To the Graduate Council:

I am submitting herewith a thesis written by Michael Dean Christian entitled Morlet Wavelets and the Phase Vocoder Applied to Musical Sound Transformation. I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Bruce MacLennan, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:

Associate Vice Chancellor
and Dean of the Graduate School

Morlet Wavelets and the Phase Vocoder Applied to Musical Sound Transformation

A Thesis

Presented for the

Master of Science Degree

The University of Tennessee, Knoxville

Michael Dean Christian

August 1996

Acknowledgments

I wish to thank my advisor, Dr. Bruce MacLennan, for the support and guidance he has given me in understanding the material pertaining to my thesis. I would also like to thank Dr. David Straight for encouraging me to pursue my interest in sound synthesis, and for his time spent on my thesis committee. My gratitude also goes out to Dr. Thomas Ryan for his thorough critique of this work, and for serving on my committee. Many thanks to James Michael Myers for challenging me to consider how this research could affect the field of Virtual Reality and how to present this material to a non-technical audience.

I can only begin to express my gratitude to my wife, Donna, for standing by me and continuing to love me while I devoted far more time and energy to my thesis than to her; to my parents, for believing in me and encouraging me when I was losing hope; and to my dogs, Tip and Karma, for bringing peace and joy into my life everytime I see them.

Abstract

One of the primary challenges of musical signal processing entails discovering methods of producing sounds which are new, unique, and perhaps unattainable through conventional acoustic orchestration, but which have attributes of conventional instruments, imparting a sense of non-mechanical "realness" to the listener.

The Phase Vocoder, an analysis / resynthesis approach, allows us to analyze the time/frequency components of a sampled sound and reconstruct the sound as a sum of time-varying frequencies, like additive synthesis. Since we may start by sampling rich sounds, the sense of "realness" may be encoded into the analysis. The Phase Vocoder approach then allows us to analyze the phase and amplitude components of the particular frequencies we're interested in and modify these in various ways, such as "cross-synthesis" techniques combining the frequency components of one sound with the amplitude components of another, time-stretching and compression of sound without frequency modification, or frequency shifting without time-stretching/ compression.

We investigate the use of the Phase Vocoder in this context, using the Morlet wavelet transform rather than the short-time Fourier transform in the analysis phase, with the overall goal of implementing such a system on a personal computer. Morlet wavelets are more efficient in terms of amount of frequency bands needed, and hence time required for analysis, over the same overall bandwidth, and seem to better respect the auditory system's response to sound, due to octave

scaling, allowing for better time resolution for high frequencies and better frequency resolution for low frequencies. We then explore the various modifications allowed by the Phase Vocoder in resynthesizing sounds.

Contents

List of Figures

Chapter 1

Introduction

The increasing technological permeation of our society has caused many individuals to consider their disciplines in a new light. Computers often give us the ability to easily simulate the effects of altering parameters relevant to our source of study. While science has both brought about and benefitted greatly from technological breakthroughs, the arts have also exhibited a restructuring, perhaps mirroring the increased relevance of science and technology in our lives. In the area of music composition, technology has given us increased awareness and control over the micro-features of sound. The notion of timbral manipulation as a compositional tool is a relatively recent phenomenon (while I feel that to some extent Tibetan chant could be classified as a form of timbre manipulation, it was not specifically used for compositional purposes). The earliest compositional usage of timbral manipulation involved advanced orchestration techniques developed in the 1930's

and 1940's, such as those employed by Penderecki and later by George Crumb. Shortly following World War II, Musique Concrete was developed in France while Electronic Music evolved in Germany. Musique Concrète, pioneered by Pierre Schaeffer and Pierre Henry, focused on manipulating recorded, acoustic sounds by methods such as tape splicing, alteration of tape speed, filtering, and applying delay and reverberation. The goal was to start with rich, evocative sounds and manipulate them. Early Electronic Music, as demonstrated in the works of Karlheinz Stockhausen, utilized analog oscillators as the simple building blocks of more complex sounds, manipulating them with such techniques as filtering, reverberation and delay, ring modulation and amplitude modulation. Computer Music was developed in the late 1950's by Lejaran Hiller at the University of Illinois. He utilized the computer in the composition of sound by creating algorithmic composition, in which probability theory was used to compose music. Xenakis expounded upon this, developing a mathematical system for organizing compositions according to stochastic processes. He further developed these ideas into methods for controlling the density and evolution of "grains of sound", based on Dennis Gabor's notion of an "acoustic quanta", when he hypothesized [26] that:

> All sound is an integration of grains, of elementary sonic particles, of sonic quanta. Each of these elementary grains has a threefold nature: duration, frequency, and intensity. All sound, even all continuous sonic variation, is conceived as an assemblage of a large number of elementary grains adequately disposed in time. So every sonic complex can be analyzed as a series of pure sinusoidal sounds even if the variations of these sinusoidal sounds are infinitely close, short, and complex. In

> the attack, body, and decline of a complex sound, thousands of pure sounds appear in a more or less short interval of time, Δt. . . . A complex sound may be imagined as a multi-colored firework in which each point of light appears and instantaneously disappears against a black sky. But in this firework there would be such a quantity of points of light organized in such a way that their rapid and teeming succession would create forms and spirals, slowly unfolding, or conversely, brief explosions setting the whole sky aflame.

1.1 Relevance to Virtual Reality and Multimedia

Music seems to me to be a bridge between higher, "rational" consciousness and the reptilian forebrain, driving mathematicians such as Pythagoras to study its mathemathical and archetectonic features, while evoking frenzy not only in "primitive" tribes, but also in many modern-day concert-goers. The abstract nature of music, disassociated from the discrete and dissecting symbolism of language, and our general perception of it as relationships in time, allow us to experience levels of consciousness difficult to attain by other means.

The power of music to influence consciousness should not be overlooked, as I believe it often is, in the design of multimedia or Virtual Reality systems. While research into 3-D graphics and haptics is imperative, given their current weaknesses in simulating their correlates in "reality" when compared to the current state of 3-D sound and sound synthesis, nevertheless I believe there remains opportunities for increased realism as well as potential functionality in other areas, such as the creation of internal images from sounds.

Aldous Huxley [12] postulated that the brain was a filtering valve for "mind-at-large" or universal consciousness. While this idea has not played a major role in Western philosophy, it forms a large part of Eastern philosophy, probably reaching its highest point in Cittamatra, or "mind-only" Buddhism. Gabor proposed his "acoustic quanta" as a refutation of Helmholtz belief that our auditory system worked according to Fourier analysis, decomposing a time-domain signal into a number of sinusoids with different frequencies and amplitudes [9]. Gabor believed that we perceive sound as frequency changes over time, rather than the timeless description afforded by the Fourier transform. In searching for an explanation of the holistic nature of memory, Karl Pribram discovered holography, also invented by Gabor, and the Gabor transform. While Gabor derived holography according to the Fourier transform, Pribram recognized the need for a time/frequency description in the brain (as did Gabor in his proposal of "acoustic quanta"), and chose to study holographic brain mappings in terms of the Gabor transform, thereby linking Gabor's seminal discoveries [23]. Daugman has shown evidence of two-dimensional Gabor representations in the visual cortex of mammals [5].

Morlet wavelets are an extension of the Gabor transform which include a scaling factor that leads to better frequency localization in the low frequencies and better time localization for high frequencies. They can be looked upon as a type of filter, known as a Constant-Q filter, where the frequency bandwidth is propotional to the center frequency, while Gabor elementary functions form Constant

Bandwidth filters. The critical bandwidths of the ear appear to closely approximate a Constant-Q filter (see [20]). If all of our sensory modalities turn out to be filtering devices, then we could say that we are all living in a "virtual" reality. This suggests an alternative approach to Virtual Reality from the common method of building up virtual worlds from simple elements, such as polygons in 3-D graphics (perhaps a graphical analogy to additive synthesis). We could instead take the subtractive or "Virtual Concrete" approach, creating new worlds by applying filters to representations of "reality".

If our filters closely approximate the filters in the sense organs, study of and modifications to this representation may have more perceptual significance than to filters designed according to other criteria. Additionally, we may learn something about how our sense organs themselves affect what we perceive, and may be able to use this information to find ways to transcend constraints placed on our sense organs. Bekesy studied the effects of vibrations applied to volunteer's knees. Varying the frequencies, and applying different frequencies to each knee, he found that he could create the perceptual experience of the vibration "jumping" from one knee to the other. He even found frequency combinations that caused the volunteers to feel vibrations "between" their knees, as if they had a sense organ there [24]. These results, as well as such subjective sensations as the "phantom limb syndrome", where people who have lost a limb report feelings in the missing limb, imply that our brains are constructing reality, at least to some

extent, outside of the context of reality itself. If our sensory modalities are indeed representing what is "out there" by frequency, or time/frequency, representations, such representations may lead to holographic interference patterns in our brains which could create such a depth of experience that the patterns inside our brain are projecting an outside world.

1.2 Considerations for Effective Sound Manipulation

When considering sound modification techniques, I feel it's important to consider: (1) The time required by the sound designer to produce musically interesting sounds. (2) The knowledge of acoustics, and of the particular technique, needed to gain a measure of refinement and understanding of the technique. (3) The hardware resources required to implement the technique, and the time required by a particular hardware implementation.

Analog implementations of additive synthesis allow for time-varying functions to be applied to the amplitudes and phases of the oscillators (commonly generating sinusoidal frequency components) used to create sounds, while digital techniques usually utilize short, sampled sounds (wavetables). The primary difficulty with this technique is the large amount of time, number of oscillators or wavetables, and acoustic understanding required to attain reasonably complex spectral evolutions. Horner, Beauchamp, and Haken have attempted to alleviate some of this

burden from the sound designer by employing wavetables which are more complex than sinusoids, thus reducing the number of wavetables necessary for producing rich sounds, and using Genetic Algorithms or Principle Components Analysis to approximate acoustic textures [11]. Subtractive synthesis, based on filtering complex spectra (typically noise-based in analog implementations and often based on acoustic samples in digital implementations) requires much less effort to create complex sounds, but also gives far less precise control. Nonlinear waveshaping techniques, such as Frequency Modulation, can be extremely efficient in terms of both the time required to produce rich spectral evolutions and the amount of memory required to define the sound on a digital computer, but the degree of control can be even less precise than subtractive synthesis.

The Phase Vocoder is an Analysis / Resynthesis approach, which allows us to analyze acoustic sounds. This allows the user to have at his/her disposal sound samples which will likely have acoustic complexity and relevance. One common resynthesis technique utilizes additive synthesis, often employing wavetables. Thus the Phase Vocoder can be considered a fusion of the three major sound manipulation techniques of the Twentieth Century; Electronic Music, Musique Concrete, and Computer Music. Since we can analyze a large number of frequency bands, and can manipulate frequency and time separately on a sample-by-sample basis, this technique gives us extreme precision for manipulating sound. However, this precision comes with a price, as it does for additive synthesis. While the Phase

Vocoder alleviates the sound designer, to some extent, from the large amount of acoustic knowledge and time required to produce complex sounds, the system resources needed can still be large. As with additive synthesis, additional channels (filter banks for the Phase Vocoder) give increased precision for manipulating frequency, but require additional time or hardware. Utilizing Morlet wavelets allows us to utilize fewer filter banks and still cover the frequency range effectively. While the frequency resolution will not be as good for higher frequencies compared to a short-time Fourier transform analysis, the increased time resolution may have more psychoacoustic relevance.

1.3 Motivation

The initial impetus behind this project was a desire to utilize either wavelets, the Gabor transform, or the related technique of Granular Synthesis, for sound modification. To my awareness, at the time there were no commercial software or hardware implementations of these techniques. While Truax had done some outstanding work with real-time digital Granular Synthesis, and the software forming the Intelligent Music Workstation from Laboratorio di Informatica Musicale (L.I.M.) had very advanced features for Wavelet analysis and processing of sound, Truax software was unavailable and ran on an older mainframe and the Intelligent Music Workstation required a separate sound card for sound playback. Having ac-

cess to a Power Macintosh with internal high-quality sound sampling and playback capabilities, it was my intention to implement wavelet-based sound processing on this platform, both for my own, and hopefully other electronic musicians, explorations into audio manipulation.

Chapter 2

Sampling

2.1 Stages Involved in Sampling

I will present the basic concepts of discrete-time sampling without regard to the particular implementation. This information is derived largely from [17] and [16]. A more technical introduction can be found in [21].

Discrete-time sampling involves converting an analog signal source into a discrete form, represented as a sequence of bits in a computer. The number of samples per unit time is determined by the sampling rate, R, and the precision of each sample is determined by the number of bits, n, at which the computer quantizes each analog value. In the case of sound, the analog input consists of air pressure variations from the atmospheric mean (i.e. amplitude changes) over time. Since these fluctuations occur both above and below the atmospheric mean,

the sampling process usually encodes these changes as positive and negative y-coordinate values on the Cartesian plane, with the x-axis representing discrete points in time. Amplitude changes are bounded by 2^n for n bits of precision. In the case of Compact Disk (CD) quality sampling $n = 16$, so 65,526 different amplitude values can be represented. Sampling can be viewed as a four-stage process, shown in Figure 2.1. In the first stage, the analog signal is converted into analog electrical variations by a transducer such as a microphone. In the second stage, a low-pass filter is applied to the signal values greater than $R/2$ Hz (we will see shortly why this is necessary). In the third stage, an analog-to-digital converter (ADC) samples the frequency band-limited waveform, quantizing it to n bits of precision. While it can be shown that theoretically, with certain restrictions which we'll cover next, a series of discrete samples can accurately represent a continuous waveform, inaccuracy is introduced when we try to limit the continuous amplitude values to n bits of precision. This is known as quantization error. In the fourth stage, the sampled values are stored in computer memory.

2.2 The Nyquist Theorem

According to the Nyquist, or Sampling, Theorem, to represent an analog signal up to F Hz in discrete form, the minimal sampling rate must be $2 \cdot F$ samples per second. This is the reason we must band limit our analog signal to the region at or

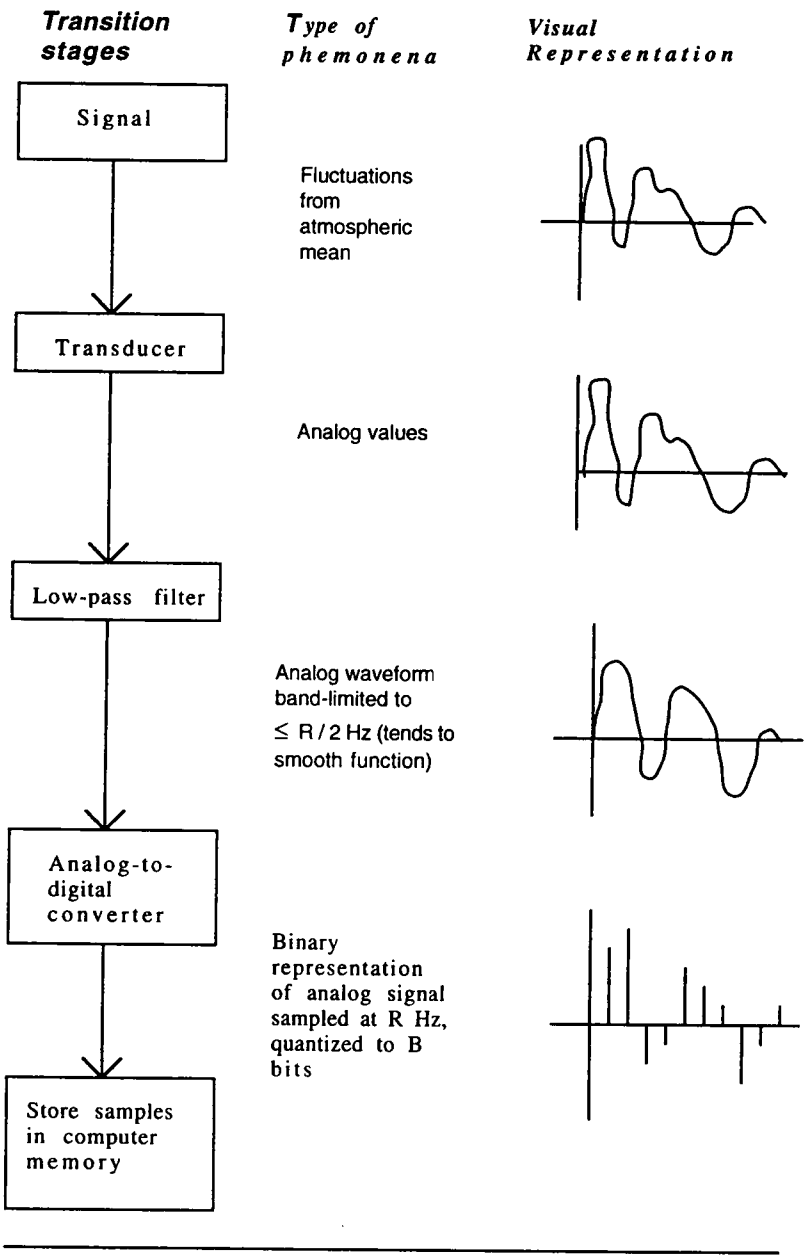

Figure 2.1: The four stages of sampling

below $R/2$ Hz, where R is the sampling rate. To see why this is so, let us consider a 2 Hz sine wave. The Nyquist Theorem tells us that we must take at least 4 samples per second to represent this signal accurately in discrete form. Figure 2.2 shows the analog signal and its discrete representation, when we use the minimal 4 samples-per-second rate. The bipolar nature of the discrete representation, with values oscillating between +1 and -1, seems to suggest that we have sampled a square wave. A square wave is defined as:

$$Sq(t) = \sum_{f=1}^{\infty} \frac{1}{f} \sin(2\pi f t),\ f \text{ odd} \tag{2.1}$$

A band-limited square wave is illustrated in Figure 2.3.

The wavelike structure of a sinusoid is analogous to the atmospheric fluctuations involved in sound propagation and mathematically represents a single frequency, with the frequency determined by the number of periods the sinusoid completes in one second. Any signal, no matter how complex, may be represented by a sum of some number, perhaps infinite, of harmonically related sinusoids (i.e. a fundamental, or lowest frequency, sinusoid and its harmonics, which are integer multiples of the fundamental). This is the basis of the Fourier transform, which we will soon cover. The Nyquist Theorem tells us that the highest frequency resolvable by discrete sampling with 4 samples-per-second sampling rate is 2 Hz. If we break apart a square wave whose fundamental is 2 Hz, we find that the

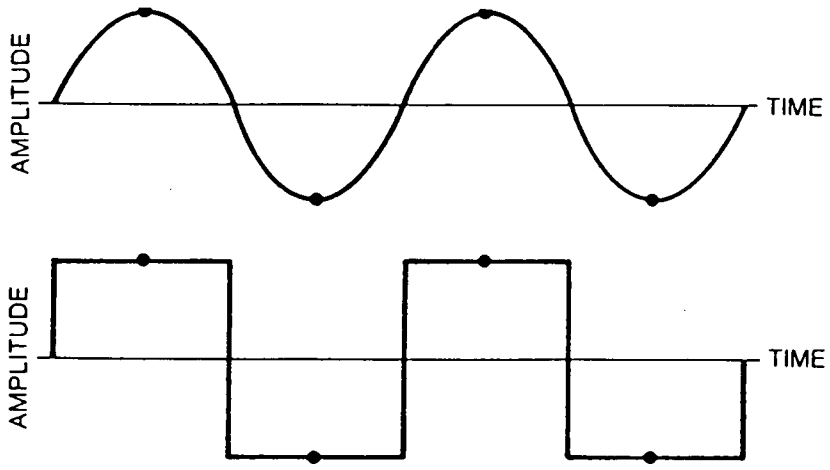

Figure 2.2: **top:** 2 Hz sine wave critically sampled at 4 samples per second. **bottom:** the bi-polar values seem to be representative of a square wave. (Source: [21])

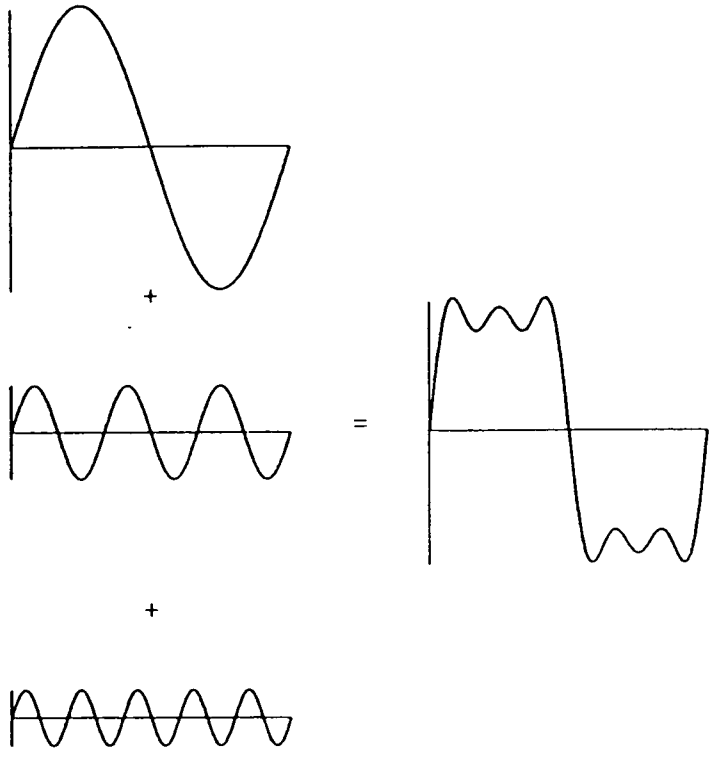

Figure 2.3: Band-limited square wave consisting of a fundamental at 2 Hz. and the third and fifth harmonic (Source: [17])

fundamental's amplitude is 1. The second component is the third harmonic of the fundamental, with a frequency of $3 \cdot 2 = 6$ Hz and an amplitude of $\frac{1}{3} = .333$. The third component is the fifth harmonic, with a frequency of 10 Hz and amplitude of $\frac{1}{5} = .20$, etc. In order to convert the sampled signal back into its analog form, which can then be propagated as sound through speakers, we must pass it thru a digital-to-analog converter (DAC). The DAC also applies a low-pass filter, eliminating frequencies above $R/2$ Hz. This will filter out all harmonics above the fundamental in our example, leaving the fundamental sinusoid at 2 Hz. Thus, only two points are required to represent a single period from analog to discrete form.

2.3 Aliasing

What would occur if we didn't band-limit our input signal, or if we make modifications to our samples which cause frequencies greater than $R/2$ Hz to occur? This is known as aliasing in the frequency domain. A common example of frequency domain aliasing is the appearance of a cars tires moving backward in a movie. Movies are shot at 24 frames per second, analogous to 24 samples per second in the audio domain. Frequency can be viewed as counter-clockwise movement on the unit circle. If we view a point on this circle which moves one time around the circle in one second, this represents one cycle per second (1 Hz) or 2π radians

per second. In general, if F is our frequency in Hz, the point will traverse $2\pi F$ radians per second, and $2\pi F$ is known as the radian frequency. In audio applications, we are typically limited by the hardware we're using to certain common sampling rates, such as 22,050 Hz and 44,100 Hz, the latter rate being the rate at which CDs are sampled. To illustrate how aliasing leads to the appearance of tires moving backward in a movie, let us consider a hardware limitation of 24 samples per second. The Nyquist Theorem tells that that the highest frequency that we can sample accurately is 12 Hz. We will consider frequencies of 1, 6, 12, 20, 24, and 30 Hz. Each sample will be taken at $2\pi F\frac{i}{R}$, where $R = 24$ and i increases from 0 to $R - 1$. With $F = 1$, samples will be taken at $2\pi\frac{0}{24} = 0$, $2\pi\frac{1}{24} = \frac{\pi}{12}$, $2\pi\frac{2}{24} = \frac{\pi}{6}$, etc., with the last sample taken at $2\pi\frac{23}{24} = 1.91666\pi$.

For $F = 3$, we see in Figure 2.4 that each sample is taken at $3/24 = 1/8$ of a revolution from the previous sample, and it appears obvious that the point is moving counter-clockwise along the unit circle with samples taken at 0, $2\pi\frac{3}{24} = \frac{\pi}{4}$, $2\pi\frac{6}{24} = \frac{\pi}{2}$, . . . , and finally $2\pi \cdot 3 \cdot \frac{23}{24} = 5.75\pi$. This would appear as counter-clockwise movement at intervals of $1/8$ of a revolution, or $\pi/4$. For $F = 6$ Hz, samples are taken at 0, $2\pi\frac{6}{24} = \frac{\pi}{2}$, $2\pi\frac{12}{24} = \pi$, . . . , and finally $2\pi \cdot 6 \cdot \frac{23}{24} = 11.5\pi$, lacking one sample of completing six full revolutions of the unit circle. Counter-clockwise movement at $1/4$ of a revolution is apparent in this case. Masters [15] makes an analogy between the perceived movement on the unit circle and the perceived movement of a clock hand illuminated by a strobe light (of

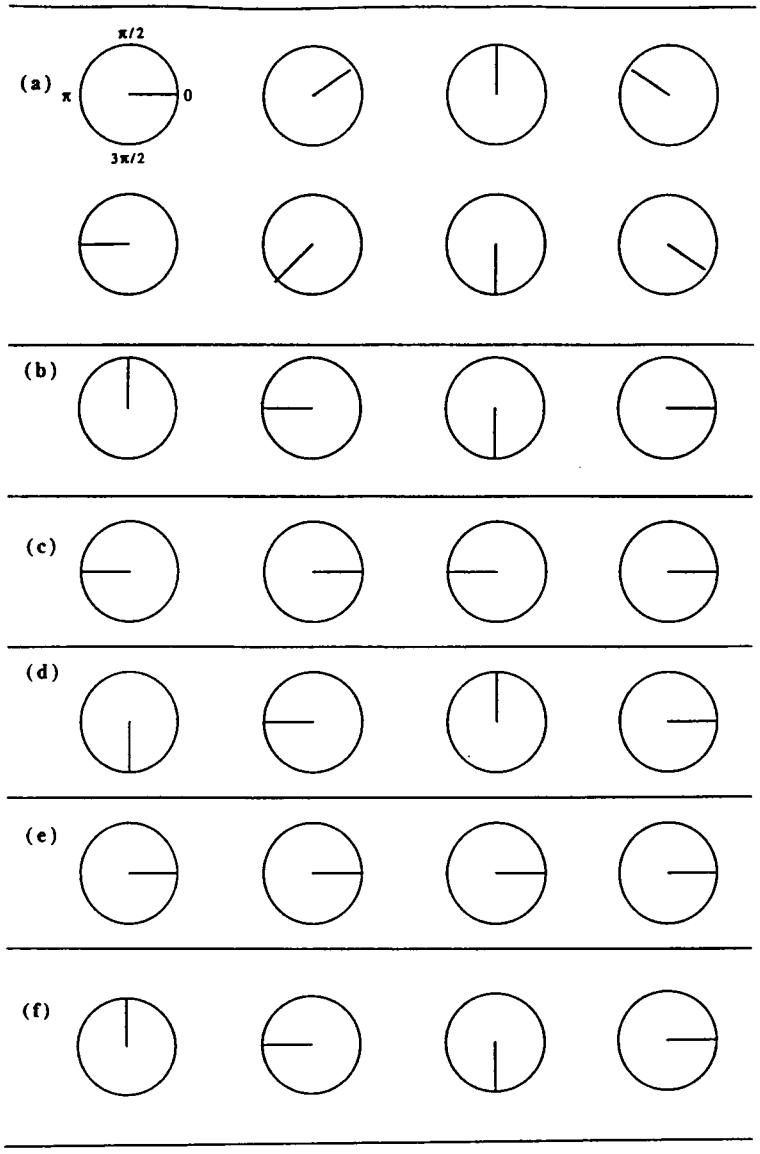

Figure 2.4: Frequency, F, depicted as motion on the unit circle, with samples taken at 24 samples per second. **(a)** For $F = 3$ Hz, counter-clockwise motion at intervals of $\pi/4$ is apparent. **(b)** For $F = 6$ Hz, apparent counter-clockwise motion at intervals of $\pi/2$. **(c)** For $F = 12$ Hz, motion at intervals of π; direction is ambiguous. **(d)** For $F = 18$ Hz, clockwise motion at intervals of $\pi/2$ is apparent. **(e)** For $F = 24$ Hz, it appears that there is no movement. **(f)** For $F = 30$ Hz, counter-clockwise motion at intervals of $\pi/4$ is apparent.

course, the clock hand is moving counter-clockwise). This is a very good analogy for discrete sampling, since the theoretically optimal sampling function is the Dirac Delta function, which has infinitesimal width and infinite energy (analogous to a very short, very bright burst of light). In the previous case, the strobe light would flash in three hour increments, so that counter-clockwise movement in three-hour increments would be apparent. As the frequency increases, the size of the increment for each sample appears to be increasing in a counter-clockwise direction, up to $F = 12Hz$. At 12 Hz, samples are taken at 0, $2\pi\frac{12}{24} = \pi$, $2\pi\frac{24}{24} = 2\pi$, $2\pi\frac{36}{24} = 3\pi$, etc., analogous to a clock whose hour hand moved in six hour increments. In this case, we can no longer tell which direction the point is moving in. This is an ambiguous case, and as we will see it is also an ambiguous case for the discrete Fourier transform (DFT). At 18 Hz, the point will appear at 0, $2\pi\frac{18}{24} = \frac{3\pi}{2}$, $2\pi\frac{36}{24} = 3\pi$, etc. In this case, the point appears to be moving in a *clockwise* direction at 1/4 revolution per sample, revealing how tires can appear to move backward when their speed increases to greater than 12 revolutions per second. Direction in a clockwise direction on the unit circle represents negative frequencies. At $F = 24$ Hz, samples will be taken at 0, 2π, 4π, etc. Since the circle is periodic with a period of 2π radians, each of these points is at the same place on the unit circle, and thus it appears that there is no movement. At $F = 30$ Hz, samples will be taken at $2\pi\frac{30}{24} = 2\pi + \frac{\pi}{4}$. Since $2\pi + \frac{\pi}{4}$ appears at the same place on the unit circle as $\frac{\pi}{4}$, it looks as though we've only moved $\frac{\pi}{4}$ radians,

which corresponds to our earlier sample of a 6 Hz signal. In [16] Moore presents the following formula for calculating the apparent frequency implied by discrete sampling of a continuous frequency:

$$F_a = F - (k+1)\frac{R}{2}, \quad k\frac{R}{2} \leq F \leq (k+2)\frac{R}{2} \tag{2.2}$$

where

Fa is the "apparent" frequency in Hz,
F is the actual frequency in Hz,
R is the sampling rate in Hz (samples per second), and
k is any odd integer which satisfies the inequality.

For our example where $F = 30Hz$, we see that the only odd integer which satisfies this inequality is 1, since $\frac{1 \cdot 24}{2} \leq 30 \leq \frac{3 \cdot 24}{2}$. In this case, the apparent frequency $Fa = 30 - \frac{(1+1)24}{2} = 6Hz$.

Aliasing is also known as foldover, and refers to frequencies above the Nyquist Frequency being shifted down and enfolded into frequencies below the Nyquist. This may be easier to visualize by representing frequencies as sinusoids. A sinusoid can be seen as an unraveling of the unit circle onto the Cartesian Coordinate Plane, with the origin centered at 0 radians for a sine wave, and $\pi/2$ radians for a cosine wave, both functions extending from negative infinity to positive infinity with periods of 2π. Figure 2.5 shows what happens when we try to sample frequencies above the Nyquist limit. While the original frequency is encoded by the system,

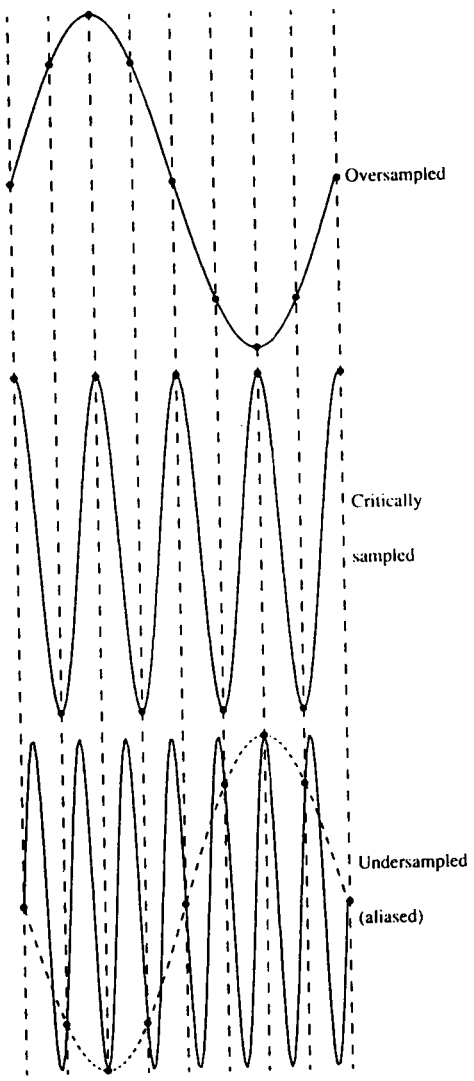

Figure 2.5: Sinusoids with frequencies of 1/8, 4/8, and 7/8 of the sampling rate. The sinusoid at 7/8 of the sampling rate is equivalent to a sinusoid at $-1/8$ of the sampling rate. (Source: [17])

the new "apparent" frequency is also encoded. The DAC will high-pass filter the samples later, leaving us with only the apparent frequency. If the signal consists of multiple frequency components, this new frequency will become enfolded into any "properly" sampled frequencies, thus corrupting the entire signal and leading to what is known as harmonic aliasing. It is very difficult, if not impossible, to remove these new frequencies after they've become encoded by the sampling process. Therefore, it is imperative that we apply a low-pass filter to the input, making sure we do not allow frequencies above the Nyquist.

Chapter 3

The Fourier Transform

We now present a discussion of the Fourier transform and its discrete version, the discrete Fourier transform (DFT). This information is derived largely from [17] and [15]. A clearer understanding of the Fourier transform requires us to pursue the topics of even and odd functions and Euler's identity first. We will also go over convolution, since the Fourier transform has important implications for convolution and the DFT implicitly relies on it.

3.1 Even and Odd Functions

Any function where $f(-x) = f(x)$ is known as an even function. A clear example of this is the cos function, which is symmetrical along the y-axis of the Cartesian plane. Another example of an even function is $f(x) = x^2$. Any function with the property that $f(-x) = -f(x)$ is referred to as an odd function. Two examples of

this type of function are $f(x) = x^3$ and the sin function, which is anti-symmetrical around the y-axis. An important mathematical property of even and odd functions is that any function may be represented as the sum of an even function and an odd function. If the function is purely an even function, then the odd part will be 0 and if the function is purely an odd function, then the even part will be 0. The reader is referred to [17, pages 63-64] for a simple proof of this property of even and odd functions.

3.2 Euler's Identity

Many functions may be stated mathematically as summation series, either finite or infinite. One group of series, the Maclaurin series, is capable of representing the sin and cos functions and the number e, among others. The Maclaurin series for e and e^x are the following:

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + ... \tag{3.1}$$

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \tag{3.2}$$

while the Maclaurin series for sin and cos are:

$$\sin(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} \tag{3.3}$$

$$= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + ... \tag{3.4}$$

$$\cos(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} \tag{3.5}$$

$$= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + ... \tag{3.6}$$

Euler's Identity describes the following relationship between complex exponentials and trigonometric functions:

$$e^{ix} = \cos(x) + i\sin(x) \tag{3.7}$$

where i represents the imaginary unit in the complex number plane, such that $i = \sqrt{-1}$. Substituting ix for x in the Maclaurin series expansion for e^x, we get:

$$\begin{aligned} e^{ix} &= 1 + ix + \frac{(ix)^2}{2!} + \frac{(ix)^3}{3!} + \frac{(ix)^4}{4!} + ... \\ &= 1 + ix - \frac{x^2}{2!} - \frac{(ix)^3}{3!} + \frac{x^4}{4!} + \frac{(ix)^5}{5!} + ... \\ &= (1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + ...) + i(x - \frac{x^3}{3!} + \frac{x^5}{5!} + ...) \\ &= \cos(x) + i\sin(x) \end{aligned}$$

Furthermore, Euler's Identity allows us to relate trigonometry to algebra, since for example, $e^{i\pi} = \cos(\pi) + i\sin(\pi) = -1 + 0 = -1$. The ability to build up arbitrary functions using the even and odd functions cos and sin, described in terms of complex exponentials is the basis of the Fourier transform.

3.3 Convolution

The convolution of two functions g and f, denoted $g * f$, is defined as:

$$g(x) * f(x) = \int_0^x g(x-v)f(v)\,dv \tag{3.8}$$

The discrete form is:

$$g(x) * f(x) = \sum_{i=0}^{n} g(i)f(n-i) \tag{3.9}$$

Since we will be relying on the discrete form in our discussion of the DFT, we will focus on this form. Let us consider convolution of a function with the unit sample function, where the unit sample function is defined as:

$$u(x) = \begin{cases} 1 & \text{if } x = 0; \\ 0 & \text{if } x \neq 0. \end{cases} \tag{3.10}$$

If we wish to delay the unit impulse by d samples, we have the following:

$$u(x-d) = \begin{cases} 1 & \text{if } x = d; \\ 0 & \text{if } x \neq d. \end{cases} \tag{3.11}$$

We can also scale the unit impulse by multiplying its values by a constant. If we convolve a function with the unit sample function, we see that the convolution is equal to the function. If we convolve the function with a delayed and/or scaled unit sample function, the result is a shifting and/or scaling of our function. Since we can view any sampled function as a number of delayed and scaled unit impulse functions, we can view convolution in the following manner: for each sampled point in the second function, we center the first function about it and scale the function by multiplying it by the value of the sampled point (see Figure 3.1). From this illustration and the definition of the convolution theorem, we see that the convolution of two functions yields a new function which generally has more elements than either of the two functions. In the discrete case, the number of elements in the new function is $|f| + |g| - 1$ elements. If the functions are not considered to be periodic, then we have what is known as a linear, or aperiodic, convolution with length $|f| + |g| - 1$, while for circular, or periodic, convolution, the period is $|f| + |g| - 1$ samples. We can also derive the convolution of the aforementioned functions by the following method: for each element in the set of one function, we multiply by all members of the other function, aligning the

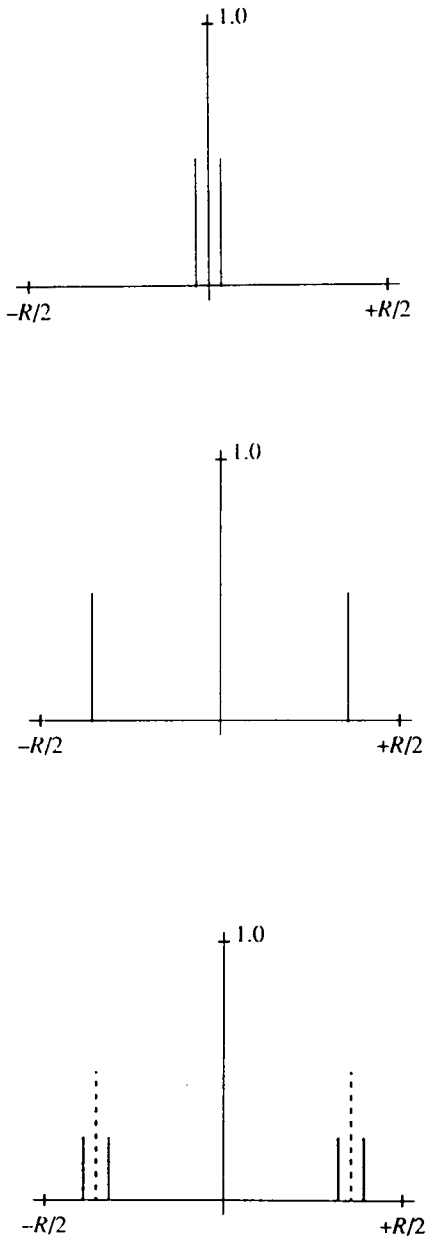

Figure 3.1: The convolution of two functions can be seen as centering one function, f, around each point of the other function,g, and scaling by that point's value. (Source: [17])

leftmost product with the multiplicand. After each element in one set has been multiplied by all members of the other set, we total the columns to get the result. Considering the two functions, $f(x) = \frac{x}{2}, -3 \leq x \leq 3$, and $g(x) = x, 1 \leq x \leq 3$, $x \in \mathbf{I}$ we have:

	$f(x)$							$*$	$g(x)$		
$=$	$\{$ -1.5	-1	-0.5	0	0.5	1	$1.5\}$	$*\{$	1	2	$3\}$
$=$	-1.5	-3	-4.5								
$+$		-1	-2	-3							
$+$			-0.5	-1	-1.5						
$+$				0	0	0					
$+$					0.5	1	1.5				
$+$						1	2	3			
$+$							1.5	3	4.5		
	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$		
$=$	-1.5	-4	-7	-4	-1	2	5	6	4.5		

and we see that we have $|f| + |g| - 1 = 7 + 3 - 1 = 9$ elements in our new function.

3.4 Continuous Fourier Transform

In our discussion of sampling, we saw that by its definition a square wave is composed of a sum of harmonically related sinusoids, with each harmonic component being an odd integer multiple of the fundamental frequency, with amplitude diminishing as the reciprocal of the harmonic number. While a square wave may be seen as a somewhat contrived example, nevertheless, the Fourier transform tells us how to compose any arbitrary function (or signal) as a sum of harmonically

related sinusoids. The Fourier transform is defined as:

$$H(f) = \int_{-\infty}^{\infty} h(t)e^{-2\pi ift}\,dt \tag{3.12}$$

and the inverse form is:

$$h(t) = \int_{-\infty}^{\infty} H(f)e^{2\pi ift}\,df \tag{3.13}$$

If $h(t)$ represents a function of time, such as a continuous audio signal , the first form tells us how to construct this same function as a function of frequency. The inverse form takes a frequency-domain representation of a function and tells us how to derive its time-domain values. Each representation is orthogonal, and there is no loss of information going between the two transformations (i.e., given $h(t)$ we can apply the forward transform to derive $H(f)$ and then apply the inverse transform to $H(f)$ and get $h(t)$ back exactly). These are very nice mathematical conditions which we will soon see do not apply to all spectral transforms (transforms which give us the spectral, or harmonic content, of a signal). However, these conditions do not necessarily mean that the Fourier transform of a signal is always a fair representation of what is actually occurring in the signal.

3.5 Discrete Fourier Transform

As we've seen, the Fourier transform is defined from negative infinity to positive infinity. For time-domain signals, this implies that the signal has no beginning or ending, as is the case for sinusoids for which we can only discuss periodicity and values referenced to some arbitrary starting point. Since we are concerned with analyzing audio signals, which generally appear to have definite starting and ending times, and since we do not have all of eternity to wait (unless we've mastered the ability to transfer consciousness into the after death state as well as move backward to the beginning of time, at which point the eternal signal began!) we need a form of the Fourier transform with a definite starting and ending time. This form is known as the discrete Fourier transform and is defined as:

$$H(k) = \frac{1}{N} \sum_{n=0}^{N-1} h(n) e^{\frac{-2\pi i k n}{N}} \quad \text{for } k = 0 \text{ to } N-1 \tag{3.14}$$

and its inverse form is:

$$h(n) = \sum_{k=0}^{N-1} H(k) e^{\frac{-2\pi i k n}{N}} \quad \text{for } n = 0 \text{ to } N-1 \tag{3.15}$$

where N is the number of samples of a signal and k is a frequency index. As we can see for both the discrete and continuous cases, the only difference in the forward and inverse forms is a change of sign in the complex exponent and the division by

N for the DFT. By Euler's Identity, we see that the change in sign reverses the sign of the imaginary sin component. The DFT requires that we divide by N after the summation in either the forward or inverse transforms (since the Fourier transform is linear, it doesn't matter when we apply the division by N). This is required for normalization. This point-by-point multiplication of the signal's samples and the complex sinusoids, and summation of the series, also known as the dot product or inner product, is a type of similarity measure. The DFT, then, is measuring the similarity between the complex sinusoids and the sampled signal, producing a single value for each sinusoidal frequency component. This value will be highest when the sampled signal contains a component at exactly the same frequency and in the same phase as the analyzing sinusoid, assuming we've measured an integral number of periods of the frequency. Because of the ambiguity of positive and negative frequency components, symmetric around the Nyquist frequency, the DFT effectively splits a frequency into positive and negative components and attributes half of the amplitude of the original frequency to each. Because sin is an odd function, its negative frequency value will be opposite in sign from its positive frequency value. This will only affect the phase of the encoded signal, however, not its amplitude.

Since the Fourier transform uses complex numbers to accomplish its task, we must use complex arithmetic to calculate amplitudes and phases. We can view this conceptually in Figure 3.2. Each analysis frequency is formed of a real and

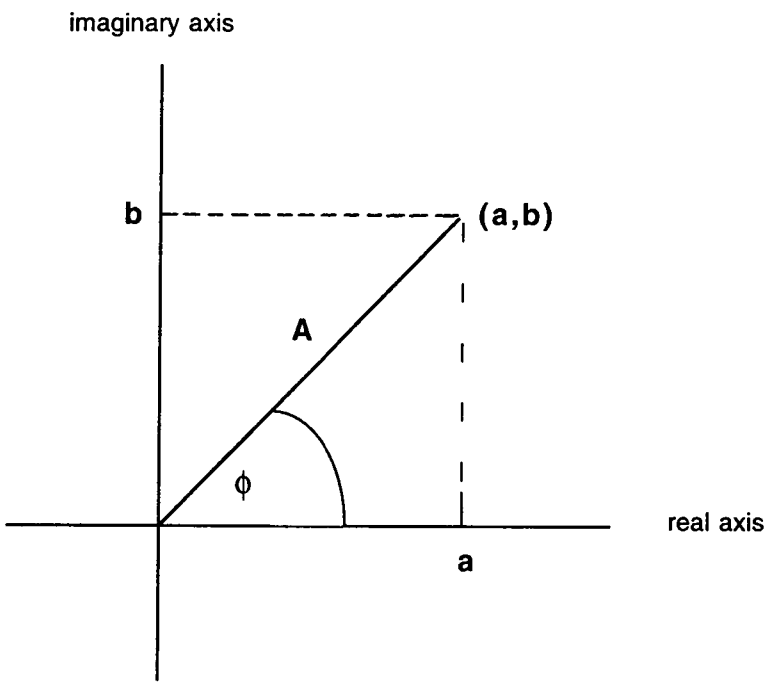

Figure 3.2: Magnitude and phase calculations using the complex coefficients of the DFT, where A is the magnitude and ϕ is the phase.

an imaginary number. These numbers can be plotted as a point on the complex number plane, with the real (cos) part determining displacement on the x-axis and the imaginary (sin) part setting the displacement on the y-axis. The magnitude of this point is the length of the line from the origin to the point, and is computed by the Pythagorean Theorem as:

$$A = \sqrt{a^2 + b^2} \tag{3.16}$$

where a is the real coefficient and b is the imaginary coefficient. Magnitude can be interpreted as amplitude in terms of audio signals. Phase is determined by the angle ϕ of this line from the x-axis. Since $\tan(\phi) = \frac{b}{a}$, we can calculate ϕ as:

$$\phi = \arctan \frac{b}{a} \tag{3.17}$$

This is illustrated in Figure 3.2. The squaring of the positive and negative complex components will cause the amplitude computed by the imaginary part to be positive. Phase will be reversed in sign, however, for negative frequency components with imaginary values. This should only occur for frequencies with sine wave components, in which case this is natural, since the sin function is odd and thus anit-symmetrical.

Although the Fourier transform, and the DFT, can represent any function or signal as a sum of sinusoids, a clear representation of the signal is usually only

discovered with the DFT when the signal is periodic, with an integral number of periods occuring per unit time or per the number of samples presented to the DFT. With the DFT the matter is further complicated by the fact that the DFT is not a true discrete representation of the continuous Fourier transform, unlike the summation series for sin, cos, and e among others. By limiting the Fourier transform to a definite beginning and ending time, we have implicitly multiplied it by a rectangular window function, defined as:

$$w(t) = \begin{cases} 1 & \text{for } t_{begin} \leq t \leq t_{end}\,; \\ 0 & \text{elsewhere.} \end{cases} \tag{3.18}$$

While this gives us excellent localization in the time domain, its frequency domain counterpart, $\text{sinc}(x) = \sin(x)/x$, is very poorly localized and can lead to what are known as analysis artifacts. To see why this is so, let us look at some common functions and their Fourier transforms, shown in Figure 3.3. Because of the reversibility of the Fourier transform, either one of each function pair can be viewed as being in the time domain, with the other in the frequency domain. If we consider the rectangular window function to be in the time domain we can notice the properties of its Fourier transform, the sinc function, in the frequency domain. The sinc function has a very wide central lobe compared to the Fourier transform of smooth, bell-shaped functions. The Fourier transform is linear, meaning that if we take the Fourier transforms of two, possibly scaled, functions and add them

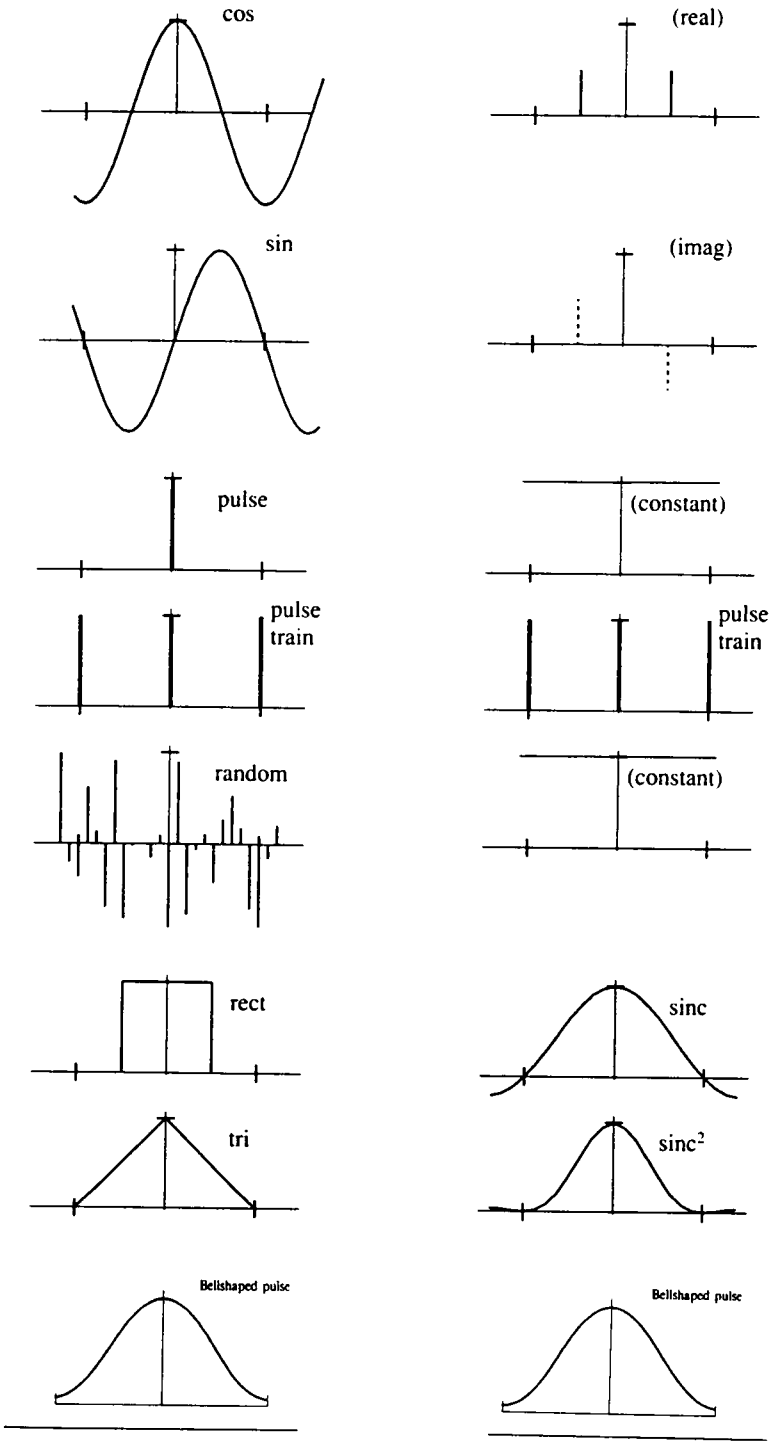

Figure 3.3: Fourier transform pairs of various functions. (Source: [17])

together, it is equivalent to scaling the transform of the first function and adding this to the scaled transform of the second function. Stated mathematically:

$$F(ag(t) + bh(t)) = aG(f) + bH(f) \tag{3.19}$$

However, the Fourier transform of the product of two functions is not equivalent to the product of the Fourier transforms of each function, but is rather the convolution of the Fourier transforms of the two functions. Stated mathematically:

$$F(g(t)h(t)) = G(f) * H(f) \tag{3.20}$$

Because of the implicit multiplication of the Fourier transform with the rectangular window function producing the DFT, we can alternately view the DFT as the convolution of the Fourier transform of the complex analyzing sinusoids with the Fourier transform of the rectangular window (i.e. the sinc function). The wide central lobe of the sinc function is an appealing property, since this function effectively lines up with each analysis frequency in the DFT because of implicit convolution. This means that it does not attenuate, to a great extent, frequencies near the analysis frequency. However, we can also note that the side lobes of the sinc function occurring between integral harmonics of the analysis frequency are also relatively wide and, more importantly, do not decrease rapidly as they move away from the analysis frequency. Moore [17] derives the amplitude response

occurring at the first side lobe of $3\pi/2$ as:

$$20 \log_{10} \left| sinc \frac{3\pi}{2} \right| \approx -13.5 \text{ dB}$$

which means that a frequency 1.5 harmonics away from the analysis frequency is attenuated by approximately 13.5 dB. For many sounds, this amount of attenuation is not enough to prevent the listener from perceiving this frequency. Perhaps even worse is the fact that the sinc function extends to infinity and continues to drop off slowly as it moves away from the analysis frequency (see Figure 3.4). While this will not corrupt the time-domain signal generated by the inverse DFT (i.e. we can still reconstruct the original signal perfectly), it can be seen as a corruption of the frequency-domain representation and can make the results of the DFT difficult, if not impossible, to interpret. Periodic functions which are harmonically related to the fundamental frequency as defined by the DFT will lie on integer points on the x-axis, which are zero crossing points for the sinc function. Thus the sinc function will not corrupt such functions. However, periodic functions which do not complete an integral number of periods per sampling time do not line up on integer multiples and will thus be corrupted by artifacts. Since the DFT is typically used for analyzing signals to try to gain a better perspective on the hidden structure of a waveform, corruption of this type often needs to be avoided.

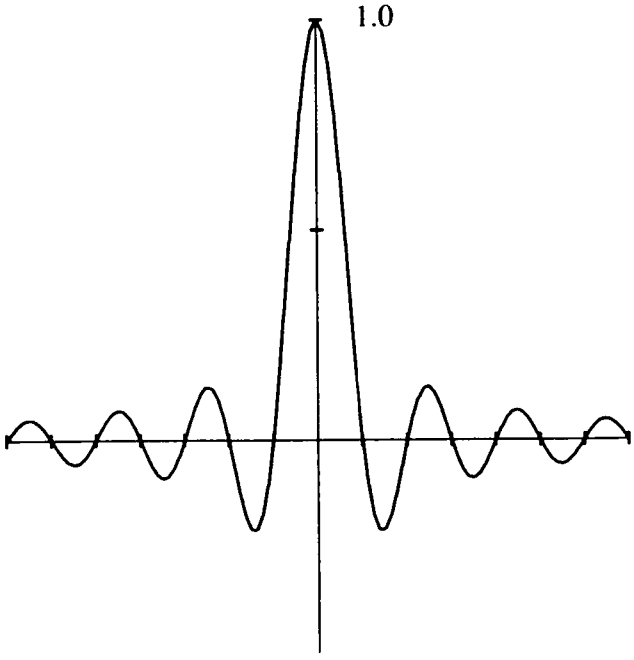

Figure 3.4: The sinc function, or Fourier transform of the rectangular window function, has a wide central lobe and side lobes which decrease slowly and extend from $-\infty$ to $+\infty$. (Source: [17])

Other window functions can be used, but each has its tradeoffs. The $sinc^2$ function is the derivative of the sinc function, and its Fourier transform is a triangle function. The $sinc^2$ function has a narrower central lobe implying that it attenuates frequencies near the analysis frequency more than the sinc function. However, its side lobe peaks drop off twice as quickly, leading to less artifacts from frequencies further away. While the $sinc^2$ function has better localization in the frequency domain, if we view it from that angle, its time-domain counterpart, the triangular window has twice the width of the rectangular window, meaning that its resolution in the time domain is not as good. Since the discrete Fourier transform tries to reconstruct whatever signal occurs during the sampling period as a sum of sinusoids, without allowing for changes occurring in the spectrum over time, the best it can do is reveal what frequency components are present in the signal, without revealing when they occur. If the time window that the DFT is using is wider, we will have a poorer description as to when certain frequency components occur. If the signal is truly periodic and harmonically related to the fundamental analysis frequency, the change in the width of the time window will not affect the analysis as long as the new width is an integer multiple of the old width, as it is in the case of the $sinc^2$ and sinc functions. But if the spectrum of the signal changes over time, the wider time window will cause the DFT to give us a less accurate representation of changes occurring over time, although the frequency resolution will be more accurate. There is an inherent tradeoff between frequency resolution

and time resolution. This fact was established by Dennis Gabor in 1946 and is the crux of our next topic. Before we move onto this topic, Time/Frequency Analysis, let us consider the implications of the rectangular window function in the time domain.

We've seen that the rectangular window function gives good time localization and can be made as good as possible by narrowing the width of the window. We can also gain an understanding of how it produces artifacts from looking at the behavior of the DFT. Because the DFT uses as its analyzing basis functions sines and cosines which are periodic, periodicity is assumed by the DFT. This implies that signals analyzed by the DFT should have periods which are integer multiples of the sample size. However, since we are only looking at some limited number of periods, this also implies that the spectral components must wrap around smoothly from the last sample point to the first sample point. Any deviation from this will lead to a jagged transition, which if we could listen to it would most likely appear as a rather nasty click in the audio domain. While the samples themselves don't wrap around, the DFT operates implicitly as if they do and thus must find unusual combinations of sinusoids to represent the signal. A function which gradually decreases the amplitude of the signal, such as a Gaussian or a Hamming Window, can be applied first to the samples to cause them to approach zero amplitude smoothly at the extremes of the sampled signal, thus allowing a smoother transition. In this case, we must also consider the convolution of the

Fourier transform of the window and the Fourier transform of the complex sinusoids. We see from our graph of various Fourier transform pairs (Figure 3.3) that most bell-shaped functions have spectral representations with much narrower central lobes, but much less pronounced side lobes compared to the sinc function, while the Gaussian remains a Gaussian, with no side lobe leakage. The Gaussian can be tailored for different central lobe widths, with wide time-domain Gaussians leading to narrower frequency-domain Gaussians and narrower time-domain Gaussians leading to wider frequency-domain versions. This tendency further illustrates the tradeoff in frequency vs. time resolution.

Chapter 4

Time / Frequency Analysis

In our previous discussion, we noted that the DFT does not give an accurate representation when analyzing a non-integral number of periods of a periodic signal. This may lead us to question how well the DFT will represent non-periodic signals, or signals with frequencies that vary over the analysis time window. While the inverse DFT will in fact give us a faithful reconstruction of such signals, the fact that the DFT only reveals what frequency components exist over the analysis window, without revealing when they occur, will hide the spectral evolution of a signal. It is well known that the attack, or onset, and decay of most musical instrument tones involves strong fluctuations in frequency, and even the so-called steady-state portions of most musical sounds are not truly periodic. For example, vocalists are typically trained to introduce vibrato (slight pitch variation) and tremolo (slight amplitude variation) when producing sustained notes. This lack

of mechanistic periodicity in music is often precisely what makes it interesting to us, and the lack thereof in many electronic instruments is what leads us to perceive them as mechanical. Therefore, if we want a more appropriate method of signal analysis, either to understand how the spectrum of a sound evolves over time or for manipulating the analysis coefficients in order to create new sounds having non-mechanistic attributes, we need to extend the capabilities of the Fourier transform. This leads us to a discussion of the short-time Fourier transform and its special case, the Gabor transform. This material is derived primarily from [14], [15], and [17]. A technical treatment of the implementation details can be found in [22].

4.1 Short-Time Fourier Transform

Dennis Gabor, the inventor of holography, was the first to propose a time-windowed, or short-time, Fourier transform. He proposed what is now known as the Gabor Transform as a theory of communication in 1946 [8]. The following year, he interpreted his theory in terms of sound and auditory perception [9]. In its general form, the continuous short-time Fourier transform (SFTF) is defined as:

$$F_x(\tau, \phi) = \int_{-\infty}^{\infty} x(t)\,\overline{g}(t-\tau)e^{-2\pi i\omega t}\,dt \tag{4.1}$$

where $\overline{g}$ is the complex conjugate of g. The discrete case can be defined as:

$$F_x(p,q) = \int_{-\infty}^{\infty} x(t)\,\overline{g}(t - p\tau_0)e^{-2\pi i q\omega_0 t}\,dt \tag{4.2}$$

where τ_0 represents discrete intervals in time and ϕ_0 represents discrete intervals in frequency.

We can see that the STFT is a function of three variables; frequency, time, and window function g. This allows us to view the STFT in two ways. If we consider a fixed point in time, the STFT would give us the Fourier Transform of a windowed waveform. Functions used for windowing typically are symmetric about a point in time t, and decrease rapidly to 0 or approximately 0. These include the Gaussian, Hamming, Hanning, and Kaiser windows. These have the effect of smoothing the waveform and driving its endpoints down to 0, or approximately 0, amplitude. When applied to the waveform before analysis with the DFT, this can have the effect of creating smooth wraparound of the signal, allowing the DFT to analyze the waveform as being periodic. More commonly the window function is viewed as modulating the real and imaginary sinusoids associated with the complex exponential, producing a damped oscillation. This is illustrated in Figures 4.1 and 4.2 We can denote the elementary function of the STFT as:

$$h_{\tau,\omega}(t) = h(t - \tau)e^{i\omega t} \tag{4.3}$$

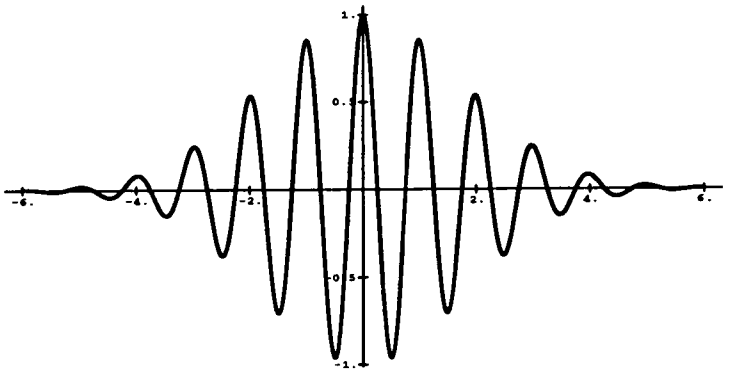

Figure 4.1: A Gaussian-modulated cosine function of the Gabor transform. (Source: [14])

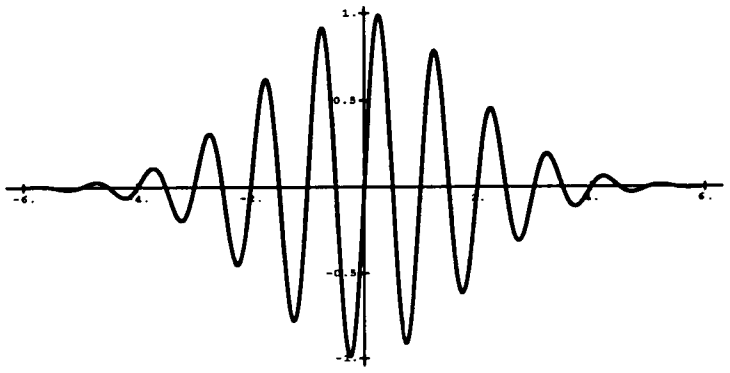

Figure 4.2: A Gaussian-modulated sine function of the Gabor transform. (Source: [14])

The STFT uses translations in time and frequency of this function, commonly referred to in wavelet literature as the mother function, as elementary analyzing functions. The convolution of the windowed sinusoids and the signal is then computed. If we consider a fixed frequency f, and knowing that multiplication in the time domain is equivalent to convolution in the frequency domain, we see that the convolution of the Fourier Transform of the analysis window with the Fourier Transform of the complex exponential effectively centers the frequency response of the window around the real and imaginary analysis frequencies. Considering all frequencies involved in the analysis, we can view the STFT as a bank of filters centered around harmonic integrals of the fundamental analysis frequency, with the filter responses determined by the window function chosen. We shall return to this topic later.

In our previous discussion of analysis windows in Chapter 3, we noted that windows which localized well in the time domain (with the rectangular window fulfilling this requirement perfectly) gave poor frequency resolution and can lead to spurious components being introduced into the analysis bandwidth. Since the Fourier Transform can be applied equivalently in the frequency domain to yield time domain counterparts, it's also true that analyzing filters yielding good frequency resolution give poor time resolution. For example, a perfect low-pass filter would pass all frequencies perfectly up to the filter's cutoff, and set all frequency components above this threshold to 0. This corresponds to a rectangular

filter window, which as we've seen previously produces the sinc function when the Fourier Transform is applied to it. Since the sinc function is defined from $-\infty$ to $+\infty$, with the side lobes decreasing slowly in magnitude, the perfect low-pass filter is therefore non-causal and extremely poor for time-domain resolution.

4.2 Gabor Uncertainty Principle

Gabor showed in his theory of communication that this tradeoff between time and frequency resolution is inescapable by generalizing the mathematical framework of the Heisenberg Uncertainty Principle of quantum mechanics. We now follow MacLennan's [14] approach in showing this. Consider a periodic signal which we would like to analyze to discover its frequency. This could be a sinusoid produced by an analog synthesizer, for example. Most musical instrument tones consist of numerous frequency components, as we've seen (with the flute being a notable exception, emphasizing primarily only the fundamental and the first harmonic), but for simplicity we'll consider a pure sinusoid. We could view the sound on an oscilloscope to get patterns such as in Figure 4.3. In that figure, Δt is the time interval over which we are analyzing the signal. One way of estimating frequency is to count the maxima occurring over the analysis time. For short time intervals, we may not have very many maxima to count. This, of course, is dependent on the frequency of the signal. The frequency analysis error, Δf, is inversely proportional

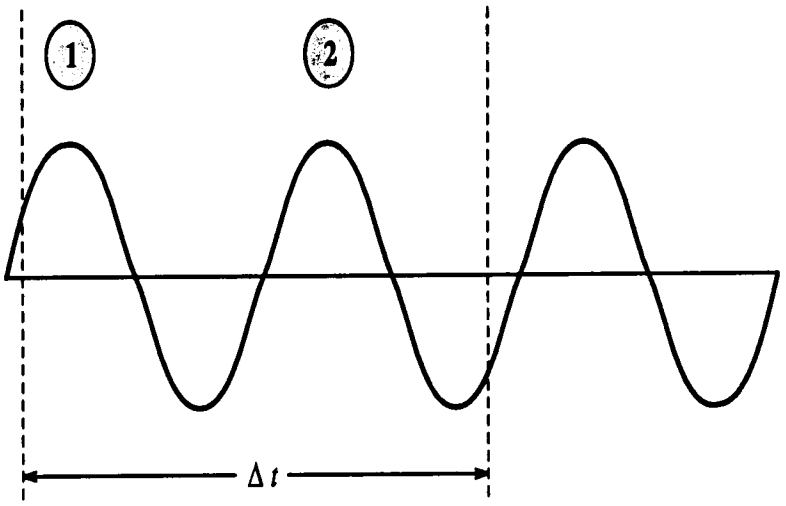

Figure 4.3: Frequency measurement using maxima counted over time, where circled numbers refer to the number of maxima. (Source: [14])

to the analysis duration, Δt. For this particular frequency measurement technique, the time required to resolve between two frequencies which differ by Δf would occur when the maxima count of the two frequencies differ by one (see Figure 4.4). Thus,

$$(f + \Delta f)\Delta t - f\Delta t \geq 1$$

which means,

$$\Delta f \Delta t \geq 1 \tag{4.4}$$

While different measurements yield different constants on the right hand side, nevertheless, since the product of frequency and time measurement is governed by a constant, increasing the measurement's accuracy in one dimension decreases the accuracy in the other.

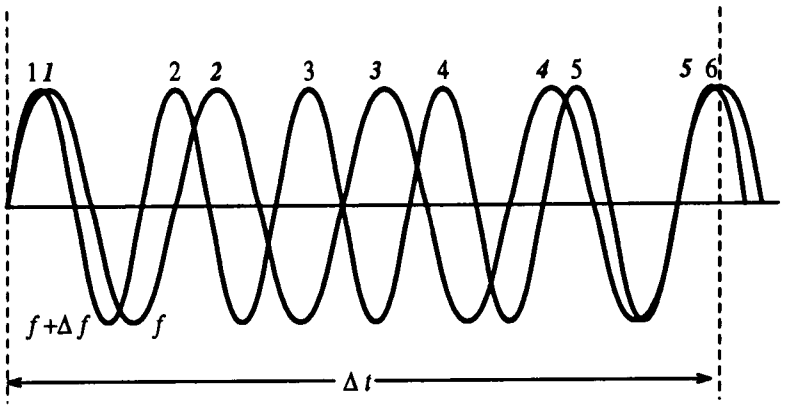

Figure 4.4: Minimum time interval required to distinquish the frequencies f and $f+\Delta f$. Italic numbers refer to maxima counted for f, while roman numerals refer to maxima for $f + \Delta f$. (Source: [14])

Since the "spread" of a signal is inversely proportional to the "spread" of its Fourier transform, (see Figure 4.5), we can reformulate the Gabor Uncertainty Principle in light of this. The spread of a function can be defined in more than one way. It can be defined as the function's standard deviation, for example. We follow MacLennan's convention of defining spreads in terms of a nominal bandwidth in the frequency domain or a nominal duration in the time domain. In this case, the nominal bandwidth refers to a rectangular region "that has the same area as the continuous spectrum and has a height equal to its amplitude at the origin," [14], while nominal duration is defined equivalently except that the area is defined in terms of the area of the signal. Since we are concerned primarily with analyzing time-varying signals, we must consider the case where the frequency components change during the time defined by the analysis window, in which case we are not measuring instantaneous frequency, but rather average

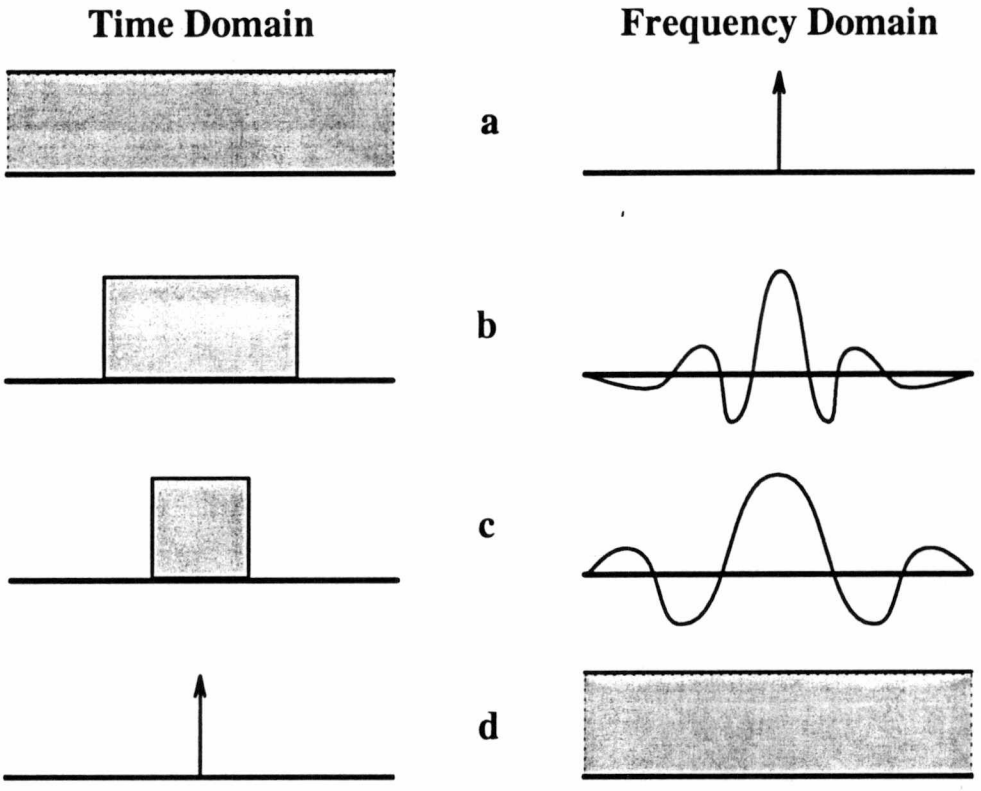

Figure 4.5: "Spreads" of a signal and its Fourier transform. **(a)**A constant signal has a unit impulse, or Dirac delta, frequency representation. **(b) and (c)** show how the frequency localization becomes wider as the time localization becomes narrower. **(d)** A unit impulse has a frequency spectrum which is a constant function. (Source: [14])

frequency over time. The nominal bandwidth measures the localization of spectral components while the nominal bandwidth measures the localization of the time domain signal. If we define the nominal duration of a general signal, with possibly negative components as:

$$\Delta t \, |\phi(0)| \;=\; \int_{-\infty}^{\infty} |\phi(t)| \, dt \tag{4.5}$$

and nominal bandwidth as:

$$\Delta f \, |\Phi(0)| \;=\; \int_{-\infty}^{\infty} |\Phi(f)| \, df \tag{4.6}$$

then we can prove the Gabor Uncertainty Principle for the spreads of a signals time and frequency components. We follow MacLennan's proof for general signals with possibly negative components:

$$\Delta t \, |\phi(0)| \;=\; \int |\phi(t)| \, dt \;\geq\; \left| \int \phi(t) \, dt \right| \;=\; |\Phi(0)|, \tag{4.7}$$

$$\Delta f \, |\Phi(0)| \;=\; \int |\Phi(t)| \, df \;\geq\; \left| \int \Phi(t) \, df \right| \;=\; |\phi(0)|, \tag{4.8}$$

Therefore, the bounds on the nominal frequency and duration spreads are:

$$\Delta f \geq \frac{|\phi(0)|}{|\Phi(0)|}, \quad \Delta t \geq \frac{|\Phi(0)|}{|\phi(0)|}. \tag{4.9}$$

This then gives us the generalized Gabor Uncertainty Principle:

$$\Delta f \Delta t \geq 1 \tag{4.10}$$

This tells us that we will be unable to distinguish frequency components differing by less than $\Delta f = \frac{1}{\Delta t}$. Therefore, if we want to be able to resolve frequency components smaller than our current Δf, we'll have to widen our analysis time-window, and vice-versa.

short-time Fourier transforms are often depicted as defining a two-dimensional coordinate system, or "Fourier space", as MacLennan [14] and Pribram [24] refer to a system whose coordinates reflect time and frequency, the domains unified by the Fourier Transform, where the x-axis typically represents time and the y-axis represents frequency. We can then view a time-frequency representation of a signal on a two-dimensional grid, where the grids are defined by the nominal bandwidth and nominal duration. The intersections of such grids form rectangles where $\Delta f \Delta t \geq 1$. (see figure 4.6) Note that if we make Δt infinitesimally small, as is the case with the Sampling Theorem, Δf becomes infinite. In other words, we have total time resolution and no frequency resolution, since the nominal bandwidth is infinite. Alternatively, if we make Δf infinitesimally small, we have excellent frequency resolution (as is the case with the continuous Fourier transform of a time-domain signal) but we have no localization in time (i.e. we can only know

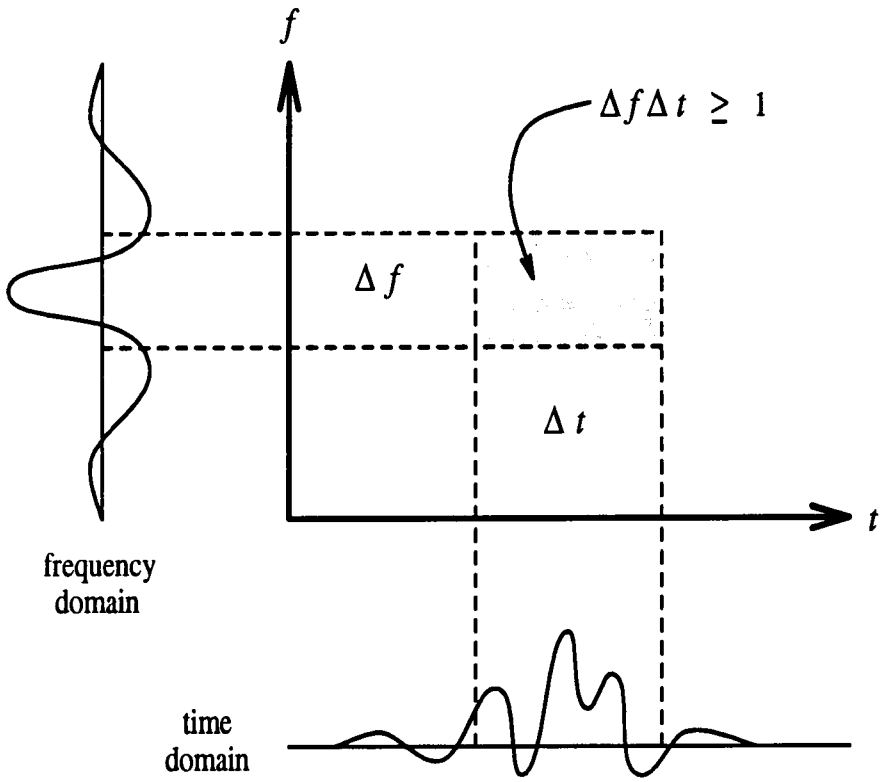

Figure 4.6: Division and uncertainty of Fourier space. The simultaneous localization of a signal in time / frequency is $\Delta f \Delta t \geq 1$. (Source: [14])

what the spectral components of a signal are, but not how they evolve over time). Gabor defined a rectangle where $\Delta f \Delta t = 1$, where uncertainty is minimized, as the elementary quantum of information, which he referred to as a *logon.*

4.3 Gabor Transform

Gabor recognized that physical signals have finite bandwidth and duration, as do our auditory and visual systems, and therefore the infinite duration / bandwidth utilized by the Fourier transform was not practical nor representative of physical signals. If the duration of a signal is limited by $M\Delta t$ and the bandwidth by $N\Delta f$, then we can transmit at most MN logons of information in the time $M\Delta t$. Gabor discovered that the rectangles could be minimized by using Gaussian-modulated complex exponentials, where the elementary function is:

$$g(x) = \pi^{-1/4} e^{\frac{-x^2}{2\sigma^2}} \tag{4.11}$$

where σ determines the localization of the Gaussian. Since the complex exponential decomposes into real cosine and imaginary sine components, this gives us two Gaussian-modulated sinusoids. The Gabor elementary function forms a complex spiral. In [19], Petersen makes an analogy to a helical spring stretched parallel to a wall and a floor. If the spring is centered at the origin and a light is then shown on the spring from above, its shadow will form the Guassian-modulated

sine, while if a light is shown from behind the spring (in relation to the wall), its shadow will form the Gaussian-modulated cosine (see Figure 4.7). Reiterating our previous definition of the STFT:

$$F_x(p,q) = \int_{-\infty}^{\infty} x(t)\,\overline{g}(t - p\tau_0)e^{-2\pi i q\omega_0 t}\,dt \tag{4.12}$$

we notice that the window function (Gaussian for the Gabor transform), is centered at multiples of τ_0, while the frequency translates by multiples of ω_0. We need to consider what values are appropriate for τ_0 and ω_0. If we set τ_0 equal to the reciprocal of the sampling rate, for example, computing spectral components for each sample, not only would the frequency resolution be compromised according to the Gabor Uncertainty Principle, but the amount of computation, time required, and final data size could be astronomical. By increasing the value of τ_0, we can view the discrete STFT as "hopping" the windowed complex exponential along the time axis at the points $n\tau_0$. By setting τ_0 equal to the diameter of the analysis window, we can reduce the amount of data and computation we have to deal with. However, this comes at a steep price. Since most commonly used analysis windows have maximum height at the center and taper to 0, or approximately 0, at the ends, it's obvious that the amplitude of the signal will be attenuated except at the center point of the analysis windows. Therefore, we will have lost information about the signal. We can remedy this, to a great extent, by

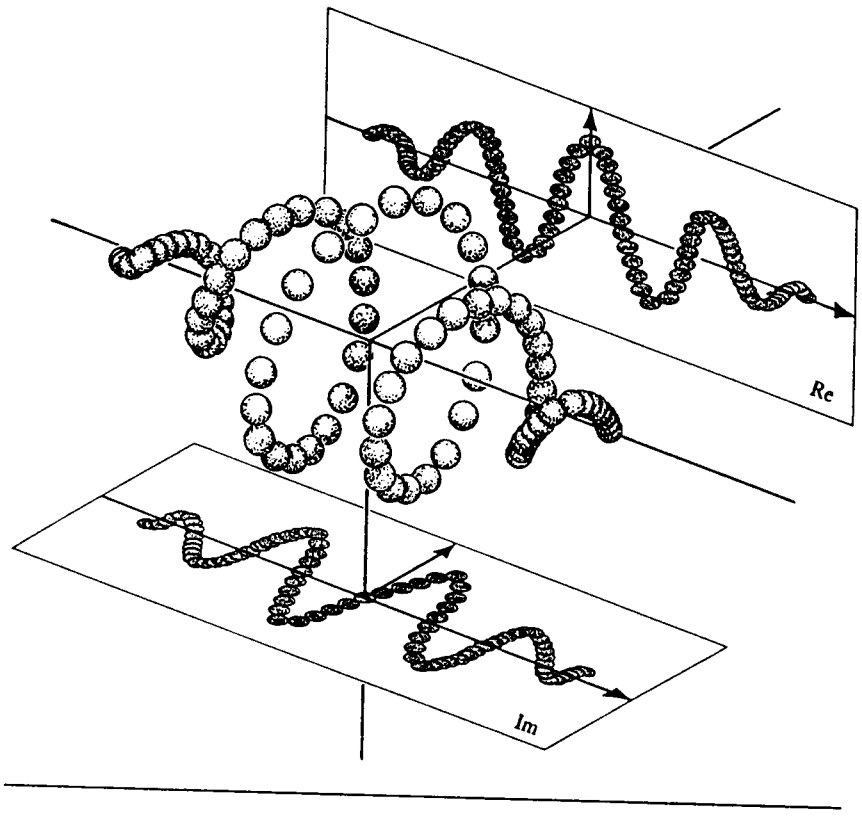

Figure 4.7: The complex spiral defined by the Gabor elementary function, and its real and imaginary projections (Source: [19])

overlapping our elementary functions, as depicted in Figure 4.8. Intuitively, we can see that this will introduce redundancy into our time-frequency analysis, since we are now taking spectrum measurements of the same samples more than once. We no longer have an orthogonal system, as we had with the Sampling Theorem and the Fourier transform. However, under certain conditions, we can still attain an "approximately orthogonal" representation, known as a frame, which we will cover soon.

By applying the Fourier transform to the Gaussian window of the Gabor elementary function, we get the following window:

$$G(f) = \sqrt{2\sigma}\pi^{\frac{1}{4}}\, e^{-2\pi^2\sigma^2 f^2} \tag{4.13}$$

which is, once again, a scaled Gaussian. As mentioned previously, the STFT can be viewed as a bank of filters. For the Gabor transform, then, the analysis bandwidths are of Gaussian form, and the extent to which the filter bandwidths overlap is inversely proportional to the amount of overlap of the time-domain elementary functions. Furthermore, their width is inversely proportional to the width of the time-domain analysis window, determined by the scale parameter σ (see Figure 4.9). Again we see the inherent tradeoff in time resolution versus frequency resolution.

The STFT, and its special case, the Gabor transform, uses windows which

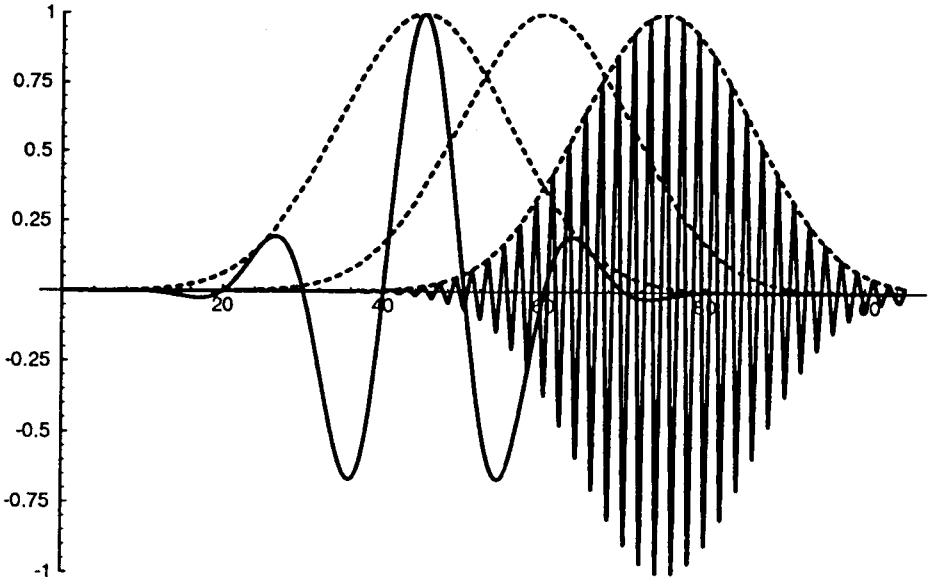

Figure 4.8: Overlapping Gabor elementary functions (Source: [15])

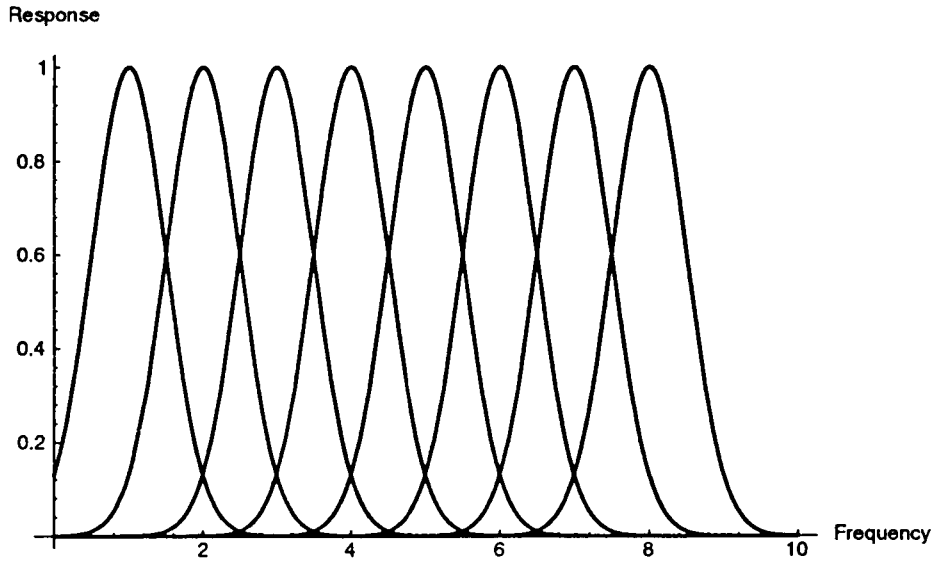

Figure 4.9: Constant Bandwidth filters defined by the Gabor transform (Source: [15])

define resolution rectangles of fixed size over both the time and frequency axes. Thus they reveal spectral components which are multiples of the fundamental analysis frequency (the lowest frequency analyzed) at fixed offsets in time, which are integrals of τ_0. Since harmonic overtones are multiples of a signal's fundamental frequency, it's clear that the STFT and Gabor transform are performing a harmonic analysis at fixed offsets of time. The extent to which they characterize the overtones of the signal correctly depends on several issues. The fundamental frequency is fixed by the choice of ω_0, because the value of ω_0 combined with the frequency spreads and possible scaling of the windows, determines how many analysis bands, N, will be required to effectively cover a given bandwidth. The fundamental frequency will then be R/N, where R is the sampling rate. If we're analyzing a monophonic sound (a sound from a single source), we need to know its fundamental frequency, which of course can vary over time, in order to properly analyze the harmonic overtone series. If we're analyzing a polyphonic sound (a signal originating from more than one source) we have to make a choice as to which fundamental to choose. Additionally, for polyphonic sounds, it may occur that overtones from more than one source fall within the frequency bandwidth of one of our analysis filters, at which point the best the STFT and Gabor transforms can do is give us the total energy falling within that bandwidth (attentuated by the analysis window), without revealing how much energy each source contributed. And of course we may have attenuation of any frequency component which does

not fall precisely in the center of the frequency analysis window, dependent on the particular window chosen and amount of overlap between frequency windows.

We've shown that there are fixed limits on the minimum spread in both time and frequency and that these are minimized for rectangles where $\Delta f \Delta t = 1$, also known as logons. The Gabor and short-time Fourier transforms utilize translations in time and frequency of an elementary function where the translations are multiples of a fixed time shift, τ_0, or frequency shift, ω_0, from the origin of Fourier space. Implicitly, then, all the resolution rectangles, spreads, are of the same size. Are there other ways to break up a two-dimensional coordinate system for the purposes of signal analysis? This question brings us to our next topic, wavelets.

Chapter 5

Wavelets

5.1 Psychoacoustic Relevance of a Time-Scale Approach

I've said that the Gabor transform can be viewed as dissecting a two-dimensional coordinate system into a gridwork of rectangles. This is also true of the STFT in general, but without using the Gabor elementary function the rectangles cannot be logons (i.e. they do not minimize the localization in both the time and frequency spreads). We've also seen that the frequency and time windows are all of the same shape and size and are thus performing, approximately, a harmonic analysis at fixed intervals of time. While this seems like a natural way to analyze pitched musical instrument tones, nevertheless it is probably not quite the way our auditory system analyzes these tones. Gabor recognized that we do not perceive music purely in terms of amplitude changes over time or as frequency

components with fixed amplitudes, but as spectral variations over time, and thus extended the idea of the logon to the notion of "acoustic quanta" [9]. While it is generally accepted that we do perceive sound and music in terms of frequency changes over time, there is evidence [20] that our auditory system resolves better in the time domain for high frequencies and better in the frequency domain for low frequencies. This is intuitively evident when we consider the relative interval between C_4 (middle C) and C_5 (octave above middle C), when $C_4 = 261.626$ Hz [2, pages 228-230]. With both the equal-tempered and the mean-tone scale, $C_5 = 523.25$, so the difference in frequency $\Delta f = 261.624 \approx C_4$. $C_6 = 1046.49$ and so the difference in frequency between C_5 and C_6, $\Delta f_2 = 523.24 \approx C_5 \approx 2 \cdot C_4$. We can see that the change in frequency from one octave to the one above it is $\Delta f/f = 1$. Filters where $\Delta f/f =$ constant are known as Constant-Q filters and are depicted in Figure 5.1. Alternatively, filters where $\Delta f =$ constant, as is the case of the STFT and Gabor transform, are known as Constant Bandwidth filters, and we've already seen their shapes (Figure 4.9). Furthermore, a pitch n octaves above another pitch will have a frequency of $2^n \cdot f$ where f is the lower pitch. Obviously, octaves are on an exponential scale.

Referring back to our frequency measurement technique of counting maxima over a given time interval Δt, if a time interval of Δt_1 is required to distinquish between f_1 and Δf_1, the time interval necessary to distinguish between the frequencies nf_1 and $n\Delta f_1$ is $\frac{1}{n}\Delta t_1$. Since this time interval is smaller for higher

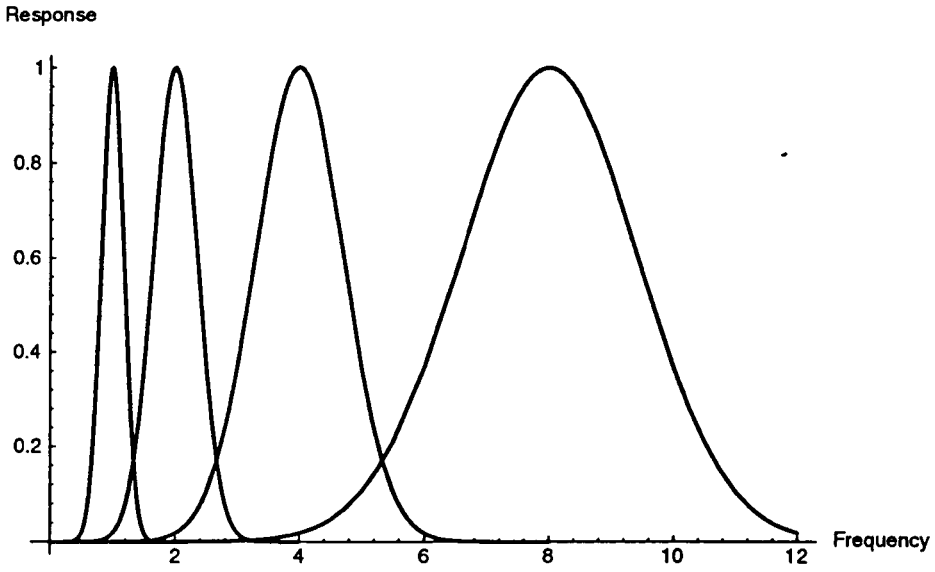

Figure 5.1: Constant-Q filters defined by the Morlet wavelet transform (Source: [15])

frequencies, it makes sense that the spread in the time domain of our analysis windows can be made smaller, leading to better time localization for higher frequencies. This property of scaling the analysis windows according to frequency has led to the term time-scale transforms for wavelets, rather than time-frequency. However, since scaling is applied according to frequency, the notion of a time-frequency transform is still implicit.

5.2 Generalized Wavelet Transforms

The continuous wavelet transform (CWT) of a signal x(t) is defined as:

$$W(\tau,a) \;=\; < g_{\tau,a}|x > \;=\; \int \overline{g}_{\tau,a}(t)\,x(t)\,dt \;=\; \int \frac{1}{\sqrt{a}}\,\overline{g}(\frac{t-\tau}{a})\,x(t)\,dt \qquad (5.1)$$

Comparing this to the definition of the continuous STFT:

$$F(\tau,a) \;=\; < h_{\tau,a}|x > \;=\; \int \overline{h}_{\tau,a}(t)\,x(t)\,dt \;=\; \int \overline{h}(t-\tau)\,e^{i\omega t}\,x(t)\,dt, \qquad (5.2)$$

notice the scaling factor, a, in the definition of the CWT. This parameter is what allows the size of the wavelet's resolution rectangles to vary. For the discrete case, the wavelet mother functions are of the form:

$$g_{p,q}(t) \;=\; \frac{1}{a_0^{q/2}}\,\overline{g}(\frac{t}{a_0^{q}} - p\tau_0) \qquad (5.3)$$

It is common practice that $q = 0$, the smallest scale, corresponds to the highest frequency analyzed. Thus there is no actual scaling of the highest frequency, since the time variable is divided by 1. It is also typical that $a_0 > 1$. A common value used is 2, in which case scaling is done according to octaves. Since the analysis window is centered at multiples of τ_0, the division by powers of a_0 leads to a wider spread in the time domain for lower frequencies. For example, if $a_0 = 2$, and the analysis window is currently centered at the 15th sample point, the amount of attenuation for a compactly-supported Mother function at time $t_1 = 12$ for the smallest scale (highest frequency) wavelet is equivalent to the amount of attenuation at time $t_2 = 6$ for the next smallest scale and to time $t_3 = 3$ for the third smallest scale. We have seen how the lattice-spacing of the two-dimensional coordinate system differs for wavelets and the STFT, and have implied that frequency resolution is better in the lower coordinates of the scale plane and time resolution is better in the coordinates corresponding to high frequencies. This can be seen in Figures 5.2 and 5.3. However, while the wavelet transform defines a rectangular region of influence or spread, in the frequency domain, it forms a triangle of influence in the time-domain [13]. This localization property has the effect of revealing phase discontinuities, or abrupt changes in frequency. For this reason, it is better adapted for analyzing rapid spectral changes, such as those commonly associated with the attack and decay of musical instrument sounds. The division by $\sqrt{a}$ normalizes the energy of the tranform across scales.

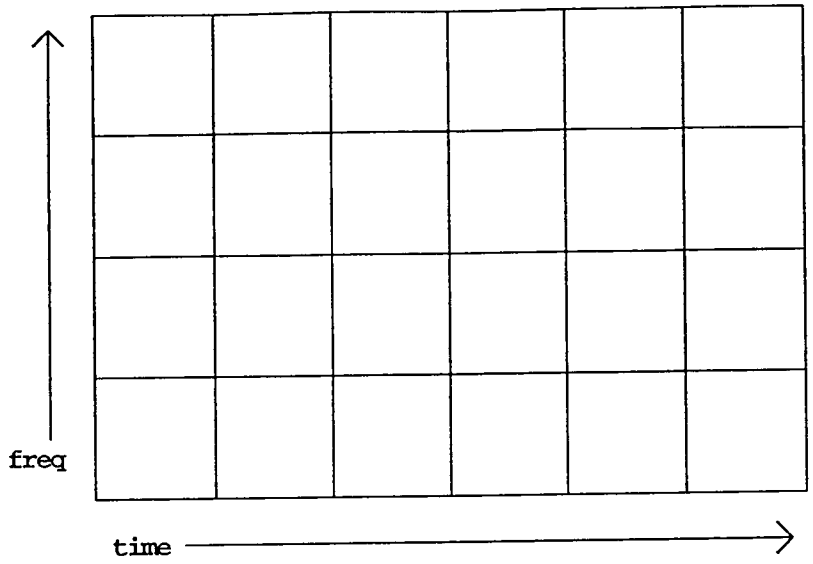

Figure 5.2: Dissection of Fourier space by the STFT

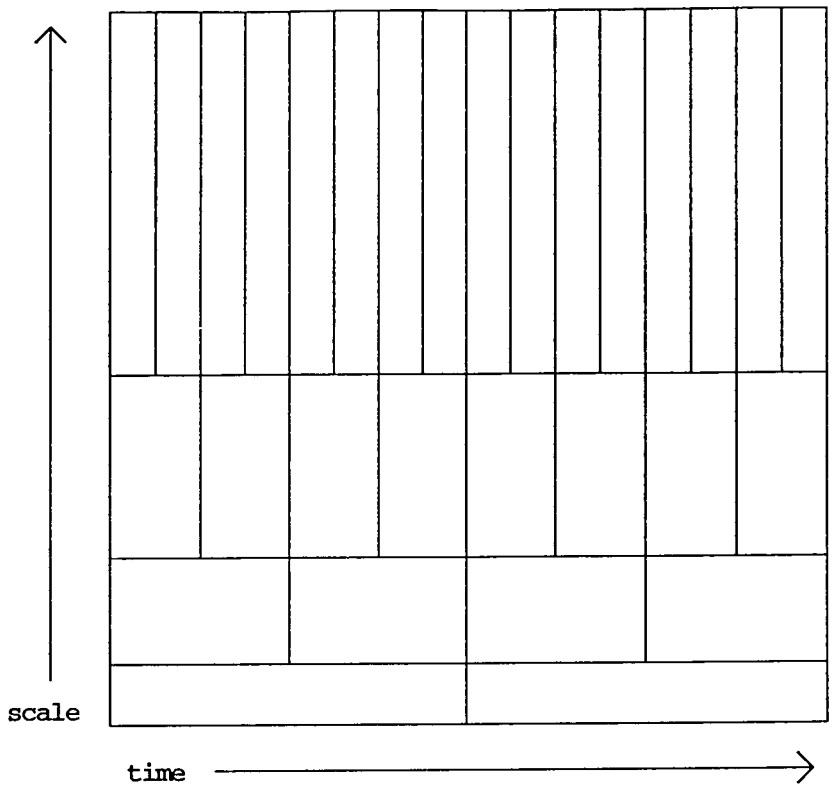

Figure 5.3: Dissection of Fourier space by the wavelet transform

There exist numerous elementary, or mother, wavelet functions, many of which have been shown to be orthogonal [see [3] for examples]. These are nice, since they allow a perfect reconstruction of the signal through use of a simple inner product between the transform's coeffiecients and the particular inverse wavelet transform. However, these wavelets do not localize as well in terms of both time and frequency as the Gabor transform, based on the Guassian envelope. They also commonly have basis functions which do not have a clear connection with frequency, unlike the sine / cosine pair of the Fourier transform or the modulated sine / cosine pair of the STFT. Morlet formalized the notion of wavelets, utilizing an elementary function very similar to the Gabor elementary function. He and Grossman extended the concept to basis functions which meet certain admissibility requirements, and coined the term wavelets for them. For continuous wavelets, an inverse transform exists, allowing the reconstruction of the original time series, if these admissibility conditions are met. The inverse wavelet transform is:

$$x(t) = \frac{1}{C_g} \int \int W(\tau, a) g_{\tau,a}(t) \frac{d\tau da}{a^2} \tag{5.4}$$

where C_g is a constant dependent on the particular wavelet chosen, such that:

$$C_g = 2\pi \int \frac{|\hat{g}(\omega)|^2}{\omega} d\omega \tag{5.5}$$

where $\hat{g}$ is the Fourier transform of g. This condition is met when C_g converges,

which for differentiable $G(\omega)$ means [13]:

$$1. \quad G(0) = \int g(t)\,dt = 0 \quad \text{(i.e. g has no DC bias).} \tag{5.6}$$

$$2. \quad |g(t)|^2\,dt < \infty \quad \text{(i.e. the squared modulus, energy, is finite).} \tag{5.7}$$

5.3 Morlet Wavelets

The Morlet wavelet mother function is of the form:

$$g(t) = \pi^{-1/4}\left(e^{i\omega t} - e^{-\omega^2/2}\right) e^{-t^2/2} \tag{5.8}$$

The Fourier transform of this function is a shifted Guassian, where the shift is necessary so that the wavelet basis contains no DC bias. Once again, we have as analyzing functions Gaussian-modulated sinusoid pairs, only this time they are subjected to a change in scale (see Figure 5.4). Also note that while this particular wavelet looks very similar to the Gabor basis function, the complex exponential is included as part of the definition of the basis function, and is not inherently part of the general definition of wavelets. The STFT consists of basis functions, where the analysis window, $h_{\tau,\omega}$, is the same Gaussian window, translated to different time locations, modulating different frequencies. Therefore, for higher frequencies, more oscillations will occur in the analysis window. The scaling of the Gaussian defined by the Morlet transform, with narrower Gaussians for higher frequencies,

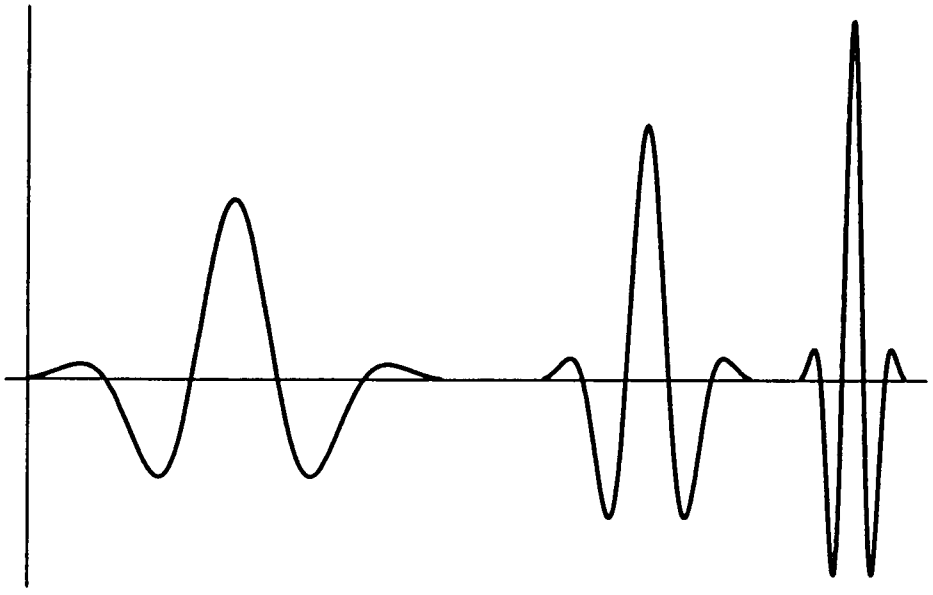

Figure 5.4: Three Morlet wavelets with increasing scale. Scaling leads to the same number of oscillations occuring in each window. (Source: [15])

causes the number of oscillation to remain constant.

Since the focus of my research is audio signal processing of musical sounds and their spectral evolution, I believe it makes sense to use a wavelet transform which has a clear connection to frequency, and have thus chosen to investigate the use of Morlet wavelets in this context. While the lack of orthogonality poses some difficulties with the computation of the wavelet coefficients and reconstruction of the time series, MacLennan notes that because there are no degrees of orthogonality, biological systems probably cannot depend on orthogonal representations [14]. For example, if the system were subsequently damaged and coefficients were lost, the representation would no longer form a basis and important information might be lost. If the system was dependent on an orthogonal basis, it might be unable to cope with the loss of information. In the case of damaged sense organs, redundancy may be useful. While both the Gabor transform and the Morlet wavelet transform do not form an orthogonal basis, they may form a generalization of a basis, known as a frame.

5.4 Frames and Reconstruction

It can be proven that continuous wavelets, including the Morlet wavelet, which meet the previously defined admissibility conditions can be used to reconstruct the original time series (see [4, pages 24-27]). However, for nonorthogonal transforms,

perfect reconstruction cannot be computed by an inner product, and requires more advanced algorithms (see [5]). When we discretize our wavelets, reconstruction is not quaranteed, based on the previous admissibility requirements. We are now shifting the centers of the time and frequency (or scale) windows at discrete intervals over the two-dimensional coordinates of time and frequency (or scale). It would be nice if we could reduce the amount of redundancy in our representation, both to reduce the amount of memory required to store the coefficients, and to make reconstruction easier. However, we need to make sure we haven't lost information by creating gaps in the spreads of the windows. If we want good localization in both time and frequency, redundancy cannot be avoided.

While we can capture all the information in the signal with a nonorthogonal discrete transform, the wavelets used do not constitute a true basis, since they are not linearly independent. The amount of redundancy in our representation can be characterized through the use of frames. If Ψ_i is a possibly infinite set of vectors (this can be extended to functions) in Hilbert space H, then Ψ_i forms an orthonormal basis iff:

$$\|f\|^2 = \Sigma_i | < f, \Psi_i > |^2 \tag{5.9}$$

In other words, the squared length of any vector, f, in H is equal to the sum of squared projections of the vector, f, onto the vectors Ψ_i. This idea can be extended to nonorthogonal bases Ψ_i if there exist constants $A > 0$, $B < \infty$ that

bound the sum of squared projections of f onto Ψ_i, for any f in H. In the case of discrete wavelets, we can replace the Ψ_i by the wavelet funtions g_{pq}. Frames are thus defined as:

$$A\|f\|^2 \leq |< f, \Psi_i >|^2 \leq B\|f\|^2 \tag{5.10}$$

If $A = B = 1$ we have an orthonormal basis. If $A = B$, known as a tight frame, their values indicate the amount of redundancy of the frame. If $A = B$, stable reconstruction can be attained. If $B > A$, the ratio B/A determines the stability and rapidity of convergence, which are best when $B/A = 1$.

Masters [15, pages 146-149, 177-179] explains the necessary conditions for generating frames for both the STFT and wavelet transforms. For the STFT, it's necessary that the spacing between the centers of the analysis windows be tight enough that no information is lost. If $\tau_0 \cdot \phi_0 > 1$, no function exists that can generate a frame. If $\tau_0 \cdot \phi_0 = 1$, a special function may exist that generates a frame, but any such function will be unable to localize well in both time and frequency. Only a function where the product is less than 1 can both generate a frame and localize well in both time and frequency. The second condition for generating a frame is:

$$\Sigma_p |g(t - p\tau_o)|^2 > 0 \quad t = 0, 1, ..., \|x\|$$

This condition is satisfied when all samples t in $x(t)$ are covered by the spread of an analysis window. The third condition is that the window must decay rapidly

as it moves away from the center, $(1+|t|)^{-(1+\epsilon)}$, $\epsilon > 0$, or faster. While these are necessary conditions for generating frames for the STFT, they are not sufficient conditions. A more rigorous treatment of the necessary and sufficient conditions can be found in [3].

For discrete wavelets, we have the same basic conditions, except that a good frame cannot necessarily be generated for all wavelet mother functions when $\tau_0 \cdot \xi_0 < 1$, where ξ_0 is the scaling parameter. Daubechies [3] also gives the sufficient conditions for generating frames for discrete wavelets, when restrictions are placed on ξ_0. When $\xi_0 = 2$, as is typical in many wavelet applications and which makes sense for audio applications since the scaling is then based on octave intervals of the well-tempered scale, a slight modification to the basic wavelet transformy is then required for many Mother functions, including the Morlet, in order to attain tight frames. This modification involves constructing more than one Mother function, in which case the Mother functions are known as voices. If we wish to split up the octave, analyzing N frequency bands per octave, we can use wavelet Mother functions (voices) of the following form, as proposed by Grossmann, Kronland-Martinet, and Morlet [10]:

$$h^j = 2^{-j/N} h(2^{-j/N} t) \quad j = 0, ..., N-1 \tag{5.11}$$

We now have N voices per octave, where the analysis centers are shifted in the

frequency dimension by $2^{-j/N}, j = 0, ..., N-1$, relative to the lowest voice, h^0, in the octave, but which all line up in the time dimension (see Figure 5.5). Twelve voices per octave would give us frequency bands associated with the twelve-tone well-tempered scale of Western music, for example. This may be useful in musical applications.

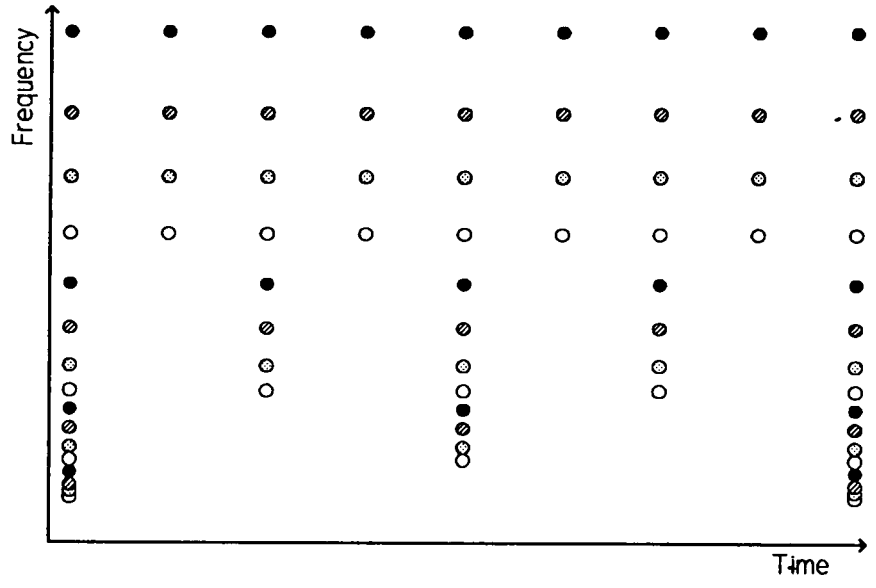

Figure 5.5: Modified spacing of wavelet grid with four voices per octave. (Source: [15]

Chapter 6

Implementation and Musical Applications

The understanding and manipulation of sound has played an important role in the areas of signal analysis and Digital Signal Processing. Gabor proposed his notion of "acoustic quanta" as a more precise description of our psycho-acoustical experience [9]. Morlet, Grossmann, and Kronland-Martinet have shown a great deal of interest in the applications of wavelets to both musical analysis and manipulation. Xenakis was the first person to implement Granular Synthesis, utilizing "grains" of sound analagous to Gabor's acoustical quanta, on a computer, and the first to propose a theory of composition based on the notion of "acoustical quanta" [26], while Truax was the first person to attain real-time Granular Synthesis on a computer [25], utilizing this for musical purposes.

My goal is to use the wavelet transform to decompose sampled sounds in order to manipulate them in perceptually interesting ways and to be able to play the altered sounds through a personal computer, possibly saving them as sampled sound files. To this end, the following criteria are necessary:

(1) Optimized algorithms: While the speed of personal computers has come a long way over the past few years, with current computational speeds exceeding most workstations and many mainframes from only a few years ago, nevertheless both the STFT and wavelet transforms can be tediously slow if programmed in a straightforward manner using convolution of the elementary grains with a large sample, which is common for sound files since one minute of CD quality sound requires approximately 5 megabytes per track.

(2) Resynthesis methods: Sampled sound files typically represent sound as discrete amplitude changes over time, and many computer and Digital Signal Processing (DSP) chips play samples using this format and converting to analog values. If we are to convert back into the time domain, we need a resynthesis method or methods to accomplish this.

(3) Musically interesting modifications: While the field of wavelet analysis / resynthesis is still young, the analogy between wavelet analysis and our auditory perception brings hope that we can find modification methods for wavelet coefficients that have perceptual significance and can lead to musically evocative sounds. Since we're using wavelets in the analysis end of the Phase Vocoder, we can draw on

what has been accomplished previously with the Phase Vocoder to affect sound.

We will now focus on these issues, beginning with optimized wavelet and STFT algorithms.

6.1 Filtering Via the FFT

We previously discussed the two primary ways of viewing the STFT; either as a windowed Fourier transform computed at discrete points in time or as a bank of filters centered at discrete points in the frequency domain. Since convolution in one domain of Fourier space is equivalent to multiplication in the other domain, the convolution of the original time series with the windowed complex sinusoids can be accomplished by multiplying the Fourier transform of the time series and the Fourier transform of the windowed complex sinusoids and then taking the inverse Fourier transform of their product. In mathematical form:

$$y(t) = h(t) * x(t) = \sum_{f=0}^{N-1} H(f)X(f)e^{\frac{-2\pi i f n}{N}} \quad \text{for } t = 0 \text{ to } N-1 \tag{6.1}$$

where $H(f)$ and $X(f)$ are the Fourier transforms of $h(t)$ and $x(t)$ respectively. Additionally, windowing the complex sinusoids can be accomplished by convolving the window's Fourier transform with the Fourier transform of the complex sinusoids and taking the inverse Fourier transform of the product. Since the Fourier transform of a pure sinusoid is nonzero for only two points in the frequency plane

(assuming we've measured an integral number of periods, in the case of the DFT), this effectively centers a copy of the window's Fourier transform at these points. As we've seen previously, for cosine waves these points are symmetric around the Nyquist frequency, while for sine waves they are anti-symmetric. Because the Fourier transform of a Gaussian remains a Gaussian, each windowed sinusiodal component of the Gabor and Morlet transforms defines two band-pass filters of Gaussian form centered at these discrete analysis frequencies. The imaginary component is known as an in-quadrature filter, while the real component is known as an in-phase filter. Their Fourier transforms are illustrated in Figures 6.1 and 6.2.

Of course for Gabor transforms these are Constant-Bandwidth while for the Morlet they are Constant-Q. The complex sinusoids of each transform are added together to form their respective elementary functions. The symmetry of the cosine and the anti-symmetry of the sine leads to cancellation below the Nyquist and reinforcement above it, causing the frequency domain filters to consist of only one Guassian centered at the frequency of the sinusoidal component plus the Nyquist rate (see Figure 6.3), with double the magnitude of its component Guassians. Thus both the Gabor and Morlet transforms can be accomplished by taking the transform of the time series, multiplying this by Gaussians centered at harmonic (scale in the case of Morlet) intervals above the Nyquist, and taking the inverse Fourier transform of the product. Utilizing the FFT to accomplish this can greatly decrease the computation time required to compute the coefficients

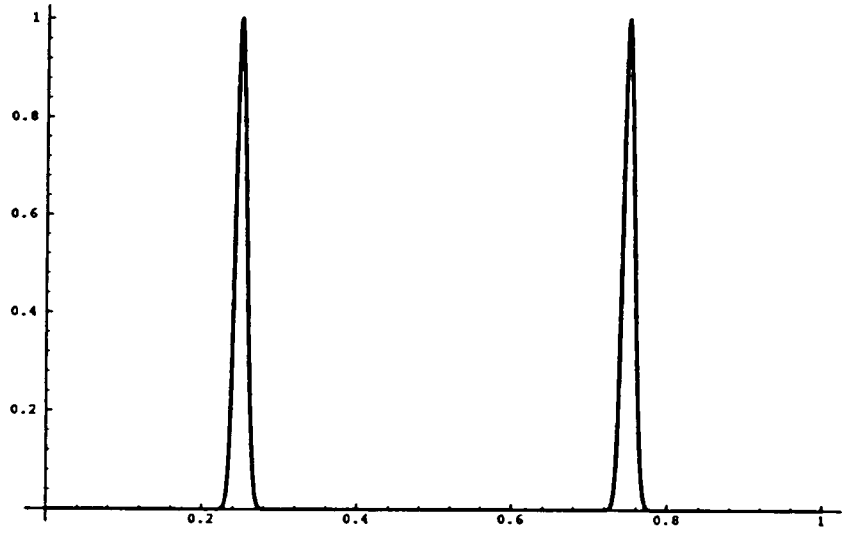

Figure 6.1: The Fourier transform of an in-phase filter. (Source: [15])

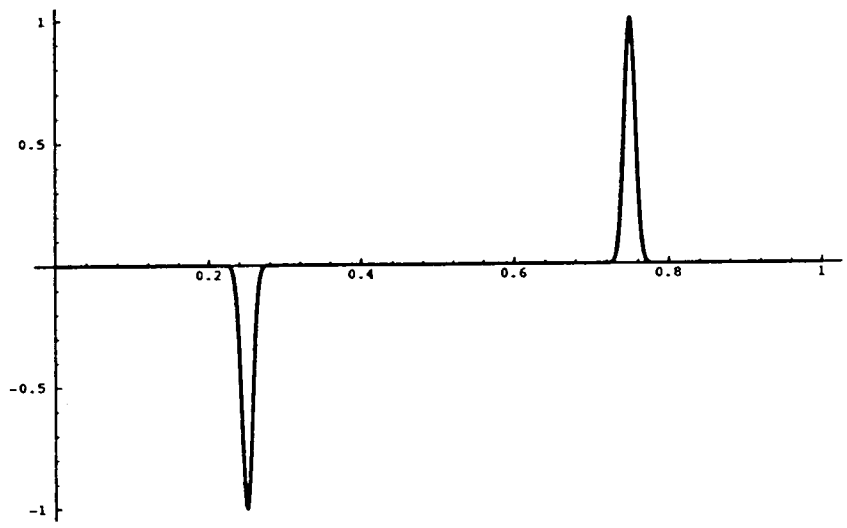

Figure 6.2: The Fourier transform of an in-quadrature filter. (Source: [15])

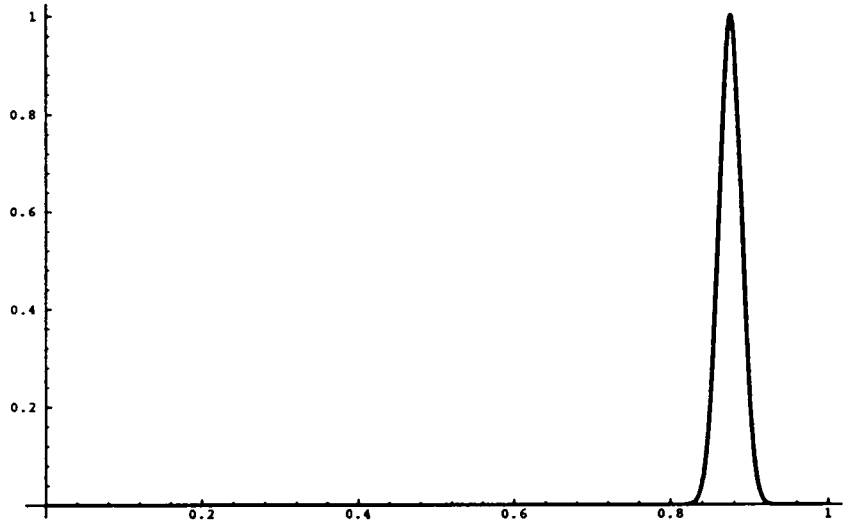

Figure 6.3: The Fourier transform of a Gabor or Morlet filter. (Source: [15])

using convolution in the time domain.

Additional computation time can be saved when we realize that we only need to compute twice as many samples as the current highest frequency being analyzed (i.e. frequencies inside the spread of the frequency-domain Gaussian), according to the Nyquist Theorum. If we analyze each sample point taken, the data size needed to store the coeffiecients would be M times as large as the original data size, where M is the number of frequencies or voices. Additionally we would need to perform M FFTs. While the FFT operates in $n \log_2(n)$ time, a large amount of computation time can still be saved if we're only performing FFTs on half as many points each time our analysis center decreases by an octave. If we need to reconstruct the signal, however, we need to restore the original number of samples before performing the inverse wavelet or STFT transforms by adding zeros where the missing samples are and applying an interpolating filter. While the perfect interpolating filter is a low-pass filter, we've noted that these are not constructible, since they extend from $-\infty$ to $+\infty$ in the time domain. However, Kaiser or Hamming windows can be applied to localize low-pass filters in time and these have led to good results [22].

If we have a good idea of the frequency bandwidth we're interested in, we can save additional storage and computation time. For example, if the signal we're analyzing has a bandwidth of 125 - 500 Hz we only need 1000 samples to adequately sample the highest frequency, 500 samples for the octave below that,

and 250 for the lowest octave. If we've sampled at CD quality, 44.1 kHz, we only need to analyze every $\lceil 44,100/500 \rceil = 89th$ sample point for the highest frequency, half this for the octave below, and half again for the octave below that. Thus we really only need to analyze $89 + 45 + 23 = 157$ samples, for one second of sound.

Because of the implicit wraparound of the DFT and FFT, the end of the signal may become encoded as part of the beginning of the signal using this technique, as it relies on linear convolution. Padding the end of the series with zeros can reduce or eliminate the effects of this. Masters [15] gives intuitive bounds on the number of zeros required to eliminate the problem. Since the FFT commonly uses kernals of size two or greater, requiring zero padding for all samples between the current sample size and the next largest power of two, the number of zeros added in order to utilize the FFT is often enough to reduce the wraparound problem.

6.2 Phase Vocoder

The Phase Vocoder is an analysis / resynthesis method which utilizes the STFT in the analysis phase. It is an extension of the Channel Vocoder designed by Dudley [6] at Bell Labs, and was jointly developed by Flanagan and Golden [7] at Bell Labs. The term vocoder refers to its usage as a voice encoder, whereby a bank of filters could be used to represent speech transmitted across phone lines. It was hoped that by using parallel filter-banks and omitting channels containing neg-

ligable amplitude information, the Channel Vocoder could transmit speech more efficiently than transmitting the original signal. However, the Channel Vocoder did not encode phase information, which led to poor quality in the speech transmitted. The Phase Vocoder was developed to remedy this situation.

The filter-bank approach to the STFT can be defined mathematically as follows, by defining the STFT as (following [22]:

$$F(n) = \frac{1}{N} \sum_{t=-\infty}^{\infty} x(t)h(n-t)e^{-i\omega t} \quad \text{for} \quad k = 0, 1, ..., N-1 \tag{6.2}$$

we can rewrite this, by substituting $n - t$ for t, as:

$$F(n) = \frac{1}{N} \sum_{t=-\infty}^{\infty} h(t)x(n-t)e^{-i\omega(n-t)} = e^{-i\omega n} \frac{1}{N} \sum_{t=-\infty}^{\infty} h(t)x(n-t)e^{-i\omega t} \tag{6.3}$$

By setting $h_k(t) = h(t)e^{i\omega_k n}$, we define the following convolution of the signal $x(t)$ with the impulse response of a set of filters, h_k, modulated by $e^{-i\omega n}$, as:

$$F_k(n) = e^{-i\omega n} \frac{1}{N} \sum_{t=-\infty}^{\infty} h_k(t)x(n-t) \quad \text{for} \quad k = 0, 1, ..., N-1 \tag{6.4}$$

The modulation by $e^{-i\omega_k n}$ can be understood, in view of the Convolution Theorum, as convolving the frequency response of the signal with the frequency response of the complex exponential. For each sinusoidal component at frequency f in the signal, this will lead to a modulated signal at $\omega_k - f$ and $\omega_k + f$, each with half

the amplitude of f, according to:

$$\cos A \cos B = \frac{1}{2}[\cos(A+B) + \cos(A-B) \tag{6.5}$$

Frequencies close to the analysis frequency, ω_k, will be shifted down near zero and up near ω_{2k}. This is illustrated in Figure 6.4. The frequency response of most commonly used STFT window functions has a lowpass characteristic. Therefore, the filtering action of the window will remove or attenuate components in the signal not near the analysis frequency while components near the analysis frequency will remain, since these have been modulated down near zero frequency. To resynthesize the signal, we need to shift the coefficients back up to their original analysis band by modulating by $e^{i\omega_k t}$ and taking the sum over all filters. This is illustrated in Figure 6.5.

As previously mentioned, the number of samples analyzed can be decimated by a factor $D = \lceil \frac{\omega_{max}}{\omega_k} \rceil$, where ω_k is the center of the analysis frequency and $\omega_{max} = \frac{R}{2}$ is the highest frequency allowable for sample rate R if we wish to avoid aliasing. Portnoff's implementation can give perfect reconstruction if both the decimation and interpolation factors are equivalent and if the lowpass filter, $h(n)$, has the following properties:

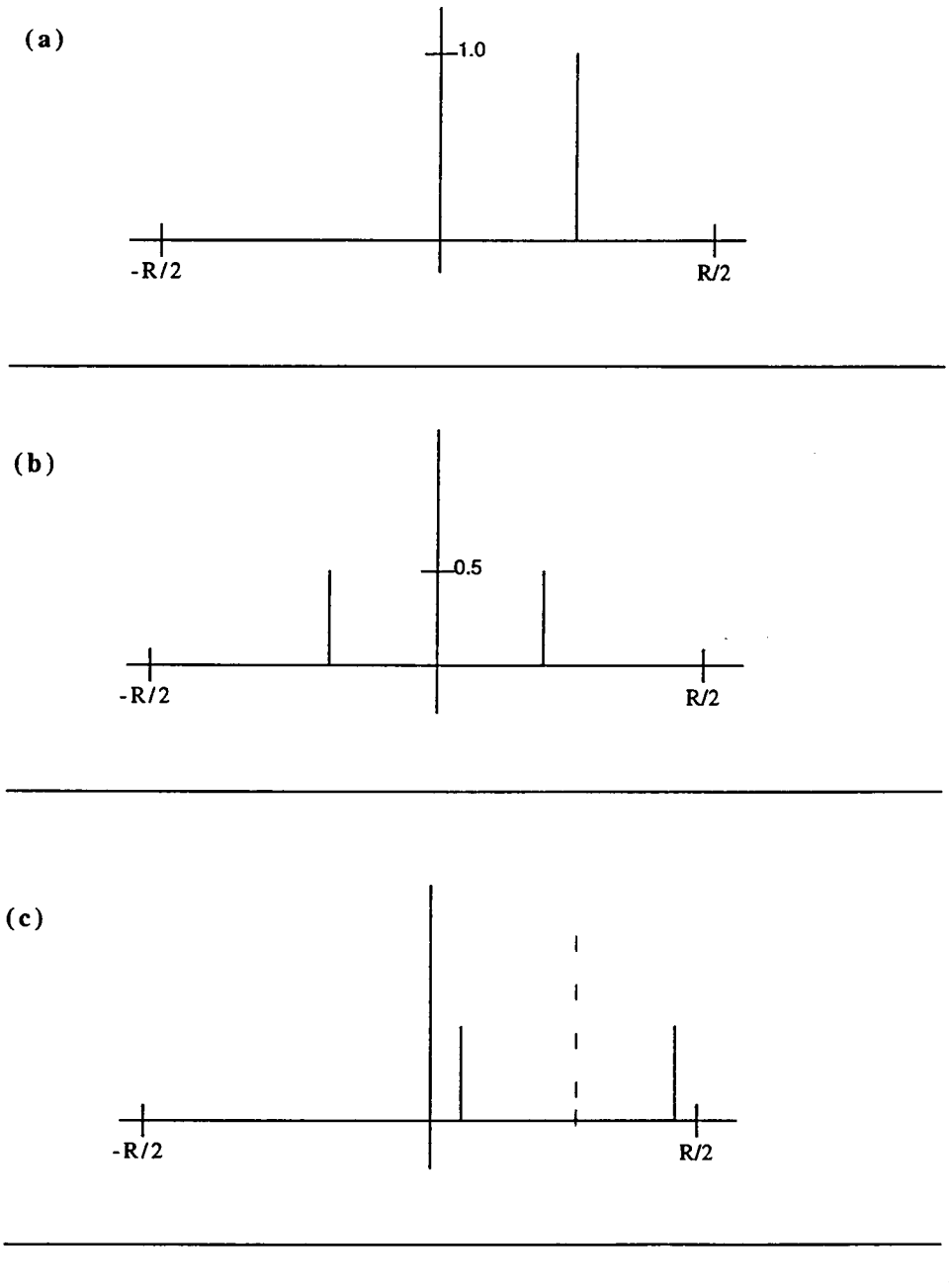

Figure 6.4: Frequency-domain convolution of analyzing Fourier sinusoid with sinusoidal component in signal. **(a)**Sinusoidal component in signal. **(b)**Frequency response of one analysis sinusoid. **(c)** Convolution of the two (solid line).

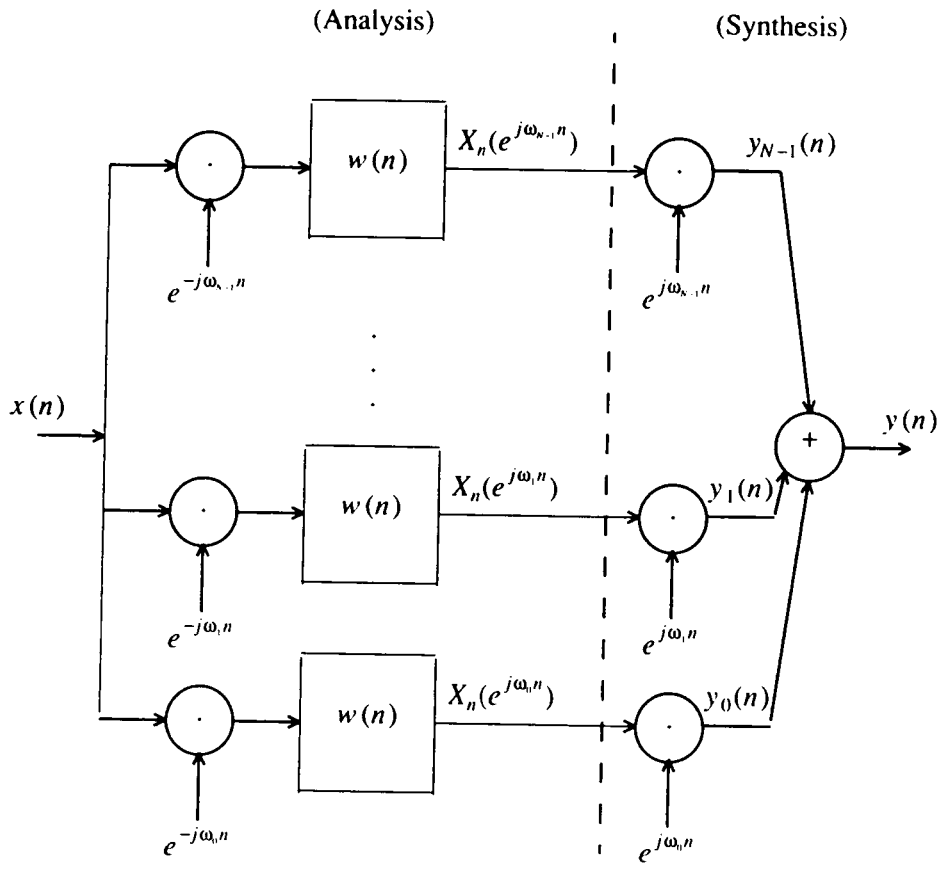

Figure 6.5: Operation of the Phase Vocoder. Each channel modulates the input signal by a unique integer multiple of the fundamental analysis frequency. Window functions ($w(n)$) commonly act as low-pass filters. In the synthesis stage, the filtered components are shifted back to their original frequencies, and their sum is then taken to reconstruct the signal. (Source: [17])

$$\begin{aligned} &(1) \quad h(0) = 1 \\ &(2) \quad h(n) = 0 \text{ for n} = \pm \text{ N}, \pm \text{ 2N}, \pm \text{ 3N}, \ldots \end{aligned} \tag{6.6}$$

where N is the number of frequency bands. These conditions tell us that although the frequency response of any one of these filters is not an ideal bandpass filter, their sum is unity. Portnoff was the first to implement the Phase Vocoder using the computationally efficient FFT, using an overlap-add method described in [22].

6.3 Resynthesis

In Chapter 2, it was pointed out that by simply changing the sign of the complex exponential, we could return from the frequency domain to the time domain. We wish to find a method or methods for reconstruction of the STFT or wavelet coefficients into a time-domain representation. This is commonly referred to as resynthesis, and we've just presented one method for accomplishing this via the Phase Vocoder. There are two other primary methods for performing resynthesis with Morlet wavelets and the STFT. The formula for general wavelets, analagous to the inverse DFT, was introduced in Chapter 5 as:

$$x(t) = \frac{1}{C_g} \int \int W(\tau, a) g_{\tau,a}(t) \frac{d\tau da}{a^2} \tag{6.7}$$

where

$$C_g = 2\pi \int \frac{|\hat{g}(\omega)|^2}{\omega} d\omega \tag{6.8}$$

The analogous formula for the STFT is:

$$x(t) = \frac{1}{2\pi} \int \int F(\tau, \omega_0) h_{\tau,\omega_0}(t)\, d\tau d\omega_0 \tag{6.9}$$

These formulas can be interpreted as a Granular Synthesis technique, whereby the coefficients computed are treated as weights for translated and dilated wavelet elementary functions or translated and modulated STFT elementary functions (grains). This is illustrated in Figure 6.6.

The other resynthesis method commonly used for the STFT is an alternative approach to the final modulation by $e^{i\omega_k t}$ in the Phase Vocoder. If we view the intial modulation of the signal by $e^{-i\omega_k t}$ as two modulations, by $\cos(\omega_k t)$ and $\sin(\omega_k t)$ respectively, then we can use the technique introduced in Chapter 2, whereby each pair of coefficients represent a point on the complex plane, with $a_t(\omega_k)$ referring to the real (cos) coefficient and $b_t(\omega_k)$ referring to the imaginary (sin) coefficient. Then for each filter center, ω_k and each point in time t, we can compute magnitude, or amplitude, as:

$$|F_{t,\omega_k(x)}| = \sqrt{a_t^2(\omega_k) + b_t^2(\omega_k)} \tag{6.10}$$

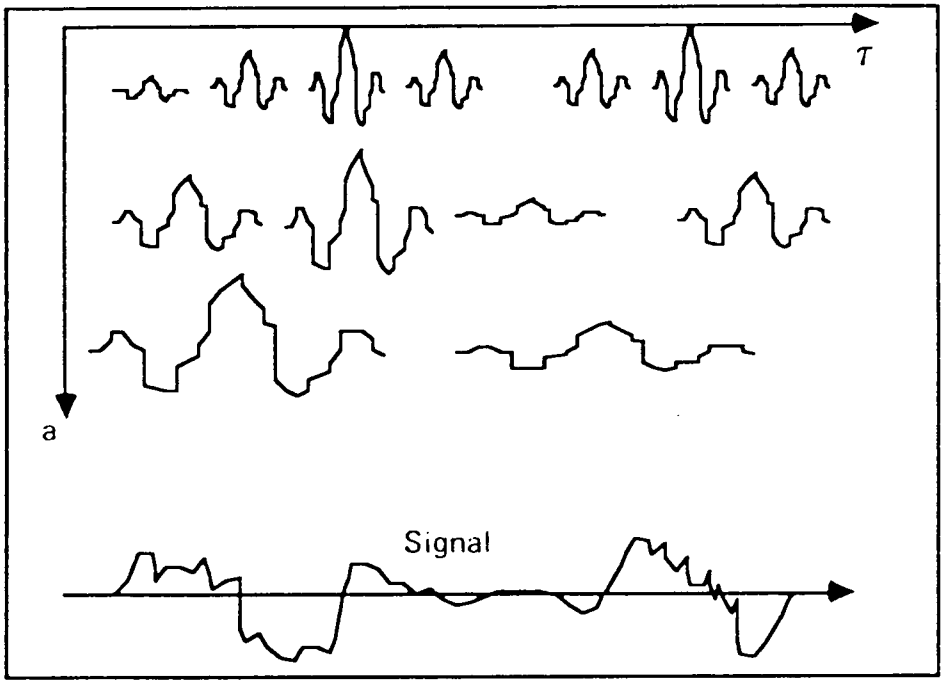

Figure 6.6: Granular resynthesis using dilated and traslated wavelets weighted by coefficients computed by the wavelet transform. (Source: [13])

while the instantaneous phase can be calculated as:

$$\theta_t(\omega_k) = \arctan \frac{b_t(\omega_k)}{a_t(\omega_k)} \tag{6.11}$$

Instantaneous frequency is then the derivative of the phase angle. Since we are dealing here with discrete samples on the unit circle, representing changes in phase, the best we can do is a discrete approximation of the derivative. Moore [17] approximates this by calculating the first-order difference of the phase angles, according to:

$$\Delta\theta_t(\omega_k) = \theta_t(\omega_k) - \theta_{t-D}(\omega_k) \tag{6.12}$$

where D is the decimation factor. Since both the STFT and the Morlet wavelet transform are complex, this will give us a complex reconstruction for each analysis channel, which for the Morlet wavelet transform is:

$$f_{\omega_k}(x) = |F_{t,\omega_k(x)}|e^{i\theta_t(\omega_k)}\frac{1}{\sqrt{a}} \tag{6.13}$$

Since audio signals are real, we only need to consider the real term and can thus reconstruct the signal as:

$$f(x) = \sum_{a=2^j} |F_{t,\omega_k(x)}|cos(\theta_t(\omega_k))\frac{1}{\sqrt{a}} \tag{6.14}$$

6.4 Musical Applications and Results

Now that we've seen how the Phase Vocoder operates in terms of analysis / resynthesis, it's time to consider how we can use this technique for sonic transformation. It was mentioned that we could decimate the number of sample points as we decreased in frequency and still return to the original sample rate by interpolation with a low-pass filter. What if we chose to decimate by the same factor, D, for each bandwidth in the analysis, avoiding channels where this decimation factor would cause us to exceed the Nyquist Frequency for that channel's center? What if, for example, we had originally sampled at 44.1 kHz and decimated all frequencies below 11 kHz by a factor of 2, and either did not analyze above this rate or set all channels above it to zero? We would have skipped over every other sample in our analysis phase. However, we've still taken measurements which will allow us to calculate instantaneous frequency and amplitude at the points analyzed. If we now use this data to reconstruct the signal according to equation 6.14, and play this back at the original 44.1 kHz sample rate, the sound will evolve over half the amount of time as the original, but will have the same frequency. The Phase Vocoder thus allows us to disassociate frequency and time in such a way that we can alter one without affecting the other. Note that if we had attempted to perform this by merely removing every other sample in the time domain, we would have increased the frequency by a factor of 2. By the same token, we can

interpolate during resynthesis by a factor $I > D$, by padding the coefficients with zeroed samples and applying a lowpass interpolating filter. In this case, we can use the Phase Vocoder as a sonic microscope to study and hear the evolution of a sound over a much larger period of time. James Moorer has used this technique to expand sounds by factors of over 100 [18]. This can be very useful in studying the complex spectral changes occurring during the attack of an instrument, such as a violin, with strong harmonic overtones. My implementation of time expansion and compression using Morlet wavelets in the analysis phase did not give audible results as good as I've heard where the STFT was used. Daniel Arfib [1] notes that time-warping with wavelets is problematic but does not elaborate upon this. My belief is that the wavelet's triangle of influence in the time-domain which reveals phase discontinuities the STFT does not detect because of the smearing effect of the window, leads to sharper, more objectionable, alterations when the time-domain is subsequently deformed.

Equation 6.14 reveals the simplicity with which we can perform frequency-shifting without alteration in time. Multiplying θ_t by any real number or integer, n, will cause the cos function to traverse the unit circle n times per second. If $n = 2$ for example, the frequency will be shifted up by one octave, while for $n = 0.5$, the frequency will be shifted down an octave. I've achieved excellent results with this procedure when using sounds without strong resonances, such as the sound of a water droplet. In this regard it should be mentioned that since the Phase Vocoder

is recording the energy of the signal in all of the bands analyzed, it's not truly measuring pitch but rather timbre. Therefore, if we shift all channels by the same amount, this will result in a timbral shift. This can lead to unwanted effects, since voices and instruments are being shifted out of their natural range, leading to the impression of a shrinking or expansion of the physical size of the instrument. Since our ears are selectively attuned to the characteristics of the human voice, this can be especially pronounced. It's generally considered that shifting the human voice down by more than a major third will lead to a mechanistic sound. Moore [17] suggests retaining the amplitudes in the channel you're shifting to. However, this can also be problematic since we may not know if the sound has any energy at the frequency we're shifting to.

Another problem associated with the Phase Vocoder is the possibility of analyzing more than one sound's harmonics within the same analysis band, when working with polyphonic input. It would be impossible then to manipulate the sounds independently and frequency shifts could lead to the possibility of creating inharmonic overtones. However, in the Twentieth-century, inharmonic overtones are not often frowned upon, and sounds created by this technique may be musically interesting. The wider frequency bands used by wavelets in the high frequencies can make this more likely to happen. This can be alleviated to some extent by using multiple voices per octave. Although I've had excellent success in resynthesizing the original sound with only one voice per octave, three to four

voices per octave has given much better results in the area of frequency shifting. Not only do the additional voices help prevent multiple instrument overtones, or perhaps more than one overtone from the same instrument, from falling into the same frequency band, they also give a more accurate measurement of frequency, since the center frequencies are likely to be less disposed from the signal's energy falling in that band. The amount of the modulation need not be a constant, but could be time varying. I've implemented ramp functions which allow the user to define beginning and end times for linear functions which modulate the frequency, causing it to rise or fall over the given interval of time. I've also included sinusoid modulation functions. If the frequencies for these are set low, this will produce vibrato, or slight pitch inflections, over the time interval and channels to which it is applied. This can be musically useful since vibrato is often employed, especially by vocalists, to vary pitch while sustaining a note so that it doesn't become redundant.

Since the sum of the magnitudes of all the Phase Vocoder channels can be understood as representing the timbre of the sound, we may wonder then how the frequency measurements relate to this. For resonant sounds the Phase Vocoder functions in a similar fashion to Linear Prediction. Linear Prediction is a technique for separating the excitation source and response characteristics of a signal. This makes sense for studying, or simulating, the human voice since it is a clear example of a source / response system, with the vocal cords acting as a source and the

skull cavity supplying the resonance. Linear Prediction works by creating a set of filters that act as the resonant response. Since these filters define the overall shape of the resonant response, they are making measurements more closely akin to pitch than timbre. For this reason, Linear Prediction typically performs better for frequency shifting than the Phase Vocoder. However, if there is no clear source / response relationship in the sound, such as with many percussive instruments, this technique is not very effective. While the STFT can perform somewhat better on non-pitched sounds, it also has difficulty because of its implicit measurements of the signal as a harmonic overtone series. wavelets are better adapted to this situation since they aren't performing a harmonic analysis and because of their better localization in the time-domain.

If we have analyzed two sounds with a clear source / response relationship, we can then perform what is known as Cross Synthesis, where one sound source is used as the excitation for the other sound's resonant response. This leads to effects such as talking guitars, using the frequency of the guitar and the resonant response of the voice. This technique takes a good deal of acoustic knowledge and trial-and-error to attain good results, since many sounds cannot be modeled well as source / response systems. In particular, the source sound needs to be free of strong resonances. Non-pitched percussion seems to work well as the excitation source, as do ocean waves, vocals, noise-based sources, and stringed instruments where the vibrating string provides the excitation source and the instrument body

produces resonances. I've gotten very interesting musical results from combining voice / electric quitar, ocean waves / voice, voice / violin, cymbals / cello, where the first of the pair was the excitation source. Some combinations that did not work as well were chaotic oscillators / voice (although this was somewhat eerie and might have worked in a horror film) and electric guitar / voice.

As was done with the frequency coefficients, we can apply constant or time-varying functions to the amplitude coefficients. If we modulate a single channel by an amount $\Delta\omega_k > 1$ this will cause that frequency band to be emphasized while modulation by $\Delta\omega_k < 1$ will attenuate the frequency band. By selectively choosing certain channels to emphasize and setting others to zero, or negligable frequencies, we can approximate lowpass, highpass, bandpass, and bandreject filters. If the modulatory functions are time-varying we can achieve results similar to Subtractive Synthesis. I've implemented these techniques with both constant and time-varying modulation, the latter utilizing ramp functions and sinusoids. By starting ramp functions at zero amplitude and having them rise slowly to one, we can achieve the effect of the sound fading in. Applying this technique, but offsetting the initial increase in the ramp begin time for each channel, we can have the sound's frequencies evolve over different time periods. By utilizing multiple breakpoints for the ramps in each channel, we can achieve very complex filtering of the sound over time.

Chapter 7

Conclusions

My research has been primarily application oriented, with the focus on implementing a general-purpose sound transformation environment for a Power Macintosh including sampling, Morlet wavelet-based analysis, temporal / spectral modifications, and resynthesis using the Phase Vocoder approach. These modified sounds can then be saved to AIFF files for sound playback. The system works well and can accomplish interesting transformations in a reasonable amount of time, although it does not currently operate in realtime, either for analysis or resynthesis. I'm currently somewhat limited by amount of RAM, 8 megabytes, with regard to transforming sounds larger than approximately 20 seconds, or applying time shifts larger than 10 or 12 times the original length for short sounds of 3-4 seconds.

7.1 Comparison of Analysis / Resynthesis Techniques

The Phase Vocoder, utilizing either the STFT or Morlet wavelet transform in the analysis phase, offers a very flexible analysis / resynthesis system, capable of combining some of the best features of Electronic Music, Musique Concrete, and Computer Music. However, as with all sound modification techniques, it has its limitations. Linear Prediction seems to perform better for pitch shifting and Cross Synthesis, and is also far more efficient, commonly requiring only a few filter coefficients and a small excitation source. Linear Prediction does not, however, seem as flexible as the Phase Vocoder. While Morlet wavelets are more efficient, seem to model the acoustic system better, and are better at representing nonpitched sounds, nevertheless, the Phase Vocoder appears to perform frequency-shifting and time-shifting better when using the STFT in the analysis phase. However, the efficiency required to implement this in a reasonable amount of time on a personal computer necessitates the use of wavelets rather than the STFT, if we're interested in analyzing the entire audio bandwidth. The field of wavelets is relatively new and additional sound modification techniques may be found which allow time and frequency shifting as good or better than the STFT allows.

7.2 Future Directions

Some additional features which I hope to implement in the future include:

(1) Faster resynthesis. While the time required for resynthesis is not excessive at the present, it could be shortened by using multiple channels of sound and the built-in wavetable routines provided by the Macintosh Sound Manager. This would require the need for synchronization of the sound channels so that they all started simultaneously, using the Macintosh Timing Manager. However, this limits the number of analysis channels we can have, since the Sound Manager will only allow allocation of a given number of channels. A compromise could be made between this technique and the present technique of adding the components of all frequency bands, by adding a limited number of frequency bands, storing in a buffer, and passing this to a sound channel allocated for wavetable synthesis. The user could be given the choice of this method or saving to a sound file for later playback.

(2) Enhanced control over sound modification. By enhancing the current graphical user interface to include graphing of the sampled sound and each channel of the analysis coefficients, I would like to then provide for cut, copy, and paste ability for moving parts of sound around or deleting them, as well as the ability to set loop points. Since we could set separate loop points for each channel's coefficients, this could allow the creation of very complex rhythmic patterns occuring in a single

sound. I would also like to provide the user with a graphing tablet with which they could draw arbitrary functions, using these to modify the amplitude and frequency components.

(3) Crossfading Cross Synthesis. The difficulty in finding sounds which work well together for Cross Synthesis has been noted. I hope to implement a technique that would allow for the combination of two or more sound's frequency components with one sound's amplitude components, or vice-versa, using linear interpolation to vary the influence of each sound's frequency components over time. By using the original sound's frequency as part of the overall frequency spectrum, this may allow a less drastic alteration of the sound. We could also use this to crossfade from the original sound to the Cross Synthesized version.

Bibliography

Bibliography

[1] D. Arfib. Analysis, transformation, and resynthesis of musical sounds with the help of a time-frequency representation. In Giovanni De Poli, Aldo Piccialli, and Curtis Roads, editors, *Representations of Musical Signals*, chapter 3, pages 87–118. MIT Press, Cambridge, MA, 1991.

[2] Richard E. Berg and David G. Stork. *The Physics of Sound.* Prentice Hall, Englewood Cliffs, NJ, 1982.

[3] Ingrid Daubechies. The wavelet transform, time-frequency localization, and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, 1990.

[4] Ingrid Daubechies. *Ten Lectures on Wavelets.* Philadelphia Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[5] J. G. Daugman. Spatial visual channels in the fourier plane. *Vision Research*, 24:891–910, 1984.

[6] H. Dudley. The vocoder. *Bell Labs Record*, 17:122–126, 1939.

[7] J. L. Flanagan and R. M. Golden. Phase vocoder. *Bell Labs Technical Journal*, 45:1493–1509, 1966.

[8] Dennis Gabor. Theory of communication. *Journal of the Institution of Electrical Engineers*, 93(3):429–457, 1946.

[9] Dennis Gabor. Acoustical quanta and the theory of hearing. *Nature*, 159(4044):591–594, 1947.

[10] A. Grossmann, R. Kronland-Martinet, and J. Morlet. Reading and understanding continuous wavelet transforms. In J. M. Combes, A. Grossmann, and Ph. Tchamitchian, editors, *Wavelets: Time-Frequency Methods and Phase Space*, chapter 1, pages 2–20. Springer-Verlag, Berlin, 1987.

[11] Andrew Horner, James Beauchamp, and Lippold Haken. Methods for multiple wavetable synthesis of musical instrument tones. *Journal of the Audio Engineering Society*, 41(5):336–356, 1993.

[12] Aldous Huxley. *The doors of perception.* Harper, New York, 1970.

[13] R. Kronland-Martinet and A. Grossmann. Application of time-frequency and time-scale methods (wavelet transforms) to the analysis, synthesis, and transformation of natural sounds. In Giovanni De Poli, Aldo Piccialli, and

Curtis Roads, editors, *Representations of Musical Signals*, chapter 2, pages 45–86. MIT Press, Cambridge, MA, 1991.

[14] Bruce MacLennan. Gabor Representations of Spatiotemporal Visual Images. Technical Report ut–cs–91–144, University of Tennessee, Knoxville, TN, September 1991.

[15] Timothy Masters. *Signal and Image Processing With Neural Networks.* John Wiley and Sons, New York, 1995.

[16] F. R. Moore. An introduction to the mathematics of digital signal processing. In John Strawn, editor, *Digital Audio Signal Processing*, chapter 1, pages 1–67. William Kaufmann, Los Altos, CA, 1985.

[17] F. Richard Moore. *Elements of Computer Music.* Prentice Hall, Englewood Cliffs, NJ, 1990.

[18] James A. Moorer. The use of the phase vocoder in computer music applications. *Journal of the Audio Engineering Society*, 24:717–727, 1978.

[19] T. L. Petersen. Spiral synthesis. In John Strawn, editor, *Digital Audio Signal Processing*, chapter 3, pages 137–147. William Kaufmann, Los Altos, CA, 1985.

[20] Tracy L. Petersen and Steven F. Boll. Critical band analysis-synthesis. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 31(3):656–663, 1983.

[21] Ken C. Pohlmann. *Principles of Digital Audio.* Howard W. Sams, Indianapolis, 1985.

[22] Michael R. Portnoff. Implementation of the digital phase vocoder using the fast fourier transform. *IEEE Proceedings on Acoustics, Speech, and Signal Processing*, 24:243–248, 1976.

[23] Karl Pribram. *Languages of the Brain.* Wadsworth Publishing, Monterey, CA, 1977.

[24] Karl Pribram. *Brain and Perception: Holonomy and Structure in Figural Processing.* Lawrence Erlbaum, Hillsdale, NJ, 1991.

[25] Barry Truax. Real-time granular synthesis with a digital signal processor. *Computer Music Journal*, 12(2):14–26, 1988.

[26] Iannis Xenakis. *Formalized Music: Thought and Mathematics in Composition.* Indiana University Press, Bloomington, IN, 1971.

vita

Vita

Michael Dean Christian was born in Charlottesville, Virginia on January 18, 1964. He lived in Bluefield, Virginia before graduating from Graham High School in June, 1982. He received a Bachelor of Arts in Music from the University of Tennessee, Knoxville in December, 1991, where his primary emphasis was in the field of Electronic Music Composition. He entered the Master of Science program in Computer Science at UT Knoxville in May, 1994, and received the Master of Science in Computer Science in August, 1996. During his time in the Masters program, he served as a Graduate Teaching Assistant in introductory computer science, two courses in the area of Neural Networks, and a graduate course in computer systems organization.