Curtis Roads

with John Strawn, Curtis Abbott, John Gordon, and Philip Greenspun

The Computer Music Tutorial

The MIT Press
Cambridge, Massachusetts
London, England

13 Spectrum Analysis

Applications of Spectrum Analysis

Spectrum Plots

- Static Spectrum Plots
- Power Spectrum
- Time-varying Spectrum Plots

Models behind Spectrum Analysis Methods

Spectrum and Timbre

Spectrum Analysis: Background

- Mechanical Spectrum Analysis
- Computer-based Spectrum Analysis
- *Heterodyne Filter Analysis*
- *The Saga of the Phase Vocoder*

The Short-time Fourier Spectrum

- Windowing the Input Signal
- Operation of the STFT
- Overlap-add Resynthesis from Analysis Data
- *Limits of Overlap-add Resynthesis*

Why Overlapping Windows?
Oscillator Bank Resynthesis
Analysis Frequencies
Time/Frequency Uncertainty
Periodicity Implies Infinitude
Time/Frequency Tradeoffs
Frequencies in between Analysis Bins
Significance of Clutter
Alternative Resynthesis Techniques

The Sonogram Representation
Sonogram Parameters

The Phase Vocoder
Phase Vocoder Parameters
Frame Size
Window Type
FFT Size and Zero-padding
Hop Size
Typical Parameter Values
Window Closing
Tracking Phase Vocoder
Operation of the TPV
Peak Tracking
Editing Analysis Envelopes
Cross-synthesis with the Phase Vocoder
Computational Cost of the Phase Vocoder
Accuracy of Resynthesis
Problem Sounds
Analysis of Inharmonic and Noisy Sounds
Deterministic plus Stochastic Techniques

Constant *Q* Filter Bank Analysis
Constant *Q* versus Traditional Fourier Analysis
Implementation of Constant *Q* Analysis

Analysis by Wavelets
Operation of Wavelet Analysis
Wavelet Display
Wavelet Resynthesis
Sound Transformation with Wavelets
Comb Wavelet Separation of Noise from Harmonic Spectrum
Comparison of Wavelet Analysis with Fourier Methods

Signal Analysis with the Wigner Distribution
Interpreting Wigner Distribution Plots
Limits of the Wigner Distribution

Non-Fourier Sound Analysis
Critiques of Fourier Spectrum Analysis
Autoregression Spectrum Analysis
Autoregressive Moving Average Analysis
Source and Parameter Analysis
Parameter Estimation
Analysis by Other Functions
Walsh Functions
Prony's Method

Auditory Models
Cochleagrams
Correlograms

Signal-understanding Systems
Pattern Recognition
Control Structure and Strategy
Examples of Signal-understanding Systems

Conclusion

Will not the creative musician be a more powerful master if he is also informed in regard to the pure science of the methods and materials of his art? Will he not be able to mix tone colors with greater skill if he understands the nature of the ingredients and the effects which they produce? (D. C. Miller 1916)

Just as an image can be described as a mixture of colors (frequencies in the visible part of the electromagnetic spectrum), a sound object can be described as a blend of elementary acoustic vibrations. One way of dissecting sound is to consider the contribution of various components, each corresponding to a certain rate of variation in air pressure. Gauging the balance among these components is called *spectrum analysis*.

A working definition of spectrum is "a measure of the distribution of signal energy as a function of frequency." Such a definition may seem straightforward, but no more general and precise definition of spectrum exists. This is because different analysis techniques measure properties that they each call "spectrum" with more-or-less diverging results. Except for isolated test cases, the practice of spectrum analysis is not an exact science (see Marple 1987 for a thorough discussion). The results are typically an approximation of the actual spectrum, so spectrum analysis is perhaps more precisely called *spectrum estimation*.

Spectrum analysis is evolving rapidly. The survey in this chapter, though broad, cannot account for every possible approach. Given the technical nature of the subject, our major concern in this chapter is to render sometimes abstruse concepts in terms of musical practice. Appendix A treats Fourier analysis in more detail and is a complement to this chapter.

Applications of Spectrum Analysis

Spectrum plots reveal the microstructure of vocal, instrumental, and synthetic sounds (Moorer, Grey, and Snell 1977; Moorer, Grey, and Strawn 1978; Piszczalski 1979a, b; Dolson 1983, 1986; Stautner 1983; Strawn 1985a, b). Thus they are essential tools for the acoustician and psychoacoustician (Risset and Wessel 1982).

Musicologists are increasingly turning to sonograms and other sound analysis techniques in order to study music performance and the structure of electronic music (Cogan 1984). This extends to automatic transcription of music—from sound to score—either in common music notation or a graphic form (Moorer 1975; Piszczalski and Galler 1977; Chafe et al. 1982; Foster et al. 1982; Haus 1983; Schloss 1985).

Real-time spectrum analysis is one type of "ear" for interactive music systems. Spectrum analysis reveals the characteristic frequency energy of

instrumental and vocal tones, thus helping to identify timbres and separate multiple sources playing at once (Maher 1990). As chapter 12 shows, the results of spectrum analysis are often valuable in pitch and rhythm recognition.

But musicians want not only to analyze sounds; they want to modify the analysis data and resynthesize variants of the original sounds. More and more sound transformation techniques begin with an analysis stage, including time compression and expansion, frequency-shifting, convolution (filtering and reverberation effects), and many types of cross-synthesis—hybrids between two sounds. Techniques based on spectrum analysis allow continuous transformation between "natural" and "synthetic" tones in resynthesis of analyzed tones (Gordon and Grey 1977; Risset 1985a, b; Serra 1989). For more on analysis/resynthesis see chapters 4 and 5.

Spectrum Plots

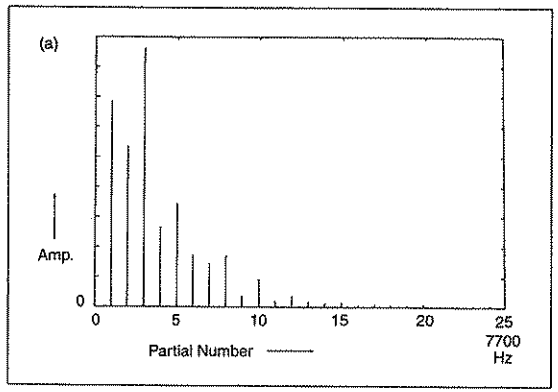
Many strategies exist to measure and plot spectra. This section looks at strategies falling into two basic categories: *static* (like a snapshot of a spectrum), and *time-varying* (like a motion-picture film of a spectrum over time).

Static Spectrum Plots

Static plots capture a still image of sound. These sonic snapshots project a two-dimensional image of amplitude versus frequency. The analysis measures the average energy in each frequency region over the time period of the analyzed segment. This time period or window can vary from a brief instant to several seconds or longer. (Later we discuss the tradeoffs of various window lengths.)

One type of static plot is a *discrete* or *line spectrum*, where a vertical line represents each frequency component. For a mostly harmonic tone, the clearest analysis is *pitch-synchronous*. This type of analysis measures the amplitude of the harmonics of a tone whose pitch can be determined beforehand. Figure 13.1a shows the line spectrum of the steady state part of a trumpet tone, measured using a pitch-synchronous technique. Notice that at the instant this spectrum was measured, the third harmonic is higher in amplitude than the fundamental.

Figure 13.1b shows another trumpet spectrum plotted on a logarithmic (dB) amplitude scale. Such a scale compresses the plot into a narrower vertical band. By tracing the outline of the peaks one can see the overall formant shape.



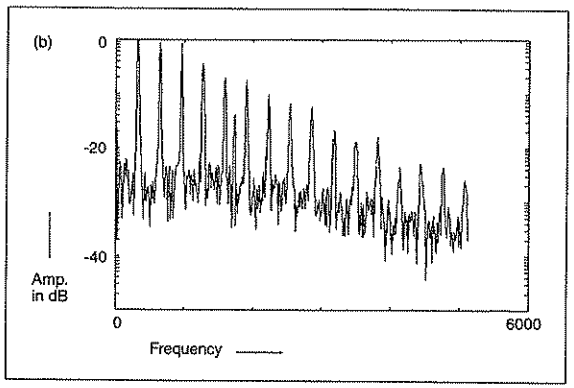


Figure 13.1 Static spectrum plots. (*a*) Line spectrum amplitude-versus-frequency plot of the sustained portion of a trumpet tone. Each line represents the strength of a harmonic of the fundamental frequency of 309 Hz. Linear amplitude scale. (*b*) Spectrum of trumpet tone in (*a*) plotted on a logarithmic (dB) scale, which compresses the plot into a narrower vertical band. (*c*) Spectrum plot in a continuous form, showing the outline of the formant peaks for a vocal sound "ah." Linear amplitude scale. (Plots courtesy of A. Piccialli, Department of Physics, University of Naples.)

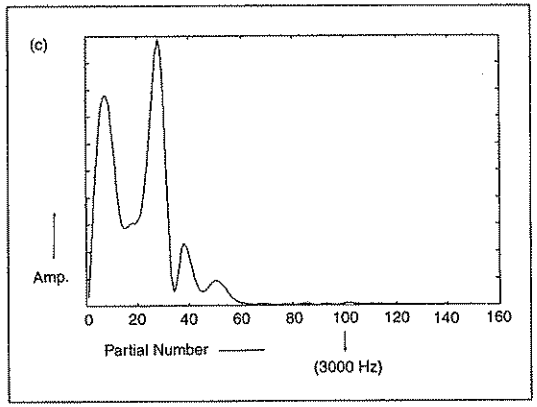


Figure 13.1 (cont.)

Figure 13.1c plots the spectrum of a vocal sound "ah" in a *continuous* form, where the discrete points measured by the analyzer have been filled in by graphical interpolation. Individual sinusoidal components are hidden, but the overall shape of the spectrum is clear.

Each type of static spectrum plot has its advantages, depending on the signal being analyzed and the goal of the analysis.

Power Spectrum

From the amplitude spectrum one can derive the *power spectrum*. Physicists define *power* as the square of the amplitude of a signal. Thus, power spectrum is the square of the amplitude spectrum. Displays of spectrum sometimes show power, rather than amplitude, because this correlates better with human perception. Yet another measure is the *power spectrum density* or PSD, which applies to continuous spectra like noise. A simple definition of the PSD is that it is the power spectrum within a specified bandwidth (Tempelaars 1977).

Time-varying Spectrum Plots

Details in the spectrum of even a single instrument tone are constantly changing, so static, timeless plots can represent only a portion of an evolving sound form. A time-varying spectrum depicts the changing blend

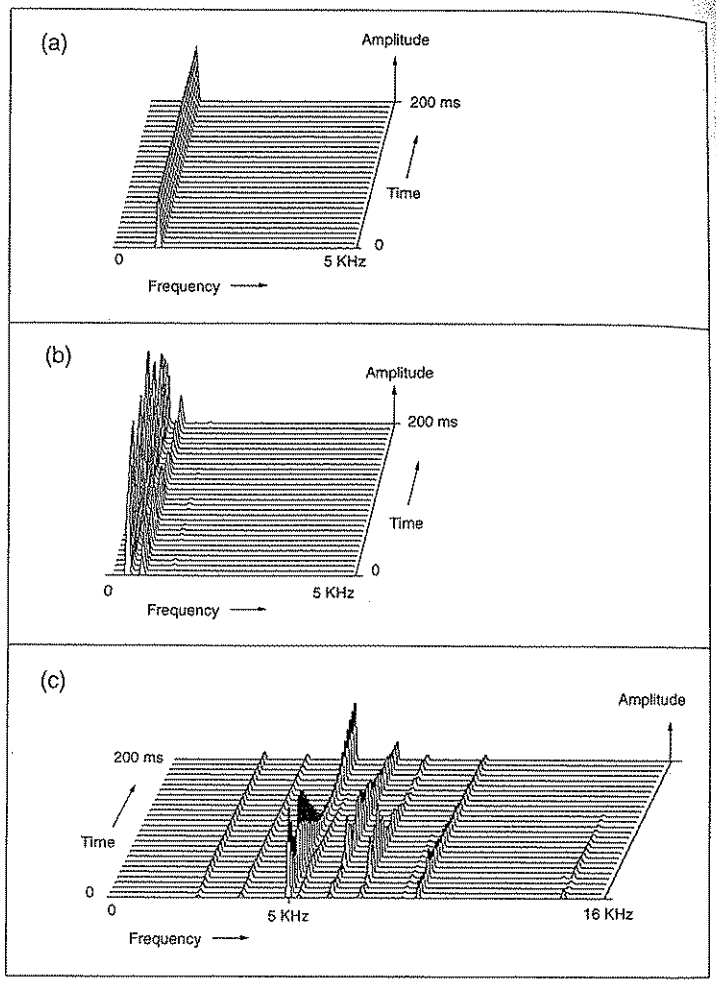


Figure 13.2 Time-varying spectra plotted on a linear amplitude scale. Time moves from front to back. (*a*) Sine wave at 1 KHz. (*b*) Flute playing fluttertongue at pitch E4. (*c*) Triangle, hit once.

of frequencies over the duration of an event. It can be plotted as a three-dimensional graph of spectrum versus time (figure 13.2). These plots essentially line up a series of static plots, one after the other.

Figure 13.3 shows two more display formats for time-varying spectrum analysis. Figure 13.3a is a still photograph from a *waterfall display*—a spectrum plot in which the time axis is moving in real time. The term waterfall display comes from the fact that this type of plot shows waves of rising and falling frequency energy in a fluidlike depiction. Figure 13.3b depicts a vocal melody.

Another way to display a time-varying spectrum is to plot a *sonogram* or *spectrogram*—a common tool in speech analysis, where it was originally called *visible speech* (Potter 1946). A sonogram shows the frequency versus time content of a signal, where frequency is plotted vertically, time is plotted horizontally, and the amplitudes of the frequencies in the spectrum are plotted in terms of the darkness of the trace. That is, intense frequency components are plotted darkly, while soft frequency components are plotted lightly (figure 13.4). We discuss the sonogram representation in more detail later.

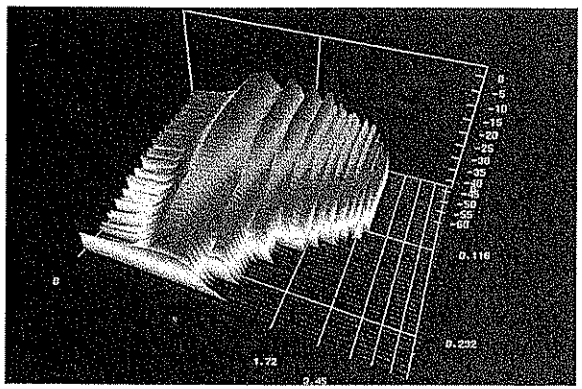
Models Behind Spectrum Analysis Methods

There does not seem to be any general or optimal paradigm to either analyze or synthesize any type of sound. One has to scrutinize the sound—quasi-periodic, sum of inharmonic components, noisy, quickly or slowly evolving—and also investigate which features of the sound are relevant to the ear. (Jean-Claude Risset 1991)

No single method of spectrum estimation is ideal for all musical applications. Fourier analysis—the most prevalent approach—is actually a family of different techniques that are still evolving. A variety of non-Fourier methods continue to be developed, as we show later.

Every sound analysis technique should be viewed as fitting the input data to an assumed model. Methods based on Fourier analysis model the input sound as a sum of harmonically related sinusoids—which it may or may not be. Other techniques model the input signal as an excitation signal filtered by resonances, as a sum of exponentially damped sinusoids or square waves, as a combination of inharmonically related sinusoids, as a set of formant peaks with added noise, or as a set of equations that represent certain behavior of a traditional instrument. Innumerable other models are conceivable. As we see in detail later, variations in performance among the different methods can often be attributed to how well the assumed model

(a)



(b)

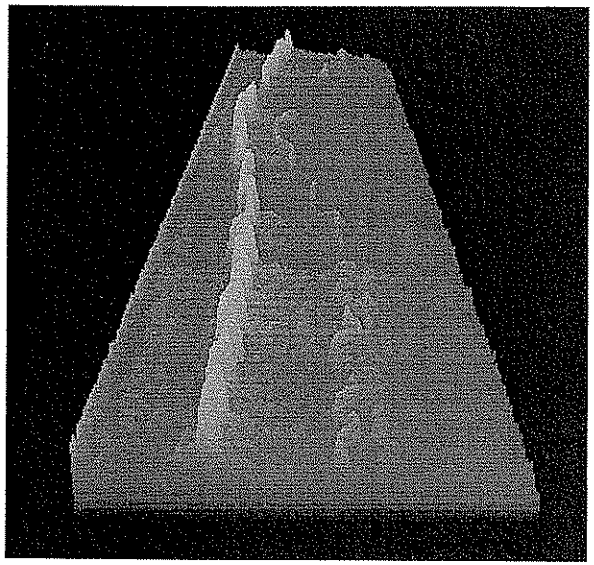


Figure 13.3 Still images from real-time waterfall displays. (*a*) Synthetic trumpet tone. Time comes from the back toward the front, with the most recent time at the front. The frequency scale is logarithmic, going from left to right. The fundamental frequency is approximately 1 KHz. Amplitude is plotted vertically on a logarithmic dB scale. (*b*) Vocal melody. Time is coming toward the reader, with the most recent time at the front. Low frequencies are at left. (Images courtesy of A. Peevers, Center for New Music and Art Technologies, University of California, Berkeley.)

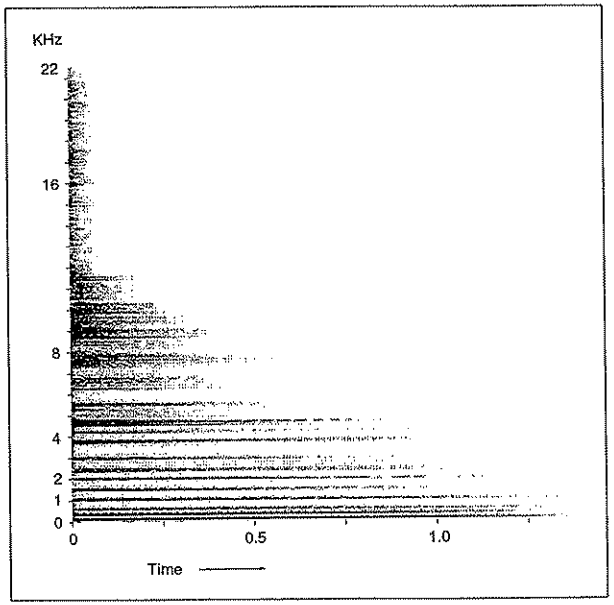


Figure 13.4 Sonogram plot of a struck tam-tam. The vertical axis is frequency, and the horizontal axis is time. This sonogram uses 1024 points of input data, and a Hamming window. The plot has a frequency resolution of 43 Hz and a time resolution of 1 ms. The analysis bandwidth of 0 to 22 KHz, and the measured dynamic range is -10 to -44.5 dB, plotted on a linear amplitude scale.

matches the process being analyzed. Hence it is important to choose the appropriate analysis method for a particular musical application.

Spectrum and Timbre

The term "timbre" is a catchall for a range of phenomena. Like the vague terms "sonority" and "Klangideal" (Apel 1972) it may some day be superseded by a more precise vocabulary of sound qualities. The classification of musical timbre is an ancient science. Early Chinese civilization developed sophisticated written descriptions of timbre, including a taxonomy of instrumental sources (metal, stone, clay, skin, silk threads, wood, gourd, and bamboo), and elaborate accounts of the different "touches" (attack forms, pulls, and vibratos) involved in playing the silk strings of the classical *chhin*

instrument (Needham, Ling, and Girdwood-Robinson 1962). Indeed, a main playing technique of the *chhin* is the production of different timbres at the same pitch.

Spectrum and timbre are related concepts, but they are not equivalent. Spectrum is a physical property that can be characterized as a distribution of energy as a function of frequency. How to measure this energy precisely is another question! Psychoacoustics uses the term "timbre" to denote perceptual mechanisms that classify sound into families. By this definition, timbre is at least as concerned with perception as it is with sound signals. It is certainly easiest to discuss timbre in the realm of traditional instrument and vocal tones, where almost all past research has focused. Only a few attempts have been made to classify the universe of sound outside this category, the most heroic being the studies of Pierre Schaeffer (1977; see also Schaeffer, Reibel, and Ferreyra 1967).

A common timbre groups tones played by an instrument at different pitches, loudnesses, and durations. No matter what notes it plays, for example, we can always tell when a piano is playing. Human perception separates each instrument's tones from other instrument tones played with the same pitch, loudness, and duration. No one has much trouble separating a marimba from a violin tone of the same pitch, loudness, and duration. Of course a single instrument may also emit many timbres, as in the range of sonorities obtained from saxophones blowed at different intensities.

Numerous factors inform timbre perception. These include the amplitude envelope of a sound (especially the attack shape), undulations due to vibrato and tremolo, formant structures, perceived loudness, duration, and the *time-varying spectral envelope* (frequency content over time) (Schaeffer 1977; Risset 1991; McAdams and Bregman 1979; McAdams 1987; Gordon and Grey 1977; Grey 1975, 1978; Barrière 1991; see also chapter 23).

In identifying the timbre of an instrumental source, the attack portion of a tone is more important perceptually than the steady state (sustained) portion (Luce 1963; Grey 1975). Traditional instrument families such as reeds, brass, strings, and percussion each have characteristic attack "signatures" that are extremely important in recognizing tones made by them.

Amplitude and duration have an influence on the perception of timbre. For example, the proportions of the frequencies in the spectrum of a flute tone at 60 dB may be the equivalent to those in a tone amplified to 120 dB, but we hear the latter only as a loud blast. Similarly, a toneburst that lasts 30 ms may have the same periodic waveshape as a tone that lasts 30 seconds, but listeners may find it difficult to say whether they represent the same source.

The point is that spectrum is not the only clue to perceived timbre. By examining the time-domain waveform carefully, one can glean much about the timbre of a sound without subjecting it to a detailed spectrum analysis (Strawn 1985b).

Spectrum Analysis: Background

In the eighteenth century, scientists and musicians were well aware that many musical sounds were characterized by harmonic vibrations around a fundamental tone, but they had no technology for analyzing these harmonics in a systematic way. Sir Isaac Newton coined the term "spectrum" in 1781 to describe the bands of color showing the different frequencies passing through a glass prism.

In 1822 the French engineer Jean-Baptiste Joseph, Baron de Fourier (1768–1830) published his landmark thesis *Analytical Theory of Heat*. In this treatise he developed the theory that complicated vibrations could be analyzed as a sum of many simultaneous simple signals. In particular, Fourier proved that any periodic function could be represented as an infinite summation of sine and cosine terms. Due to the integer ratio relationship between the sinusoidal frequencies in Fourier analysis, this became known as *harmonic analysis*. (For a brief history of Fourier analysis, see appendix A.) In 1843, Georg Ohm (1789–1854) of the Polytechnic Institute of Nürnberg was the first to apply Fourier's theory to acoustical signals (Miller 1935). Later, the German scientist H. L. F. Helmholtz (1821–1894) surmised that instrumental timbre is largely determined by the harmonic Fourier series of the steady state portion of instrumental tones (Helmholtz 1863). Helmholtz developed a method of harmonic analysis based on mechanical-acoustic resonators.

Translating Helmholtz's term *Klangfarbe* ("sound color"), the British physicist John Tyndall coined the term *clang-tint* to describe timbre as "an admixture of two or more tones" and carried out imaginative experiments in order to visualize sound signals, such as "singing flames" and "singing water jets" (Tyndall 1875).

Mechanical Spectrum Analysis

Manually operated mechanical waveform analyzers were developed in the late nineteenth and early twentieth centuries (Miller 1916; see also appendix A). Backhaus (1932) developed an analysis system for a single harmonic at

a time. This consisted of a carbon microphone connected to the input of a tunable bandpass filter. The output of the filter was routed to an amplifier, whose output was in turn connected to a pen and drum recorder. Backhaus tuned the filter to the frequency of the harmonic of interest and commanded an instrumentalist to play a note. As the musician played, Backhaus cranked a drum while a pen traced the output of the filter for that frequency on a roll of paper. The resulting trace was taken to represent the behavior of a single harmonic. Meyer and Buchmann (1931) developed a similar system.

Advances in the design of oscilloscopes in the 1940s generated a wave of new research. Scientists photographed waveforms from the oscilloscope screen and then manually traced their outline into mechanical Fourier analyzers.

A theoretical leap forward was described in Norbert Wiener's classic paper on *generalized harmonic analysis* (Wiener 1930), which shifted the emphasis of Fourier analysis from harmonic components to a continuous spectrum. Among other results, Wiener showed, by analogy to white light, that white noise was composed of all frequencies in equal amounts. Blackman and Tukey (1958) described a practical implementation of Wiener's approach using sampled data. After the advent of computers in the early 1950s, the Blackman-Tukey approach was the most popular spectrum analysis method until the introduction of the *fast Fourier transform* (FFT) in 1965, sometimes credited to Cooley and Tukey (1965). (See Singleton 1967, Rabiner and Gold 1975, and appendix A for more on the history of the FFT.)

Most precomputer analyses, such as those of Miller (1916) and Hall (1937) averaged out the time-varying characteristics of instrumental tone. As in the research of Helmholtz, these studies presumed that the steady state spectrum (sustained or "held" part of a note) played a dominant role in timbre perception. As mentioned earlier, it is now recognized that the first half-second of the attack portion of a tone is more important perceptually than the steady state portion to the identification of an instrumental note.

Dennis Gabor's pioneering contributions to sound analysis (1946, 1947) had a delayed impact, but are now viewed as seminal, particularly because he presented a method for analysis of time-varying signals. In Gabor's theories, sound can be analyzed simultaneously in the time and frequency domain into units he called *quanta*—now called *grains, wavelets,* or *windows,* depending on the analysis system being used. See chapter 5 for more on grains. Wavelet analysis and windows are discussed later in this chapter.

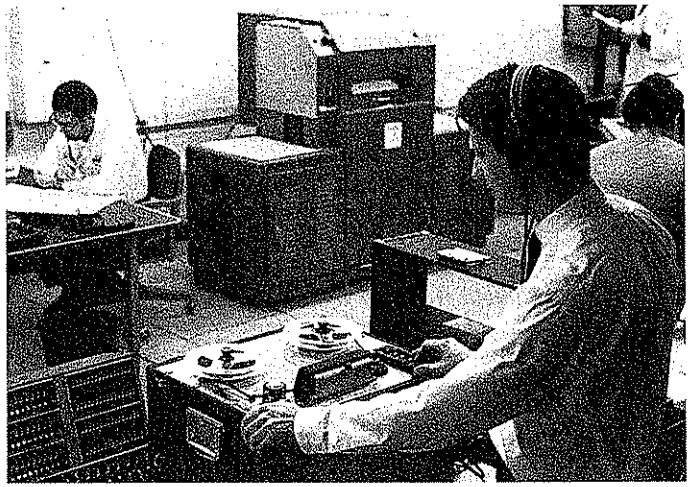


Figure 13.5 James Beauchamp performing sound analysis experiments at the University of Illinois, ca. 1966.

Computer-based Spectrum Analysis

Early experiments in computer analysis of musical instrument tones required heroic efforts. Analog-to-digital converters were rare, computers were scarce, theory was immature, and analysis programs had to be cobbled from scratch on punched paper cards (figure 13.5). Against these obstacles, computer-based analysis and synthesis developed in the 1960s yielded more detailed results than did analog models. At Bell Telephone Laboratories, Max Mathews and Jean-Claude Risset analyzed brass instruments using a *pitch-synchronous* analysis program (Mathews, Miller, and David 1961; Risset 1966; Risset and Mathews 1969). Pitch-synchronous analysis breaks the input waveform into *pseudoperiodic segments.* It estimates the pitch of each pseudoperiodic segment. The size of the *analysis segment* is adjusted relative to the estimated pitch period. The harmonic Fourier spectrum is then calculated on the analysis segment as though the sound were periodic; as though the pitch is quasi-constant throughout the analysis segment. This program generated time-varying amplitude functions for each harmonic of a given fundamental. Luce's (1963) doctoral research at the Massachusetts Institute of Technology implemented another pitch-synchronous approach to analysis/resynthesis of instrumental tones.

Several years later, Peter Zinovieff and his colleagues at EMS, London, developed a hybrid (analog-digital) real-time Fourier analyzer/resynthesizer for musical sound (Grogorno 1984).

Heterodyne Filter Analysis

The next step in computer analysis of musical tones involved *heterodyne filters* (Freedman 1965, 1967; Beauchamp 1969, 1975; Moorer 1973, 1975). The heterodyne filter approach is good for resolving harmonics (or quasi-harmonics) of a given fundamental frequency. This implies that the fundamental frequency is estimated in a prior stage of analysis. The heterodyne filter multiplies an input waveform by a sine and a cosine wave at harmonic frequencies and then sums the results over a short time period to obtain amplitude and phase data.

Figure 13.6a shows the operation of the heterodyne method. The input signal is multiplied by an analysis sine wave. In figure 13.6a, the frequency of the two signals exactly match, so the energy is completely positive, indicating strong energy at the analysis frequency. In 13.6b the two frequencies are not the same, so we obtain a waveform that is basically symmetrical

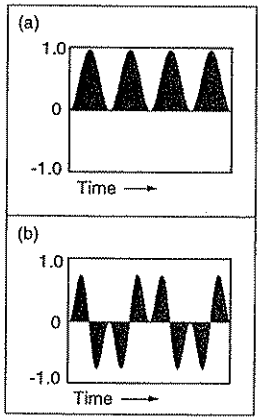


Figure 13.6 Heterodyne filter analysis. (*a*) Product of an input signal (a 100 Hz sine wave) with an analysis signal (a 100 Hz sine wave). The result is entirely positive, indicating strong energy at 100 Hz. (*b*) Product of an input signal (a 200 Hz sine wave) with an analysis signal (a 100 Hz sine wave). The result is scattered positive and negative energy, indicating no strong energy at 100 Hz in the input signal.

about the amplitude axis. When the heterodyne filter sums this waveform over a short time period it basically cancels itself out.

After a period of experimentation in the 1970s, the limits of the heterodyne method became well known. Moorer showed that the heterodyne filter approach is confused by fast attack times (less than 50 ms) and pitch changes (e.g., glissando, portamento, vibrato) greater than 2 percent (about a quarter tone). Although Beauchamp (1981) implemented a *tracking* version of the heterodyne filter that could follow changing frequency trajectories (similar in spirit to the tracking phase vocoder discussed later), the heterodyne approach has been supplanted by other methods.

The Saga of the Phase Vocoder

One of the most popular techniques for analysis/resynthesis of spectra is the *phase vocoder* (PV). Flanagan and Golden of Bell Telephone Laboratories developed the first PV program in 1966. It was originally intended to be a coding method for reducing the bandwidth of speech signals. Far from compressing audio data, however, the PV causes a data explosion! That is, the raw analysis data are much greater than the original signal data.

The PV is computationally intensive. Early implementations required so much computing time that the PV was not applied in practical applications for many years. Working at the Massachusetts Institute of Technology, Portnoff (1976, 1978) developed a relatively efficient PV, proving that it could be implemented using the FFT. He experimented with sound transformations of speech such as time compression and expansion. This led to Moorer's landmark paper on the application of the PV in computer music (Moorer 1978).

During the 1970s and 1980s, computer-based spectrum analysis yielded significant insights into the microstructure of instrumental and vocal tones (Moorer, Grey, and Snell 1977; Moorer, Grey, and Strawn 1978; Piszczalski 1979a, b; Dolson 1983; Stautner 1983; Strawn 1985b). In the 1990s spectrum analysis has evolved from an esoteric technical specialty to a familiar tool in the musician's studio—for analysis, transcription, and sound transformation. The next sections discuss various forms of spectrum analysis, including the short-time Fourier transform and the phase vocoder. Then we present extensions of Fourier analysis, including constant Q filter banks and the wavelet transform. Although Fourier methods dominate spectrum analysis, other methods have gained ground in recent years. So we also survey these "non-Fourier techniques" later in this chapter. (For a technical overview of spectrum analysis written in an anecdotal style, see Robinson 1982.)

The Short-time Fourier Spectrum

The *Fourier transform* (FT) is a mathematical procedure that maps any continuous-time (analog) waveform to a corresponding infinite Fourier series summation of elementary sinusoidal waves, each at a specific amplitude and phase. In other words, the FT converts its input signals into a corresponding spectrum representation. To adapt Fourier analysis to the practical world of sampled, finite-duration, time-varying signals, researchers molded the FT into the *short-time Fourier transform* or STFT (Schroeder and Atal 1962; Flanagan 1972; Allen and Rabiner 1977; Schafer and Rabiner 1973b).

Windowing the Input Signal

As a preparation for spectrum analysis, the STFT imposes a sequence of *time windows* upon the input signal (figure 13.7). That is, it breaks the input signal into "short-time" (i.e., brief) segments bounded in time by a window function. A window is nothing more than a specific type of envelope designed for spectrum analysis. The duration of the window is usually in the range of 1 ms to 1 second, and the segments sometimes overlap. By analyzing the spectrum of each windowed segment separately, one obtains a sequence of measurements that constitute a time-varying spectrum.

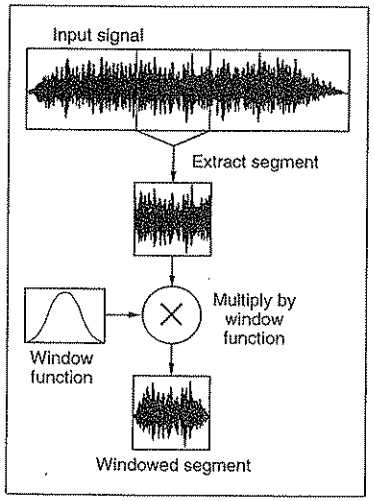


Figure 13.7 Windowing an input signal.

The windowing process is the source of the adjective "short-time" in "short-time Fourier transform." Unfortunately, windowing has the side effect of distorting the spectrum measurement. This is because the spectrum analyzer is measuring not purely the input signal, but rather, the product of the input signal and the window. The spectrum that results is the convolution of the spectra of the input and the window signals. We see the implications of this later. (Chapter 10 explains convolution. Appendix A discusses windowing in more detail.)

Operation of the STFT

After windowing, the STFT applies the *discrete Fourier transform* (DFT) to each windowed segment. Here all we need say about the DFT is that it is a type of Fourier transform algorithm that can handle discrete-time or sampled signals. Its output is a discrete-frequency spectrum, that is, a measure of energy at a set of specific equally spaced frequencies. (See appendix A for an introduction to the DFT.)

The fast Fourier transform or FFT, mentioned earlier in the historical section, is simply an efficient implementation of the DFT. Thus most practical implementations of the STFT apply the FFT algorithm to each windowed segment. Figure 13.8 diagrams the STFT.

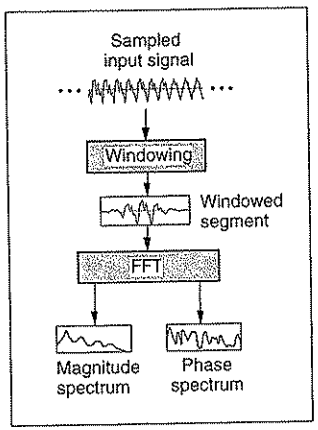


Figure 13.8 Overview of the short-time Fourier transform (STFT).

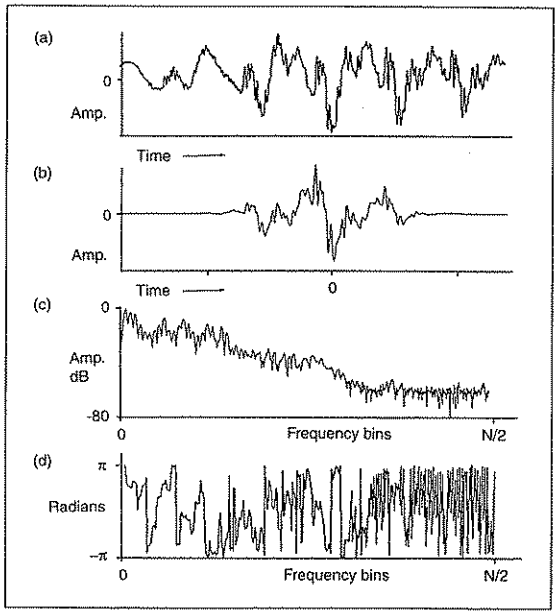


Figure 13.9 STFT signals. (*a*) Input waveform. (*b*) Windowed segment. (*c*) Magnitude spectrum plotted over the range 0 to −80 dB. (*d*) Phase spectrum plotted over the range $-\pi$ to π. (After Serra 1989.)

Each block of data generated by the FFT is called a *frame*, by analogy to the successive frames of a film. Each frame contains two things: (1) a *magnitude spectrum* that depicts the amplitude of every analyzed frequency component, and (2) a *phase spectrum* that shows the initial phase value for every frequency component. All of the plots in figures 13.1–13.4 are magnitude spectrum plots.

We could visualize each of these two spectra as histograms with a vertical line for each frequency component along the abcissa. The vertical line represents amplitude in the case of the magnitude spectrum, and starting phase (between $-\pi$ and π) in the case of the phase spectrum (figure 13.9). The magnitude spectrum is relatively easy to read. When the phase spectrum is "normalized" to the range of $-\pi$ and π it is called the *wrapped phase* representation. For many signals, it appears to the eye like a random function. An *unwrapped phase* projection may be more meaningful visually. Appendix A explains the concepts of wrapped and unwrapped phase.

To summarize, the application of the STFT to a stream of input samples results in a series of frames that make up a time-varying spectrum.

Overlap-add Resynthesis from Analysis Data

To resynthesize the original time-domain signal, the STFT can reconstruct each windowed waveform segment from its spectrum components by applying the *inverse discrete Fourier transform* (IDFT) to each frame. The IDFT takes each magnitude and phase component and generates a corresponding time-domain signal with the same envelope as the analysis window.

Then by overlapping and adding these resynthesized windows, typically at their −3 dB points (see chapter 5 for an explanation of this term), one obtains a signal that is a close approximation of the original. Figure 13.10 depicts the overlap-add process in schematic form. (Appendix A explains both the IDFT and overlap-add resynthesis in more detail.)

We use the qualification "close approximation" as a way of comparing practical implementations of the STFT with mathematical theory. In theory, resynthesis from the STFT is an identity operation, replicating the input sample by sample (Portnoff 1976). If it were an identity operation in practice, we could copy signals through an STFT any number of times with no generation loss. However, even good implementations of the STFT lose a small amount of information. This loss may not be audible after one pass through the STFT.

Limits of Overlap-add Resynthesis

Resynthesis with the plain *overlap-add* (OA) method is of limited use from the standpoint of musical transformation. This is because the OA process is designed for the case where the windows sum perfectly to a constant. As Allen and Rabiner (1977) showed, any additive or multiplicative transformations that disturb the perfect summation criterion at the final stage of the OA cause side effects that will probably be audible. Time expansion by stretching the distance between windows, for example, may introduce comb filter or reverberation effects, depending on the number of frequency channels or *bins* used in the analysis. Using speech or singing as a source, many transformations result in robotic, ringing voices of limited use.

One way to lessen these unwanted artifacts is to stipulate a great deal of overlap among successive windows in the analysis stage, as explained in the next section. The method of "improved overlap-add" resynthesis is another

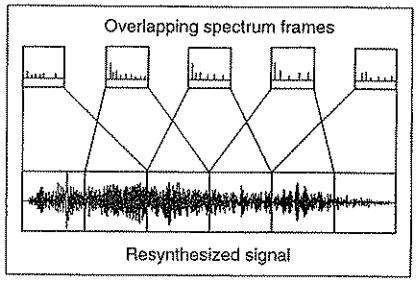


Figure 13.10 Overlap-add resynthesis. The gray areas indicate overlapping spectrum frames. Note: for visual clarity, we show only five frames. In practice it is typical to use more than 100 frames per second of analyzed sound.

strategy for overcoming these problems (George and Smith 1992; see also the description later in this chapter).

Why Overlapping Windows?

The motivation behind the overlapping analysis windows in the STFT can be confusing. After all, theory says that we can analyze a segment of any length and exactly resynthesize the segment from the analysis data. Evidently we can analyze in one pass Stravinsky's *Le sacre du printemps* using a 30-minute-long window, and reconstruct the entire piece from this analysis. This being the case, why bother to break the analysis into small, overlapping segments?

The reasons are several. The analysis of a monaural sound sampled at 44.1 KHz and lasting 30 minutes would result in a spectrum of over 79 million points. A visual inspection of this enormous spectrum would eventually tell us all the frequencies that occurred over a 30-minute duration, but would not tell us when precisely they occurred; this temporal information is embedded deep in the mathematical combination of the magnitude and phase spectra, hidden to the eye. Thus the first thing that windowing helps with is the visualization of the spectrum. By limiting the analysis to short segments (less than a tenth of a second, typically), each analysis plots fewer points, and we know more accurately when these frequencies occurred.

A second reason for using short-time envelopes is to conserve memory. Consider an analysis of a 30-minute chunk of sound swallowed in one gulp. Assuming 16-bit samples, one would need a computer with at least 79 million 16-bit words of random-access memory (RAM) just to hold the input while the computer calculates the FFT. By breaking the input into bite-sized

segments it becomes easy to calculate the FFT on each small segment at a time.

A third reason for short-time windows is that one obtains results quicker. For *Le sacre du printemps* one would have to wait up to 30 minutes just to read in the input signal, plus however long it takes to calculate an FFT on a 79 million point input signal. Windowing the input lets one obtain initial results after a few milliseconds of the input has been read in, opening up applications for real-time spectrum analysis.

These three reasons explain the segmentation, but why overlap the windows? As explained earlier, smooth bell-shaped windows minimize the distortion that occurs in windowing. And of course, bell-shaped windows must overlap somewhat in order to capture the signal without gaps. But even greater overlap is often desirable, more than is dictated by the perfect summation criterion. Why is this? Increasing the overlap factor is equivalent to *oversampling the spectrum*, and this protects against the aliasing artifacts that can occur in transformations such as time-stretching and cross-synthesis. An overlap factor of eight or more is recommended when the goal is transforming the input signal.

Later we discuss basic criteria for selecting a window and setting its length. Appendix A goes into the subject of windowing in more detail. Next we present an alternative to the overlap-add resynthesis model.

Oscillator Bank Resynthesis

Sinusoidal additive resynthesis (SAR) (or *oscillator bank* resynthesis) differs from the overlap-add approach. Rather than summing the sine waves at each frame—as in the OA resynthesis model—SAR applies a bank of oscillators driven by amplitude and frequency envelopes that span across frame boundaries (figure 13.11). This implies that the analysis data must be converted beforehand into such envelopes. Fortunately, the conversion from analysis data (magnitude and phase) to synthesis data (amplitude and frequency) takes little calculation time.

The advantage of the SAR model is that envelopes are much more robust under musical transformation than the raw spectrum frames. Within broad limits, one can stretch, shrink, rescale, or shift the envelopes without worrying about artifacts in the resynthesis process; the perfect summation criterion of the OA model can be ignored. A disadvantage of SAR is that it is not as efficient computationally as OA methods.

A tracking phase vocoder can be seen as a SAR method since it also constructs frequency envelopes for additive sine wave synthesis. We discuss this approach in more detail in the section on the phase vocoder later.

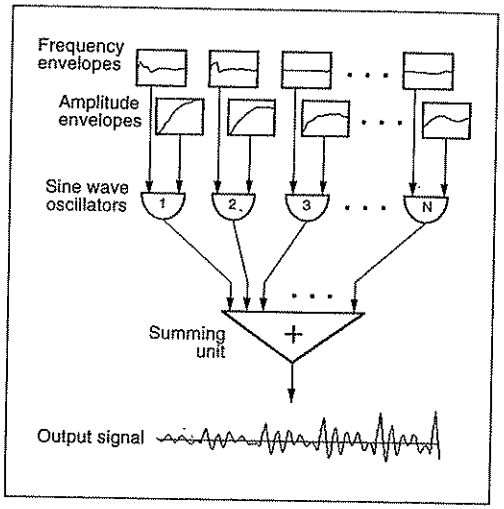


Figure 13.11 Oscillator bank resynthesis. The analysis data have been converted into a set of continuous amplitude and frequency envelopes. The number of oscillators needed for the resynthesis grows and shrinks depending on the complexity of the sound.

Analysis Frequencies

One can think of the STFT as applying a bank of filters at equally spaced frequency intervals to the windowed input signal. The frequencies are spaced at integer multiples (i.e., harmonics) of

$$\frac{\textit{sampling frequency}}{N}$$

where N is the size of the analyzed segment. (As we will later see, the value of N is usually greater than the actual number of sound samples analyzed; for now we assume they are the same length.) Thus if the sampling frequency is 50 KHz and the window length is 1000 samples, the analysis frequencies are spaced at intervals 50,000/1000 = 50 Hz apart, starting at 0 Hz. The analyzer at 0 Hz measures the *direct current* or *DC offset* of the signal, a constant that can shift the entire signal above or below the center point of zero amplitude.

Audio signals are bandlimited to half the sampling rate (25 KHz in this case), and so we care about only half of the analysis bins. (As mentioned earlier, a bin is a frequency channel in the parlance of signal processing.) The effective frequency resolution of an STFT is thus $N/2$ bins spread equally across the audio bandwidth, starting at 0 Hz and ending at the Nyquist frequency. In our example, the number of usable audio frequency bins is 500, spaced 50 Hz apart.

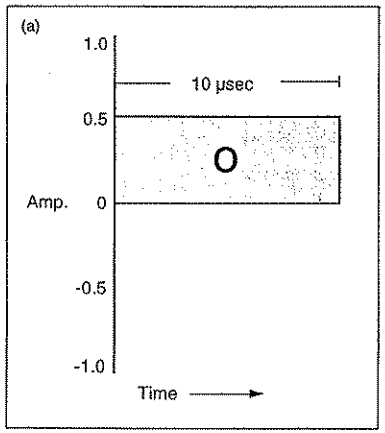
Time/Frequency Uncertainty

All windowed spectrum analyses are hampered by a fundamental *uncertainty principle* between time and frequency resolution, first recognized by quantum physicists such as Werner Heisenberg in the early part of the twentieth century (Robinson 1982). This principle means that if we want high resolution in the time domain (i.e., we want to know precisely when an event occurs), we sacrifice frequency resolution. In other words, we can tell that an event occurred at a precise time but we cannot say exactly what frequencies it contained. Conversely, if we want high resolution in the frequency domain (i.e., we want to know the precise frequency of a component), we sacrifice time resolution. That is, we can pinpoint frequency content only over a long time interval. It is important to grasp this relationship in order to interpret the results of Fourier analysis.

Periodicity Implies Infinitude

Fourier analysis starts from the abstract premise that if a signal contains only one frequency, then that signal must be a sinusoid that is infinite in duration. Purity of frequency—absolute periodicity—implies infinitude. As soon as one limits the duration of this sine wave, the only way that Fourier analysis can account for this is to consider the signal as a sum of many infinite-length sinusoids that just happen to cancel each other out in such a way as to result in a limited-duration sine wave! While this characterization of frequency neatens the mathematics, it does not jibe with our most basic experiences with sound. As Gabor (1946) pointed out, if the concept of frequency is used only to refer to infinitely long signals, then the concept of changing frequency is impossible!

Still, we can understand one aspect of the abstract Fourier representation by a thought experiment. Using a sound editor, imagine that we zoom into the limit of the time domain of a digital system. In the shortest "instant" of time we see an individual sample point (the shaded rectangle marked **O** in figure 13.12a). We know exactly when this sample occurs, so we have high temporal resolution. But we cannot see what waveform it may be a part of; it could be a part of a wave at any frequency within the Nyquist range



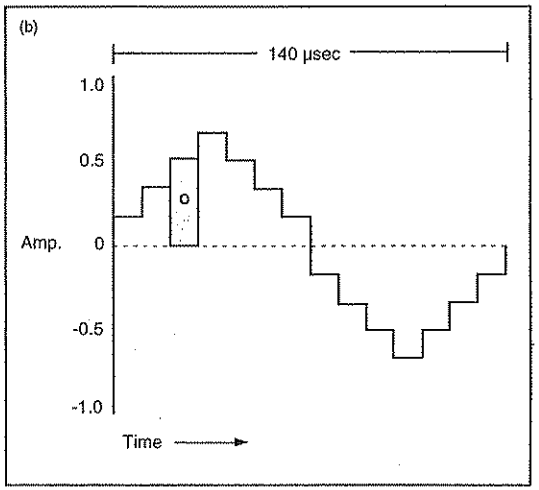


Figure 13.12 Frequency uncertainty at a small timescale. (*a*) The shaded box O represents a zoom into a precise sample period in a system with good time resolution (a 10 μsec sample period implies a sampling rate of 100 KHz). No frequency information is revealed at this time resolution; we lose any sense of what larger waveform this might be a part of. Thus a frequency estimation from one or a few samples is bound to be only a rough guess. (*b*) Zooming out to a timescale of 140 μsec gives a much better picture of the overall waveform and the local frequency period.

of the system. As we zoom out (figure 13.12b), we have more samples to analyze, and so the more sure we can be about what possible frequencies they might represent. But since Fourier analysis calculates the spectrum for the entire analyzed segment at a time, spectrum displays of long segments leave uncertainty as to when a particular frequency occurred. Once again, frequency precision comes at the expense of temporal imprecision.

Filter design provides more clues. Recall from chapter 10 that the number of delay stages influences the sharpness of a filter. In order to isolate a very narrow band, such as a single frequency component, we need extremely sharp edges in the filter response. This implies that one needs to look back into the distant past of the signal in order to extract a pure frequency. Another way of saying this is that such a filter has a long *impulse response*. (See chapter 10 for an explanation of impulse response.)

Time/Frequency Tradeoffs

The FFT divides up the audible frequency space into $N/2$ frequency bins, where N is the length in samples of the analysis window. Hence there is a tradeoff between the number of frequency bins and the length of the analysis window (figure 13.13). For example, if N is 512 samples, then the number of frequencies that can be analyzed is limited to 256. Assuming a sampling rate of 44.1 KHz, we obtain 256 bins equally spaced over the bandwidth 0 Hz to the Nyquist frequency 22.05 KHz. Increasing the sampling rate only widens the measurable bandwidth. It does not increase the frequency resolution of the analysis.

Table 13.1 demonstrates the balance between time and frequency resolution. If we want high time accuracy (say 1 ms or about 44 samples at a 44.1 KHz sampling rate), we must be satisfied with only 44/2 or 22 frequency bins. Dividing up the audio bandwidth from 0 to 22.05 KHz by 22 frequency bins, we obtain 22,050/22 or about 1000 Hz of frequency resolution. That is, if we want to know exactly when events occur on the scale of 1 ms, then our frequency resolution is limited to the gross scale of 1000-Hz-wide frequency bands. By sacrificing more time resolution, and widening the analysis interval to 30 ms, one can spot frequencies within a 33 Hz bandwidth. For high resolution in frequency (1 Hz), one must stretch the time interval to 1 second (44,100 samples)!

Because of this limitation in windowed STFT analysis, researchers are examining hybrids of time-domain and frequency domain analysis, *multiresolution analysis*, or non-Fourier methods to try to resolve both dimensions at high resolution. Later sections discuss these approaches.

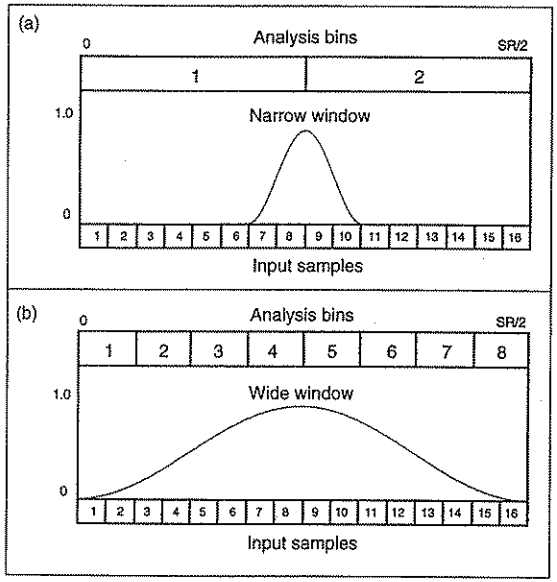


Figure 13.13 Relationship of window size to the number of frequency analysis bins. (*a*) A narrow window of four samples can resolve only two frequencies. (*b*) a wider window of sixteen samples divides the spectrum into eight bins.

Table 13.1 Time versus frequency resolution in windowed spectrum analysis

Length of time window (in ms)	Frequency resolution (analysis bandwidth) (in Hz)
1	1000
2	500
3	330
10	100
20	50
30	33
100	10
200	5
300	3
1000 (1 sec)	1
2000	0.5
3000	0.3

Frequencies in between Analysis Bins

The STFT knows only about a discrete set of frequencies spaced at equal intervals across the audio bandwidth. The spacing of these frequencies depends on the length of the analysis window. This length is effectively the "fundamental period" of the analysis. Such a model works well for sounds that are harmonic or quasi-harmonic where the harmonics align closely with the bins of the analysis. But what happens to frequencies that fall in between the equally spaced analysis bins of the STFT? This is the case for inharmonic sounds such as gongs or noisy sounds such as snare drums.

Let us call the frequency to analyzed *f*. When *f* coincides with the center of an analysis channel, all its energy is concentrated in that channel, and so it is accurately measured. When *f* is close to but not precisely coincident with the center, energy is scattered into all other analysis channels, but with a concentration remaining close to *f*. Figure 13.14 shows three snapshots of a frequency sweeping from 2 to 3 Hz, which can be generalized to other frequency ranges. The leakage spilling into all frequency bins from components in between bins is a well-known source of unreliability in the spectrum estimates produced by the STFT. When more than one component is in between bins, *beating effects* (periodic cancellation and reinforcement) may occur in both the frequency and amplitude traces. The result is that the analysis shows fluctuating energy in frequency components that are not physically present in the input signal.

Significance of Clutter

If the signal is resynthesized directly from the analysis data, the extra frequency components and beating effects pose no problem; they are benign artifacts of the STFT analysis that are resolved in resynthesis. Beating effects are merely the way that the STFT represents in the frequency domain a time-varying spectrum. In the resynthesis, some components add constructively and some add destructively (canceling each other out), so that the resynthesized result is a close approximation of the original. (Again, in theory it is an identity, but small errors creep into practical applications.)

Beating and other anomalies are harmless when the signal is directly resynthesized, but they obscure attempts to inspect the spectrum visually, or transform it. For this reason, the artifacts of analysis are called *clutter*. Dolson (1983) and Strawn (1985a) assay the significance of clutter in analysis of musical instrument tones. Gerzon (1991) presents a theory of "superresolving" spectrum analyzers that offer to improve resolution in both time

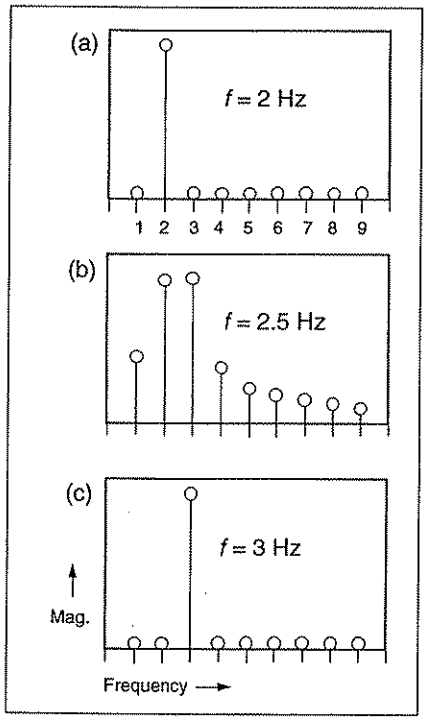


Figure 13.14 Three STFT "snapshots" of a sound changing frequency from 2 to 3 Hz. The STFT in this case has analysis bins spaced at 1 Hz intervals. When the input frequency is 2.5 Hz, it falls in between the equally spaced frequency bins of the analyzer, and the energy is spread across the entire spectrum. (After Hutchins 1984.)

and frequency, at the expense of increased clutter, which, Gerzon argues, has some perceptual significance.

Alternative Resynthesis Techniques

Two alternatives to the standard techniques of resynthesis merit a brief word here. The first is an adaptive method that offers improved resolution and more robust transformations; the second offers greatly increased resynthesis speed.

Analysis-by-synthesis/overlap-add (ABS/OLA) refines the STFT with overlap-add resynthesis by incorporating an error analysis procedure

(George and Smith 1992). This procedure compares the original signal with the resynthesized signal. When the error is above a given threshold, the procedure adjusts the amplitudes, frequencies, and phases in the analysis frame to approximate the original more closely. This adaptive process may occur repeatedly until the signal is more-or-less precisely reconstructed. As a result the ABS/OLA method can handle attack transients, inharmonic spectra, and effects such as vibrato with greater accuracy than the plain overlap-add method. It also permits more robust musical transformations. As we will see later, a method called the tracking phase vocoder has similar benefits.

The "FFT^{-1}" method is a special hybrid of overlap-add and oscillator bank resynthesis optimized for real-time operation. The method is so named because the resynthesis is carried out by the inverse FFT, which is sometimes abbreviated FFT^{-1}. It starts from previously calculated oscillator bank resynthesis data. It then converts these data by an efficient algorithm into an overlap-add model with data reduction and optimization steps that greatly speed up resynthesis. See Rodet and Depalle (1992) and French patent 900935 for details.

The Sonogram Representation

A *sonogram, sonograph*, or *spectrogram* is a well-known spectrum display technique in speech research, having been used for decades to analyze utterances. A sonogram shows an overview of the spectrum of several seconds of sound. This enables the viewer to see general features such as the onset of notes or phonemes, formant peaks, and major transitions. A trained viewer can read a speech sonogram. See Cogan (1984) for an example of using sonograms in the analysis of music. The sonogram representation has also been employed as an interface for spectrum editing (Eckel 1990; see chapter 16).

The original sonogram was Backhaus's (1932) system, described earlier in the background section on spectrum analysis; see also Koenig et al. (1946). In the 1950s the Kay Sonograph was a standard device for making sonograms. It consisted of a number of narrow bandpass analog filters and a recording system that printed dark bars on a roll of paper. The bars grew thicker in proportion to the energy output from each filter. Today sonograms are generally implemented with the STFT.

Figure 13.4 showed a sonogram, representing a sound signal as a two-dimensional display of time versus "frequency + amplitude". The vertical

dimension depicts frequency (higher frequencies are higher up in the diagram) and shades of gray indicate the amplitude, with dark shades indicating greater intensity.

Sonogram Parameters

The parameters of the modern sonogram are the same as those of the STFT, except for certain display parameters. Adjustments to these parameters make a great difference in the output image:

1. Range of amplitudes and the type of scale used, whether linear or logarithmic.
2. Range of frequencies and the type of scale used, whether linear or logarithmic.
3. Time advance of the analysis window, also called *hop size* (in samples) or *window overlap factor*. This determines the time distance between successive columns in the output display. (We discuss this parameter in more detail in the section on the phase vocoder.)
4. Number of samples to analyze and the size of the FFT analysis window; the resolution of time and frequency depend on these parameters.
5. Number of frequency channels to display, which determines the number of rows in the graphical output and is related to the range and scale of the frequency domain; this cannot exceed the resolution imposed by the window size.
6. Window type—see the discussion in the section on the phase vocoder and in appendix A.

Parameter 4 includes two parameters; the FFT window size is usually greater than the actual number of sound samples analyzed, the difference being padded with zero-valued samples. (See the section on phase vocoder analysis parameters.) These parameters have the most dramatic effect on the display. A short window results in a vertically oriented display, indicating

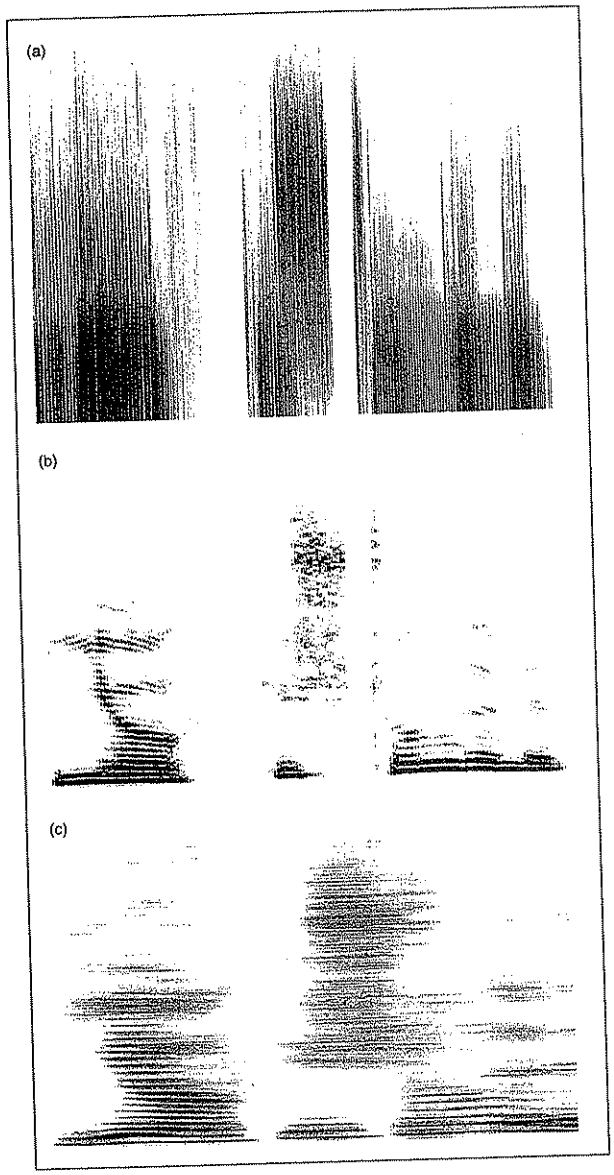


Figure 13.15 Time-versus-frequency tradeoffs in sonogram analysis and display. All displays show speech sound sampled at 44.1 KHz. (*a*) Analysis window is 32 samples long, time resolution is 0.725 ms, and frequency resolution is 1378 Hz. (*b*) Analysis window is 1024 samples long, time resolution is 23.22 ms, frequency resolution is 43.07 Hz. (*c*) Analysis window is 8192 samples long, time resolution is 185.8 ms, frequency resolution is 5.383 Hz. (Sonograms provided by Gerhard Eckel using his SpecDraw program.)

the precise onset time of events but blurring the frequency reading (figure 13.15a). A medium length window resolves both time and frequency features fairly well, indicating the presence of formant frequencies (figure 13.15b). A long window generates a horizontally oriented display, as individual frequency bands come into clear view, but their position in time is smeared along the horizontal axis (figure 13.15c).

The speech sonogram has to be modified to handle the more stringent demands of music. Musical sonograms tend to be longer than speech sonograms, including sections or entire pieces. The dynamic range of music is much wider than speech. Also, as Lundén and Ungvary (1991) point out, speech sonograms are oriented toward an accurate physical representation of the spectrum, whereas musicians are more interested in a perceptual view that is in accord with what we can hear. The cochleagram display, explained later, may be a more accurate perceptual picture. For a critical analysis of traditional sonograms from the standpoint of accuracy, see Loughlin, Atlas, and Pitton (1992).

The Phase Vocoder

The phase vocoder has emerged as an increasingly popular sound analysis tool, being packaged in several widely distributed software packages. (Gordon and Strawn 1985 and Moore 1990 contain annotated code for practical phase vocoders.) One can view the PV as passing a windowed input signal through a bank of parallel bandpass filters spread out at equal intervals across the audio bandwidth. These filters measure the amplitude and phase of a sinusoidal signal in each frequency band. Through a subsequent operation (explained in appendix A), these values can be converted into two envelopes: one for the amplitude of the sine, and one for the frequency of the sine. This corresponds to the case of oscillator bank resynthesis previously discussed. Various implementations of the PV offer tools for modifying these envelopes, allowing musical transformations of analyzed sounds.

In theory, analysis and resynthesis via the phase vocoder is a sample-by sample clone (Portnoff 1976). In practice, there is usually a slight loss of information, which may not be audible in one analysis/resynthesis pass. In any case, a musician's use of the PV inevitably involves modification of the analysis data before resynthesis. For what the composer seeks in the output is not a clone of the input, but a musical transformation that maintains a sense of the identity of the source. That is, if the input signal is a

spoken voice, one usually wants it to sound like a spoken voice even after being transformed. One can also use the PV for radical distortions that destroy the identity of the input signal, but more efficient distortion algorithms are easily found, such as the modulations discussed in chapter 6.

See chapter 5 for a description of the first vocoder. For more on the PV, including descriptions of practical implementations, see Portnoff 1976, 1978, 1980; Holtzman 1978; Moorer 1978; Moore 1990; Dolson 1983, 1986; Gordon and Strawn 1985; Strawn 1985b; Strawn 1987; Serra 1989; Depalle and Poirot 1991; Erbe 1992; Walker and Fitz 1992; Beauchamp 1993.

Phase Vocoder Parameters

The quality of a given PV analysis depends on the parameter settings chosen by the user. These settings must be adjusted according to the nature of the sounds being analyzed and the type of results that are expected. The main parameters of the PV are the folllowing:

1. Frame size—number of input samples to be analyzed at a time
2. Window type—selection of a window shape from among the standard types (see the discussion later)
3. FFT size—the actual number of samples fed to the FFT algorithm; usually the nearest power of two that is double the frame size, where the unit of FFT size is referred to by *points*, as in a "1024-point FFT" (equivalent to "1024-sample FFT")
4. Hop size or overlap factor—time advance from one frame to the next

Now we discuss each parameter in turn. Then in the following section we give rules of thumb for setting these parameters.

Frame Size

The frame size (in samples) is important for two reasons. The first is that the frame size determines one aspect of the tradeoff in time/frequency resolution. The larger the frame size, the greater the number of frequency bins, but the lower the time resolution, and vice versa. If we are trying to analyze sounds in the lower octaves with great frequency accuracy, large frame sizes are unavoidable. Since the FFT computes the average spectrum content within a frame, the onset time of any spectrum changes within a frame is lost when the spectrum is plotted or transformed. (If the signal is simply resynthesized, the temporal information is restored.) For high-frequency

sounds, small frames are adequate, which are also more accurate in time resolution.

The second reason frame size is important is that large FFTs are slower to calculate than small FFTs. Following the rule of thumb that the calculation time for an FFT is proportional to $N \times \log_2(N)$, where N is the length of the input signal (Rabiner and Gold 1975), it takes more than a thousand times as long to calculate a 32,768-point FFT, for example, than a 64-point FFT. The latency of a long FFT may be too onerous in a real-time system.

Window Type

Most PVs give the option of using one of a family of standard window types, including Hamming, Hanning (or Hann; see Marple 1987), truncated Gaussian, Blackman-Harris, and Kaiser (Harris 1978; Nuttall 1981; see also appendix A). All are quasi-bell-shaped, and all work reasonably well for general musical analysis/resynthesis. For analyses where precision is important (such as creating a systematic catalog of spectra for instrumental tones) the choice of analysis window may be more critical. This is because windowing introduces distortion, and each type of window "bends" the analysis plots in a slightly different way. For more on windows see appendix A.

FFT Size and Zero-padding

The choice of FFT size depends on the transformation one plans to apply to the input sound. A safe figure for cross-synthesis is the nearest power of two that is double the frame size. For example, a frame size of 128 samples would mandate an FFT size of 256. The other 128 samples in the FFT are set to zero—a process called *zero-padding* (see appendix A).

Hop Size

The hop size is the number of samples that the analyzer jumps along the input waveform each time it takes a new spectrum measurement (figure 13.16). The shorter the hop size, the more successive windows overlap. Thus some PVs specify this parameter as an overlap factor that describes how many analysis windows cover each other. Regardless of how it is specified, the hop size is usually a fraction of the frame size. A certain amount of overlap (e.g., eight times) is necessary to ensure an accurate resynthesis. Even more overlap may improve accuracy when the analysis data are going to be transformed, but the computational cost is proportionally greater.

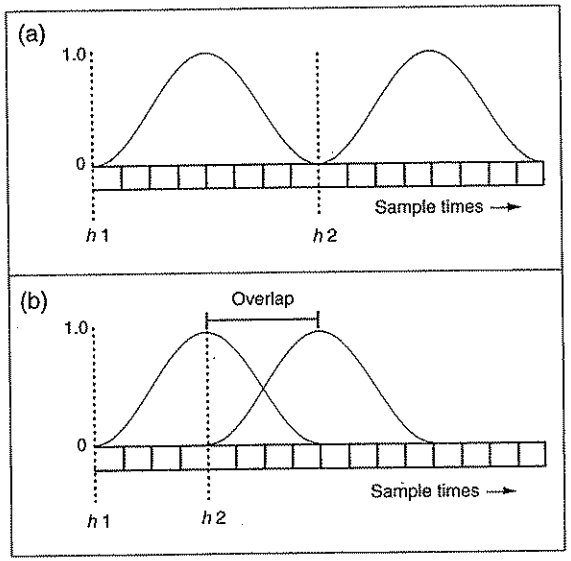


Figure 13.16 Varying hop size for analysis windows that are eight samples long. $h1$ and $h2$ are the starting times for each window. (*a*) Nonoverlapping windows when hop size = window size. (*b*) Overlapping windows when hop size is less than window size. In this case the hop size is four samples.

Typical Parameter Values

No parameter settings of the PV are ideal for all sounds. But when the parameters are set within a certain range, a variety of traditional instrumental sounds can be analyzed and resynthesized with reasonable fidelity. Here are some rules of thumb for PV parameter settings that may serve as a starting point for more "tuned" analyses:

1. Frame size—large enough to capture four periods of the lowest frequency of interest (Depalle and Poirot 1991). This is particularly important if the sound is time-stretched; too small a frame size means that individual pitch bursts are moved apart, changing the pitch, although formants are preserved.
2. Window type—any standard type except rectangular.
3. FFT size—double the frame size, in samples.

4. Hop size—if the analysis data are going to be time-distorted, the recommended hop size is an eighth of the frame size, in samples (i.e., eight times overlap). In general, the minimum technical criterion is that all windows add to a constant, that is, all data are equally weighted. This typically implies an overlap at the −3 dB point of the particular window type chosen, from which can be derived the hop size.

Window Closing

Once is not enough. (S. J. Marple 1987).

Any given setting of the window size results in an analysis biased toward harmonics of the period defined by that window size. Frequency components that fall outside the frequency bins associated with a given window size will be estimated incorrectly. Thus some spectrum analysis procedures run the same signal through the analyzer repeatedly with different settings for the window size. A procedure that starts from high time and low frequency resolution and works progressively to low time and high frequency resolution is called *window closing* (Marple 1987).

Some STFT analyzers try to estimate the pitch of the signal in order to determine the optimal window size. As mentioned earlier, pitch-synchronous analysis works well if the sound to be analyzed has a basically harmonic structure.

Tracking Phase Vocoder

Many current implementations of the PV are called tracking phase vocoders (TPVs) because they follow or track the most prominent peaks in the spectrum over time (Dolson 1983; McAulay and Quatieri 1986; Quatieri and McAulay 1986; Serra 1989; Maher and Beauchamp 1990; Walker and Fitz 1992). Unlike the ordinary phase vocoder, in which the resynthesis frequencies are limited to harmonics of the analysis window, the TPV follows changes in frequencies. The result of peak tracking is a set of amplitude and frequency envelopes that drive a bank of sinusoidal oscillators in the resynthesis stage.

The tracking process follows only the most prominent frequency components. For these components, the result is a more accurate analysis than that done with an equally spaced bank of filters (the traditional STFT implementation). The other benefit is that the tracking process creates frequency and amplitude envelopes for these components, which make them more robust under transformation than overlap-add frames. A disadvantage is

that the quality of the analysis may depend more heavily on proper parameter settings than in the regular STFT.

Operation of the TPV

A TPV carries out the following steps:

1. Compute the STFT using the frame size, window type, FFT size, and hop size specified by the user
2. Derive the squared magnitude spectrum in dB
3. Find the bin numbers of the peaks in the spectrum
4. Calculate the magnitude and phase of each frequency peak
5. Assign each peak to a *frequency track* by matching the peaks of the previous frame with those of the current frame (see the description of peak tracking later)
6. Apply any desired modifications to the analysis parameters
7. If additive resynthesis is requested, generate a sine wave for each frequency track and sum all sine wave components to create an output signal; the instantaneous amplitude, phase, and frequency of each sinusoidal component is calculated by interpolating values from frame to frame (or use the alternative resynthesis methods described earlier)

Peak Tracking

The tracking phase vocoder follows the most prominent frequency trajectories in the spectrum. Like other aspects of sound analysis, the precise method of peak tracking should vary depending on the sound. The tracking algorithm works best when it is tuned to the type of sound being analyzed—speech, harmonic spectrum, smooth inharmonic spectrum, noisy, etc. This section briefly explains more about the tracking process as a guide to setting the analysis parameters.

The first stage in peak tracking is peak identification. A simple control that sets the *minimum peak height* focuses the identification process on the most significant landmarks in the spectrum (figure 13.17a). The rest of the algorithm tries to apply a set of *frequency guides* that advance in time (figure 13.17b). The guides are hypotheses only; later the algorithm will decide which guides are confirmed frequency tracks. The algorithm continues the guides by finding the peak closest in frequency to its current value. The alternatives are as follows:

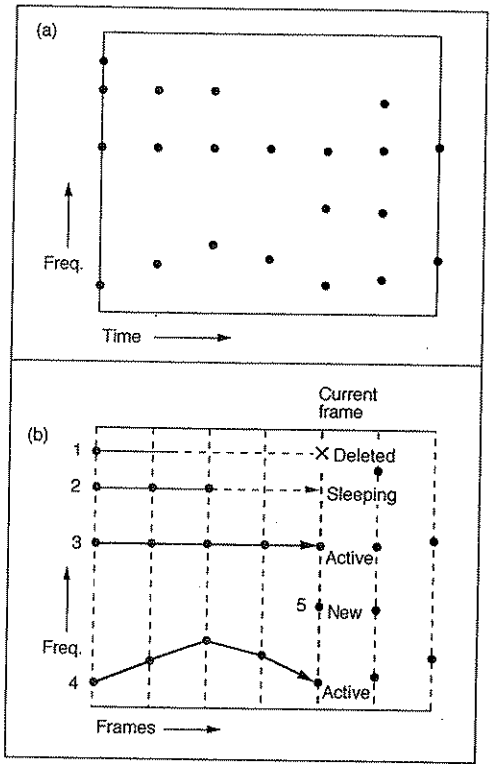


Figure 13.17 Peak identification and tracking. (*a*) Isolation of a set of spectrum peaks. (*b*) Fitting frequency guides to peaks. Guide 1 at the top did not wake up after three frames, so it is deleted. Guide 2 is still sleeping. Guides 3 and 4 are active. Guide 5 starts from a new peak.

- If it finds a match, the guide continues.
- If a guide cannot be continued during a frame it is considered to be "sleeping."
- If the guide does not wake up after a certain number of frames—which may be specified by the user—then it is deleted. It may be possible to switch on *guide hysteresis*, which continues tracking a guide that falls slightly below the specified *amplitude range*. Hysteresis alleviates the audible problem of "switching" guides that repeatedly fade slightly, are cut to zero by the peak tracker, and fade in again (Walker and Fitz 1992). With hysteresis the guide is synthesized at its actual value, which may be less than the amplitude range, instead of with zero amplitude.
- If there is a conflict between guides, the closest guide wins, and the "loser" looks for another peak within the *maximum peak deviation*, a frequency band specified by the user.
- If there are peaks that are not accounted for by current guides, then a new guide begins.

The process of windowing may compromise the accuracy of the tracking, particularly in rapidly moving waveforms such as attack transients. Processing sounds with a sharp attack in time-reversed order helps the tracking algorithm (Serra 1989). This gives the partial trackers a chance to lock onto their stable frequency trajectories before encountering the chaos of the attack, resulting in less distortion. The data can be reversed back to its normal order before resynthesis.

The next section discusses step 6, modification of the TPV analysis envelopes.

Editing Analysis Envelopes

Changing the parameters of the resynthesis creates transformations in the sound. By modifying the hop size in the playback, for example, one can implement time expansion and compression effects. Due to the underlying sinusoidal model, however, when a time expansion is performed on a complex attack or a noisy sound, individual sine waves emerge and the noisy quality is lost. The spectral modeling synthesis of Serra (1989), described later, addresses this problem.

To create sophisticated musical transformations one must edit the analysis data generated by the TPV—the frequency, amplitude, and phase curves (Moorer 1978; Dolson 1983; Gordon and Strawn 1985). This laborious process of transmutation is greatly aided by automatic data reduction procedures and graphical editor programs. (See chapter 4 for information on data reduction in additive synthesis, and see the section on spectrum editors in chapter 16.) Table 4.1 in chapter 4 lists some of the musical effects made possible by modification of PV spectrum data.

Cross-synthesis with the Phase Vocoder

Another possibility for sound transformation with less editing is *cross-synthesis*. Cross-synthesis is not one technique; it takes a number of forms. The

most common form uses the magnitude functions from one spectrum to control the magnitude functions of another. That is, the strength of each frequency component in sound *A* scales the strength of the corresponding frequency component in sound *B*. This is implemented by mutiplying each point in spectrum *A* by each corresponding point in spectrum *B*. Another term for this type of cross-synthesis is *filtering by convolution* (see chapter 10 for more on convolution). Musically, cross-synthesis is most effective when one of the sounds being filtered has a broad bandwidth, like a noise source. By using a phase vocoder with two inputs, cross-synthesis is basically automatic (Depalle and Poirot 1991). Another type of cross-synthesis uses the magnitude functions from one sound with the phase functions of another sound to create a hybrid sound effect (Boyer and Kronland-Martinet 1989).

Musical guidelines for cross-synthesis with the PV are much the same as for cross-synthesis by fast convolution. See chapter 10 for more on these guidelines.

Computational Cost of the Phase Vocoder

The phase vocoder is one of the more computationally expensive operations available to musicians, particularly when tracking is carried out. The tracking phase vocoder soaks up large quantities of computer power even though the inner core is implemented using the efficient FFT algorithm. The PV also generates a large amount of analysis data; in some cases this is many times greater than the size of the sample data being analyzed. A panoply of techniques may be applied to reduce computation and conserve space. For example, the envelopes generated by the TPV may be computed at a lower sampling rate. This may not compromise the audio quality since these control functions tend to change more slowly than the audio sampling rate. Before resynthesis they can be restored to the original sampling rate by interpolation. Other *data reduction* methods can also be applied; see the discussion of data reduction in chapter 4.

Accuracy of Resynthesis

The accuracy of all Fourier-based resynthesis is limited by the resolution of the analysis procedures. Small distortions introduced by numerical round-off, windowing, peak-tracking, undersampling of envelope functions, and other aspects of the analysis introduce errors. In a well-implemented PV, when the analysis parameters are properly adjusted by a skilled engineer and no modifications are made to the analysis data, the error is negligible perceptually.

The tracking PV, on the other hand, interprets the raw analysis data in constructing its tracks. It discards all information that does not contribute to a track. This sifting may leave out significant portions of sound energy, particularly noisy, transient energy. This can be demonstrated by subtracting the resynthesized version from the original signal to yield a *residual signal* (Strawn 1987a; Gish 1978, 1992; Serra 1989). One can consider this residual or difference to be analysis/resynthesis error. It is common to refer to the resynthesized, quasi-harmonic portion as the "clean" part of the signal and the error or noise component as the "dirty" part of the signal. For many sounds (i.e., those with fast transients such as in cymbal crashes), the errors are quite audible. That is, the "clean" signal sounds unnaturally "sanitized" or sinusoidal, and the "dirty" signal, when heard separately, contains the missing grit. (See the section on analysis of inharmonic and noisy sounds in a moment.)

For efficiency, some PVs have the option of discarding phase information, saving only the amplitude and frequency data. This results in a data reduction and corresponding savings in computation time, but also degrades the accuracy of the resynthesis. Without proper phase data, a resynthesized waveform, for example, does not resemble the original, although it has the same basic frequency content (Serra 1989). In certain steady state sounds, a rearrangement of phases may not be audible. But for high-fidelity reproduction of transients and quasi-steady-state tones, phase data help reassemble short-lived and changing components in their proper order and are therefore valuable.

Problem Sounds

The PV handles harmonic, static, or smoothly changing tones best. Transformations such as timescale expansion and compression on these sounds result in natural sounding effects. Certain sounds, however, are inherently difficult to modify with PV techniques. These include noisy sounds such as raspy or breathy voices, motors, any sound that is rapidly changing on a timescale of a few milliseconds, and sounds that contain room noise. Transformations on these types of sounds may result in echoes, flutter, unwanted resonances, and undesirable colored reverberation effects. These are mainly due to phase distortions that occur when the analysis data is transformed.

Analysis of Inharmonic and Noisy Sounds

Demonstrations prove that tracking phase vocoders can analyze and resynthesize many inharmonic sounds, including bird songs (Serra and Smith

1990), and tuned percussion tones (gongs, marimba, xylophone, etc.). But since the TPV is based on Fourier analysis, it must translate noisy and inharmonic signals into combinations of periodic sinusoidal functions. Particularly for noisy signals, this can be a costly process from a storage and computational standpoint. To synthesize a simple noise band, for example, requires an ever-changing blend of dozens of sine waves. Storing the control functions for these sines fills up a great deal of space. In some TPVs this amounts to more than ten times as many bytes as the original sound samples. Resynthesizing the sines demands a tremendous amount of computation. Moreover, since the transformations allowed by the TPV are based on a sinusoidal model, operations on noisy sounds often result in clusters of sinusoids that have lost their noisy quality.

Deterministic Plus Stochastic Techniques

To handle such signals better, the TPV has been extended to make it more effective in musical applications. Serra (1989) added filtered noise to the inharmonic sinusoidal model in *spectral modeling synthesis* (SMS). (See also chapter 4 and Serra and Smith 1990.) As figure 13.18 shows, SMS reduces the analysis data into a *deterministic* component (prominent narrowband components of the original sound) and a *stochastic* component. The deterministic component tracks the most prominent frequencies in the spectrum. SMS resynthesizes these tracked frequencies with sine waves. The tracking follows only the most prominent frequency components, discarding other energy in the signal. Thus SMS also analyzes the *residue* (or *residual*), which is the difference between the deterministic component and the original spectrum. This is used to synthesize the stochastic component of the signal. The residual is analyzed and approximated by a collection of simplified spectrum envelopes. One can think of the resynthesis as passing white noise through filters controlled by these envelopes. In the implementation, however, SMS uses sine waves with random phase values, which is equivalent to the filtered noise interpretation.

The SMS representation, using spectrum envelopes and sine waves, rather than a filter bank, makes it easier to modify the stochastic part in order to transform the sound. Graphical operations on envelopes are intuitive to a musician, whereas changing filter coefficients leads to technical complications. A problem with SMS is that the perceptual link between the deterministic and stochastic parts is delicate; editing the two parts separately may lead to a loss of perceived fusion between them.

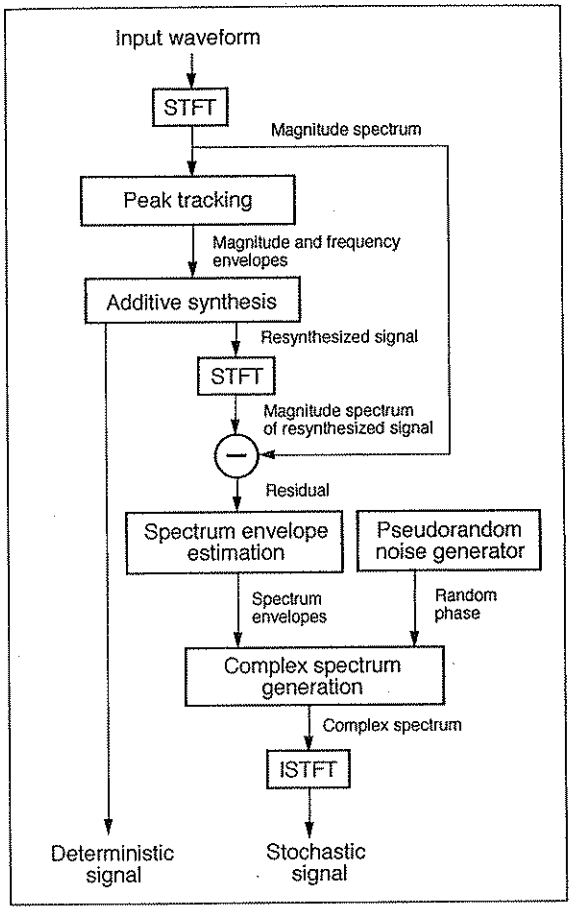


Figure 13.18 Analysis part of X. Serra's spectral modeling synthesis technique. The deterministic part follows a strictly sinusoidal additive synthesis approach. The stochastic part of the signal derives from the difference between the resynthesis of the deterministic (quasi-harmonic) part and the STFT of the input waveform. The system simplifies each residual component by fitting an envelope to it. The envelope representation makes the stochastic part easier to modify by musicians. The resynthesis of the stochastic part then uses these envelopes with a random phase component—equivalent to filtered white noise.

Constant Q Filter Bank Analysis

Various spectrum analysis methods can be grouped under the rubric of *constant Q* filter bank techniques—applied in audio research since the late 1970s (Petersen 1980; Petersen and Boll 1983; Schwede 1983; Musicus, Stautner, and Anderson 1984). Within this family are the so-called *auditory transform* (Stautner 1983) and the *bounded-Q frequency transform* (Mont-Reynaud 1985a; Chafe et al. 1985). The wavelet transform, discussed in the next section, could also be classified as a constant Q technique.

Recall from chapter 5 that Q can be defined for a bandpass filter as the ratio of its center frequency to its bandwidth. In a constant Q filter bank, each filter has a similar or the same Q. Thus the bandwidth of the high-frequency filters is much broader than those of the low-frequency filters, because, like musical intervals, constant Q analyzers work on a logarithmic frequency scale. For example, a one-third octave filter bank is a constant Q device.

Constant Q Versus Traditional Fourier Analysis

The constant Q filter bank's logarithmic frequency analysis is different from regular Fourier analyzers. Fourier analysis divides the spectrum into a set of equally spaced *frequency bins*, where there are half as many bins as there are samples taken as input (for real signals, negative frequency components duplicate the positive frequency components). In Fourier analysis, the width of a bin is a constant equal to the Nyquist rate divided by the number of bins. For example, for a 1024-point FFT at a sampling rate of 48 KHz, the width of a bin is 24,000/1024, or 23.43 Hz.

When the results of the FFT are translated to a logarithmic scale (such as musical octaves) it is clear that the resolution is worst in the lower octaves. To separate two low-frequency tones E1 (41.2 Hz) and F1 (43.65 Hz) that are a semitone apart requires a large time window (e.g., 2^{14} or 16,384 samples). But to use the same resolution at higher frequencies is a waste, since human beings have difficulty distinguishing between two tones that are 2.45 Hz apart in the octave between 10 and 20 KHz. Hence there is a mismatch between the logarithmic continuum of frequencies that we hear and the linear frequency scale of FFT analysis. This problem is addressed by methods like the constant Q transform, in which the bandwidth varies proportionally with frequency. That is, the analysis bands are thin for low frequencies and wide for high frequencies (figure 13.19). Thus in constant Q analysis the length of the analysis window varies according to the frequency being analyzed. Long windows analyze low frequencies, and short windows analyze high frequencies.

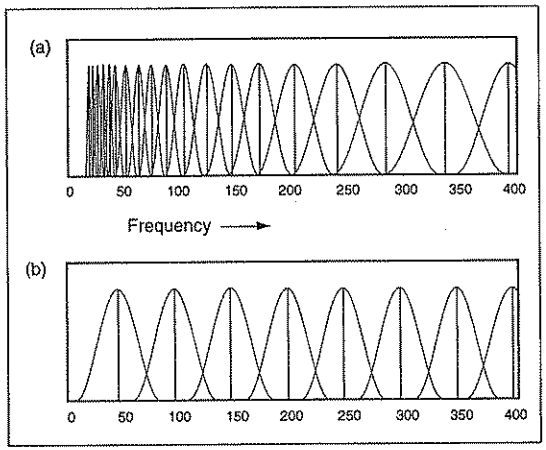


Figure 13.19 Spacing of filters for constant Q versus Fourier techniques. (*a*) Using only 43 filters (19 of which are shown), the constant Q method achieves 1/4-octave frequency resolution from 20 Hz to 21 KHz. (*b*) Fourier filter spacing, with a band every 46 Hz. Using almost 12 times as many filters (512, or which 8 are shown), Fourier methods still do not have the low-frequency resolution as constant Q methods. The Fourier method will have 46 Hz resolution throughout the audio bandwidth, even in the highest octave where the ear cannot accurately resolve these differences.

Constant Q filter banks do not avoid the uncertainty relationship between time and frequency, discussed earlier, but temporal uncertainty is concentrated in the lower octaves, where the analysis bands are narrow, and therefore the windows and the filter impulse responses are long. Since sonic transients (attacks) tend to contain high-frequency components, a constant Q response has the advantage of time localization in high frequencies with frequency localization in low frequencies.

Another attractive feature of constant Q techniques is that the human ear has a frequency response that resembles constant Q response, particularly above 500 Hz (Scharf 1961, 1970). That is, the auditory system performs a type of filter bank analysis with a frequency dependent bandwidth. These measured auditory bandwidths are of such a fundamental nature that they are called *critical bands*. (See chapter 23 for more on critical bands.) Figure 13.20 plots center frequencies versus bandwidths for a bank of 23 bandpass filters used in the so-called auditory transform, which was based on an