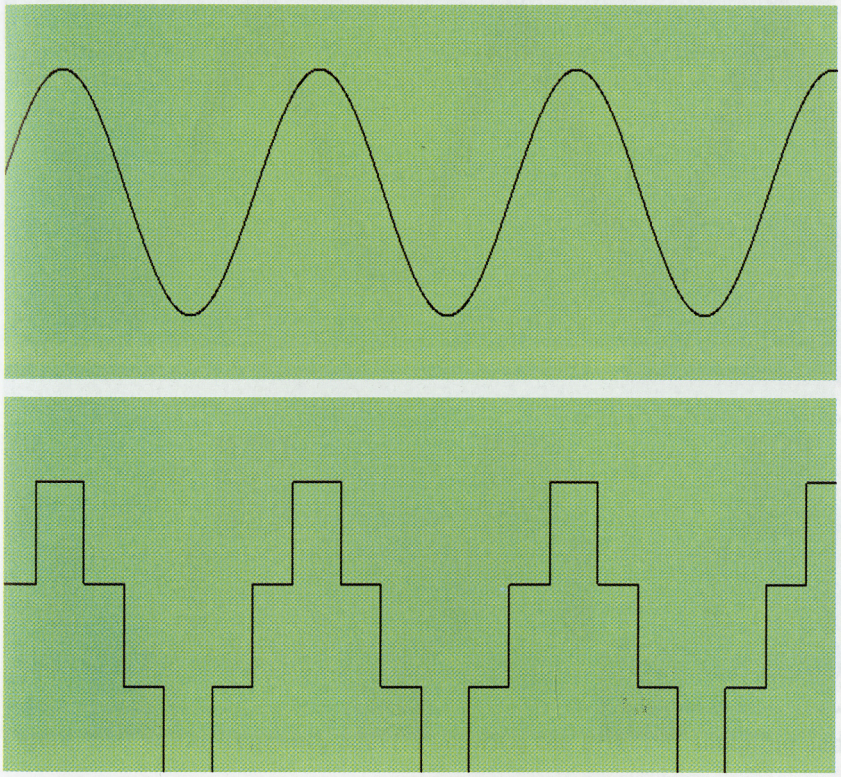


Figure 9.8 Undersampling a pure sine wave

by samples that jump between just a few quantized values. This is shown in Figure 9.8. The upper waveform is a pure sine wave; below it is a digitized version, where only four levels are available to accommodate the amplitude range of the original signal.† Evidently, the sampled waveform is a poor approximation of the original. The approximation could be improved by increasing the number of bits for each sample, but a more economical technique, resembling the anti-aliasing applied when rendering vector graphics, is usually employed. Its operation is somewhat counter-intuitive.

Before sampling, a small amount of random noise is added to the analogue signal. The word 'dithering' (which we used with a somewhat different meaning in Chapter 6) is used in the audio field to refer to this injection of noise. The effect on sampling is illustrated in Figure 9.9. The upper waveform is the original sine wave with added dither;‡ the lower waveform is a sampled version of this dithered signal. What has happened is that the presence of the noise has caused the samples to alternate rapidly between quantization levels, instead of jumping cleanly and abruptly from one to the next, as they

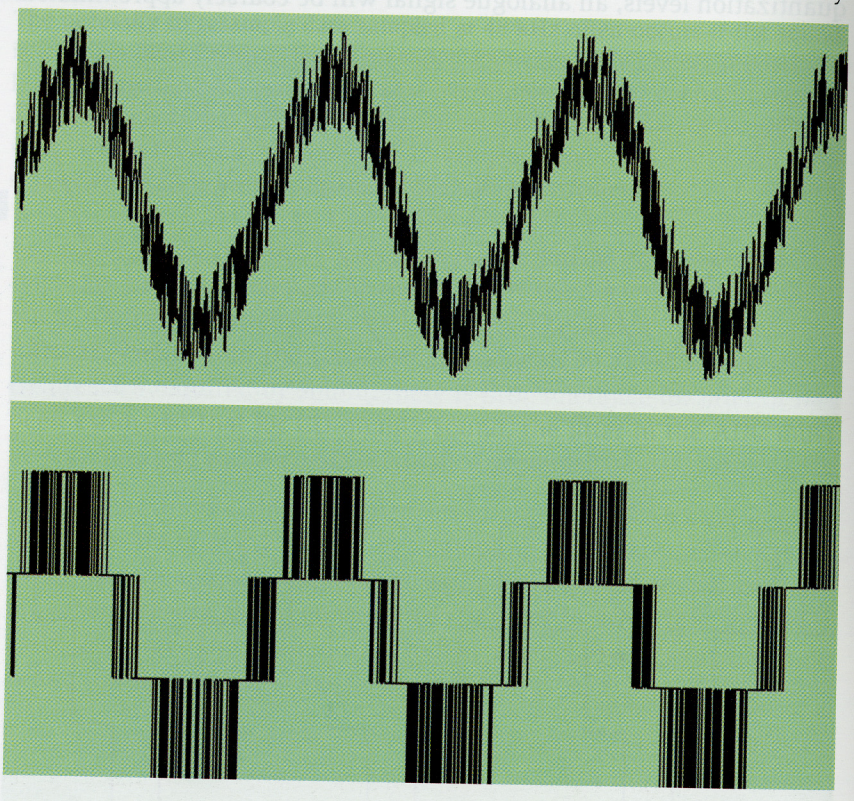


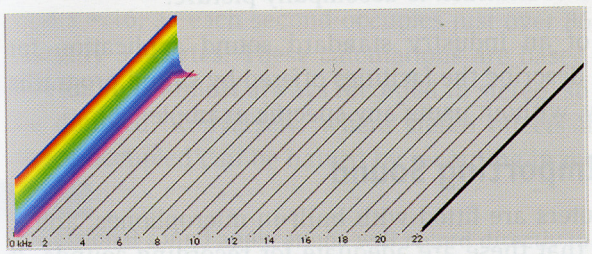
Figure 9.9 Dithering

† If you want to be scrupulous, since these images were prepared using a digital audio application, the top waveform is a 16-bit sampled sine wave (a very good approximation), the lower is the same waveform downsampled to 2 bits.

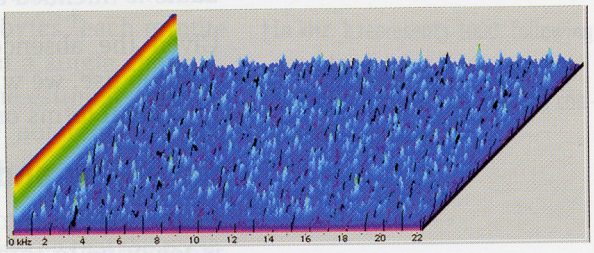
‡ We have used rather more noise than is normal, in order to show the effect more clearly.

do in Figure 9.8. The sharp transitions have been softened. Putting it another way, the quantization error has been randomized. The price to be paid for the resulting improvement in sound quality is the additional random noise that has been introduced. This is, however, less intrusive than the quantization noise it has eliminated.

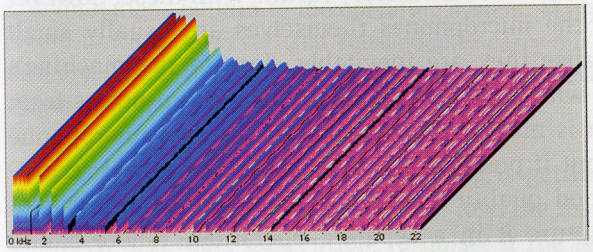
Figure 9.10 illustrates the effect of sampling and dithering on the signal's frequency spectrum. In these pictures, the horizontal x-axis represents frequency, the vertical y-axis amplitude, with the colours being used as an extra visual indication of intensity, and the back-to-front z-axis represents time. The first spectrum is the pure sine wave; as you would expect, it is a spike at the wave's frequency, which is constant over time. To its right is the spectrum of the sampled signal: spurious frequencies and noise have been introduced. These correspond to the frequency components of the sharp edges. Below the pure sine wave is the spectrum of the dithered version. The extra noise is randomly distributed across frequencies and over time. In the bottom left is the sampled version of this signal. The pure frequency has re-emerged clearly, but random noise is present where before there was none. However, although this noise will be audible, the ear will be able to discern the signal through it, because the noise is random. Where the undithered signal was sampled, the noise was concentrated near to the signal frequency, in a way that is much less easily ignored.



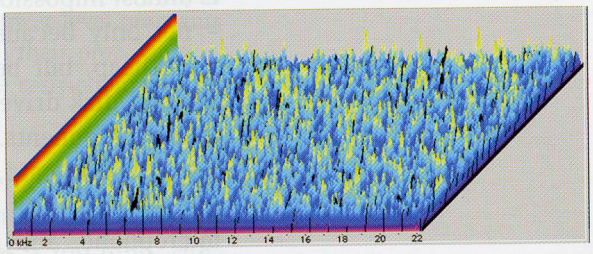
Pure sine wave



Sine wave with dithering noise



Undersampled sine wave



Undersampled dithered sine wave

Figure 9.10 Audio frequency spectra showing the effect of sampling and dithering

Processing Sound

With the addition of suitable audio input, output and processing hardware and software, a desktop computer can perform the functions of a modern multi-track recording studio. Such professional facilities are expensive and demanding on resources, as you would expect. They are also as complex as a recording studio, with user interfaces that are as intimidating to the novice as the huge mixing consoles of conventional studios. Fortunately, for multimedia, more modest facilities are usually adequate.

There is presently no single sound application that has the *de facto* status of a cross-platform desktop standard, in the way that Photoshop and Dreamweaver, for example, do in their respective fields. Several different packages, some of which require special hardware support, are in use. Most of the well known ones are biased towards music, with integrated support for MIDI sequencing (see page 317) and multi-track recording. Several more modest programs provide simple recording and effects processing facilities; where hardware support is not provided, real-time effects are not usually achievable. Video editing packages usually include some integrated sound editing and processing facilities, and some offer basic sound recording. These facilities may be adequate for multimedia production in the absence of special sound software, and are especially convenient when the audio is intended as a soundtrack to accompany picture.

Given the absence of an industry standard sound application for desktop use, we will describe the facilities offered by sound programs in general terms only, without using any specific example.

Recording and Importing Sound

Many desktop computers are fitted with built-in microphones, and it is tempting to think that these are adequate for recording sounds. It is almost impossible to obtain satisfactory results with these, however – not only because the microphones themselves are usually small and cheap, but because they are inevitably close to the machine's fan and disk drives, which means that they will pick up noises from these components. It is much better to plug an external microphone into a sound card, but if possible, you should do the actual recording onto DAT (or good quality analogue tape) using a professional microphone, and capture it in a separate operation. Where sound quality is important, or for recording music to a high standard, it will be necessary to use a properly equipped studio. Although a computer

can form the basis of a studio, it must be augmented with microphones and other equipment in a suitable acoustic environment, so it is not really practical for a multimedia producer to set up a studio for one-off recordings. It may be necessary to hire a professional studio, which offers the advantage that professional personnel will generally be available.

Before recording, it is necessary to select a sampling rate and sample size. Where the sound originates in analogue form, the choice will be determined by considerations of file size and bandwidth, which will depend on the final use to which the sound is to be put, and the facilities available for sound processing. As a general rule, the highest possible sampling rate and sample size should be used, to minimize deterioration of the signal when it is processed. If a compromise must be made, the effect on quality of reducing the sample size is more drastic than that of reducing the sampling rate: the same reduction in size can be produced by halving the sampling rate or halving the sample size; the former is better. If the signal is originally a digital one – the digital output from a DAT recorder, for example – the sample size should be matched to the incoming rate, if possible.

A simple calculation suffices to show the size of digitized audio. The sampling rate is the number of samples generated each second, so if the rate is r Hz and the sample size is s bits, each second of digitized sound will occupy $rs/8$ bytes. Hence, for CD-quality, $r = 44.1\times10^3$ and $s = 16$, so each second occupies just over 86 kbytes,† each minute roughly 5 Mbytes. These calculations are based on a single channel, but audio is almost always recorded in stereo, so the estimates should be doubled. Conversely, where stereo effects are not required, the space occupied can be halved by recording in mono.

† In kHz, k represents 1000, following normal usage, but in kbytes, the k is 1024, in accordance with the conventions of computing.

Professional sound applications will record directly to disk, so that the possible length of recordings is limited only by the available disk space and any file size limitations built in to the operating system. Many lower-level programs record to RAM, however, and subsequently carry out all their processing in memory. While this is more efficient, it imposes severe restrictions on the length of sound that can be managed.

The most vexatious aspect of recording is getting the levels right. If the level of the incoming signal is too low, the resulting recording will be quiet, and more susceptible to noise; if the level is too high, *clipping* will occur; that is, at some points, the amplitude of the incoming signal will exceed the maximum value that can be recorded. The

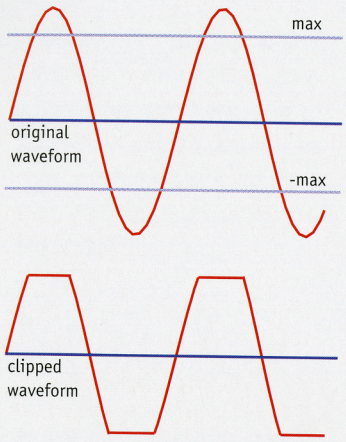


Figure 9.11 Clipping

value of the corresponding sample will be set to the maximum, so the recorded waveform will apparently be clipped off straight at this threshold. (Figure 9.11 shows the effect on a pure sine wave.) The result is heard as a particularly unpleasant sort of distortion. Ideally, a signal should be recorded at the highest possible level that avoids clipping. Sound applications usually provide level meters, so that the level can be monitored, with clipping alerts. Where the sound card supports it, a gain control can be used to alter the level. Some sound cards do not provide this function, so that the only option is to adjust the output level of the equipment from which the signal originates. Setting the level correctly is easier said than done, especially where live recordings are being made: to preserve the dynamic range of the recording, the same gain must be used throughout, but the optimum can only be determined at the loudest point. When the sound is live, this cannot be known in advance, and only experience can be used to choose gain settings. Where the material already exists on tape or CD, it is possible – and usually necessary – to make several passes in order to find the best values.

Some software includes automatic gain controls, which vary the gain dynamically according to the amplitude of the signal, in order to prevent clipping. They must, therefore, reduce the volume of louder passages, so as a side-effect, they reduce the dynamic range of the recording. This is generally undesirable, but may be necessary if suitable levels cannot be maintained throughout the recording.

It may be obvious, but it seems worth emphasizing: once a signal has been clipped, nothing can be done to restore it. Reducing the amplitude subsequently just produces a smaller clipped signal. There is no way to recover the lost waveform. Similarly, although sound programs often provide a facility for 'normalizing' a sound after recording, by amplifying it as much as possible without causing clipping, this stretches the dynamic range of the original without adding any more detail. In practice it may be necessary to use this facility, or to select and amplify particularly quiet passages within a sound editing application after the recording has been made. In principle, though, the gain should always be set correctly, both when recording to tape, and when recording or capturing to disk.

A technically simpler alternative to recording sound is to import it from an audio CD. Although audio CDs use a different format from CD-ROM, they are nevertheless a structured collection of digital data, so they can be read by suitable software. QuickTime includes an audio CD import component that allows any sound application based on

QuickTime to open tracks on a CD just like any other file. This is the simplest way of importing sounds, but most recorded music is copyrighted, so it is necessary to obtain permissions first. Copyright-free collections can be obtained, much like royalty-free image libraries, although they tend to the anodyne. Composers and musicians with access to professional recording facilities may supply their work on CD, avoiding the need for the multimedia producer to deal with the sound recording process. However, even when importing sounds from CDs there can be difficulty in getting the levels right.

The Internet is a rich source of ready-made sounds, but many are made available illegally, and others may not be legally reproduced without payment of a fee. Increasingly, though, record companies are arriving at mechanisms to provide music online, in the form of MP3 or AAC files (see below). While it may be legal to download these files and listen to them, it remains generally illegal to use them in any published form without obtaining clearance from the copyright holders.

Sound Editing and Effects

We can identify several classes of operation that we might want to apply to recorded sounds. Most of them have counterparts in video editing, and are performed for similar reasons.

First, there is editing, in the sense of trimming, combining and rearranging clips. The essentially time-based nature of sound naturally lends itself to an editing interface based on a timeline. A typical sound editing window is divided into tracks, in imitation of the separate tape tracks used on traditional recording equipment, providing a clear graphic representation of the sound through time. The sound in each track may usually be displayed as a waveform; the time and amplitude axes can be scaled, allowing the sound to be examined in varying degrees of detail. Editing is done by cutting and pasting, or dragging and dropping, selected parts of the track. Each stereo recording will occupy two tracks, one for each channel. During the editing process many tracks may be used to combine sounds from separate recordings. Subsequently, these will be mixed down onto one or two tracks, for the final mono or stereo output. When mixing, the relative levels of each of the tracks can be adjusted to produce the desired balance – between different instruments, for example.

A special type of edit has become common in audio: the creation of loops. Very short loops are needed to create voices for the electronic musical instruments known as samplers (whose functions are increas-

ingly performed by software). Here, the idea is to create a section of sound that represents the sustained tone of an instrument, such as a guitar, so that arbitrarily long notes can be produced by interpolating copies of the section between a sample of the instrument's attack and one of its decay. It is vital that the sustained sample loops cleanly; there must not be abrupt discontinuities between its end and start, otherwise audible clicks will occur where the copies fit together. Although some software makes such loops automatically, using built-in heuristics such as choosing zero crossings for each end of the loop, the best results require a detailed examination of the waveform by a person. Longer loops are used in certain styles of dance music – techno and drum'n'bass, for example – which are based on the combination of repeating sections. Again, there is a requirement for clean looping, but this time at the coarser level of rhythmic continuity. Software is also available that puts together even longer loops from a pre-recorded library, pitch- and time-shifting them so they are in the same key and tempo, to allow non-composers to produce music of a sort.

As well as editing, audio has its equivalent of post-production: altering sounds to correct defects, enhance quality, or otherwise modify their character. Just as image correction is described in terms of filters, which are a digital equivalent of traditional optical devices, so sound alteration is described in terms of gates and filters, by analogy with the established technology. Whereas analogue gates and filters are based on circuitry whose response produces a desired effect, digital processing is performed by algorithmic manipulation of the samples making up the signal. The range of effects, and the degree of control over them, that can be achieved in this way is much greater than is possible with analogue circuits. Several standard plug-in formats are in use that allow effects to be shared among programs. Although it is not an audio application, Premiere's effects plug-in format is becoming widely used; at a more professional level, the formats associated with Cubase VST and with DigiDesign ProTools are popular.

The most frequently required correction is the removal of unwanted noise. For example, in Figure 9.1, it might be considered desirable to remove the background noises that were unavoidably picked up by the microphone, since the recording was made in the open. A *noise gate* is a blunt instrument that is used for this purpose. It eliminates all samples whose value falls below a specified threshold; samples above the threshold are left alone. As well as specifying the threshold, it is usual to specify a minimum time that must elapse before a

sequence of low amplitude samples counts as a silence, and a similar limit before a sequence whose values exceed the threshold counts as sound. This prevents the gate being turned on or off by transient glitches. By setting the threshold just above the maximum value of the background noise, the gaps between words will become entirely silent. Since the noise gate has no effect on the speaker's words, the accompanying background noise will cut in and out as he speaks, which may well turn out to be more distracting than the original noise. This illustrates a general problem with noise removal: the noise is intimately combined with the signal, and although people can discriminate between the two, computer programs generally cannot.

Noise gates can be effective at removing hiss from music, since, in this case, the noise is hidden except in silent passages, where it will be removed by the noise gate. There are more sophisticated ways of reducing noise than the all-or-nothing filtering of the noise gate, though. Filters that remove certain bands of frequencies can be applied to noise that falls within a specific frequency range. *Low pass* filters, which allow low frequencies to pass through them, removing high frequencies, can be used to take out hiss; *high pass filters*, which pass the high frequencies and block the low, are used to remove 'rumble':

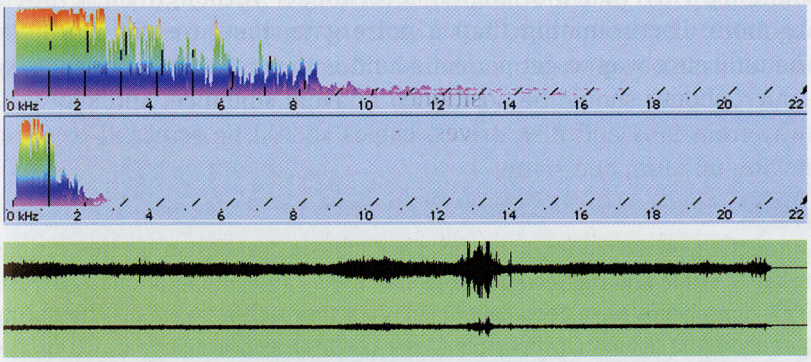


Figure 9.12 Low pass filtering

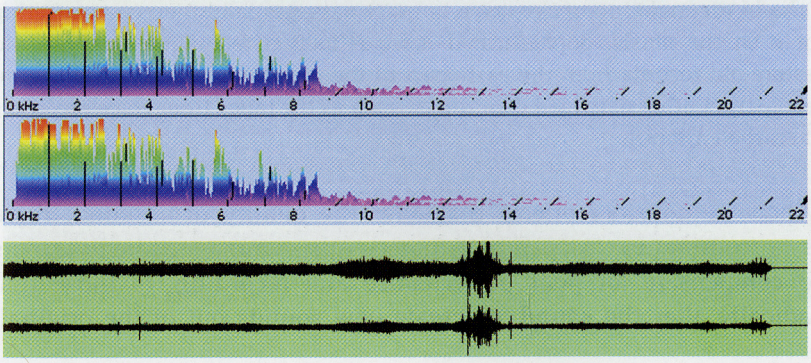


Figure 9.13 High pass filtering

low frequency noise caused by mechanical vibrations. Figures 9.12 and 9.13 show the effect of low and high pass filters on the spectrum and waveform of the sea sound from Figure 9.7. (The upper spectrum and waveform in each figure are the original sound; the lower after filtering.) A *notch filter* removes a single narrow frequency band. The commonest use of notch filters is to remove hum picked up from the mains: this will have a frequency of exactly 50 or 60 Hz, depending on the part of the world in which the noise was recorded. Some sophisticated programs offer the user the ultimate facility of being able to redraw the waveform, rubbing out the spikes that correspond to clicks, and so on. To do this effectively, however, requires considerable experience and the ability to interpret the visual display of a waveform in acoustic terms, which, as the examples shown earlier demonstrate, is not always easy.

Specialized filters are available for dealing with certain common recording defects. A *de-esser* is a filter that is intended to remove the sibilance that results from speaking or singing into a microphone placed too close to the performer. *Click repairers* are intended to remove clicks from recordings taken from damaged or dirty vinyl records. (There are also effects plug-ins that attempt to add authentic-sounding vinyl noise to digital recordings.) Although these filters are more discriminating than a noise gate, they are not infallible. The only sure way to get perfect sound is to start with a perfect take – microphones should be positioned to avoid sibilance, and kept well away from fans and disk drives, cables should be screened to avoid picking up hum, and so on.

Although the noise reduction facilities available in desktop sound applications are fairly crude and ineffectual, more elaborate – and computationally expensive – approaches have been developed. One approach is based on attempting to analyze the acoustic properties of the original recording apparatus on the basis of the make-up of the noise in quiet passages, and then compensating for it in the music. Sophisticated noise reduction techniques are used to restore old records from the early part of the twentieth century, and also to reconstruct other damaged recordings, such as the tapes from voice recorders of crashed aircraft.

When we consider effects that alter the quality of a sound, there is a continuum from those that perform minor embellishments to compensate for poor performance and recording, to those that radically alter the sound, or create new sounds out of the original.

A single effect may be used in different ways, at different points in this continuum, depending on the values of parameters that affect its operation. For example, a *reverb* effect is produced digitally by adding copies of a signal, delayed in time and attenuated, to the original. These copies model reflections from surrounding surfaces, with the delay corresponding to the size of the enclosing space and the degree of attenuation modelling surfaces with different acoustic reflectivity. By using small delays and low reflectivity, a recording can be made to sound as if it had been made inside a small room. This degree of reverb is often a necessary enhancement when the output from electric instruments has been recorded directly without going through a speaker and microphone. Although cleaner recordings are produced this way, they are often too dry acoustically to sound convincing. Longer reverb times can produce the illusion of a concert hall or a stadium. Still longer times, with the delayed signals being amplified instead of attenuated, can be used creatively to generate sustained rhythm patterns from a single chord or note. Figure 9.14 shows the effect of adding an echo to our sea sound.

Other effects can be put to a variety of uses in a similar way. These include *graphic equalization*, which transforms the spectrum of a sound using a bank of filters, each controlled by its own slider, and each affecting a fairly narrow band of frequencies. (Analogue graphic equalizers are commonly found on mid-range domestic sound

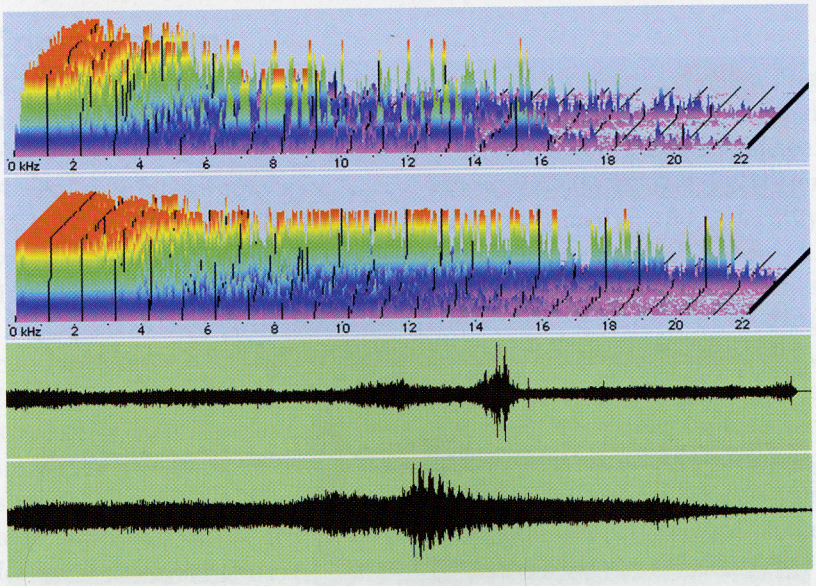


Figure 9.14 Echo

systems.) These can be used to compensate for recording equipment with idiosyncratic frequency response, or to artificially enhance the bass, for example, to produce a desired frequency balance. *Envelope shaping* operations change the outline of a waveform. The most general envelope shapers allow the user to draw a new envelope around the waveform, altering its attack and decay and introducing arbitrary fluctuations of amplitude. Specialized versions of envelope shaping include *faders*, which allow a sound's volume to be gradually increased or decreased, and *tremolo*, which causes the amplitude to oscillate periodically from zero to its maximum value.†

† To classical musicians, 'tremolo' means the rapid repetition of a single note – this does produce a periodic oscillation of amplitude. The 'tremolo arm' fitted to Fender Stratocasters and other electric guitars actually produces a periodic change of *pitch*, more accurately referred to as 'vibrato'.

Time stretching and *pitch alteration* are two closely related effects that are especially well-suited to digital sound. With analogue recordings, altering the duration of a sound can only be achieved by altering the speed at which it is played back, and this alters the pitch. With digital sound, the duration can be changed without altering the pitch, by inserting or removing samples. Conversely, the pitch can be altered without affecting the duration.

Time stretching is required when sound is being synchronized to video or another sound. If, for example, a voice-over is slightly too long to fit over the scene it describes, the soundtrack can be shrunk in time, without raising the pitch of the speaker's voice, which would happen if the voice track was simply played at a faster speed. Time stretching can also be applied to music, to alter its tempo. This makes

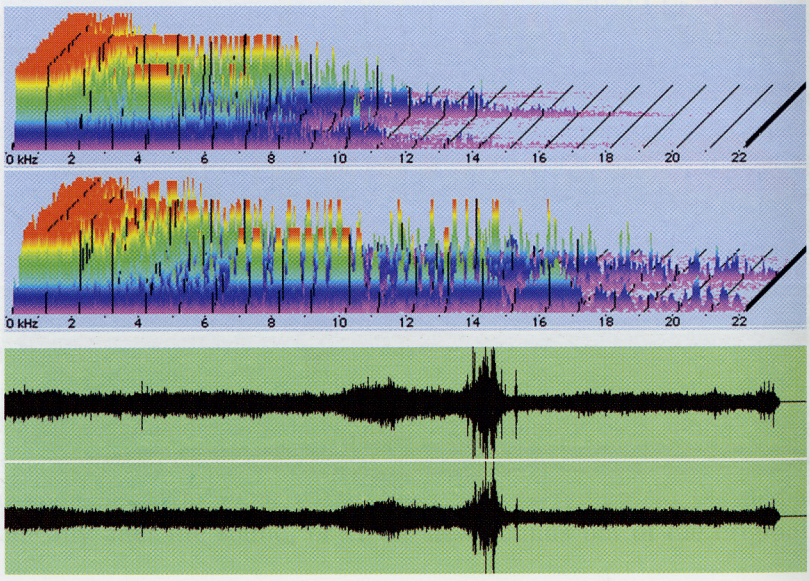


Figure 9.15 Pitch shifting

it possible to combine loops that were sampled from pieces originally played at different tempos.

Pitch alteration can be used in several ways. It can be applied uniformly to alter the pitch of an instrument, compensating for an out-of-tune guitar, for example. It can be applied periodically to add a vibrato (periodic fluctuation of pitch) to a voice or instrument, or it can be applied gradually, to produce a 'bent note', in the same way a blues guitarist changes the tone of a note by bending the string while it sounds. The all-important shape of the bend can be specified by drawing a curve showing how the pitch changes over time. Pitch alteration can also be used to transpose music into a different key; again, this allows samples from disparate sources to be combined harmoniously. Figure 9.15 shows the result of shifting the pitch of the sea up an octave (i.e. doubling the frequencies).

Beyond these effects lie what are euphemistically called 'creative' sound effects. Effects such as flanging, phasing, chorus, ring modulation, reversal, Doppler shift, and wah-wah, which were pioneered in the 1960s on albums such as the Beatles' *Sergeant Pepper's Lonely Hearts Club Band* and Jimi Hendrix's *Electric Ladyland*, have been reproduced digitally, and joined by new extreme effects such as roboticization. These effects, if used judiciously, can enhance a recording, but they are easily over-used, and are generally best enjoyed in private.

Compression

While the data rate for CD-quality audio is nothing like as demanding as that for video, it still exceeds the bandwidth of dial-up Internet connections, and lengthy recordings rapidly consume disk space. A single three-minute song, recorded in stereo, will occupy over 25 Mbytes. Hence, where audio is used in multimedia, and especially when it is delivered over the Internet, there is a need for compression. The complex and unpredictable nature of sound waveforms makes them difficult to compress using lossless methods. Huffman coding can be effective in cases where the amplitude of the sound mainly falls below the maximum level that can be represented in the sample size being used. In that case, the signal could have been represented in a smaller sample size, and the Huffman algorithm, by assigning short codes to the values it does encounter, will effectively do this automatically. This is a special case, though, and, in general, some form of lossy compression will be required.

An obvious compression technique that can be applied to speech is the removal of silence. That is, instead of using 44,100 samples with the value of zero for each second of silence (assuming a 44.1 kHz sampling rate) we record the length of the silence. This technique appears to be a special case of run-length encoding, which, as we said in Chapter 5, is lossless. However, as Figure 9.1 shows, 'silence' is rarely absolute. We would obtain little compression if we simply run-length encoded samples whose value was exactly zero; instead, we must treat samples falling below a threshold as if they were zero. The effect of doing this is equivalent to applying a noise gate, and is not strictly lossless, since the decompressed signal will not be identical to the original.

The principles behind lossy audio compression are different from those used in lossy image compression, because of the differences in the way we perceive the two media. In particular, whereas the high frequencies associated with rapid changes of colour in an image can safely be discarded, the high frequencies associated with rapid changes of sound are highly significant, so some other principle must be used to decide what data can be discarded.

Speech Compression

Telephone companies have been using digital audio since the early 1960s, and have been forced by the limited bandwidth of telephone lines to develop compression techniques that can be effectively applied to speech. An important contribution of this early work is the technique known as *companding*. The idea is to use non-linear quantization levels, with the higher levels spaced further apart than the low ones, so that quiet sounds are represented in greater detail than louder ones. This matches the way in which we perceive differences in volume.

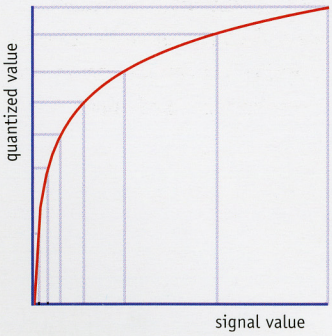


Figure 9.16 Non-linear quantization

Figure 9.16 shows an example of non-linear quantization. The signal value required to produce an increase of one in the quantized value goes up logarithmically. This produces compression, because fewer bits are needed to represent the full range of possible input values than a linear quantization scheme would require. When the signal is reconstructed an inverse process of expansion is required, hence the name 'companding' – itself a compressed version of 'compressing/expanding'.

Different non-linear companding functions can be used. The principal important ones are defined by ITU Recommendations for use

in telecommunications. Recommendation G.711 defines a function called the μ-*law*, which is used in North America and Japan. It has been adopted for compressing audio on Sun and NeXT systems, and files compressed in accordance with it are commonly found on the Internet. A different ITU Recommendation is used in the rest of the world, based on a function known as the *A-law*.

Telephone signals are usually sampled at only 8 kHz. At this rate, μ-law compression is able to squeeze a dynamic range of 12 bits into just 8 bits, giving a one-third reduction in data rate.

The μ-law is defined by the equation:

$$y = \log(1 + \mu x)/\log(1 + \mu) \text{ for } x \geq 0$$

where μ is a parameter that determines the amount of companding; $\mu = 255$ is used for telephony.

The A-law is:

$$y = \begin{cases} Ax/(1 + \log A) & \text{for } 0 \leq |x| < 1/A \\ (1 + \log Ax)/(1 + \log A) & \text{for } 1/A \leq |x| < 1 \end{cases}$$

Another important technique that was originally developed for, and is widely used in, the telecommunications industry is *Adaptive Differential Pulse Code Modulation (ADPCM)*.† This is related to inter-frame compression of video, in that it is based on storing the difference between consecutive samples, instead of the absolute value of each sample. Because of the different nature of audio and video, and its origins in hardware encoding of transmitted signals, ADPCM works somewhat less straightforwardly than a simple scheme based on the difference between samples.

Storing differences will only produce compression if the differences can be stored in fewer bits than the sample. Audio waveforms can change rapidly, so, unlike consecutive video frames, there is no reason to assume that the difference will necessarily be much less than the value. Basic *Differential Pulse Code Modulation (DPCM)* therefore computes a predicted value for a sample, based on preceding samples, and stores the difference between the prediction and the actual value. If the prediction is good, the difference will be small. *Adaptive DPCM* obtains further compression by dynamically varying the step size used to represent the quantized differences. Large differences are quantized using large steps, small differences using small steps, so the amount of detail that is preserved scales with the size of the

† Pulse Code Modulation is the term used in audio and communications circles for encoding digital data as a sequence of pulses representing ones and zeros. Whereas this is more or less the only sensible representation for computer use, alternatives, such as Pulse Width Modulation, exist where the data is to be represented as a stream for transmission, rather than as stored values.

difference. The details of how this is done are complicated, but as with companding, the effect is to make efficient use of bits to store information, taking account of its rate of change.

ITU Recommendation G.721 specifies a form of ADPCM representation for use in telephony, with data rates of 16 kbps and 32 kbps. Lower rates can be obtained by a much more radical approach to compression. *Linear Predictive Coding* uses a mathematical model of the state of the vocal tract as its representation of speech. Instead of transmitting the speech as audio samples, it sends parameters describing the corresponding state of the model. At the receiving end, these parameters can be used to construct the speech, by applying them to the model. The details of the model and how the parameters are derived from the speech lie beyond the scope of this book. Speech compressed in this way can be transmitted at speeds as low as 2.4 kbps. Because the sound is reconstructed algorithmically, it has a machine-like quality, so it is only suitable for applications where the content of the speech is more important than a faithful rendition of someone's voice.

Perceptually Based Compression

The secret of effective lossy compression is to identify data that doesn't matter – in the sense of not affecting perception of the signal – and to throw it away. If an audio signal is digitized in a straightforward way, data corresponding to sounds that are inaudible may be included in the digitized version. This is because the signal records all the physical variations in air pressure that cause sound, but the perception of sound is a sensation produced in the brain, via the ear, and the ear and brain do not respond to the sound waves in a simple way.

Two phenomena in particular cause some sounds not to be heard, despite being physically present. Both are familiar experiences: a sound may be too quiet to be heard, or it may be obscured by some other sound. Neither phenomenon is quite as straightforward as it might appear.

The *threshold of hearing* is the minimum level at which a sound can be heard. It varies non-linearly with frequency, as shown in Figure 9.17. A very low or very high frequency sound must be much louder than a mid-range tone to be heard. It is surely no coincidence that we are most sensitive to sounds in the frequency range that corresponds to human speech. When compressing sound, there is no point in retaining sounds that fall below the threshold of hearing, so a com-

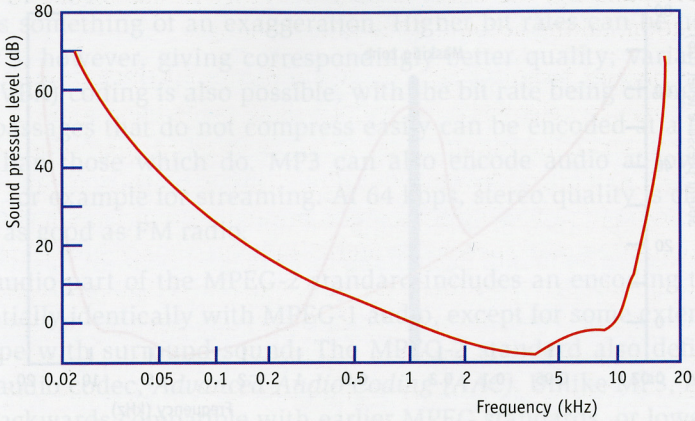


Figure 9.17 The threshold of hearing

pression algorithm can discard the corresponding data. To do this, the algorithm must use a *psycho-acoustical model* – a mathematical description of aspects of the way the ear and brain perceive sounds. In this case, what is needed is a description of the way the threshold of hearing varies with frequency.

Loud tones can obscure softer tones that occur at the same time.† This is not simply a case of the loud tone 'drowning out' the softer one; the effect is more complex, and depends on the relative frequencies of the two tones. *Masking*, as this phenomenon is known, can be conveniently described as a modification of the threshold of hearing curve in the region of a loud tone. As Figure 9.18 shows, the threshold is raised in the neighbourhood of the masking tone. The raised portion or *masking curve* is non-linear, and asymmetrical, rising faster than it falls. Any sound that lies within the masking curve will be inaudible, even though it rises above the unmodified threshold of hearing. Thus, there is an additional opportunity to discard data. Masking can be used more cleverly, though. Because masking hides noise as well as some components of the signal, quantization noise can be masked. Where a masking sound is present, the signal can be quantized relatively coarsely, using fewer bits than would otherwise be needed, because the resulting quantization noise can be hidden under the masking curve.

It is evident that the phenomena just described offer the potential for additional compression. It is not obvious how a compression algorithm can be implemented to take advantage of this potential.

† In fact, they can also obscure softer tones that occur a little later or, strange as it may seem, slightly earlier.

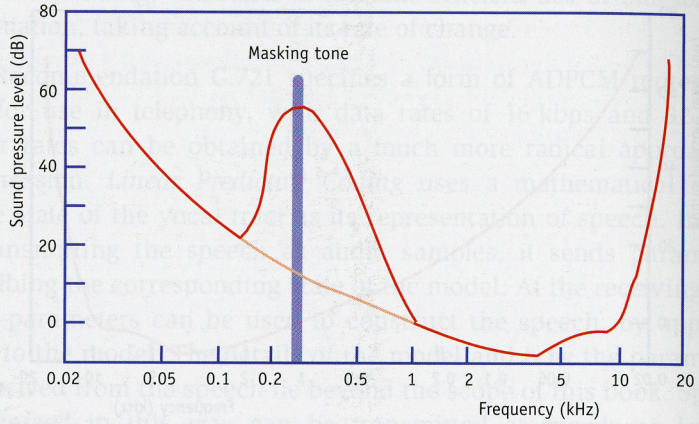


Figure 9.18 Masking

The approach usually adopted is to use a bank of filters to split the signal into bands of frequencies; 32 bands are commonly used. The average signal level in each band is calculated, and using these values and a psycho-acoustical model, a masking level for each band is computed. That is, it is assumed that the masking curve within each band can be approximated by a single value. If the signal in a band falls entirely below its masking level, that band is discarded. Otherwise, the signal is quantized using the least number of bits that causes the quantization noise to be masked.

Turning the preceding sketch into a working algorithm involves many technical details that lie beyond the scope of this book. The best known algorithms that have been developed are those specified for audio compression in the MPEG standards. MPEG-1 and MPEG-2 are primarily video standards, but, since most video has sound associated with it, they also include audio compression. MPEG audio has been so successful that it is often used on its own purely for compressing sound, especially music.

MPEG-1 specifies three *layers* of audio compression. All three layers are based on the principles just outlined. The encoding process increases in complexity from Layer 1 to Layer 3, while as a result, the data rate of the compressed audio decreases: the quality obtained at 192 kbps for each channel at Layer 1 only needs 128 kbps at Layer 2, and 64 kbps at Layer 3. (These data rates will be doubled for stereo.) MPEG-1 Layer 3 audio, or *MP3* as it is usually called,† achieves compression ratios of around 10:1, while maintaining high quality.

† MP3 does not, despite what you may sometimes read, stand for MPEG-3. There is no MPEG-3.

A typical track from a CD can be compressed to under 3 Mbytes. The sound quality at this rate is sometimes claimed to be 'CD quality', but this is something of an exaggeration. Higher bit rates can be used at Layer 3, however, giving correspondingly better quality; variable bit rate (VBR) coding is also possible, with the bit rate being changed, so that passages that do not compress easily can be encoded at a higher rate than those which do. MP3 can also encode audio at lower bit rates, for example for streaming. At 64 kbps, stereo quality is claimed to be as good as FM radio.

The audio part of the MPEG-2 standard includes an encoding that is essentially identically with MPEG-1 audio, except for some extensions to cope with surround sound. The MPEG-2 standard also defined a new audio codec, *Advanced Audio Coding (AAC)*. Unlike MP3, AAC is not backwards compatible with earlier MPEG standards, or lower layers. By abandoning backwards compatibility, AAC was able to achieve higher compression ratios at lower bit rates than MP3. Like MP3, AAC is based on perceptual coding, but it uses additional techniques and a more complicated implementation. Subjective listening tests consistently rate AAC quality as superior to MP3 at the same bit rates, and the same subjective quality is attained by AAC at lower rates than MP3. For instance, AAC audio at 96 kbps is considered to be superior to MP3 at 128 kbps. AAC has been incorporated and extended in MPEG-4, where it forms the basis of coding of natural audio (as distinct from speech and synthesized sound).

Lossy compression always sounds like a dubious practice – how can you discard information without affecting the quality? In the case of MPEG audio, the argument is that the information that has been discarded is inaudible. This contention is based on extensive listening tests, and is supported by the rapid acceptance of MP3 as a format for downloading music. (It should also be borne in mind that, although some people care obsessively about the quality of audio reproduction, most people aren't very particular, as witnessed by the enduring popularity of the analogue compact audio cassette.) As with any lossy form of compression, though, MPEG audio will deteriorate progressively if it is decompressed and recompressed a number of times. It is therefore only suitable as a delivery format, and should not be used during production, when uncompressed audio should be used whenever possible.

Formats

Most of the development of digital audio has taken place in the recording and broadcast industries, where the emphasis is on physical data representations on media such as compact disc and digital audio tape, and on data streams for transmission and playback. There are standards in these areas that are widely adhered to. The use of digital sound on computers is a much less thoroughly regulated area, where a wide range of incompatible proprietary formats and *ad hoc* standards can be found. Each of the three major platforms has its own sound file format: AIFF for MacOS, WAV (or WAVE) for Windows, and AU† for Unix, but support for all three by applications is common on all platforms. The standardizing influence of the Internet has been less pronounced in audio than it is in graphics. QuickTime, Windows Media and RealAudio are all widely used as container formats for audio compressed with different codecs. The popularity of music swapping services using MP3 has led to its emergence as the leading audio format on the Internet.

† More properly, WAV is the Audio Waveform file format, actually a variety of Microsoft RIFF file, and AU is the NeXT/Sun audio file format.

MP3

MP3 has its own file format, in which the compressed audio stream is split into chunks called 'frames', each of which has a header, giving details of the bit rate, sampling frequency and other parameters. The file may also include metadata tags, oriented towards musical content, giving the title of a track, the artist performing it, the album from which it is taken, and so on. MP3 files have been widely used for downloading and storing music on computers and mobile music players. (The growth of legal music download services, as against informal peer-to-peer swapping services that have caused so much agitation in the music industry, may see MP3 superseded as the format of choice.)

MP3 is, however, primarily an encoding, not a file format, and MP3 data may be stored in other types of file. In particular, QuickTime may include audio tracks encoded with MP3, and Flash's SWF movies use MP3 to compress any sound they may include.

Streaming Audio Formats

In Chapter 7, we explained that streamed video resembles broadcast television. Streamed audio resembles broadcast radio. That is, sound is delivered over a network and played as it arrives, without having to be stored on the user's machine first. As with video, this allows live