or playback systems becomes a distinctive feature of the musical process. Sampling an external sound is only ever an approximation to its natural state. Digitizing a natural sound may capture its detail in a highly effective way, opening it up for manipulation and transformation but, however fine-grained the sampling rate may be, there is always a sense of distance. Such issues of quality can generate whole aesthetics, such as the ‘retro’ 8-bit sound that is evocative of the early days of computer soundcards and still popular today.

The digital method of using numbers to encode sound is efficient because it is not subject to the same kinds of errors introduced by the physical properties of the analog medium. It is non-linear and, at least in theory, does not suffer degradation when identical copies are made (although a signal can sometimes be degraded at other stages in the process). This is the advantage of digital over analog. However, whether a sound comes from an external source (such as an instrument, voice or any other acoustic sound) or whether the sound input is a microphone or a cable (line), it will at some point have to be converted into digital form to be handled by a computer or other digital device, and the same applies in reverse at the other end of the chain.

A simple way to explore all the above ideas is to record an acoustic signal (a voice or musical instrument) with a good microphone plugged directly into the computer. By using software which allows for different settings of bit rate, sampling rate and resolution, the aural consequences of digitizing sound can be investigated. Setting the level deliberately high will also give the clipping effect. Digital audio, like film, is an illusion, because it consists only of a set of values which, when correctly decoded by the technology, translates into a sounding result. Of course, analog sound is also a kind of illusion, but its apparent reality derives from the nature of the medium onto which it is literally written. The underlying reality of digital audio is mathematics, whereas the underlying reality of analog audio is physics. This does not mean that the physical consequences of the digital manipulation of sound are any less, but the abstraction that sits at the heart of digital sound-processing also lies behind many attempts by instrument-makers and musicians to introduce more physicality and gesture into the act of performing digital music.

**Fourier Transforms**

Digitization is the basic process by which analog sound is encoded as digital audio, but sound organization and manipulation depends upon the computer’s ability accurately to analyse the contents of a sound file. The most common method is called a *Fourier transform* (FT).¹ This is a digital representation of a sound in the time and frequency domains that reflects the detail of its changing spectrum to a level of accuracy that is limited by the processing power and speed of the computer. A good way to envisage an FT is to imagine a spreadsheet containing a number of rows and columns of cells. The rows represent narrow bands of harmonics, grouped by frequency range, and are called *bins*. The columns contain single digital samples. At a sample rate of 44,100 Hz (CD quality), this means each column lasts 1/44,100ths of a second. Each cell, therefore, contains a tiny amount of information about a sample and its constituent frequencies. In other words, there is a very small amount of energy in a cell.

Now, in order to analyse a sound for longer than the duration of one sample, it is necessary to increase the number of columns in the imaginary spreadsheet. This is called the *buffer size*, *window length* or *window size* and typically increases by a factor of two,
thus: 512 samples, 1024, 2048, 4096, etc. Raising the number of rows in the imaginary spreadsheet can similarly increase the frequency resolution. This has the effect of narrowing the frequency range of each bin. The size of the window, therefore, gives the frequency resolution and the time resolution. The buffer size divided by the sample rate gives the duration of the buffer in milliseconds. The sample rate divided by the buffer size gives the frequency range of each bin.

In practice, there is a trade-off between these two resolutions. A bigger buffer gives more accurate pitch definition but less accurate time definition. A smaller buffer gives less accurate pitch definition and more accurate time definition (see Table 6.1).

Figures 6.2, 6.3 and 6.4 show three sonograms of the same sound, of 3 seconds' duration, with increasing window sizes. Time is shown along the x-axis, frequency up the y-axis. Notice how the bins become smaller.

<table>
<thead>
<tr>
<th>S/R</th>
<th>Window size time res (samp)</th>
<th>Window size time res (ms)</th>
<th>Bin width freq res (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>44100</td>
<td>32</td>
<td>0.73</td>
<td>1378.13</td>
</tr>
<tr>
<td>44100</td>
<td>64</td>
<td>1.45</td>
<td>689.06</td>
</tr>
<tr>
<td>44100</td>
<td>128</td>
<td>2.9</td>
<td>344.53</td>
</tr>
<tr>
<td>44100</td>
<td>256</td>
<td>5.8</td>
<td>172.27</td>
</tr>
<tr>
<td>44100</td>
<td>512</td>
<td>11.61</td>
<td>86.13</td>
</tr>
<tr>
<td>44100</td>
<td>1024</td>
<td>23.22</td>
<td>43.07</td>
</tr>
<tr>
<td>44100</td>
<td>2048</td>
<td>46.44</td>
<td>21.53</td>
</tr>
<tr>
<td>44100</td>
<td>4096</td>
<td>92.88</td>
<td>10.77</td>
</tr>
<tr>
<td>44100</td>
<td>8192</td>
<td>185.76</td>
<td>5.38</td>
</tr>
<tr>
<td>44100</td>
<td>16384</td>
<td>371.52</td>
<td>2.69</td>
</tr>
<tr>
<td>44100</td>
<td>32768</td>
<td>743.04</td>
<td>1.35</td>
</tr>
</tbody>
</table>

\[
\frac{1000}{\text{(sample rate ÷ time res (samp))}} \quad \frac{\text{sample rate ÷ time res (samp)}}{}
\]

*Figure 6.2 Sonogram – window size: 64 © Peter Batchelor*
An FT therefore, analyses the sound into a continuous spectrum of its constituent harmonics. It does this by calculating the frequency, amplitude and phase of the various sine waves within the given time frame. Because the amount of data produced by this process is large and the number of calculations per second can be prohibitively great, an algorithm has been developed to enable FTs to be calculated quickly. This is called a fast Fourier transform, or FFT, and is the most common form of digitally processed sound. Input to an FFT is usually given an envelope whose shape can be chosen by the user. There are several of these window shapes, of which the best are: Hamming, Hanning and Blackman.

Why is understanding such complex digital analysis so important? Because it provides the means by which musicians may shape and control their sound, which is one of the main goals. A thorough knowledge of the way digital sound is created, handled and manipulated is an essential skill. Furthermore, this knowledge and understanding will suggest creative possibilities. Consider the following example to illustrate the point.

The kind of errors thrown up in the timing processes of D to A conversion may be called glitches. However, a glitch is also just a fault in a system, such as an unwanted electrical pulse or a programming error. The idea could be extended to the surface
noise on an LP, for example, or the accidental sound made by touching the tip of a jack plug before connecting it to an instrument. In the late 1990s, various artists and musicians became interested in these unwanted sounds and began using them, mainly in place of percussion. The result has been a burgeoning ‘glitch’ music genre, which operates on the margins of the techno scene. A survey of the Mille Plateaux record label’s ‘Clicks and Cuts’ series will give an idea of the possibilities opened up by these artists. The best among them know what they are doing, just as surely as the best violinist knows how to get a good sound from the violin. These are not just arbitrary uses of digital artefacts: the digital musician combines engineering skills with artistic innovation.

FFT is not the only way to manipulate the spectral content of a sound. In the past decade or so, an alternative called the wavelet transform has begun to appear. These transforms parallel Heisenberg’s Uncertainty Principle, which states that the momentum and position of a moving particle cannot be known simultaneously. Whilst the FT works fine for stationary signals, that is to say, it renders the time and frequency information necessary for the spectrum of a sound to be understood discretely, as though it happened once and remained unchanged, the wavelet transform attempts to perform the same function for sounds whose spectra vary over time. Since Heisenberg’s Uncertainty Principle excludes the possibility of complete knowledge of this, the Continuous Wavelet Transform (CWT) just gives as good a resolution as possible. The finer the resolution, the more computation required, with the result that the quality of CWT implementations, such as pitch-shifters and time-stretchers, in commercial software packages is variable.

---

**Project 18 (Elementary): FFT Processing**

*Introduction*

Digital sound processing might appear to transform a sound in a particular way (altering the pitch, slowing down, etc.), but this is merely the aural consequence of the production of a new collection of data. Any digital process can produce strange or unwanted artefacts, sounds that do not contribute to the aural impression the process is designed to create. In most cases, good software means that this is not a problem. This project, however, positively explores the creative possibilities that endless repetitions of a process can produce.

*The Project*

Take a recorded sound and apply an effect (timestretch, reverb, pitch shift, distortion, anything) repeating the effect over and over again until the sound is nothing like the original sound. Retain a copy of each step in the process. Now listen back to the sounds one at a time in reverse order.

*Notes*

Notice how the process diverges further and further from how it might be imagined to sound. Rather than going deeper ‘inside’ the sound, the process is in fact generating a new sound. Ask each time: What is heard? What does it mean? What is its source?
RECOMMENDED LISTENING

  Pioneering band that worked with damaged audio products and mutilated CDs to
  create glitch music.

  The definitive series of glitch recordings.

FILE FORMATS

Once a sound has been analysed by the computer, it may be manipulated and trans-
formed. However, the sheer quantity of data in a typical audio file can be prohibitively
high. In order to maintain an aurally convincing illusion while being readable by software
and, where necessary, reducing the size of the file, the audio data is normally organized
into a specific format. These file formats generally use some kind of compression, which
comes in two forms: lossless and lossy.

Lossless compression does not sacrifice any audio information but still manages to
reduce the amount of data in the raw file. Typical lossless file formats include .wav
(wave-form audio format) on the PC, and .aiff (audio interchange file format) on the
Mac.

A well-known example of lossy compression is the mp3, or, to give it its full title,
MPEG audio layer 3, format. MPEG stands for ‘Moving Pictures Experts Group’, which
is the body charged with setting standards for the encoding of audio and video. Mp3 is
typical of lossy compression formats, in that it discards musical information that may be
regarded as less important or even irrelevant to the listener, such as very quiet material
simultaneous with or immediately following very loud material.

Lossy compression formats use various types of encryption, which produce generally
better sounding results as the number of kilobits per second (kbps) is increased. For mp3
encoding, 128 kbps is generally considered the lowest rate to produce good quality, but
this is a matter of opinion and will vary according to the music that is being encoded
and the ears of the individual listener. Other examples of lossy compression include the
successors to mp3, WMA (windows audio media), AAC (advanced audio compression)
used by Apple for its iTunes and iPod products, and Ogg Vorbis, which is an open, pat-
et-free compression.
Project 19 (Elementary): File Formats

Introduction

This project is quite easy to achieve, but is a very useful way to understand the musical implications of different audio file formats. Musicians can tend to accept what the computer presents to the ear as a standard, but in fact there is considerable variation between file formats, often depending on the type of music that is encoded. Even lossless formats can exhibit certain audible characteristics, depending on the sample rate and bit rate.

The Project

Choose some music containing a wide range of contrasting dynamic (loud and soft) levels and sounds. Now use a software package to save the file in various formats and vary the settings to achieve different quality levels. Listen carefully and comparatively to the results. It might be a good idea to try this with several different examples. The aim is to try to hear what has changed, what has been lost, and what has been emphasized. An extension of this project is to create some music which is designed specifically for a certain file format.

Notes

In an mp3 culture, certain kinds of music adapt better to the file format. This project will quickly establish what those might be and why that is the case. This has implications for music that does not work so well in that format. Formats will change and music will adapt, but one interesting question to explore is whether the arrival of lossy compression formats has changed musical culture, particularly with reference to the kind of music that is made?

FURTHER READING

  A standard work on the subject written from a technical perspective which sets out the primary information for musical applications as well.

REPRESENTING

The preceding discussion looked at the way computers represent sound to themselves. However, the way in which digital music is represented to us, the users, is somewhat different and no less fundamental to our understanding. There are many forms of representation, from visualizations to code, from text files to metadata or keywords. Analysis of these may develop complex representations, leading to musical information retrieval systems and search possibilities by pattern-matching (rhythmic, pitch, melodic, spectrum, etc.), automated playlist-building, music recognition and so on. However, the most common form of representation is through a visual display on an interface. Such representations are often modelled on traditional electronic displays, but exploit the processing capabilities of the computer to deliver enriched information.
Wave-form Diagrams

The basic representation of a sound is the wave-form diagram. Figure 6.5 is a simple flat wave-form diagram of a pure sine wave, a single harmonic, oscillating at a given amplitude and phase. The lack of any additional harmonics means that the wave form is smooth and regular, and the resulting sound is extremely pure. The second image represents exactly the same sine wave but shows the harmonic spectrum, which in this case consists, predictably, of a single peak at the relevant frequency.

![Wave-form Diagram](image)

Figure 6.5 Sine wave © Peter Batchelor

A more complex wave-form diagram, such as the one shown in Figure 6.6, gives a summary of the harmonic activity in a sound, resulting in a single line which uses a process called 'spectral averaging' to convey a certain amount of information about (among other things) periodicity, or lack of it. This is useful for getting a general impression of the timbral character and, especially, for identifying pitch, but what is needed for a more sophisticated understanding is detailed information about all the harmonics and their complex relationships over time; in other words, the spectrum. There are a number of solutions to this, but the most commonly encountered is the spectrogram (also sometimes called a sonogram).

Spectrograms

Spectrograms are easy to produce digitally and provide ready information about the spectrum of any sound. Some software enables audible spectrograms, so that the user may listen to an individual harmonic. A typical spectrogram represents time along the x-axis and frequency along the y-axis, but the most important and distinctive feature is the amplitude of the various harmonics, which are usually shown as striations of varying depths or
intensity. These may be colour (rather like a weather map) or monochrome, but in general the more dense or intense the striation, the stronger that particular harmonic will be in the spectrum. Figure 6.7 shows a two-dimensional (2D) spectrogram of a piano playing a note, followed by another a fifth above. The harmonic series may be clearly seen.

Figure 6.8 shows a 3D spectrogram of a sound showing the amplitude of each of its harmonics and their changing relationships over time, represented by the individual
envelopes. Notice that the characteristics of the attack and decay of the sounds are as clear as the central continuant. This is a great advantage of spectrograms, whereas the conventional wave-form diagram is most useful in the continuant only. The perception of timbre is often located during the attack stage. There have been many psychoacoustic experiments to show that listeners cannot distinguish one musical instrument from another, for example, if the attack is removed and only the continuant is heard.

Figure 6.9 shows a 2D spectrogram of the spoken word ‘sound’, showing frequency information represented vertically and time horizontally. Notice the higher frequencies are most dense during the sibilant ‘s’ sound, whereas lower frequencies predominate during the rounded ‘ound’.

SYNTHESIZING

The grandfather of digital synthesis, Max Mathews, observed that it is ‘very hard to create new timbres we hear as interesting, powerful and beautiful’. Whereas recording a sample has the advantage that the process captures the extraneous aspects, the ‘dirt’ of the sound, a synthetic sound created from within the computer has a certain ‘cleanliness’. The extent to which this may be attractive or valuable is an aesthetic matter. We seem to have an innate tendency to prize sounds that come from nature and much effort is
devoted in digital music to giving synthetic sounds that same 'natural' quality. On the other hand, a sampled sound is in the end identical to a synthesized sound, inasmuch as it consists of samples. Processing a sample may remove all trace of its natural origins, creating something that sounds synthetic. Does this reduce its capacity to be 'interesting, powerful and beautiful'? And does this mean that a purely synthesized sound may never be 'interesting, powerful and beautiful'?

**INFOGRAPHIC: SINUSOIDS**

The sine wave, or sinusoid, is most useful in synthesis, since it can readily be isolated or generated using electronic or digital equipment. However, synthesis does not normally restrict itself to pure sine waves but uses more complex variants, such as:

- **square waves** comprising the odd numbered harmonics of a fundamental
- **triangle waves** consisting of a fundamental frequency, with odd-numbered harmonics (these roll off in amplitude more quickly than in a square wave, producing a smoother sound)
- **sawtooth waves**, which normally ramp upwards over time, then sharply drop, although there are some which ramp downwards over time then sharply rise.

Notice how in Figures 6.10, 6.11 and 6.12 the name of the wave form derives from its visual representation rather than from its sound. Already, even with such simple waves, the representation has become the way in which the sound is understood.
Figure 6.10 Square wave © Peter Batchelor

Figure 6.11 Triangle wave © Peter Batchelor
Most synthesis relies upon the analytical capabilities of computers described earlier to produce effective results. The creation of ever better algorithms for the many types of synthesizing process has preoccupied musicians and scientists since the 1950s. The earliest examples had their roots in traditional electronic processes, such as *additive* and *subtractive* synthesis. In additive synthesis, sine tones are combined and built up to create new sounds, whereas in subtractive synthesis filters are applied to a source in order to sculpt similarly new sounds. Both processes may readily be performed in the digital domain, and many digital synthesizers self-consciously set out to emulate their earlier analog counterparts, often seeking to capture the ‘warmth’ and unpredictability of those instruments.

Frequency Modulation, or *FM*, synthesis was introduced in 1967–1968 by John Chowning at Stanford University, and essentially consisted of using one sinusoid to modulate another, thereby creating more complex spectra. This technique formed the basic constituent of many of the most successful digital synthesizers, such as the Yamaha DX7, which was released in 1983. The modulation technique has applications beyond FM synthesis itself, and often forms the basis of creative experimentation in synthesis. For example, Phase Modulation exploits the fact that the rate at which the phase of a wave changes is the same as the frequency expressed in cycles per second. New variants of the FM algorithm have been introduced, including time-varying parameters and feedback. It remains a fertile area for digital synthesis.

*Wavetable* synthesis is another approach that builds upon the analytical capabilities of FTs. By storing complex wave forms in tables they may be reused at any point in the process of synthesis and in any order. The FT analyses the components of a wave form by identifying the sinusoids and their relative strengths within frequency bands in a single cycle. The resulting wave form is stored and made available for subsequent looking up. The various types of table include: *transient* wavetables that capture the evolution of harmonics in a sound over time; *formant* wavetables that reflect harmonic changes; *wave sequence* wavetables that have changeable harmonic content; and *speech* wavetables.