Rhythm Analyzer

A Technical Look at
Swing Rhythm
in Music

by

KENNETH ALAN LINDSAY

A THESIS

Presented to the Department of Computer Science in partial fulfillment
of the requirements for the degree of

Master of Science in Mathematics and Computer Science

Ashland, Oregon

June, 2006

APPROVAL PAGE

Rhythm Analyzer: A Technical Look at Swing Rhythm in Music

A Thesis prepared by Kenneth Alan Lindsay in partial fulfillment for the degree of Masters of Science in Mathematics and Computer Science.

This thesis has been approved and accepted by:

Pete Nordquist, Thesis Advisor	Date
Dr. Curtis Feist, Math Advisor	Date
Todd Barton, Music Advisor	Date
Dr. Lynn Ackler, DSP Advisor	Date
Dr. Joseph L. Graf, Dean of Sciences	Date

Dedication

It don't mean a thing, if it ain't got that swing.

-- ***Duke Ellington and Irving Mills***

Acknowledgments

Thanks to my Brasilian music teachers and friends:

Jorge Alabe, who got more rhythm out of one surdo than I'd ever heard in one place in my life, one sunday afternoon in Oakland, years ago. Jorge also invited me down to New Orleans to play with Casa Samba in the Heritage Jazz Festival even though I was pretty much a beginner. Being onstage in New Orleans, even in the back row, is a great thrill.

Mestre Beiçola, who always brings his love of life out in his performances, and gave me the experience of performing with Imperatriz Leopoldinense, the Number One Escola de Samba in Rio de Janeiro in the millennium year 2000.

Kim Atkinson for triggering the idea about focusing on swing feeling.

Mestre Biquinho, Jim Fitzgerald, Carlinhos Pandeiro de Oura, Curtis Pierre, Boca Rum, Mark Lamson, Carlos Aceituno, Marcio de Ile Aye, Gamo de Paz, Naoyuki Sawada, Jacare, Guello, Carlos Oliveira, John Santos, Justino.

Thanks to the technicals:

Pete Nordquist, who always had fun with this project, and was majorly enthusiastic.
Curtis Feist, who helped me overcome math anxiety.
Todd Barton, and Terry Longshore, for crucial insight into music.
Lynn Ackler, for DSP tutoring and such like.
Dr. Kemble Yates, he didn't make differential equations and numerical analysis easy, but he did make them understandable.
Dr. Dean Ayers, his domain knowledge greatly improved my treatment of audiology.
H.H. XIVth Dalai Lama, I think my writing style improved greatly after I read some of his books.
Clayton Press, who introduced me to ***Stir it up*** by Bob Marley and opened up some significant doors for my musical thinking.
Dr. Steve Bryson, whose cogent comments on the rough draft enabled me to eliminate many obscurations, obfuscations and errors, and elucidate my explanations more clearly.

Rhythm Analyzer
A Technical Look at Swing Rhythm in Music

by

Kenneth Alan Lindsay

ABSTRACT

We investigate the nature of swing rhythm in music by using computer analysis techniques. Swing is not a genre of music, but rather a style of performance. The same musical piece (data) can be played in swing or straight time. The musical notes and structure would be identical in both performances, but the notes' temporal patterns have slight, significant differences between straight and swing performances. We demonstrate a technical approach for analyzing these differences, and show examples of several styles of swing, including American Swing, Brasilian Samba, and Jamaican Reggae. Compared to American swing, Brasilian swing, or *swingee* as Brasilians call it, shows significantly more complex patterns of timing variations.

Unlike much of the work in the computer industry, computer music rarely strays far from the human experience. As such, it is a useful bridge between the purely technical and the purely human, especially emotional response. In particular, we enjoy swing music because its basic nature is an expression of fun and enjoyment of life.

Table of Contents

CHAPTER 1. INTRODUCTION

1.1 Purpose and Research Strategy

The purpose of the current work is to produce useful techniques for extracting and recognizing certain features in music. Our primary interest is in rhythm, distinguishing amongst various types of percussive note events, and characterizing these feature sets for determining what makes some music have a swing feeling, while other music does not.

1.1.1 Time and Frequency

Sound is typically described in terms of time and frequency. The human auditory system, like laboratory or recording studio devices, is stimulated by vibrations of air molecules against some type of transducer (eardrum, microphone) which converts the air vibrations into another form, such as electrical signals in circuitry, or nerve impulses, which are both electrical and chemical in nature. The patterns in circuitry are easily analyzed using Digital Signal Processing (DSP) techniques, and this information is a useful framework for understanding the details that gives the swing feel to music.

Information in the time domain (e.g. input audio stream) can be converted to information in the frequency domain (pitches, or tones). An event with a one millisecond (1/1000 of a second) repetition rate corresponds to a frequency of one thousand cycles per second (1000 Hz, or 1 Khz). The range of frequencies audible to the human ear is approximately 20 Hz to 20,000 Hz (20 KHz). Much of the information of interest for music and speech is in the range of 100 Hz to 5000 Hz.

A set of frequencies derived from a time domain input stream is called the *spectrum* of the input data. The time and frequency forms of information are mathematically equivalent, but often one form is more convenient than the other to use for a particular

purpose. The spectrum is closely related to how we perceive sounds. This is explored in chapter 4.

1.2 Musical Audio Events

Most popular music can be broadly described in terms of rhythm and pitch, which are encapsulated in *musical events*. Rhythm is the temporal relationships of musical events. Pitch is the simple frequency content of these events. Timbre is a complex variant of pitch, and is used to describe the *qualities* of the sound, enabling distinction between trumpet and piano for example, even if they play the same musical note, or pitch.

Not all changes in music are adequately described in terms of separate events. Many forms of music have important features that change smoothly from one set of frequencies to another, or that smoothly modulate the loudness or pitch of a note. These changes are often subtle and more difficult to analyze than sharp percussive events. We have focussed on recognizing and analyzing distinct percussive note events, but our techniques can be extended for analyzing these more subtle musical patterns. We believe these subtle changes are highly correlated with human emotional response to music and consider this an important area for future research.

For the current work we recognize musical events in terms of rapid changes (faster than fifty milliseconds) of pitch and power level[1]. These changes are generally complex rather than simple. They are derived either from broad portions of the frequency spectrum, or specific subsets of correlated frequencies.

1.2.1 Note Identification

A musical note event is characterized by a rapid power change (as short as one millisecond) in a set of frequencies, called ***onset*** or ***attack,*** followed by a longer period of mostly steady frequencies, generally called ***decay-sustain-release*** (DSR) collectively.

---

[1] Power level is essentially the same as amplitude for the purpose of this thesis. The typical definition of power is amplitude squared (amplitude times amplitude). The shape of the waveform changes somewhat, but the features we are extracting are about the same.

The DSR period for percussion events ranges from tens to hundreds of milliseconds. Onset is often associated with large changes in loudness or power of the audio signal. This large quick change is characteristic of most percussive note events, although there are exceptions, which we discuss later.

We have developed computer algorithms for extracting and identifying a variety of percussive note events. We visually analyze the events representing a musical sample in order to discern temporal relationships between notes. These temporal relationships are the fundamental nature of rhythm.

1.2.2 Rhythm

After several note events are identified in terms of pitch, onset time and duration, they can be represented as a time series and the next level of information extraction, rhythm, can be performed. Once we have an informational representation of the rhythm, we can use it to characterize the ***style*** of the music. In particular, we investigate ***swing*** vs. ***straight*** time. Swing rhythm is found in Jazz, Blues, and many other styles with African roots including Cuban, Brasilian, and Caribbean music. Straight time is typified by some classical European music styles. These are not the only two forms of rhythmic style. Many examples of music exist that have temporal variations that are not well classified as either swing or straight time.

As a convenience, we refer to straight time, especially musical tablature based on the standard European notation, as Mozart-Bach (MB) notation or time. This is not to say that Mozart and Bach did not employ rhythmic variation and expressiveness in their music, but merely that the standardization of pitch and timing notation can be traced to that era in European music (1700's).

(Bengsston, 1987) presents a perspective on the features and limitations of CCMN (Current Common Musical Notation), which is essentially what we are calling MB notation. He observes that learning by notation rather than by experience can impede a young

musician's ability to play authentically, due to the misinterpretation of the printed information which of course is only a guide to the musical data, not to the performance.

We distinguish between *rhythm* which means the temporal data of a set of musical notes (e.g., as written in tablature), and *rhythmic style* which is a form of expression that a human performer may use when playing the musical data. Style is generally indicated by a linguistic comment on the musical score such as *rubato* or *with a swing feel*. This is a form of meta-information meaningful to an educated performer who is familiar with the particular comment and style. Such linguistic comments are essentially useless to some one who is unfamiliar with either the music style or the meaning of the comment.

In chapter 5 we present a variation of standard MB notation that we believe is useful and informative for conveying the feeling of various swing styles, both to skilled musicians and beginners. Performance of music using variations of the temporal patterns as written in MB notation (1/4 notes, 1/8 notes and so on) is called *rhythmic expression*. Recently, computer algorithms have been developed that emulate the temporal variations as played by a human performer, i.e., swing and other rhythmic expression.

1.2.3 Related Work

We review a number of prior works in rhythm processing and music recognition. We also look at psychological research on human perception of music and time. We have discovered that reading old material, even if it is technically weak or obsolete, can be useful for several reasons. First, some of our own currently cherished dogmas about what is important may appear absurd to future researchers similarly to how we may consider the work of earlier researchers to be naive or ignorant. This can help produce an understanding of the evolution of knowledge in a complex technical area like music analysis, and can also facilitate the open mind that is essential for good scientific research.

Sometimes old insights or observations can lead to very useful ideas when put in a modern context. (Strawn, 1985) includes lengthy discussion about whether 12-bit encod-

ing of audio information is adequate for reproducing high fidelity music, and questions if 16-bit encoding is needed or a waste of compute resources. We see this as silly today, but it leads to the idea of analysis of compute costs for pattern recognition work, which may be quite practical using 12-bit audio. Moorer in (Strawn, 1985) includes very strong technical opinions based on the idea of exact frequency sets derived from Fourier Analysis, forgetting apparently that Fourier Analysis is merely a model for data and information, and should not be mistaken for the information *per se.* The frequency analysis strategy of the human ear is quite different from the results generated by Fourier Analysis, and this provides useful ideas for DSP and pattern recognition, like using instantaneous frequency metrics or nonharmonic Fourier series rather than standard Fourier series in the extraction of musical features. Ideas like these can lead to better quality algorithms, lower compute costs for similar results, or both. (Strawn, 1985) also reports that *vocoder* (voice encoder) techniques had improved substantially since the 1960's, primarily because the newer vocoders use phase information from the audio, whereas the earlier vocoders did not. Similarly, although our current work, and most or all of the research literature, ignores phase information available from spectral analysis, we believe that there is much useful potential in this discarded information. There is some evidence for believing the human ear takes advantage of phase information in separating and distinguishing information that comes from difference sources, even though they are completely blended in the input stream.

1.2.4 Psychology and Perception

Music is fundamentally a human experience. Physicists and psychologists have studied human perception of music since the 19th century. There is evidence for an underlying commonality of temporal perception in humans, and spontaneous production of rhythmic patterns that starts in early childhood. This work is useful in demonstrating, for example, that most commonly used tempos in music are within a temporal range that exists at a low level in the human perceptual apparatus, independent of music itself (Fraisse,

1982). Eventually this sort of research may shed light on human perception at the neurological level, working from the cognitive levels outwards toward the pattern recognition and data collection systems of the audio cortex. We think that percussion and rhythm sounds provide a simple and tractable approach to mapping the human auditory system from the outside in, much as flashing dots on a computer screen have proved useful for mapping the human visual cortex. In the 21st century, this is a practical field of research.

An important aspect of music that distinguishes it from other human symbol systems such as language, is the close connection between emotional response and the perception of musical performance. While it can be reduced to symbolic notation on a page, the essence of music is found in expressive live performances and perception of these. Much modern pop music is produced using robotic sequencing software, but this music does not evoke the complex emotional responses from the human information system that music performed by human beings does. The motion picture industry has a highly developed infrastructure to support the production of music that evokes these important and often subconscious viewer reactions. Rhythmic expressiveness is an important aspect of such music, and our technical analysis of rhythm might be used to help quantify features that are correlated with different emotional responses.

Emotions are not generally considered a part of computer science, but of course they are an important part of the human information system. To this end, our work focuses on immediately practical techniques such as those that can let a computer serve as a technical tutor for humans to better learn, play and understand rhythmic complexities and subtleties, and thus better enjoy music. Music is sometimes used as part of medical therapy for emotional and psychological issues. Emotional well being (or lack of it) is a multi billion dollar industry in the modern world, and we suggest that learning to play and appreciate music better is a useful alternative to pills and therapy.

1.2.5 Computer Science in Music

Computers are used for many purposes related to music, but we focus primarily on two: computer analysis of music, and computer production of music. Practical production of music preceded detailed analysis techniques, but both are fairly mature now. We look at the state of the art in computer production of swing feeling in music, which is a feature available in some commercial software products. We present our research and results for computer analysis of music, and explore immediately practical applications to music production software.

The two information science techniques most commonly used for music analysis are DSP (digital signal processing) and pattern recognition. Various DSP techniques can render a stream of raw audio data (e.g., a format like CD audio, which is one or more channels of 16 bit integer data points sampled at 44,100 points/second) into a different format, such as a frequency spectrum, which is more useful for a particular purpose. We typically use the frequency spectrum of a music sample to identify note events from specific instruments. Pattern recognition, like DSP, is a field with many techniques. We currently use a few fairly simple pattern recognition techniques that are adequate for the current work, and we have also investigated more advanced techniques such as neural nets and statistical analysis.

The purpose of computer analysis of music can be conceptualized hierarchically. Starting with real world signals (musical audio data) we want to extract *information, knowledge, and understanding* about what is contained in the data. As humans we perform this parsing more or less automatically, but to create an information hierarchy in a computer we need to explicitly perform the data manipulation tasks. A typical scenario is that data is processed by DSP to extract information features, such as the frequencies found in the signal, temporal changes of power and frequency, phase relationships amongst the frequencies and so on. These information features are used to extract knowledge about the piece of music, e.g., what is the rhythm conveyed in a sequence of beats

by a particular percussion instrument, what is the fundamental pitch of the note played by the trumpet, how do the trumpet's overtones combine to produce the timbre or quality of sound (e.g. smooth and mellow, bright, punchy etc). The information features can be combined to generate a framework for knowledge about the music such as where are the main beats, what are the relationships of major and minor beats. Other researchers (Guoyon, Klapuri, Tzanetakis et al.) have used this sort of information to determine the meter, key, and time signature. Finally we can use this knowledge to answer the crucial question: *does this piece of music swing, or is it square?* Duke Ellington and other musical experts have expressed the notion that this is where the *meaning* resides. Meaning and *understanding* are higher level abstractions in the data, information, knowledge framework. Please note that these are not intended as rigid categories, but are merely a model for human conceptualization of music. This fuzzy classification methodology is often used in information science and artificial intelligence to describe knowledge of a system at various levels of abstraction.

While computers have been used for music production since the 1960's, their utility for recognizing patterns in music was not very practical until the 1990's. Some pattern recognition work was done in the 1970's and 1980's, but was limited to research labs. The development of digital music as a common form of distribution has led to great interest in automating the recognition of musical patterns for the practical purposes of searching musical databases using a symbol system appropriate to the salient native elements in music, and marketing of music based on similarity metrics accessible by such techniques. The most prevalent use of computers in music is the machinery which transforms digital data to acoustic form for listeners. In February 2006, Apple Computer celebrated selling one billion songs from its iTunes online music store. We report this as tangential information relevant to the current work, since people listening to music is what drives the expanding computer music market in all its forms.

1.3 Cultural Background: Swing vs. Straight Time

Many musicians, when asked about swing in music, will initially indicate that it is a ***feeling*** and follow this by a more technical detail such as "triplet eighth notes"[2] or "six against four rhythm"[3] or "there are as many kinds of swing as there are drummers"[4] and so forth. The underlying similarity is that swing music produces a different physical and emotional response in many people than do straight time performances. Swing is a desirable feature in music, indicated by the popularity of this style in a wide variety of musical genres during the 20th century. (Gabrielsson, 2000) reports that in his research listeners prefer music which has rhythmic expressiveness such as swing, and that they often react negatively to rhythms with completely uniform timing.

We use a human rather than technical specification for swing: it is a property of musical performance that induces a more or less energetic rhythmic motion in listeners. This can be foot tapping, dancing, bouncing or swaying while seated or standing, or other participatory behavior. Both computer science and psychology researchers commonly use such a metric to define swing. We believe that this effect is an ancient piece of the human condition, and may predate the emergence of hominids. Geese for example, synchronize their wing flapping when flying in formation, as do horses running in an orderly herd (not a stampede). These can easily be analyzed in terms of system optimization. These synchronization effects, sometimes called *entrainment,* are similar to a group of musicians synchronizing to a leader, such as the drum master in Brasilian batucada.[5]

Another good metric for testing if a music sample swings is to make an audio loop of a short section of the piece and play the loop endlessly in a player like Quicktime. If listening to the loop becomes tedious, or begins to sound mechanical or repetitive after

---

[2] Chris Wood, professional musician, founder of "Samba Like it Hot!" bateria. Ashland, OR USA.

[3] Shawn Moore, musician. Ashland, OR.

[4] Statements similar to this are made by almost every musician surveyed about swing

[5] Todd Barton, professional composer and musician. Oregon Shakespeare Festival and SOU. Ashland, OR.

only a few repetitions, then it probably doesn't swing much. We found that we could listen to many of the analyzed loops (which have a good swing feeling) repeatedly and they did not become tiresome. We did not conduct any extensive survey of many listeners as a psychology researcher might do, but we believe this observation about tedium vs interest is well contained in the mainstream of music research (Gabrielsson, 2000).

Having a working definition of swing, we now ask the question, *where does swing come from?* Listening to the timing of a horse canter or human walking gives a good perceptual insight into swing: it is rooted in motion itself. Being found in animal motion other than human leads to the conclusion that swing predates language. A recording of my dog running shows strong resemblance to Brasilian pandeiro rhythm. Sounds generated by the motion of a vehicle such as a streetcar or railroad show how synchronized polyrhythms emerge as a natural result of the bouncing of the vehicle, and the asymmetric nature of the patterns suggests an origin of the swing style. These vehicles can be regarded as information systems, as surely as a database server is. Mathematical modeling of such dynamical systems is explored in chapter 6.

As modern music came to be dominated by sequencers and robotic rhythms like house or rave music, young listeners have not learned about temporal variation in the way that someone like Louis Armstrong may have learned it, riding on the New Orleans streetcars with their rhythmic but imprecise clackety sounds. This is a cultural loss, and our work shows how computers can be used to ameliorate this deficiency. We can provide technical feedback about the temporal patterns of swing. Mathematical models can generate timing variations for rhythmic modification in music production. These can be used by both music teachers and students to facilitate learning about swing.

1.3.1 Notes Inegales

European music has had temporal variations in music performance for many centuries. A style from the 17th and 18th centuries was called *Notes Inegales*, meaning une-

qual notes (i.e. note timings) in French. This belongs to the general category of rhythmic expressiveness in music. It is not clear whether the influence of this style had any direct effect on the development of modern swing such as American Jazz and Blues. Brasil had a strong influx of European music in the early 19th century because the Portuguese Emperor and his Court moved to Brasil when Napoleon invaded Portugal[6]. Certainly one can find strong influences from the European tradition that came down in Brasilian folk and formal music traditions. The main influences in Brasilian music are rooted in the African traditions, but there is also blending between European and African styles.

1.3.2 American, Brasilian and Other Types of Swing

Traditional American Swing is generally quick tempo and energetic, but we use a broader definition to easily include Samba, Reggae and other styles: swing is that quality in rhythmic performance that causes people to move with the music, whether consciously or unconsciously. The motivation is not to try to precisely categorize musical style, but rather to lay a foundation for finding similarities between music from different cultures, so a listener of one type of swing might find and enjoy other types, e.g., when searching a music database using a swing criterion. There is also the important phenomenon of syncretism of different cultures or traditions which synthesizes a new style by combining aspects of two or more extant styles. Samba-Reggae, Soukous and Hi-Life are popular styles that have evolved from combinations of other forms. We expect much new development of this sort of music in the 21st century as global travel and internet access broadens exposure to other cultures. Our work is useful for documenting similarities and differences between various types of swing style.

The ***swing ratio*** is a measure of slight but consistent time differences between pairs of successive "evenly" spaced note events as written in tablature. Use of this term goes back at least to (Cholakis, 1995), and has been investigated by (Anders & Sund-

---

[6] Harvey Weinappel, professional musician and musicologist, Los Angeles, CA USA.

strom, 1999), (Anders & Sundstrom, 2002) and (Birch, 2003). The ratio is obtained for a particular musical sample by statistical analysis of the patterns of "long" to "short" notes. This simple concept is very useful for analyzing American Swing and Jazz.

For example, given a drum score with a series of 1/8th notes on the cymbal, rather than play all notes evenly, a drummer might interpret the score by playing a *short-long-short-long* timing pattern, usually associated with an accent of the same count, e.g., all long notes have accents. A similar modification can be played inside a triplet pattern, as in the Blues. By crowding the downbeat and backbeat (typically the 1 or 3 of a 4 count measure) with a shorter note, or leaning away from the beat rather than into it, musicians can change the perceptual feel of a piece from how it would sound and feel if the rhythm was played evenly. We note that swing is also created by sources other than drums. Louis Armstrong and Benny Goodman express swing very clearly in their horn parts, as does Duke Ellington in his piano. Some research has been done into the swing that is created by the Gestaltic[7] performance of a group of musicians, called ***ensemble swing.*** Ensemble swing has been investigated by (Friberg & Sundstrom, 2002).

We have observed that the *dynamics* of an instrument's notes, especially the surdo in Brasilian music, contribute to swing by the timing of changes in loudness and timbre that would not be classified as note events, but rather are temporal elements inside a single note event. We have looked most closely at Brasilian swing, or *swingee (swing-ghee)*. We consistently find that swingee is substantially more temporally complex than can be expressed with a simple swing ratio, and we document some typical details in chapter 5.

1.3.3 Patterns of Temporal Variation

It is incorrect to regard swing time as less precise than straight time. We will show examples by Ray Charles, Paul Simon and others that demonstrate several aspects of the

---

[7] Gestalt comes from the work of Fritz Perls, who developed psychological theories and therapies intended to teach people to look at situations as a big picture (unified) rather than a lot of separate details. We use it to indicate a unified quality in music which is emergent when a group of musicians are all "in the groove."

precision (or looseness) of swing feeling. In traditional music lessons, students are admonished to play the notes in precise clockwork manner, in time with the metronome. This exactness usually entails counting to four using identical time differences. Swing music has an exact framework of this sort for defining the large scale structure of the rhythm. However the notes between the foundational downbeats will often occur at times other than the canonical quarter note locations. Classic American Swing and Jazz rely strongly on changing quarter note intervals to some form of triplets, and similarly for other factor of 2 subdivisions like eighth notes, sixteenth etc. In Brasilian swingee some notes are played at non MB locations and also on triplet subdivisions. We use the term triplet to describe any temporal subdivision which shows a factor of 3, typically in a 2 or 4 beat meter. This may or may not exactly coincide with standard music notation usage.

In learning Brasilian rhythms as well as in analyzing them technically, we find it useful to use the metaphor of ***rhythmic targets*** to anchor the music precisely in time. We then look at the relationship patterns between subdivision notes and the target(s) in order to accurately play a rhythmic pattern in the proper style, tempo and meter, as well as playing the correct rhythm (data) *per se*. To the best of our knowledge, the concept of rhythmic targets is presented in this paper for the first time.

No matter how well one plays the data, if it ain't got the swing, it don't mean a thing. This is far more than just a word trick. (Hamer, 2000) mentions how the 1959 London production of ***West Side Story*** stage play by Leonard Bernstein was stymied because of the difficulty of finding a drummer who could adequately play the rhythms in the intended jazzy style. This was caused by deficient music reading abilities of jazz drummers, and the inability of classically trained drummers to play the score correctly. Although the classical drummers could read the score perfectly well, they did not know how to play the rhythms in a swing style.

The classic swing ***riff*** might be described by a verbal pattern like

tzzzhhhh, *tch-ta-****tzzzhhhh,*** *tch-ta-****tzzzhhhh,*** *tch-ta-****tzzzhhhh,*** *tch-ta-****tzzzhhhh***, ...

where bold font is the accented downbeat, and the time lapse between between elements of the pattern is not the same for all note events. In fact, this is the hi-hat cymbal rhythm in Duke Ellington's ***It Don't Mean a Thing if it Ain't Got That Swing.*** Most people who have listened to much American music have heard rhythmic patterns like this, and we don't render it in musical tablature because the point here is for you, the reader, to remember what this rhythm sounds and feels like. We intend this as a transmission of non symbolic information. If written in MB notation, the beats would indicate that the meter is in 4/4 time, but the ***feel*** of the rhythm as played includes a triplet timing, especially between the pickup beat (penultimate) and the accented (final, bolded) beat in each repetition of the pattern. Even though the rhythm has a triplet feel, it has no similarity to 3/4 time signature music like waltz, or 12/8 blues with their foundational count of 3. Accurately mapping these timing variations to clear temporal locations in the music is the essence of our research to characterize the swing feel.

Swing also appears in certain kinds of dance. Tap dancing clearly distinguishes between straight and swing rhythm. Most tap dance music is in either 4/4 or 2/4 time. Straight tapping is done on the canonical MB beats corresponding to quarter note type subdivision, including very fast 1/6th and 1/32nd notes. Swung beats in tap are counted with a subdivision of 3 inside the 2 or 4 meter, e.g.*uh* ***one*** *and, uh* ***two*** *and, uh ...* .[8]

1.4 Information Science and DSP Techniques

The main branches of information science that are useful in this work are signal processing (DSP) and pattern recognition. We use DSP techniques to transform audio data into a form where temporal and spectral features are more accessible. Using spectral

---

[8] Jim Giancarlo, professional choreographer and dance teacher. Artistic Director of Oregon Cabaret Theatre, Dance Instructor at Southern Oregon University. Ashland, OR USA

information we extract the note events together with their timing information. Timing information is the basis for recognizing swing, and for classifying rhythmic patterns.

The computer music research literature mentions using many DSP techniques including fast Fourier transform (FFT), short time Fourier transform (STFT, a variation of FFT), wavelets, zero crossings, frequency filtering, sub-band processing, principle and independent components analysis (PCA and ICA), and various statistical methods. We have primarily investigated wavelets, zero crossings and STFT. We find STFT to be the most practical for our work. Filtering and sub-band processing look quite promising and practical, but time limits prevented us from investigating them in depth.

The STFT produces a *spectrogram* that is a visual guide to the moment by moment changes in frequency content of the audio sample. The length of the FFT is crucial for producing waveforms that are easily parsed for note events. The FFT acts as a kind of smoothing filter for the complex and rapid changes in the audio input stream. Longer FFTs smooth more, and short ones smooth less. We find that short FFTs (less than 1024 samples) are generally not useful because the waveforms generated from these sequences are not smooth enough to reliably recognize note events. This is similar to the problem of looking directly at the waveform of the raw audio signal and trying to recognize patterns: there is too much activity in the waveform. Detecting note events is less a problem for clearly separated individual events, but for most music, several instruments are contributing to the audio signal at any particular time. Separating these mixed sources requires specialized techniques which we do not currently use, as well as high resolution of the time and frequency data. We use fairly high resolution in both time (1 to 10 milliseconds) and frequency (10 to 50 Hz).

1.4.1 Fast Fourier Transform (FFT)

Fourier analysis is a mathematical technique that transforms raw audio data in the time domain to a set of frequencies in the audio spectrum, or frequency domain. The

frequency domain form of the information is very practical for our work. In DSP, the Cooley-Tukey FFT algorithm is a commonly used algorithm that efficiently computes the spectrum of the input data. The FFT approximates the theoretical resolution of a continuous Fourier transform. The FFT is several orders of magnitude faster than the continuous transform, and also much faster than other discrete Fourier transforms. The primary trick of the Cooley-Tukey algorithm is to take advantage of certain symmetries in the Fourier transform. The data in the time domain is multiplied by complex exponential functions (essentially, sines and cosines of different frequencies) as part of the Fourier transform. The complex exponential functions can be composed by using other complex exponentials of different frequencies, much the same way that the number 1/4 can be factored as 1/2 x 1/2. By organizing these factorizations properly and re-using some of the exponentials many times rather than recomputing them each time they are needed, the compute cost of the FFT algorithm is greatly reduced. (Brigham, 1974 ; Elliot & Rao, 1982)

1.4.2 Pattern Recognition

We use the spectra extracted by DSP to identify different musical instruments by their tonal (frequency) content. The general strategy is to extract short, simple features from large, complex data sets. These time/frequency features are good for identifying a note event, such as an increase of power level in a frequency range during a time interval. This is quite useful for identification of percussion and drums. To recognize these patterns, we mostly use thresholding techniques, based on the spectral power density curves, which are plots of power vs time in a well chosen frequency range. We also use the first and second derivatives of these waveforms. These techniques are very practical but have limitations, especially for complex music samples such as several instruments playing at once, or melodic instruments with complex spectra. For these more challenging musical samples, we plan to use neural net approaches in the next stage of this work, since they are computationally efficient pattern recognizers with great adaptability.

1.5 Structure of this Thesis Document

Our work in computer music analysis has been somewhat broad ranging rather than tightly focussed on a specific narrow topic. (Plomp, 2002) has recommended that researchers not become mired in technical details to such extent that they risk seeing only trees and not the forest. The details are important of course, but so is the big picture. We present information about both small scale and large scale views of the complex topic of human perception of music, and the associated computer analysis of the musical data.

For readers who have limited time to spend or who are not keenly interested in excessive technical details about Information Science applied to computer music analysis, we recommend reading section 3.1 first which is a practical introduction to our technical approach, and then skipping directly to chapter 5 which presents the main body of our results. After this exposure, the reader may be interested in looking more closely at the technical details of signal processing and pattern recognition. The appendices on Brasilian music and the psychophysics of human hearing may also be of general interest.

In chapter 2 we survey some of the research in the field of computer analysis and recognition of music, especially swing research. We note that one of the principle differences between our work and all other research we have read is that we focus exclusively on music as a set of distinct events, whereas most or all of other research takes a statistical approach to music analysis. We believe strongly that analyzing musical events individually rather than *in toto* is a very important paradigm because this is the primary way that humans produce and consume music. Statistical and gestaltic analysis also has useful application in understanding music, but we do not work much with this paradigm. We also note the connection between swing rhythm and bodily motion which has been investigated by many researchers including (Gabrielsson, 1987) and (Waadeland, 2004).

Chapter 3 describes the variety of DSP techniques we have investigated, and in particular includes detailed descriptions of our FFT work.

In chapter 4 we describe our pattern recognition techniques which are useful but not particularly sophisticated. We also survey some pattern recognition techniques used by other researchers in the computer music analysis field.

Chapter 5 presents the main body of our original work, which is detailed analysis of several specific examples of different genres of music. We present results that use a much finer grained model of time than do most researchers in this field. We have found strong evidence that temporal granularity should be no more than 5 to 10 milliseconds for adequate understanding of critical details of rhythmic timing. Most other researchers use 10 to 20 milliseconds as the lower limit of their temporal subdivision. In particular, we present evidence that a highly experienced musician such as Ray Charles has temporal perception which has less than 5 millisecond granularity. We also present evidence that ensemble swing depends at least in part on interactions between musicians with the ability to perceive and manipulate time differences in this range of 5 to 15 milliseconds. We also present examples of alternative musical notation which gives a quantitative guide to playing swing rhythms authentically, and a technique for automated generation of swing timing variations.

In chapter 6 we present ideas for closely related future work, including some of the deficiencies we have found in Fourier analysis.

In the appendices we present some peripheral material which is germane to our broad view that parsing musical information should not be restricted to a purely computer data processing model. This includes observations about our own experience learning and playing music, information from professional musicians, the code for our algorithm, a discography of the music we have investigated, some information about Brasilian culture focusing on music and dance, and a brief description of some of the standard knowledge of the workings of the human auditory perceptual system.

In appendix E, we focus on the front-end parts of the hearing system such as the ear and cochlea because these are directly analogous to DSP extraction of information from digital audio data. We note that there are several parallel mechanisms for transforming sound vibrations into the neural patterns which enter the human brain and that eventually become our conscious perception of sound events. We make special note of the fundamental *nonlinear* qualities of the human audio data acquisition system, in contrast to analysis using computers which is predominantly based in linear mathematics. We also consider human factors related to music perception. Psychology research has produced a large body of knowledge about intrinsic properties of the human mind and its natural inclination towards producing rhythmic patterns. The human feelings and knowledge triggered by music are also very interesting but we do not pursue them deeply, deferring to the vast literature on neuroscience which is beyond the scope of this thesis. The connection between music and human emotion has been noted and investigated in both psychological and musicological literature. We find the emotional aspect quite interesting, and include some opinions based on our experiences performing and listening to music, but not as part of the main thrust of the current work.

CHAPTER 2. RELATED WORK

Computers have been used for music and audio recognition since the 1960's, but until the 1990's cost and performance issues limited widespread application. In the middle 1980's, researchers started to use computers for extraction of musical patterns (Strawn, 1985). (Goto, 1994) implemented a system that could recognize basic beats in some simple music. This primarily consisted of picking out the ***boom-chuck!*** beat of pop music. The system ran on a supercomputer which at the time meant 64 300 MHz SPARC CPUs. Pattern recognition used traditional AI techniques (agents and symbolic modeling) applied to music which was preprocessed by simple DSP techniques. The system did not run in real time, despite the substantial compute power available, and was neither very robust nor accurate.

The rise of the Internet generated substantial interest in automated classification of musical genres. Initially the work was motivated by intellectual concerns, but the possibility of practical commercial results quickly gave rise to the idea of Music Information Retrieval (MIR). It was hoped that MIR could be used to help listeners browse audio databases in a practical manner, i.e., to help listeners find new music they like and so encourage them to purchase this music. Several conferences (ISMIR, DAFX, EUROSIP, AES, ASA, ACM Multimedia) have arisen in this field, but commercially robust systems remain elusive.

2.1 Onset Detection and Event Identification

The extraction of useful patterns from music generally uses one or more of three approaches: characterization of the sound quality (timbre), detection of note events, and identification of patterns based on temporal and tonal qualities. These techniques are most useful for rhythmic music passages, and basically ignore the more subtle problem of

detecting and classifying smooth changes in music such as breath modulation in voice or wind instruments, timbre changes, continuous changes in frequency and/or loudness changes of a protracted sound event complex. There are several reasons for focusing on the simpler tasks of extracting percussive type events. The primary reason in our experience stems from the problem of resolution, primarily in frequency, but also temporal resolution in some cases. To distinguish frequencies more precisely a longer FFT is needed. This involves a tradeoff with time resolution. We explore some ideas for improving this situation in chapter 6.

A technically better solution is to use optimized DSP hardware rather than DSP software techniques to perform the first stages of frequency extraction. In human and animal hearing, the cochlea performs frequency identification directly, and subsequent neural processing enhances the information content of the perceptual stream. Note that while the goal of FFT or frequency specific hardware is the same, the method of processing is very different. Fourier analysis can only give a static picture of a specific set of frequencies that approximate an audio sample over a specific, somewhat lengthy time interval. The time frame of practical audio FFTs is from about 20 up to 200 milliseconds. Short FFT time frames are most useful for extracting fine timing details, but long FFTs are required to extract fine details in the frequency spectrum. The cochlea gives information about the instantaneous frequency content of a signal derived from each sound wave cycle. The wavefronts happen rapidly, one for each cycle of the incoming waveform. The time frame of the wavefront can be 50 milliseconds or more, and goes down to the sub millisecond range, depending on how rapidly the incoming wavefronts occur. The rapidity of wavefronts is determined by which frequencies dominate the incoming audio signal. We don’t know of any similar technique in DSP.

In the cochlea the shape of the wavefront gives information about the frequency content of the signal much more rapidly than is possible with Fourier analysis, which must process many milliseconds of data. We believe that this represents a fundamental

limitation to DSP approaches, and favor development of a MEMS (micro electro-mechanical system) device as a kind of silicon cochlea. Such devices have been developed for high end aerospace and military applications for several decades. Recently, (Schwartz, et al., 1999) have constructed a VLSI chip which acts as a silicon cochlea, supporting the field of hearing remediation (e.g. hearing aids, implants). Only if these are mass produced in large scale will they become practical for consumer products, and so far, DSP techniques are considered adequate for many practical purposes.

Although CD quality sound is the current standard for high fidelity commercial music, we note that professional recording studios already use higher quality digital sound formats, up to 24 bit sample depth and 192 Ksamples/sec. It is possible that adaptive pattern recognition or nonlinear techniques will extend the limits of DSP. In any case, such developments are driven by economic more than technical considerations.

2.2 Music Information Retrieval (MIR)

MIR typically focuses on statistical feature matching for tempo, meter, harmonic structure, musical key, chord progression and other musical metrics. Several researchers (Klapuri, 2004), (Tzanetakis, 2002), (Dixon, 1999) have developed techniques that extract and match such characteristics for different genres of music. These systems can distinguish classical music from Jazz or Pop, and even to a certain extent distinguish Rock ‘n Roll from Heavy Metal and make similar gross distinctions between obviously different genres. In some cases the classification rate is above 90%, but the Holy Grail of fully automated classification for many varieties of music has not yet been accomplished.

Even if classification methods become sophisticated enough to reliably distinguish genres that are rhythmically, harmonically and melodically quite different, it is a very different and more difficult matter to distinguish between highly regarded original music like Miles Davis, and generic sounding “Elevator Music” copies of the same piece. Indeed, this can be difficult even for human listeners especially if they are inexperienced

in a musical style. Few people would seriously suggest using automated computer software to generate the synopsis on the back of a book -- this is a task for people to do, and will remain so for quite some time. Similarly, using computers for tasks like distinguishing the Afro Cuban All Stars or Emmie Lou Harris from a sappy cover of the same tune may remain impractical for many years.

All this criticism being said, we believe that MIR already has interesting capabilities, and this effort will soon yield practical results, letting us search for music by copy/pasting a few audio samples into the search engine, much as we search for linguistic information in Google. However, because of the current limitations for MIR, we have chosen to concentrate on other areas of music recognition technology which seem to us to be more immediately practical.

2.3 Swing Analysis

(Gabrielsson, 1987) presents results of a conference of research into timing variations in music, and the effects of such rhythmic expressiveness on human listeners. (Hamer 2000) presents a report describing technical research on the characteristics of swing rhythms done at the Swedish Royal Institute of Technology by Anders Friburg in the late 1990's. Other research into swing has been done by a small but growing number of researchers. (Guoyon, 2005), includes a detailed survey of swing research, MIR, and other information science work in the field of music knowledge and understanding.

In all the literature we surveyed, the patterns of temporal variations in swing music are modeled as an arithmetic ratio of "long" notes to "short" notes, called the ***swing ratio***. This means that if the score presents a rhythm as a set of eighth notes for example, that *alternating* notes are played with slightly more time or less time than the score indicates, in order to achieve the swing feel. Guoyon presents algorithms to make these temporal modification under program control. Our research indicates quite clearly that this model of the swing rhythm style is overly simplified and in particular, thoroughly inade-

quate for characterizing *swingee,* or Brasilian swing. We also believe the swing ratio does not accurately model the general situation of swing in American music either, although it is suitable to specify swing in many cases, e.g. the swing in ***Fever,*** by Ray Charles and Natalie Cole (2004) which we analyze shortly. Bear in mind that these simple cases are often very good pieces of music, with excellent toe tapping swing. We do not intend any pejorative attitude towards the swing ratio concept, but from our work we present a more complex model that we believe is more widely applicable.

We believe our approach is novel in that we concentrate on first principles: direct detailed analysis of timing variations among individual note events, rather than a statistical approach. Statistical analysis is useful to extract patterns and metrics in musical samples where these patterns conform to the statistical model, such as finding a swing ratio of short to long notes when the short-long pattern is played consistently and evenly -- i.e. swing exists only on one level of the hierarchy. The Brasilian music we've looked at most closely has patterns of temporal variations at different time scales, including between successive notes, between successive downbeats and offbeats and parallel temporal variations shared between notes that are not immediately adjacent. We also found that not all the instruments play the same type of swing pattern. We conclude that swing often exists at several hierarchical levels in the music, and that different instruments may swing in different ways, synchronized by a set of commonly shared musical targets which generally correspond to MB temporal locations. The MB targets are fixed but the notes played by the musicians for these temporal targets may shift slightly in time, centered around the fixed time locations of the MB notation. These non uniform temporal shifts determine an overall "loose" vs "tight" feel to the rhythm. While the temporal variations in a rhythmic pattern may be non uniform by one metric, they may be very *consistent* in the sense that they are repeated more or less exactly, i.e. the pattern at the next higher level of the swing hierarchy is a uniform and consistent pattern. American Swing can also have these qualities. ***Graceland*** has a loose feel, whereas ***Fever*** is very tight. We give details in chapter 5.

Most or all previous research has framed the problem of rhythm analysis and identification (including swing) strictly in terms of standard subdivisions of the meter in MB notation. This has produced some interesting results including the *beat histogram* which applies an FFT to data about time differences between all note events in a piece of music (Tzanetakis, 2002). Our approach is bottom up rather than top down, and we prefer to give the rhythmic patterns as much freedom as they deserve rather than forcing them to fit into the top down MB metrical subdivision model.

2.4 Swing and Motion

(Waadeland, 2004), (Gabrielsson, 1987) and others have investigated the connection between motion and swing rhythm. Waadeland presents analysis of many drummers, comparing their body motion with the rhythms they produce. Gabrielsson presents a collection of papers from the Third International Conference on Event Perception and Action sponsored by the Royal Swedish Music Academy, which includes a variety of research, opinions and conclusions about the relations between motion and rhythm. In all cases the connection between dynamics of bodily motion and production of swing or other rhythmic expression is well established. This should come as no surprise. The basic nature of any dynamical system is that it is extremely difficult to achieve perfect symmetry, and even the most meticulously crafted mechanical systems (e.g. Swiss watches with gears made from jewels) have a certain amount of lopsidedness to their action.

In Brasilian music, using this body english effect for producing swing rhythm is almost universal. Indeed, musicians of many varieties of music move their bodies as part of their musical performance, whether they are a classical string quartet or the Gospel Choir in a revival Church. We are actually baffled that anyone would think that there is *not* a fundamental connection between motion and rhythm, but it has taken work by numerous researchers over several decades, as far back as (Seashore, 1938), for this very obvious effect to be accepted as a real phenomenon.

CHAPTER 3. DSP WORK

We investigated several DSP techniques for this project. The most practical we found is the Short Time Fourier Transform (STFT). STFT performs a sequence of windowed FFTs on a data set, with the next FFT starting a fixed time delta after the current FFT. Generally we used a Hanning window (a variation of a Gaussian bell curve, see figure 3.3), slightly shorter than the length of the FFT. A few experiments using other windowing strategies suggest that this is an area that could be substantially optimized, but these refinements are not within the scope of this thesis.

The time duration of most audio samples we look at is less than twenty seconds, which means there are several hundred thousand individual data points representing the audio signal. For simplicity we look at mono signals, because the rhythmic patterns we study can be considered as a single stream of note events. More subtle analysis of music should process all available sources of information: both stereo channels, comparison of phase information for correlated harmonics representing a single instrument, or changes of power level and frequencies in a sound that indicates features like tremolo or vibrato.

Wavelet processing is an interesting modern technique (1990's) which blends time and frequency processing into a single framework, decomposing the signal into a metric space that uses the set of wavelets as basis functions. We have done lengthy work with wavelets, motivated by the appealing concept of performing time and frequency processing together, plus the potentially high performance that wavelets deliver in some cases. Our investigation did not lead us to a good unified approach using wavelets, and so we returned to using the simpler and more straightforward FFT as our primary DSP tool.

3.1 Spectra and Time Series

Figure 3.1 shows a spectral analysis of an audio sample that is typical of all samples passed through our algorithm. We chose it to introduce our work with spectra and STFTs because it clearly portrays both simple and advanced musical events that we want to identify. This figure shows Natalie Cole singing a chorus of ***Fever***. Elapsed sample time is displayed in seconds along the lower (***X)*** axis (turn the page sideways) where the lyrics are also shown. The total time for the ***Fever*** sample is about 14.3 seconds. We compute several thousand FFTs on the sample in order to get a representation of how the spectrum changes in time. An FFT is computed at the beginning of the sample, then the FFT window is shifted forward in time by a small delta and another FFT is computed, and so on until the end of the sample. All FFTs use the same window size. In this example the window size is 2048 data points of the original music sample (2 Ksamples). The time shift between FFTs is much smaller than the FFT window size. In this example we use 441 data points of the audio sample as our time delta. At 44,100 Ks/sec, an FFT shift of 441 points means 10 milliseconds resolution in the specgram. A shift of 132 points equals 3 milliseconds, and so on. More detailed analysis is presented in the next section.

A 2 Ks audio sample gives an FFT spectrum of 1025 evenly spaced frequencies. The number of frequencies is computed as (N/2) + 1 where N is the size of the FFT window (2048). The frequencies found by the FFT are displayed on the ***Y*** axis, from 20 Hz to 22,500 Hz which is the Nyquist frequency of the CD sampling rate. Dividing 22,500 by 1025 gives a frequency resolution of about 22 Hz for the 2 Ks FFT. A 1 Ks FFT gives frequency resolution of about 44 Hz, and a 4 Ks FFT yields 11 Hz resolution. There is a tradeoff between resolution in the time and frequency dimensions. While this frequency resolution is sufficient for the current work, it is quite limiting. There are some important efficiency and optimization issues that we explore in the section on future work.

The somewhat regularly spaced sharp vertical red lines are primarily finger snap and conga events, and also include some portions of other drum sounds. The yellow tips

extending into the blue “sky” that are *exactly* evenly spaced are Ray Charles snapping his fingers on the back beat. Specgram colors are determined by the audio power of each frequency at each time point, normalized to [0,1]. Blue is low and red is high power.

In the lower third of the diagram are patterns of wavy red lines, regularly spaced in the vertical axis. This is the spectrum of Natalie Cole’s voice. You can visually correlate these frequency features with the lyrics written below. The concentration of red at the bottom are the frequencies generated mostly by bass drum and bass guitar. These signals are less well defined than a human singer or melodic instrument like a trumpet. This is partly a limitation of the FFT frequency resolution, partly typical of features that can be extracted at low frequencies and partly spectral structure of the sounds of these instruments *per se*. Both the physics of sound waves and the physiology of the human ear limit the information available at these low frequencies. These physical limits are probably part of why the human voice uses a higher range of frequencies, 200 Hz to several thousand Hz, to encode the majority of speech information.

The details of the correlated frequencies of Natalie Cole’s voice are the data that give rise to our perception of *timbre* which is the quality of sound that we associate with identifiable audio events such as words or phonemes (speech), and musical instrument identification. As noted in (Dowling & Harwood, 1986) the higher frequencies (200 Hz to 5000 Hz) encode most of the information about timbre as heard by humans. These frequencies are directly perceived in the human ear by stimulation of frequency sensing “hair” cells in the cochlea. The frequency responses of the hairs is determined by their location in the cochlea, with high frequencies being sensed near the eardrum, and lower frequencies sensed at the far end of the cochlea. Pitch (or tone) is encoded more in the lower frequencies (20 Hz to 2000 Hz) which are sensed both by the frequency sensitive hairs as well as the beat phenomenon (see Appendix E). This is a reason why most melodic information is in the middle and lower frequencies: the precision of our pitch perception diminishes as the frequency goes above about 5000 Hz. Our sensitivity to higher

pitched sounds is not diminished -- hearing the snap of a twig makes the difference between the tiger having you for lunch or not. At high frequencies however, direction and distance are more important information than precise identification of pitch. (Dowling & Harwood, 1986) discuss this aspect of human hearing in some detail. They also mention, as does (Buser, et al. 1992), that human tone perception is ***not*** perfectly correlated with the measured frequency.

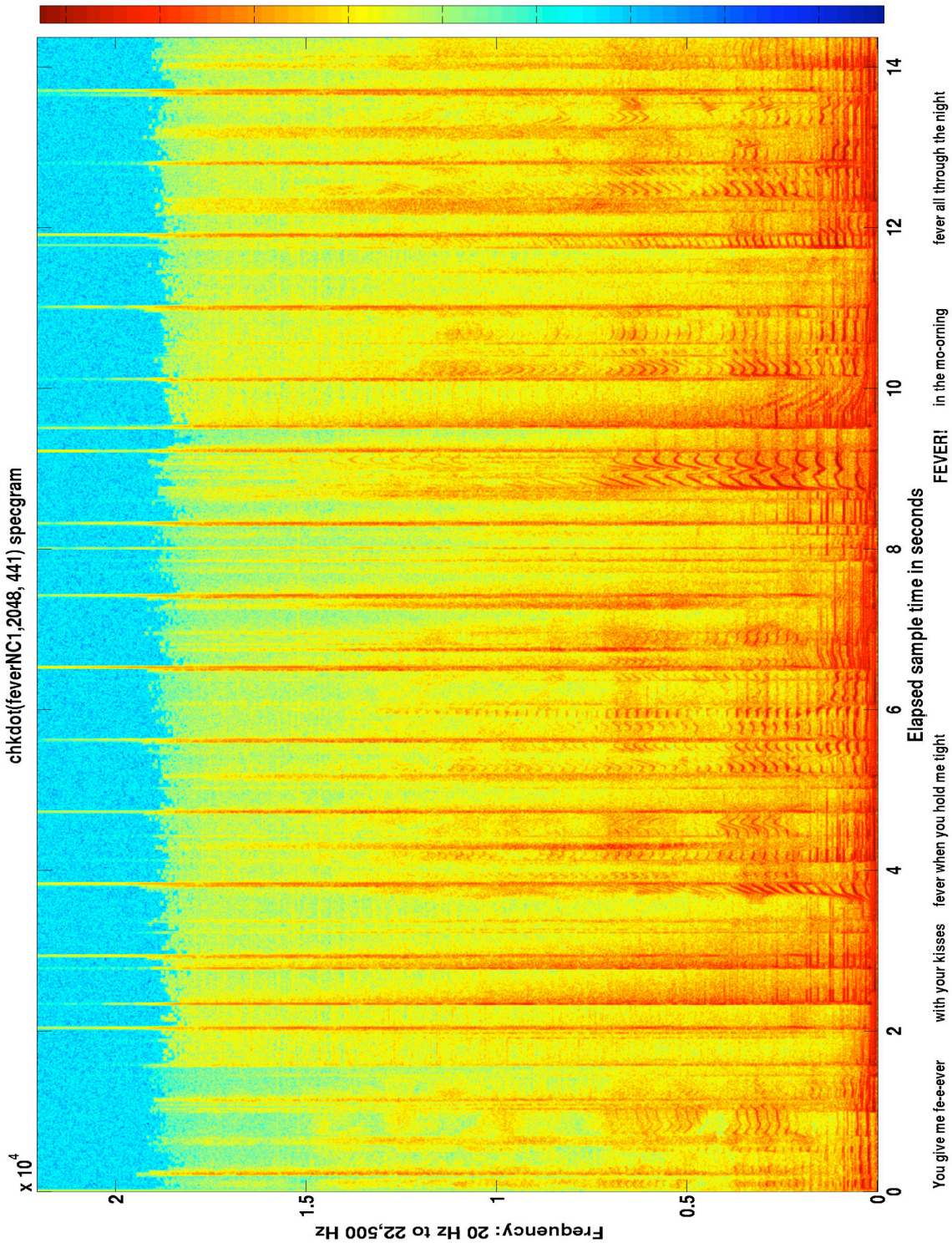

Figure 3.1 Spectrogram for Natalie Cole singing the chorus of ***Fever***

3.2 FFT and STFT

An FFT returns an array of the frequencies contained in a musical sample during a particular short time slice of data. The frequencies change quickly and incessantly throughout any "interesting" sound. Of course there is music with very slow moving changes of frequency content which is also interesting, e.g. Pink Floyd, or classical Pastoral music. Our techniques could be applied successfully to this sort of music, but in this work we only investigate music with quick events. In order to generate a picture of how the frequency spectrum of a music sample changes in time, we apply the FFT repeatedly with a slight time change between successive FFTs. This is commonly referred to as Short Time Fourier Transform (STFT), yielding a spectrogram or, as Matlab calls it, specgram. The time/frequency tradeoff is very important for making an STFT useful. A short FFT gives coarser frequency resolution and finer time resolution. Conversely, a long FFT give more frequency detail, but for a sample whose frequencies are averaged together over a longer time, yielding a less precise view of the temporal changes. Depending on the frequency content of a piece of music, adjusting the frequency and time granularities brings the resultant specgram into clearer "focus," in the sense of rendering particular details more or less clearly. These adjustments are very much like focusing a camera lens, but rather than focusing a spatial image, we change between looking more precisely at time or looking more precisely at frequency. There is a fundamental limit to the total precision, governed by the equivalent of the Heisenburg uncertainty principal.

The STFT approach we use performs a sequence of overlapping FFTs on the musical audio data. The FFT window size (time slice) for any particular run of the algorithm on a music sample is constant, e.g. a time slice of 1024, 2048 or 4096 samples of input data (1024 samples == 1 Kilo sample or 1 Ks). These three window sizes correspond to time intervals of 23, 46 and 93 milliseconds respectively, at the CD sample rate of 44,100 samples per second. The FFTs yield a frequency resolution of 44, 22 or 11 hertz respectively for 1 Ks, 2 Ks and 4 Ks lengths. We tested a few samples with 8192 length FFTs,

but in these cases the time span of the FFT significantly blurs the frequency details, and were generally not very useful. Similarly, using a 512 sample window for the FFT produces a very choppy picture of how the music sample changes in time, which makes detecting note events difficult due to the numerous small spikes in the temporal waveforms that we generate from the STFT.

The time slice and frequency information for a particular set of STFT parameters can be computed with simple algebra. For a time slice (i.e. FFT window size) of N_{ts} data points, and sampling frequency F_s samples/sec, the time occupied by the FFT window T_{fft} is given by $T_{fft} = N_{ts} / F_s$. For convenience and efficiency we use powers of 2 for the size of N_{ts} , e.g. $1024 = 2^{10}$, $2048 = 2^{11}$, $4096 = 2^{12}$. In some cases we use an FFT window which is the sum of numbers that are powers of 2, e.g. $3072 = 2^{11} + 2^{10}$. After performing the FFT, we obtain a vector of frequencies contained in the audio sample. The number of frequencies N_F is determined by the FFT window size, counted in audio data points. The formula is $N_F = (N_{ts} / 2) + 1$, so a 2048 point FFT yields 1025 distinct frequencies. The specific frequencies contained in the frequency vector are integer multiples of the "fundamental" frequency of this particular FFT, based on the formula $F_{fund} = 1 / T_{fft}$, so the frequency set is given by $F_{\{\}} = \{ n * F_{fund} : n = 1, 2, 3, ... N_F \}$.

The time shift between FFTs is the same for each run of the algorithm. Depending on what temporal resolution we want, we may run a sample several times with different time shifts in order to find an optimal resolution for specific musical features. In some cases we will present results with several different time/frequency settings for the same music sample. Typically we used time shifts between 3 and 10 milliseconds, although we tested some music samples using as short as 0.5 milliseconds time granularity.

The frequencies in the spectrum of the FFT are equally or *linearly* spaced: the difference between adjacent frequencies in cycles per second (Hz) is the same whether they are low frequencies or high frequencies. This is inherent to the design of Fourier Analy-

sis, and we consider it a disadvantage, which we explore in chapter 6. The frequency spacing in the human perceptual apparatus, and also in the spacing of note pitches in music is *exponential*. This means that the "distance" between adjacent notes as measured in Hz increases for higher frequencies and decreases for lower frequencies. Given two pairs of different adjacent notes on the keyboard, there are no two pairs that have the same frequency distance in Hz between note N-1 and N-2 compared to N-3 and N-4. For example, the number of notes between middle C and the C above or below is the same: seven white keys and five black keys on a piano, i.e., twelve half steps. The frequency in Hz of the tone corresponding to these C notes is doubled if you go up the scale or divided in half if you go down. This means that the frequency spacing of the notes within each octave is also doubled or divided in half compared to the corresponding note one octave below or above the note being currently examined. In practical terms, there are too many frequencies measured by the FFT in the upper part of the spectrum, and not enough different frequencies delivered by the FFT in the lower part of the spectrum.

3.3 Windows and Filters

Windows are scaling functions used in conjunction with the FFT algorithm for the purpose of improving results and decreasing false artifacts in the transformed data. A window is typically a simple function such as a gaussian curve that modifies the current slice of audio data, prior to the FFT. This modification is simply a sample by sample multiplication of the audio data and the window data. This yields a data slice of the same length as the original data that is smoothly reduced to zero at the beginning and end of the slice. The main effect of this pre-processing is to reduce or remove ***aliasing*** artifacts. These are the result of wraparound effects in the Fourier transform when it converts the audio data from the time domain to the frequency domain. For a detailed technical analysis, see (Brigham, 1974), (Press et al., 2002), and (Hamming, 1983). Figure 3.3 shows a sample of audio data (green), gaussian window function (blue) and the composite (red),

ready for passing into the FFT. Hundreds or thousands of such slices are processed for every musical sample processed by the STFT.

Filters are functions applied to incoming data which change the frequency content of a data sample by reducing or amplifying some range of frequencies. This may simplify subsequent processing of the data sample, or the filtering step can produce useful results directly such as measuring the power of the signal in the range of the filter. Our analysis of the compute costs of filter processing compared to FFT processing led us to prefer the FFT for the current work. The FFT approach was similar in compute cost and substantially simpler in system design, saving development time.

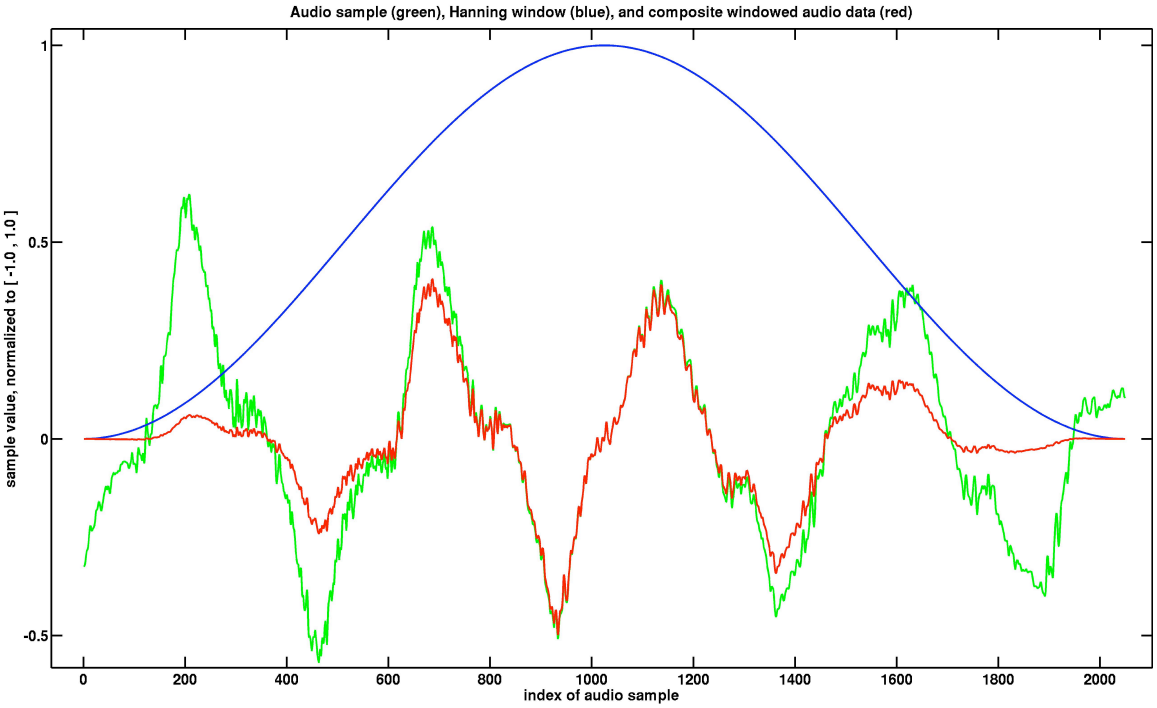

Figure 3.3 Audio Data, Window Function and Composite Result for FFT

3.4 ICA (Independent Components Analysis)

Independent components analysis is a recently developed technique useful for separating several sources of data that are mixed together in one or more data streams

(blind source separation or BSS). We briefly investigated this approach during work on note identification. Figure 3.4 shows how ICA might be used for identification of an instrument's note events by compositing a short data sample of the desired instrument sound with a longer data stream. In essence, this is a form of autocorrelation, but one that matches statistical patterns rather than exact waveforms. The bold, clear vertical line of correlated data points and the elliptical "galaxy" indicate that the data stream included the sound of the pandeiro in this case. This is a promising area for future research.

(Anemüller & Gramss, 1999) used artificial neural networks in preference to ICA for the task of source separation, claiming fast learning (about one second) for their algorithm to be able to distinguish two mixed sounds recorded in an anechoic chamber. Their network topology was a variation of feed forward multi-layer perception.

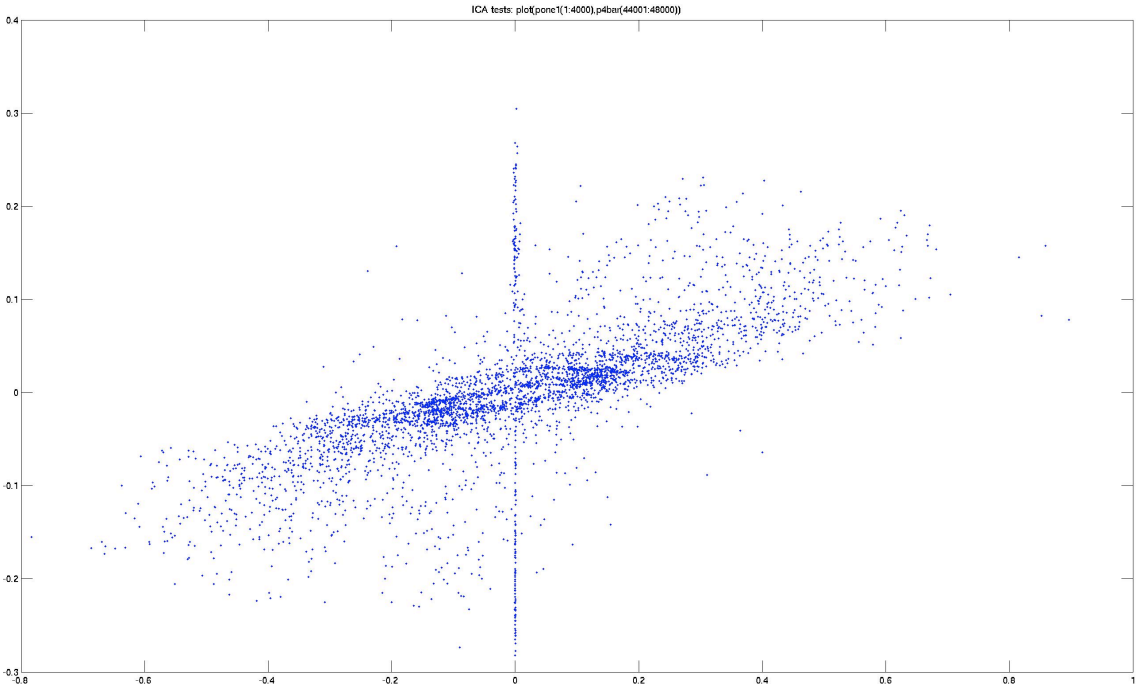

Figure 3.4 ICA Autocorrelation Plot Showing Identification of Pandeiro

3.5 Wavelets

Wavelet analysis decomposes a single large data set into several smaller data sets. These are determined, analogously to Fourier analysis, in terms of basis functions span-

ning a function space (Hilbert or Banach space). The function space for our work is simply the waveforms of the audio samples being analyzed. Fourier analysis creates a set of sine and cosine waveforms and their amplitude coefficients which, when added together, recreate the original waveform. Similarly, wavelet analysis uses a mother wavelet and copies of the mother wavelet which are scaled and translated so that the resultant set of waveforms and their coefficients will accurately represent the original waveform.

We investigated wavelets hoping to use them as a source of features for identification of note events contained in the data set. Wavelets could be used to identify note events, while also identifying the temporal location of these events in the audio stream. Our preliminary investigation and several papers in the computer music research field indicate that wavelets could be a useful analysis framework for musical note events and audio streams. We chose to abandon this line of research due to its technical difficulty and because of efficiency and scalability concerns. While an individual musical note event is both tractable and practical to analyze using wavelets, the extension to analyzing a complex audio stream with multiple instruments would entail a compute cost that we think scales at least as $\boldsymbol{O}(N^2)$ or worse for a number N of different note types. In contrast, Fourier analysis using the FFT is an $\boldsymbol{O}(N \log(N))$ operation. These numbers are merely indicative, and a complete analysis would involve considering both the number of steps in either the wavelet or FFT process, as well as the true compute cost of each step. We believe that wavelets could be used in an effective and efficient manner, but would require a deep knowledge of mathematics (Hilbert space etc) that is beyond our expertise.

3.6 Zero Crossings

Zero crossings are time points where the input audio signal power level (voltage or sound pressure level) goes from positive to negative or vice versa. The data points themselves may not equal zero exactly, in which case we noted the time points of sign changes and plotted the time of the first data point after the sign change as a zero crossing

event. While this technique is commonly cited in the literature, its utility was not immediately obvious for our work and after a short investigation moved on to other techniques.

Figure 3.6 shows a short sample of a pandeiro *pee* note event, with zero crossings marked as sets of blue dots along the two horizontal lines $\boldsymbol{Y} = \pm 0.8$. The audio sample is plotted in cyan. We count the zero crossings and show this count by the red, black, blue and magenta dotted lines. The blue line shows the count in a moving 40 data point window, red uses an 80 point window, black uses a 160 point window and magenta uses a 320 point window. The lines for the counts are normalized to fit into the same vertical scale as the audio waveform. Thus the lines show the relative count rather than the absolute count of zero crossings in their respective windows.

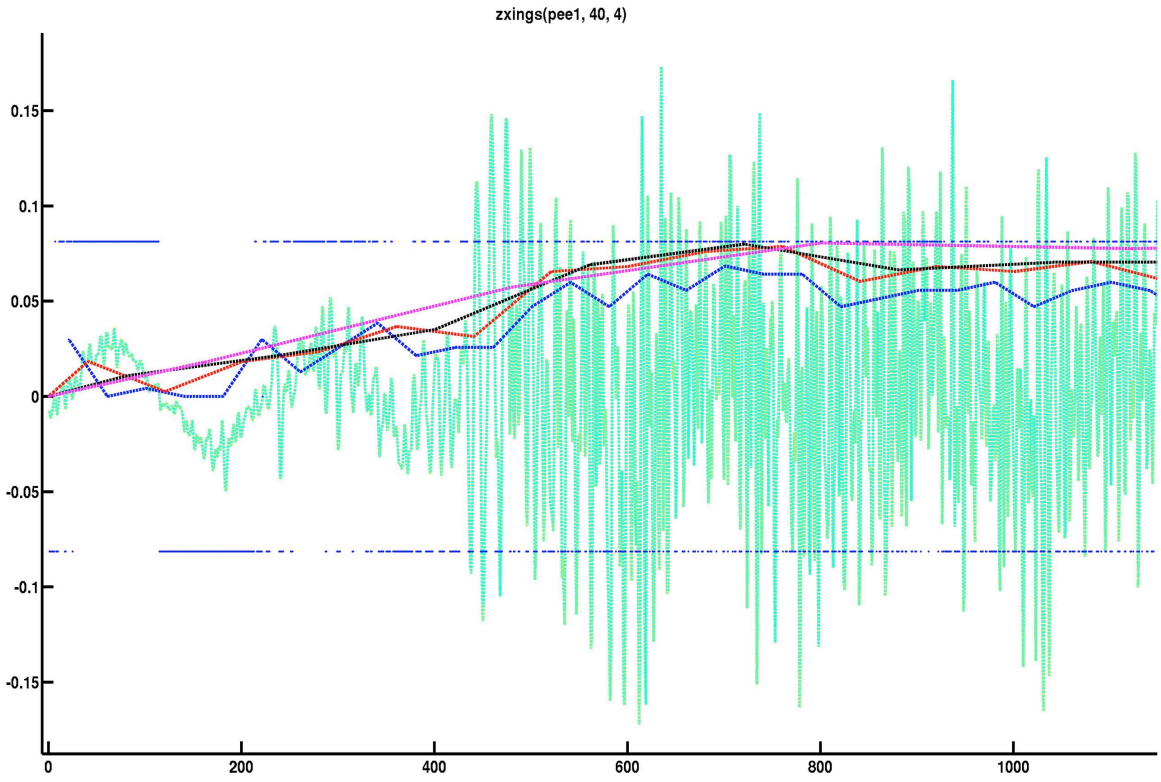

Figure 3.6 Zero Crossings in a Pandeiro Note Event (close-up)

3.7 Signal and Noise

For the most part, the music recognition literature differentiates percussion sounds from melodic and other instruments. This is a reasonable distinction because harmonically correlated sounds such as pitched or melodic instruments are fundamentally different from most percussion note events, which tend to have strong non harmonic features and characteristics. The jargon used to describe these distinctions, however, is sometimes misleading and should be amended in favor of more accurate language.

While there is some discussion about the pitch qualities of percussive sounds, percussion and drum sounds are commonly referred to as "noise". We believe this is essentially an ignorant viewpoint. Noise is merely information that is not properly understood. The canonical form of noise, white noise, is an idealized gaussian distribution of all frequencies with very useful properties. Noise as a name for some more generalized category of information takes on a wide variety of characteristics. In audio production, a low level noise signal, e.g. water or wind sounds, is commonly used as "bed" or foundation for the mix of a soundtrack. This provides subliminal shaping of the listener's perception of the meaning of the soundtrack and if done skillfully, greatly enhances the believability of the soundtrack. Usually this is *very* skillfully mixed to the point where most listeners are not explicitly aware of this shaping of perception. Another common use of noise is in the visual effects industry for film, where libraries of "film grain" are always used to help blend the computer generated graphics with real world scenes. Early productions in the 1980's and 1990's (e.g. ***The Abyss*** or ***Babylon 5***) would sometimes omit the film grain and the special effects from these early works have a very "clean" quality, whereas modern computer effects are blended much more skillfully with real world footage. Again, most people who are not professionals in the field are rarely aware of these subtleties, but the presences of this noise work greatly enhances the immersive believability of the audio or visual piece.

While percussion sounds are complex statistical entities having far less harmonic correlation than melodic instruments, there is little difficulty for our human perceptual system to distinguish between most types of individual percussion note events, even if they are very similar to each other, or mixed with several other instruments' sounds. Thus these sounds are not noise in the sense of being random or unpredictable.

In our work we easily distinguish different drum note events by using a simple frequency based approach, without resorting to any statistics. We have also seen examples where the simple frequency summing technique fails to separate two somewhat similar sounds, such as caixa and shaker. From our survey of the research literature, we expect that refining our note identification process by using simple statistics will produce useful improvements with moderate effort. We discuss this a bit further in chapter 6.

3.8 Description of Our DSP Algorithm

Our DSP algorithm (`chkdot.m`) performs an STFT on musical audio data, followed by note event identification logic, and marking of timing patterns. This is implemented as a Matlab script, listed in the appendix. For input data, the code takes a vector of digital audio data, the FFT length, and time delta for shifting to the next FFT. This stage produces the specgram which we then inspect to determine what frequency ranges to use for identifying note events. Optimizing tradeoffs between time and frequency resolution often requires testing several different sets of parameters to get a truly useful specgram. The transformed spectral data, lists of frequencies in the spectrum and time points of the shifted and overlapped FFTs are retained by Matlab as script internal data and used in subsequent passes through the algorithm for analyzing the specgram. Since computing the specgram can be as much as several hundred times more compute cost than running a time/frequency analysis, this allows us to perform multiple analyses of a single useful specgram, in order to get the best results possible in minimum time. This design can be easily adapted for use in a GUI, which we have not yet implemented.

Subsequent passes through `chkdot` use several vectors and matrices for guiding the DSP and note ID logic. These include a vector that specifies which frequency ranges to use for identifying note events (event tracks), a vector specifying how note events should be counted in the pulse track, a vector specifying in which secondary event tracks to mark events, a vector indicating how to subdivide the sample time based on the primary events detected in the pulse track, and a matrix of threshold values to use on the waveform in each event track. Thresholds can be specified for the waveform itself and its first and second derivatives. For each time slice, in each frequency range, we sum the values of each FFT for the frequency ranges specified in the frequency vector. This gives a sequence of points that represents, for each time frame, the signal's audio power in the current frequency range. These points are plotted as time series that show the changing power levels of the audio signal in the several frequency ranges specified.

The primary pattern recognition logic, after the STFT, uses thresholds of the amplitude changes between the time frame data points. We use the power level of the signal, and the first and second derivatives implemented as first and second order difference equations. Figure 3.8 shows a composite of the waveforms for the standard pandeiro batida, along with the first and second derivatives of the waveform.

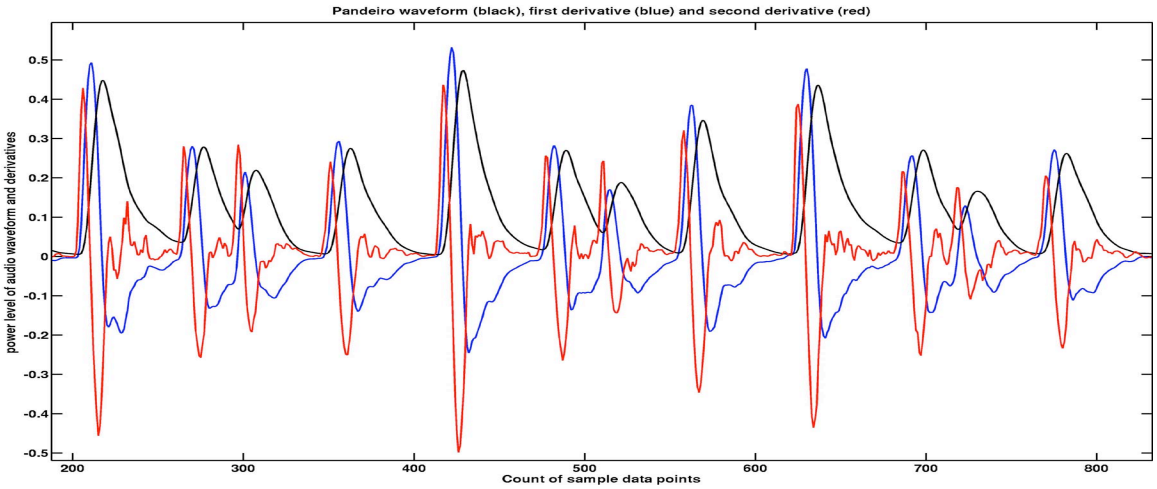

Figure 3.8 Pandeiro Waveform, First and Second Derivatives

CHAPTER 4. PATTERN RECOGNITION

After an audio signal is processed by STFT, yielding a set of frequency features, the features need to be classified into different types of note events, and the note events are time stamped according to elapsed audio sample time. We use a time granularity that is shorter than the time duration that seems noticeable for professional musicians -- see the example of Ray Charles' version of ***Fever*** in chapter 5. Our preferred temporal resolution of 1 to ten milliseconds also gives fine enough resolution for detecting rapid note onset events (percussive events). Other researchers have used longer time granularity, but we believe important information is lost in the coarser resolution provided by a time step of ten, twenty or thirty milliseconds. Psychology research reports that human perception of time has even longer granularity (around 100 milliseconds) but we believe that in the realtime context of music, human time perception is much quicker than these reports indicate. More detailed research into the perception of repetitive rhythmic events may shed light on this topic. (Gabrielsson, 1987) includes several papers describing such work.

4.1 Feature Vectors vs. Raw or Processed Data

A feature *vector* is a correlated set of information features that represents a type of note event. The primary features we use are power envelope waveforms for several frequency ranges in the STFT processed data, and the mathematical derivatives of these waveforms. These are sufficient to parse many types of percussion events in sparse musical samples. The literature indicates that adding statistical metrics to the feature vector can enhance the number of types of note events that can be reliably extracted.

For complex note events, such as a beat that has two or more types of instrument notes played simultaneously (or overlapping in time), we next plan to try using sets of overlapping frequency ranges. The frequency set of one instrument would overlap some

with another instrument for the identification process. The literature commonly mentions using simple statistical methods to extract information about temporal patterns of changes for a frequency range, or differences in the distribution of the frequency power curve for a particular time slice. We had some initial success using this idea of overlapping frequency bands vs separate frequency bands to define musical event channels, but time constraints prevented fully developing this feature. Extending our current quite practical frequency approach by using overlapping bands is appealing both for the short development time and low compute costs. We believe we could expand our vocabulary of easily recognized note events using low compute cost techniques for simple separation of simultaneous note events. A more complex audio mix would require advanced techniques.

4.2 Description of Our Pattern Recognition Techniques

After the STFT data set is generated, we select frequency bands (sub-bands) that are visually correlated with note events which we can hear when playing the audio sample. We first look for a pulse rhythm: a relatively simple and regular pattern which lays down the large scale metrical structure of the rhythm. Two principle pulse keepers in Brasilian music are the pandeiro and surdo, which parallel the role of the bass drum or bass guitar in American Jazz, Blues etc. In Reggae, the *kip* which is usually played on electric guitar performs the role of musical pulse. The role of all of these instruments is to create a heartbeat in the music that the other instruments use as an anchor for their more complex rhythms. The pulse events are usually easy to extract by summing low frequencies in the STFT, typically 500 Hz or less. In some cases there are complexities which make it difficult to use the low frequency heartbeat as a pulse track. ***Fever*** falls into this category because the drums and bass guitar do not play a simple rhythm and are not clearly extracted using our frequency summing technique, due to the coarse low frequency resolution delivered by the FFT.. The human ear easily discriminates sounds in this frequency range because its frequency resolution is better than 1 Hz in this region. For an FFT to achieve this resolution, a time interval of 1.0 second or more would be needed. Such an

FFT would be largely useless for extracting temporal information about note events. Our experiments in adjusting the FFT parameters show that longer FFTs also tend to blur the frequency information as well as the time information.

To extract the rhythmic features for ***Fever*** we used a different track for the pulse track, in this case, Ray Charles' finger snaps. Because the finger snap is on the backbeat rather than the downbeat, we added logic that a negative index for the pulse band means to chose that pulse band number, but to use it as a backbeat rather than a downbeat. This technique worked quite well and we also applied it to ***Stir it up*** and ***Could you be loved?*** by Bob Marley.

After deciding which frequencies to use for sub-band summations, we add the values of data points for each spectral sub-band at each point in time for the entire time interval of the musical sample We normalize these sums to the [0, 1] interval for ease in processing and plotting. In the `chkdot` plots, the frequency sub-bands are stacked vertically from low frequencies to high frequencies, as determined by the frequency vector input parameter. The top band in the plot is the sum of all sub-bands, i.e. the total power in the signal.

The set of time series plots thus created corresponds to the power envelopes of the music sample in the several frequency sub-bands, one point for each FFT time slice. The amplitude of these envelopes is used by our thresholding logic to detect the time location of the onset of a note event, its peak and initial decay from peak. We mark the note event with a red diamond at the peak data point. Choosing the location for the peak the way is reasonable but also somewhat arbitrary. Some note events are well represented this way, but other note event types may be ambiguous. There are also note events where the high frequency portion of the note onset is not exactly simultaneous with the low frequency part of the note sound. Some note events use the dynamics of the sound amplitude as part of the rhythmic pattern itself. Surdo plays many note events of this type.

There is a small amount of stochastic uncertainty in the note event time location when the time series waveforms are not smooth. Short time peaks can cause our logic to register perturbations in the waveform as note events, which is not a desirable result. In these cases we first adjust the frequency and threshold parameters carefully to extract note events as reliably as possible. Other solutions, such as applying a smoothing filter to the frequency sub-band waveforms, are obvious techniques to develop but we did not have the time to do so for the current work.

Once the note events are suitably detected and marked, the time differences between events are collected into a vector of time deltas. These time deltas are used to determine the locations of the red diamond markers on the `chkdot` plots. We also create a secondary event plot, `diffdot`, which highlights the relationship between patterns of *changes* in time deltas that occur between the note events that make the rhythmic pattern. It is very important to be aware that on the `diffdot` plots, the ***X*** axis is elapsed sample time (as on the specgram and `chkdot` plots), but the ***Y*** axis is the time difference *between* successive notes in the pulse band and the secondary events bands.

CHAPTER 5. MUSIC SAMPLES

We have investigated dozens of examples of music during this project. In this section we present detailed analyses of several of these music samples, with observations and conclusions based on the detailed examples as well as the broader set of samples which are not reported in detail. The original versions of the samples all have swing feel. In some of the detailed studies we also look at "straightened" versions which have been constructed from the originals, but with note events shifted slightly to remove or reduce the swing feel. In many cases, the swing is clearly related to a triplet rhythm, where some of the notes are played on beats that are subdivided by three rather than two or four. In other cases, particularly the Brasilian music, the simple triplet subdivision may be present, but there are also other subdivisions such as 5/12 and 7/24. Additionally, the Brasilian music often has slight differences in timing between the first and second half of a musical phrase, enhancing the swing feel. This style can be found in some American swing, such as ***Graceland*** by Paul Simon. Jazz or classic Swing tends to be rhythmically much tighter than songs like ***Graceland***, and the triplets are often very exact[1].

We want to be explicit about our opinion that there exist many types of swing. The research literature which looks at swing has mostly addressed music like American Jazz, and the concept of the *swing ratio* was developed to describe the characteristic shortening of some of the note events (mostly drums). As noted in the Appendix, professional musicians often classify different swing styles by which culture the music comes from: Cuba, USA, Brasil, etc. The swing in Reggae seems to be more enigmatic, and we show some results later in this chapter from our analysis of Bob Marley's music.

---

[1] Stu Fessant, professional musician, producer and recording engineer, Indigo Groove Studio. Portland, OR USA.

We have studied Brasilian rhythms more extensively than any other style, and now describe some of the details which we have discovered in this music style. This information is included to provide the reader some context about how one style of swing differs from another. Discussions with professional musicians makes it clear to us that each swing style is likely to have a collection of details such as we describe for Brasilian *swingee.* Not being experts in all music styles, we omit such details for other styles.

Generally Brasilian music does not emphasize a simple backbeat like American music does. Rather, an analogous construct is indicated by which ***side*** of the samba is referenced. In our experience there are two sides, and, as a drummer, one only hears about it if one is playing on the *wrong side.* This illustrates the principle of interlocking batidas, or *ensemble swing*. Each instrument plays its rhythm with its own flavor of swingee, collectively anchored at a few specific MB time locations. These combine in the performance to create a quick and complex sequence of tension/resolution effects. When played correctly, it gives Brasilian music a very smooth feeling despite its complexity. When one or more player is on the wrong side, the effect is to produce a chronic tension or ***pull*** in the music. The sidedness is not limited to a *one-two* metaphor. Most batidas have two sides, but the length of each rhythmic repetition may not match the lengths of rhythms of other instruments. This produces hierarchical complexity. For example, the pandeiro plays a constant *one-two-three-four* pattern whose timing (duration, rhythmic variation) varies slightly from phrase to phrase. The surdo also has a *one-two-three-four* structure, but each beat of the surdo corresponds to one entire phrase of the pandeiro. If the push and pull between surdo and pandeiro has a consistent feel, then the two batidas mesh like gears in a well-oiled but somewhat worn out machine. If one rhythm pushes when the other pulls, the resultant rhythm will not sound as smooth.

The rhythms of other instruments (e.g., tamborim) can be started at their canonical downbeat, or the two sides of the batida can be swapped which gives an even more syncopated feel but which is still smooth. The tamborim player may start the batida a

sixteenth note ahead of the surdo/pandeiro downbeat or, more commonly, a sixteenth note after the downbeat. This type of playing around the beat is sometimes called *teleco-teco* which is an onomatopoeia for the sound of the batida.[2] Again, this produces a more syncopated feel than playing the standard tamborim rhythm, but one which still flows smoothly with the surdo/pandeiro. If the tamborim starts its pattern on the two, three or four of the pandeiro, it is still "in time" in the sense that all the 1/8 or 1/4 notes between the two instruments are played at common MB time anchors, but the accents and rhythm of the tamborim may cause a pull, with the overall feeling that something is not quite right. This is a subtle thing that I am only beginning to understand. I have not analyzed the music to this level with our algorithm but, after years of listening, understanding this notion has definitely found a sensible location in the information space in my head.

Several music software applications are available that address production of swing rhythm. We have used two of these, and processed our straight versions of some music samples using the swing algorithms in the software. We present results analysing the original swing and straight versions. We have made artificial swing versions of some samples, but the analysis of these is not presented in the current paper. Some of these "roboswing" samples are virtually identical to real samples (if carefully crafted). Creating these by hand was labor intensive and relatively tedious. For real production of artificial swing music, good algorithms are needed. We explore this in the chapter on future work.

5.1 Analyzed Music Samples

We chose our musical samples entirely by subjective considerations. Basically we picked songs we like, and that we believe have a substantial swing based on our perception. We processed short sections of the songs that represent the rhythms, making seamless loops of the audio. The loop may not represent the full range of complexity of all the rhythmic patterns contained in the songs, but they clearly show the technical details that

---

[2] Jake Raar, musician from Samba-Jah performing group. Eugene, OR USA.

give rise to swing feel. We distinguish between a high level metric, the feel of swing, and lower level patterns -- the particular rhythmic patterns that generate the swing feel.

In processing the audio samples for making loops, we shaved or added very short time sections of music to try to match the rhythm exactly as the loop jumped from the end of one repetition to the beginning of the next repetition. We discovered that very short discrepancies in timing are clearly audible, and disrupt the feeling of the rhythm. These errors may be as short as 5 or 10 milliseconds, but can be heard as a timing artifact, primarily in the pulse, each time the loop starts its repetition. The difference between swing and straight feel in a sample can be caused by time differences of less than 50 to 70 milliseconds in a few note events. These timing artifacts are a very different feature from merely having a sound "glitch" such as a pop or click due to clumsy editing. Generally it is mandatory for both the beginning and end points of the loop to be at zero signal power level to avoid audio glitches. Avoiding a rhythmic anomaly is a question of getting the time length of the loop sample exactly lined up with the patterns of elapsed time in the music so the note events occur at consistent temporal locations. As we already pointed out, the human perceptual apparatus is very astute at detecting such unnatural features as are caused by the loop length not matching the time cycle of the rhythmic repetition.

The samples we investigate in detail are ***Fever*** performed by Ray Charles and Natalie Cole (2004), ***It Don't Mean a Thing (if it Ain't Got That Swing)*** played by Louis Armstrong and Duke Ellington (1962), ***Graceland*** by Paul Simon (1986), a typical pandeiro rhythm from Brasilian Samba, two additional Brasilian rhythms which are more complex than the pandeiro sample, and ***Stir it up*** by Bob Marley (1973).

5.2 MIDI for Straight Time

MIDI (Musical Instrument Digital Interface) is a widely used protocol in computer music production and research. MIDI includes specifications for communications between musical devices (synthesizers, drum machines, sequencers), as well as a file

format for note and timing information. MIDI lets a composer specify the pitch, tempo and meter of note events, and can connect these computer events to output devices such as a synthesizer that produces note sounds.

We used the MIDI capabilities in GarageBand music production software from Apple Computer to produce straight versions of the pandeiro rhythm. We also used GarageBand to produce artificial swing versions of the pandeiro samples.

5.3 Detailed Analysis of Swing Samples

In this section we present analysis results from our algorithm for several musical samples that show pertinent details of swing timing variations. We compare original samples with straightened versions of the same samples, and describe the types of details that are apparent in the graphs when inspected closely. Some timing information may not be obvious except by close-up inspection of the plots.

5.3.1 Fever

Fever is a classic R&B song with backbeat and a 2/4 or 4/4 feeling. The 2004 Ray Charles version preserves the original rhythmic meter, but the conga plays with an exact triplet subdivision style, giving a strong and very hip swing feel, despite having no explicit feeling of swing in the sense of classic American Swing. We listened to this song many times before it consciously occurred to us that the extreme hipness of Ray's version is more than just a well played backbeat -- a richer version of rock and roll as it were. Well it weren't. When we ran this sample through `chkdot` and looked closely, we discovered that many of the conga notes are played *exactly* on the triplet pickups to the downbeat and backbeat. By exact we mean within a 3 millisecond time granularity. Other identified notes events are mostly on exact 1/4 subdivisions. The pulse is Ray Charles snapping his fingers. The timing variation of these events is less than 5 milliseconds.

To create the straight version, we edited the digital audio signal by hand to move as many of the conga notes as was practical, given the subtlety of the audio mix. The time

difference between the triplet location and the straight 1/4 note location is slightly shorter than 70 milliseconds. The straight version sounds good but has a distinctly clunky feel compared to the original. Straightening the first half was fairly easy because the music is sparse and there is little overlap of note events from different instruments. The second half was not entirely straightened because its more complex mix meant that some instruments' notes overlapped others in a way that could not be separated without creating objectionable artifacts in the sample. In addition, the drummer skids his brushes around the snare drum with a strong but subtle rhythm that pervades the mix, and also causes artifacts if edited. The editing task involved moving appropriate (*swung*) note events forward or backward in time. Some of the swing notes could not be moved because either they were inextricably blended with another note event, or else the temporal location which would have been their landing place was already occupied by a note event and putting the conga note at that time location would obliterate or severely distort the other note event.

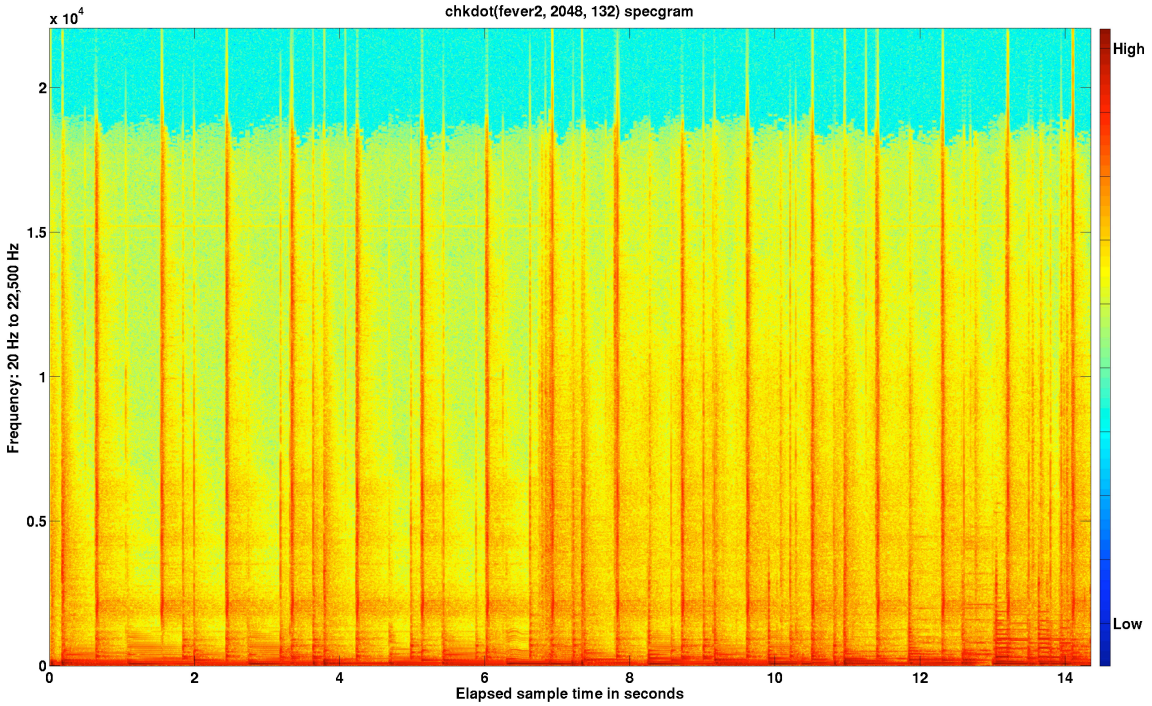

Figure 5.3.1.1 Specgram for Introduction to ***Fever***

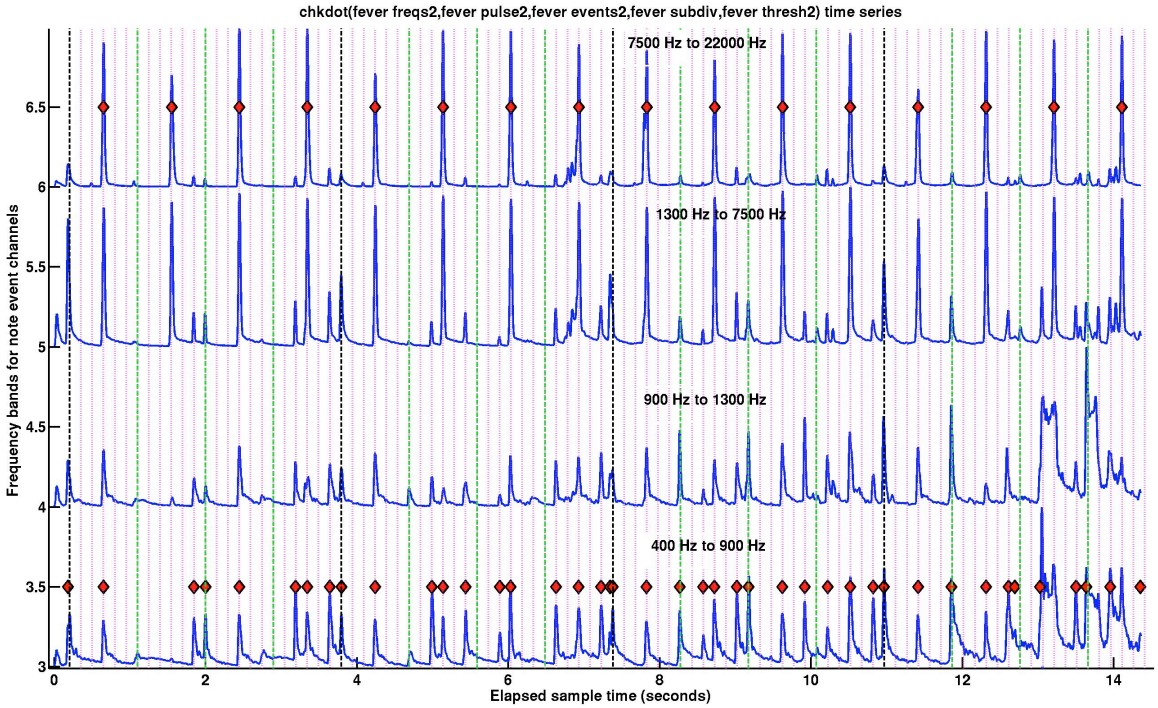

Figure 5.3.1.2 Time Series Plot for Events in Original Version of Fever

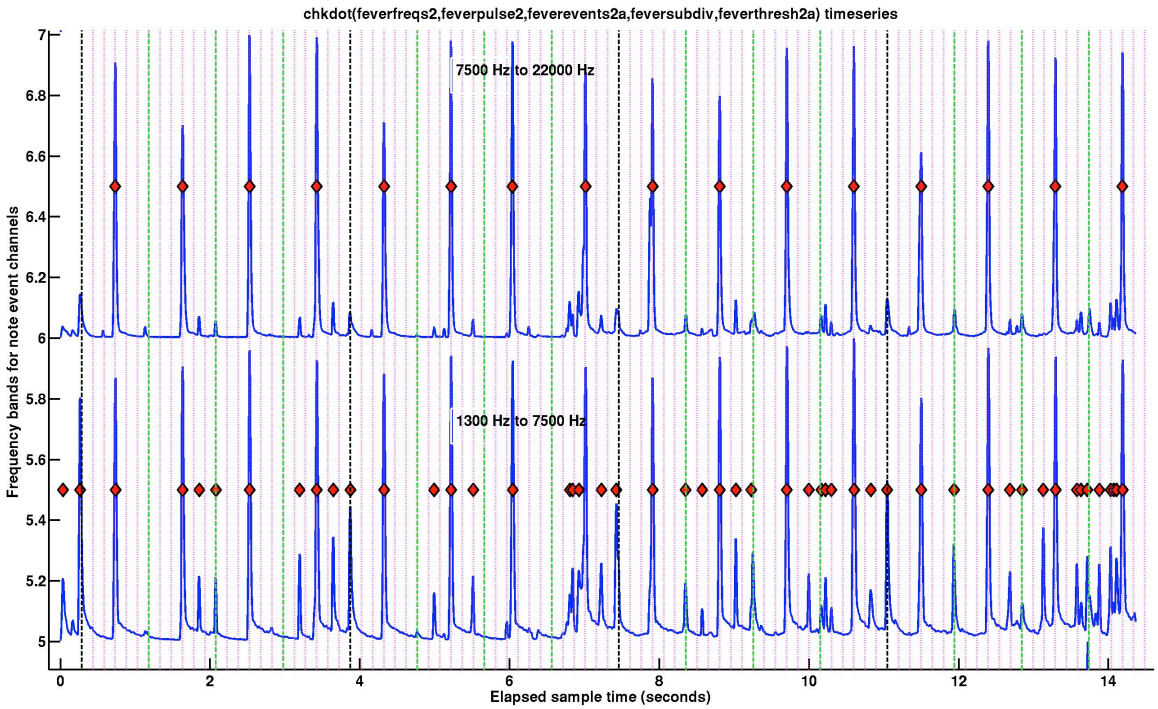

Figure 5.3.1.3 Time Series Plot for Events in Straight Version of Fever

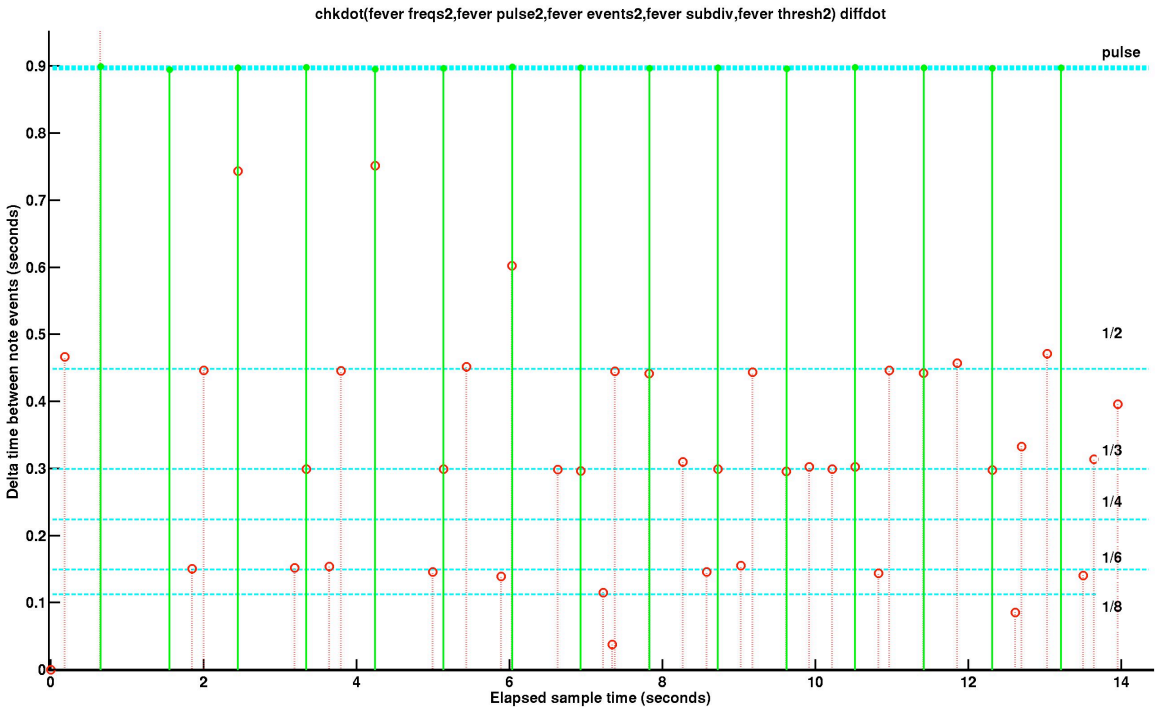

Figure 5.3.1.4 Note Timing Chart for Events in Original Version of Fever

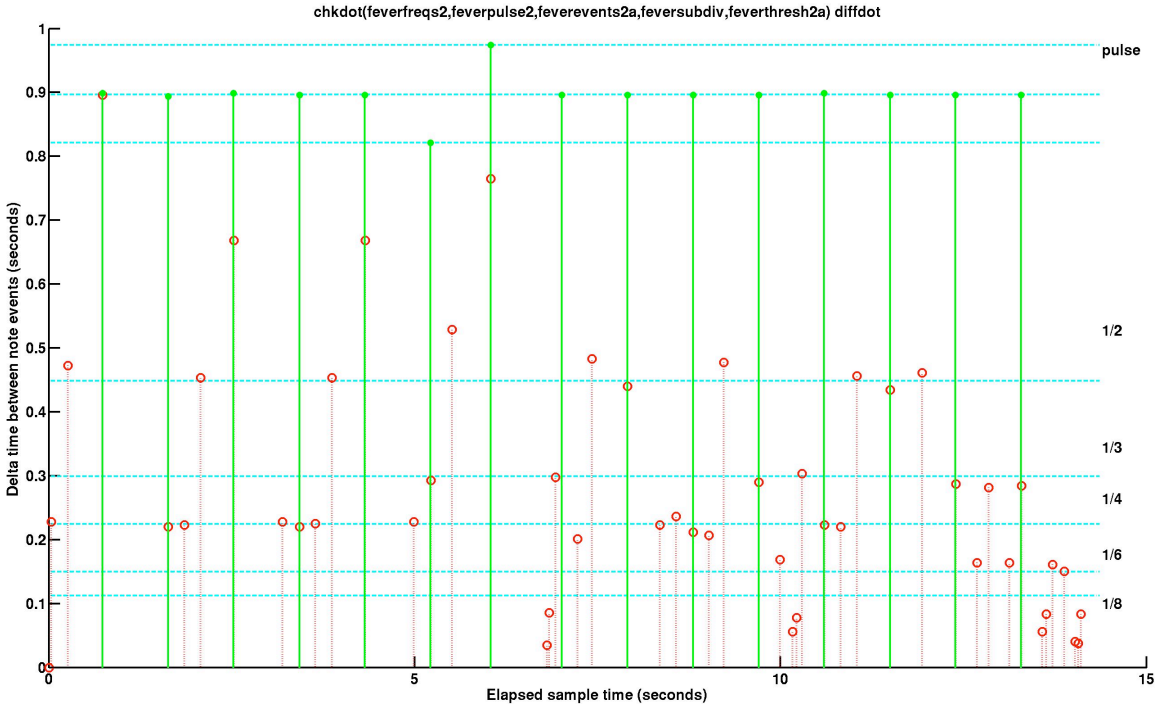

Figure 5.3.1.5 Note Timing Chart for Events in Straight Version of Fever

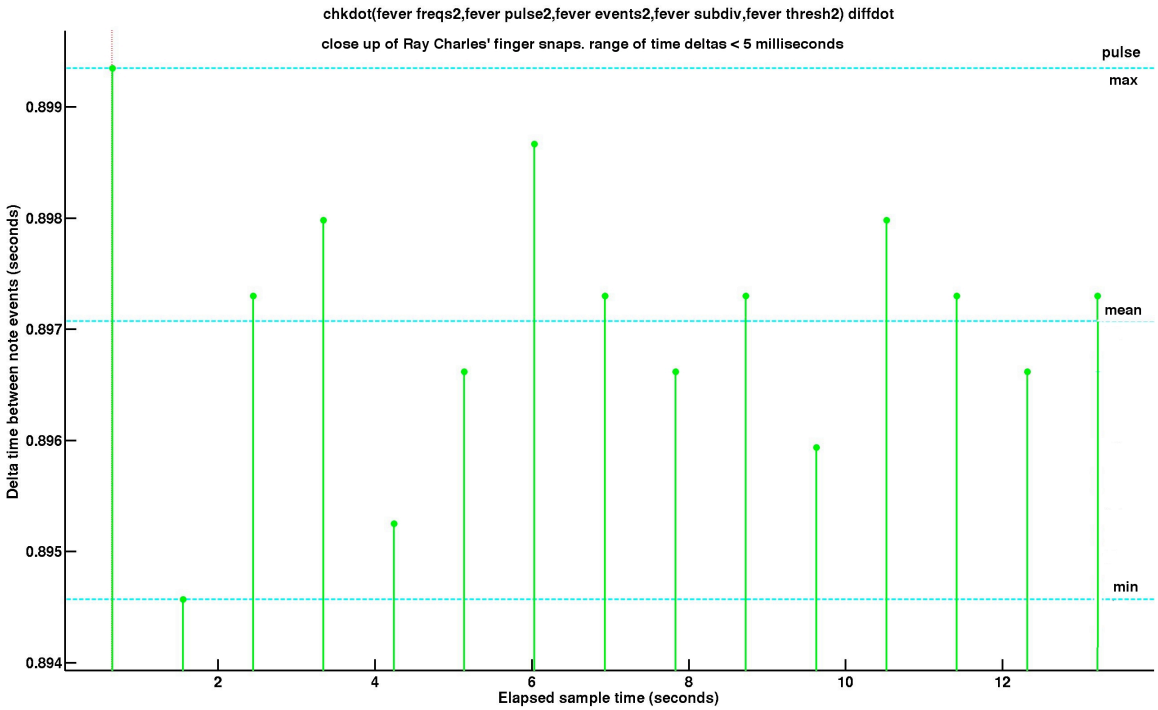

Figure 5.3.1.6 Close-up: `diffdot` Pulse Events for Original Version of ***Fever***
Variance of time deltas for Ray Charles' finger snaps is less than 5 milliseconds.

If you look closely at figure 5.3.1.2, in the pulse track at the top, straight up from the 8 second mark, you will see a small double peak. Figure 5.3.1.7 shows a close-up of this slight performance error in the second phrase.One of these events is a finger snap, and the other a conga. Everywhere else in this music sample, we found exact temporal alignment between these two instruments, but in this case, the conga plays 30 milliseconds too soon. We conclude that Ray Charles either did not hear this discrepancy during recording (unlikely), or that he was aware of it but found it acceptable. Indeed we challenge anyone to actually perceive it by direct listening (the audio sample is posted on the web). We include this anomaly because it represents an important data point in the specification of lower limits to the human audio perception system. Figure 5.3.1.8 shows an even closer view. The small bump that is visible between the first note event (conga) and the larger finger snap peak is the snare drum which has a small component of its sound in the higher frequency range where we measure the finger snap and conga.

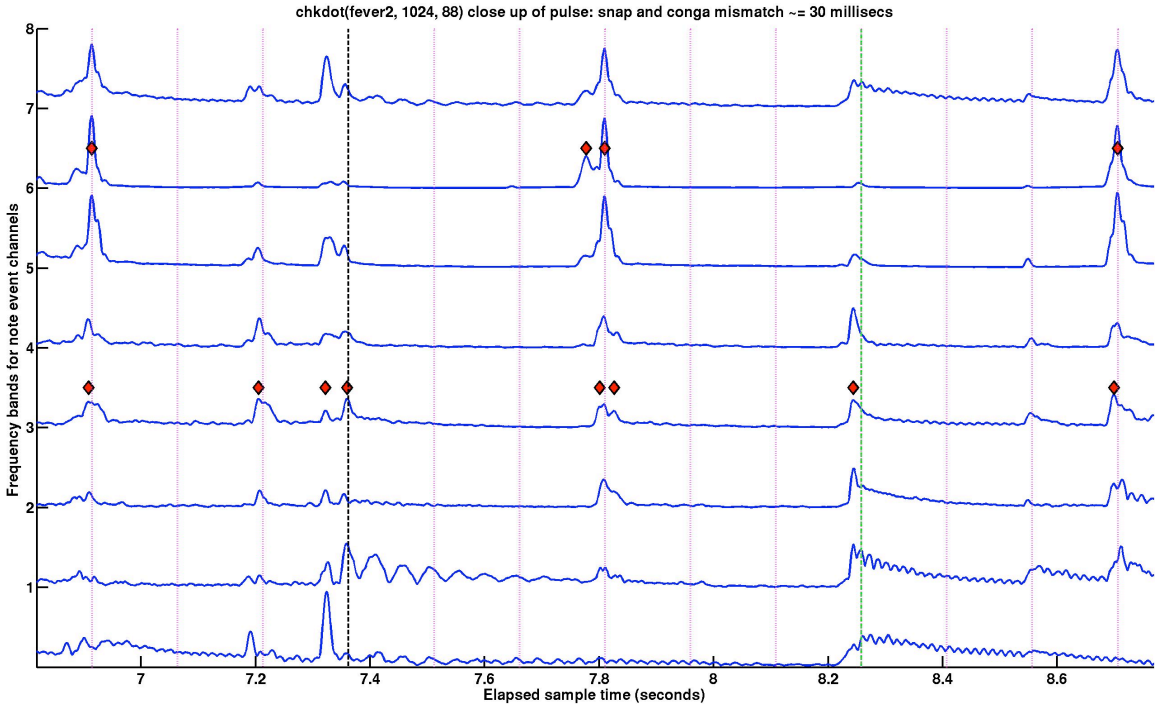

Figure 5.3.1.7 ***Fever*** missed conga note

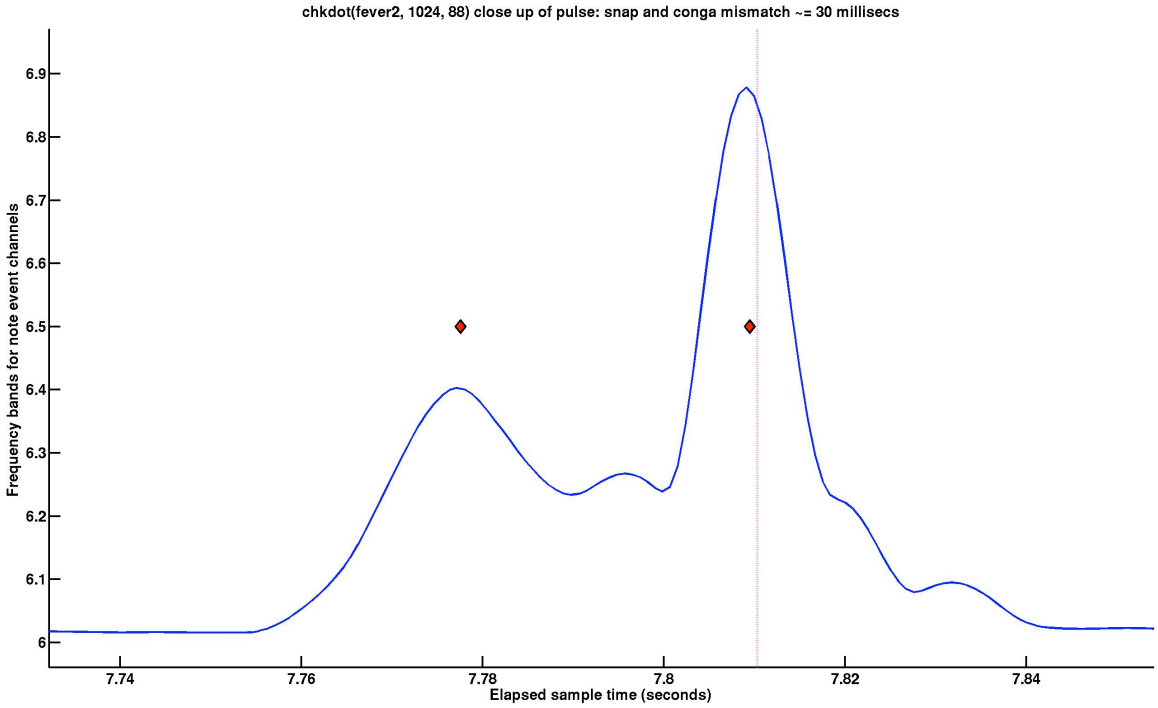

Figure 5.3.1.8 Extreme Close-up of Conga/Finger Snap Timing Anomaly

5.3.2 ***Graceland***: “Loose” Tempo

Graceland by Paul Simon is a good example of a swing feel that mimics riding on a railroad. We discovered that the several instruments in the introduction bounce around the MB beat locations in a loose swing while staying tightly synchronized with each other. Figures 5.3.2.1-3 show specgrams for the 8 bar intro to ***Graceland.*** The large scale specgram shows the subdivision of time clearly, but the spectrum appears quite broad and relatively featureless from the perspective of extracting note events. Zooming in on the low frequencies in the second and third plots show a great amount of detail visible below 1500 Hz. This shows how the resolution of the FFT is crucial for picking good frequency bands and features. The time resolution is about 10 milliseconds.

Figure 5.3.2.4 shows a ten frequency band `chkdot` time series plot for the ***Graceland*** sample. Bass drum is used as the pulse, and electric guitar is the secondary events channel. Figure 5.3.2.5-6 show `diffdot` plots of the time deltas for the pulse and guitar channels. Both event channels show significant variations in the timing. The pulse channel starts with greater variation, and settles into a somewhat tighter pattern by the second half of the sample (second 4 bar phrase). The electric guitar is much more consistent in timing variations. Close inspection shows approximately 50 millisecond range of time deltas in both event channels.

The `chkdot` subdivisions (green lines) show a triplet pattern. The pulse events in the lowest frequency band land close to the downbeat and backbeat MB lines, but drift slightly forward and backward in time. This gives a looser feel than the extremely tight swing in ***Fever*** which has every beat synchronized to less than 10 milliseconds. Looking at the ***Graceland*** `chkdot` plot, it is clear that basically every note event is played on a quarter beat subdivision. Ordinarily this would tend to sound somewhat square. The consistent variation in the electric guitar timing seems to provide a swing feel without any explicit presence of triplet subdivisions.

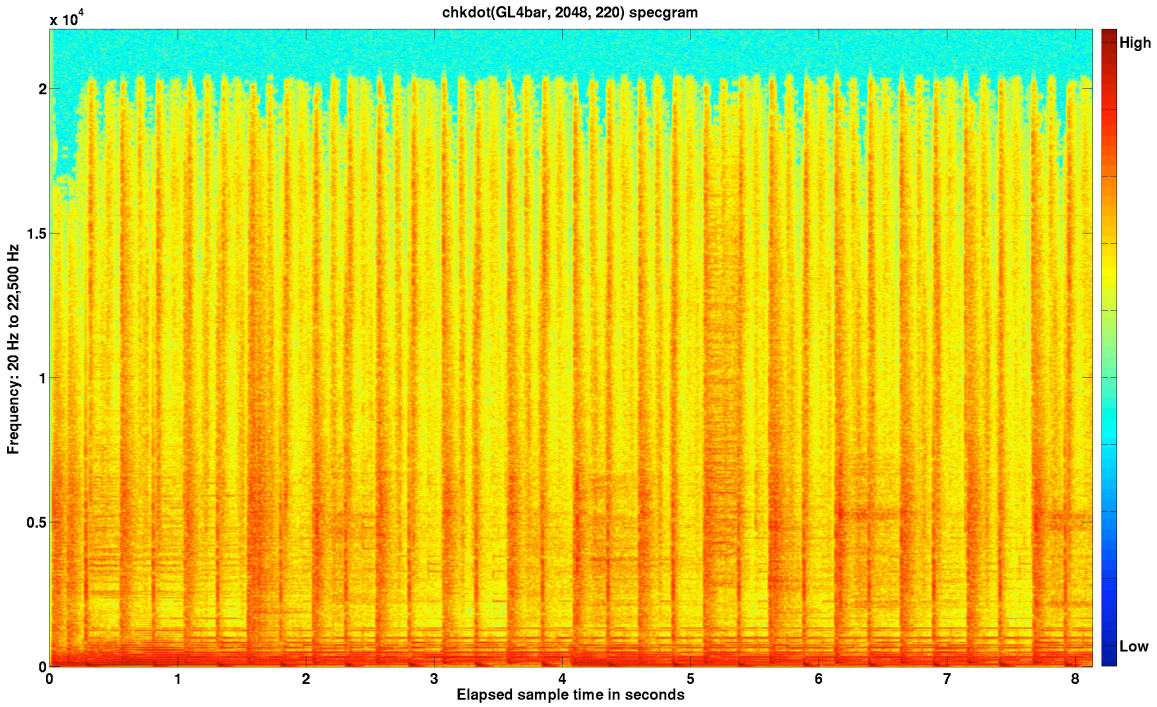

Figure 5.3.2.1 Specgram for ***Graceland***

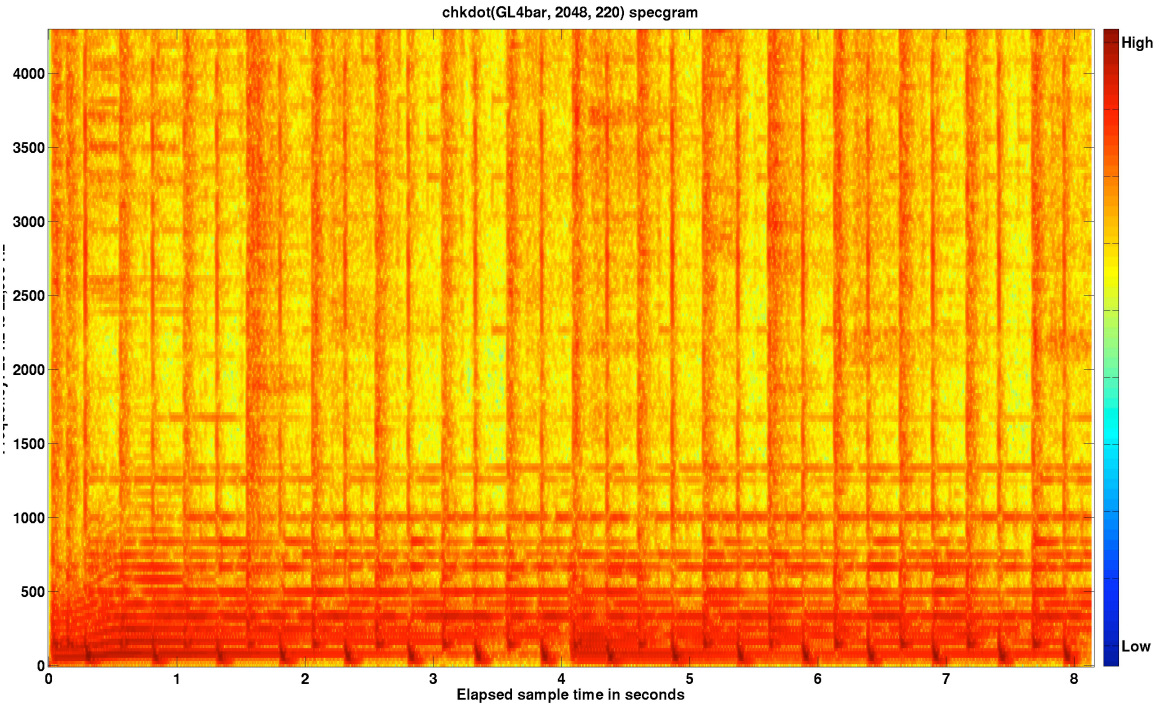

Figure 5.3.2.2 Specgram for ***Graceland*** (close-up one)

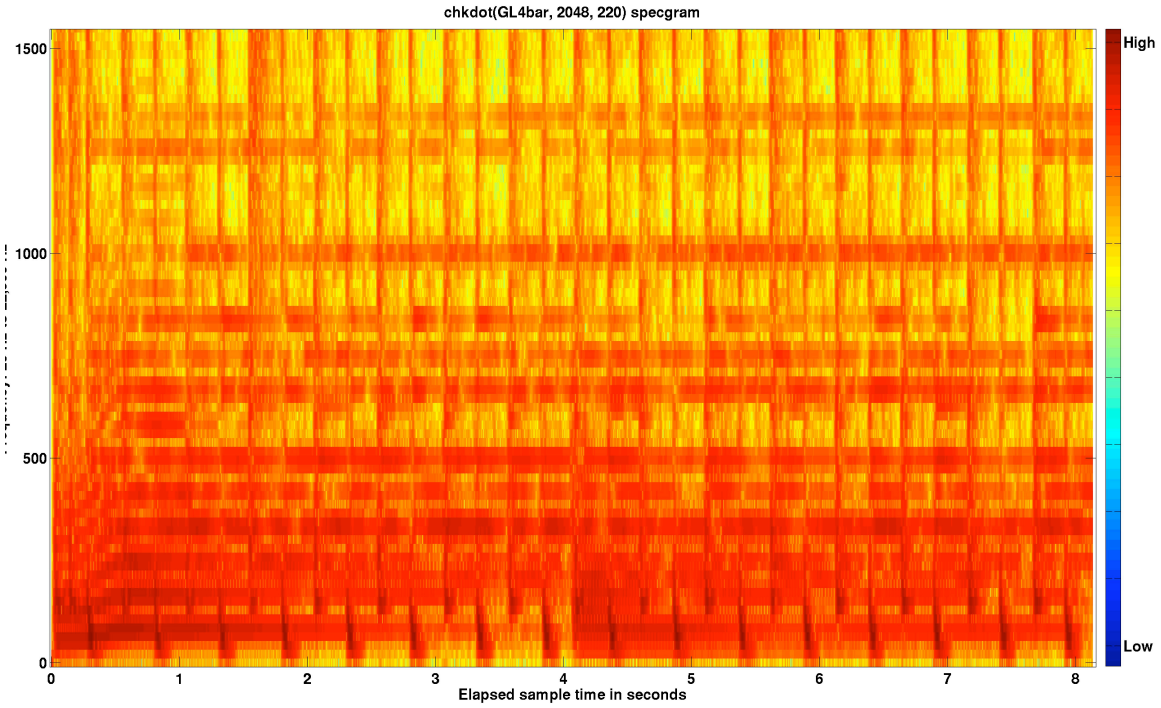

Figure 5.3.2.3 Specgram for ***Graceland*** (close-up two)

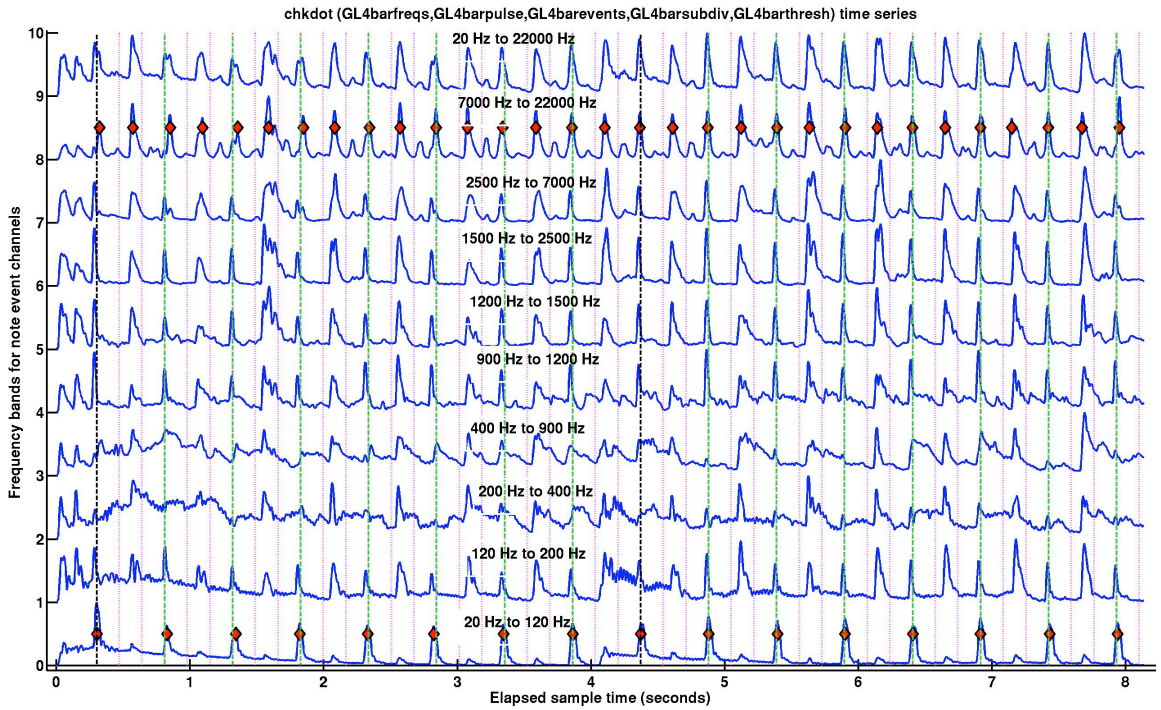

Figure 5.3.2.4 ***Graceland*** bass drum and electric guitar events

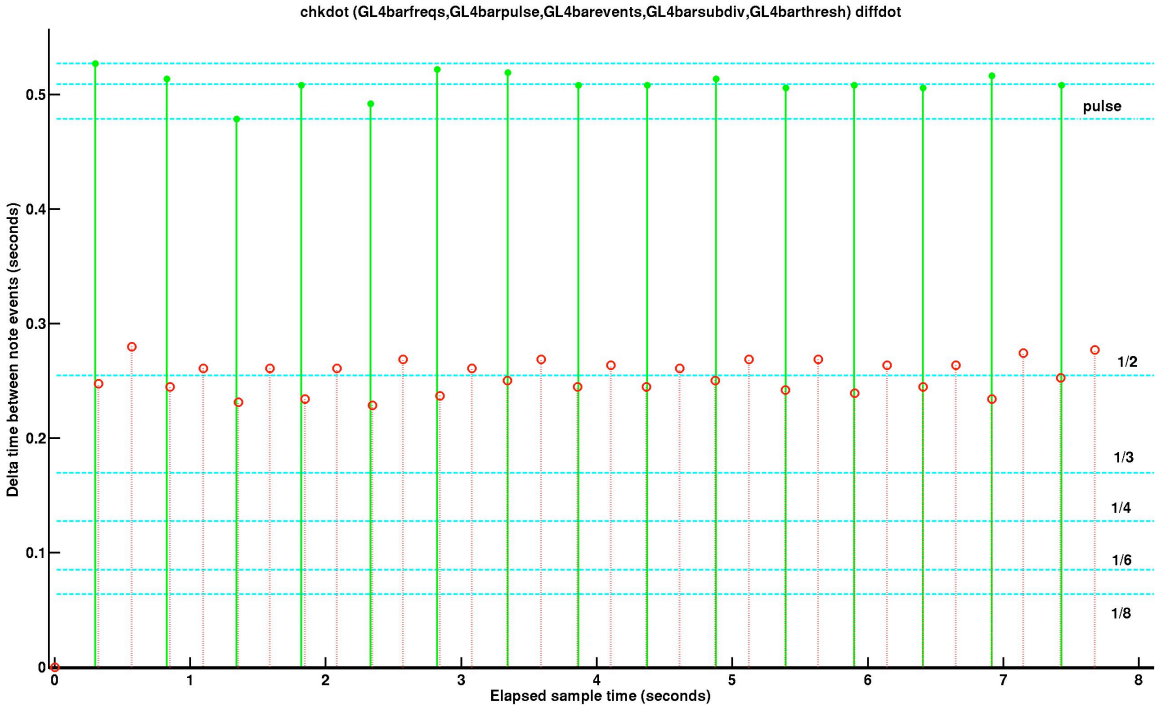

Figure 5.3.2.5 ***Graceland*** Note Event Time Deltas (`diffdot`)

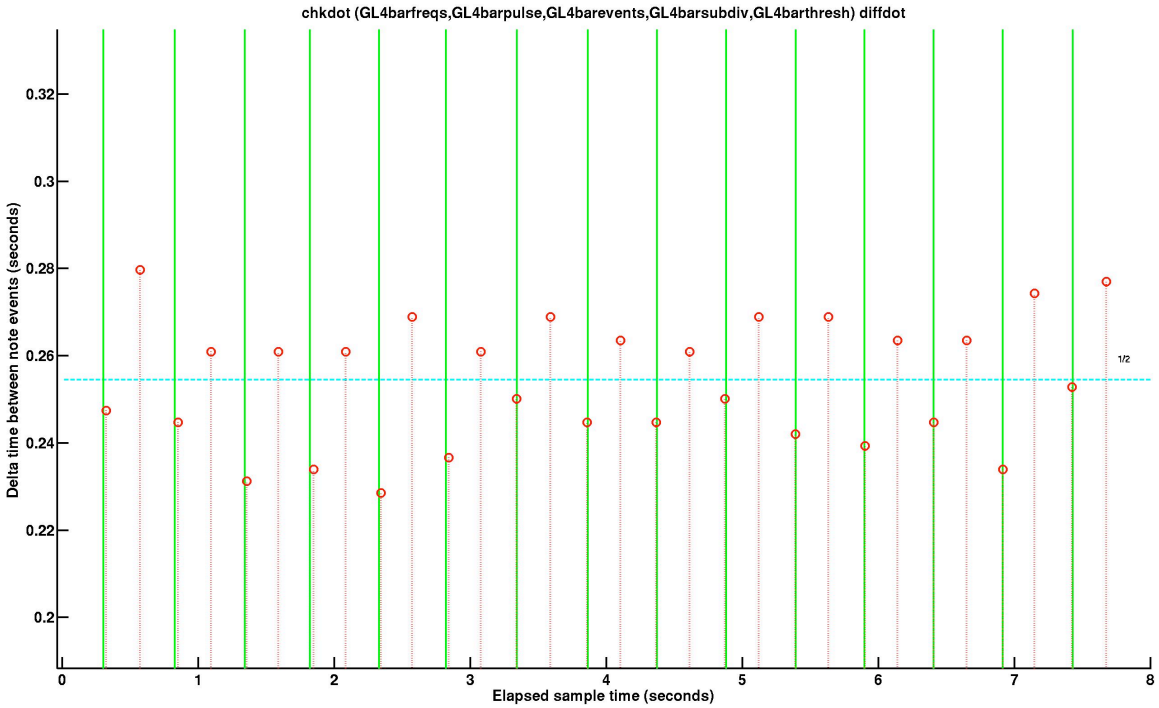

Figure 5.3.2.6 ***Graceland*** Close-up of Electric Guitar Time Deltas (`diffdot`)

5.3.3 Pandeiro

The pandeiro is a Brasilian hand drum very similar to the instrument called a tambourine in American music. Pandeiro can and does play almost every rhythmic part in Samba and Pagode: surdo, caixa, tamborim (not a tambourine, see Appendix), ganza (shaker) and cuica in addition to a variety of rhythms mostly unique to the pandeiro.[3]

The *basic* pandeiro batida is a simple 1-2-3-4 pattern played continuously with slight variations denoting which phrase of a larger pattern is being played. This pandeiro batida is invariably taught as straight time: *one-ee-and-uh* played with thumb (*one*), fingertips (*ee*), palm heel (*and*), fingertips (*uh*), over and over. American students generally have a difficult time learning to play the pandeiro. Part of this difficulty is related to posture: holding the pandeiro, Brasilian style, is as difficult as holding a violin, but with the stress on the left hand rather than chin and shoulder. The other difficulty, which became clear during the course of this research, is that most pandeiro teachers, whether Brasilian, American or other national origin, underemphasize the single most important insight: these four notes are *not* played with even time differences. As can be seen in the `chkdot` time series plots, the *uh* note is always played on the triplet pickup to the downbeat, rather than the straight quarter note. This timing variation gives the pandeiro batida a strong swing feeling. The triplet pickup to a downbeat or backbeat is a very common feature of American Swing, and also quite common in Brasilian music. In addition, the second and third notes (*ee* and *and*) are played in two very odd locations in the first half of the phrase. Neither of these is played on either a triplet, quarter or eighth note location, and there are slight time variations between repetitions of the basic batida. The pattern of these time variations is consistent by some measure, since in the `diffdot` plots the pattern is clearly a repeating waveform, rather than some kind of random pattern.

---

[3] Airto Moreira, Carlinhos Pandeiro de Oura, Jovino Santos Neto, professional Brasilian musicians/composers.

In addition to the sub-phrase temporal variations, most pandeiro players also play a slight timing difference between the length of the first and second phrases of the pandeiro batida, which enhances the lopsided swing feeling. The `diffdot` pulse events plot also shows this larger scale timing pattern. A correspondence can be imagined between the two `diffdot` patterns. This correspondence relates to making the overall composite pattern feel like a smooth swing, rather than pulling on the rhythm. The difficulty of describing this adequately in language reinforces the assertion that swing is an intuitive feeling rather than an analytical construct of counting exact subdivisions. We could analyze the timing patterns exhaustively, but it wouldn't help play the batida correctly.

The typical explanation of all this hierarchical coupling of temporal patterns, after the student has become semi-competent at holding the instrument and playing the basic notes, is that the pandeiro teacher says "*Now, play with swingee!"*

In figures 5.3.3.1-8 we show plots contrasting the original swingee version of the batida with a straight version generated in a MIDI file. The first two figures are specgrams of the spectra of the samples. The spectra are relatively the same, but the spacing of the vertical bands that represent the note events are more evenly spaced in the straight version. The second set of plots (`chkdot`) show time series of the audio signal decomposed into three frequency sub-bands. The pulse (*one* notes) is shown in the bottom frequency band. We detect these downbeat and offbeat notes from the strong low frequencies generated when the thumb hits the pandeiro skin. The other notes have the high frequencies of the jingles rattling. The *one* notes are played with two tonal variations to demarcate the two sides of the two bar phrase, called the open tone and the closed tone. Since the *one* notes also cause jingles to rattle, these notes show up in the frequency band for *ee, and, uh* notes. Observe how the note events in the straight version line up with the quarter subdivision lines. In the original swingee version, the only note event that is on an MB quarter note subdivision is the pulse.

The small amount of variation visible in the straight version is due to using hand edited samples that have slight artifacts, and using more than one sample version of each note event so the note events are not all generated by identical samples for their position in the batida: *one, ee, and, uh*. The swingee samples, played by a human musician, show temporal variations at several hierarchical levels of the rhythmic structure. `diffdot` shows about 5% - 8% variation in the time delta between downbeats, with a clear repeating pattern that resembles a slightly modulated sine wave. This set of time variations could be roughly modeled by using the swing ratio concept, if it was extended to include variations which are more than a simple ratio. The subdivision notes (*ee, and, uh)* show a more complex variation which is clearly beyond the swing ratio to model adequately.

Next are the `diffdot` plots, which show the time difference between adjacent note events on the ***Y*** axis. The green vertical markers represent the series of downbeats and the red markers are the *ee, and, uh* events. Elapsed sample time is on the ***X*** axis. The straight version has all non-pulse notes clustered around the 1/4 subdivision line. The swingee version shows a repeating pattern of timing variations for both the pulse and non-pulse note events. While the *uh* note events in the `chkdot` plots are quite precisely on the triplet pickup MB markers, in the `diffdot` plots these are spread between the 1/3 and 1/4 subdivision markers. This is because the `chkdot` markers are on the absolute MB time subdivisions, while the `diffdot` markers are relative to each other, and so the `diffdot` subdivisions depends on the timing variations in the pulse events, giving a broader spread of time deltas than in the `chkdot` plots.

These complex time variations are typical features of swingee for all the samples of Brasilian music we've analyzed. Keep in mind that the basic pandeiro part is one of the simplest rhythms found in Brasilian music. The swing ratio model of timing variations is completely inadequate to describe these types of rhythms. In chapter 6, we explore an idea for generating these temporal variations using Fourier series.

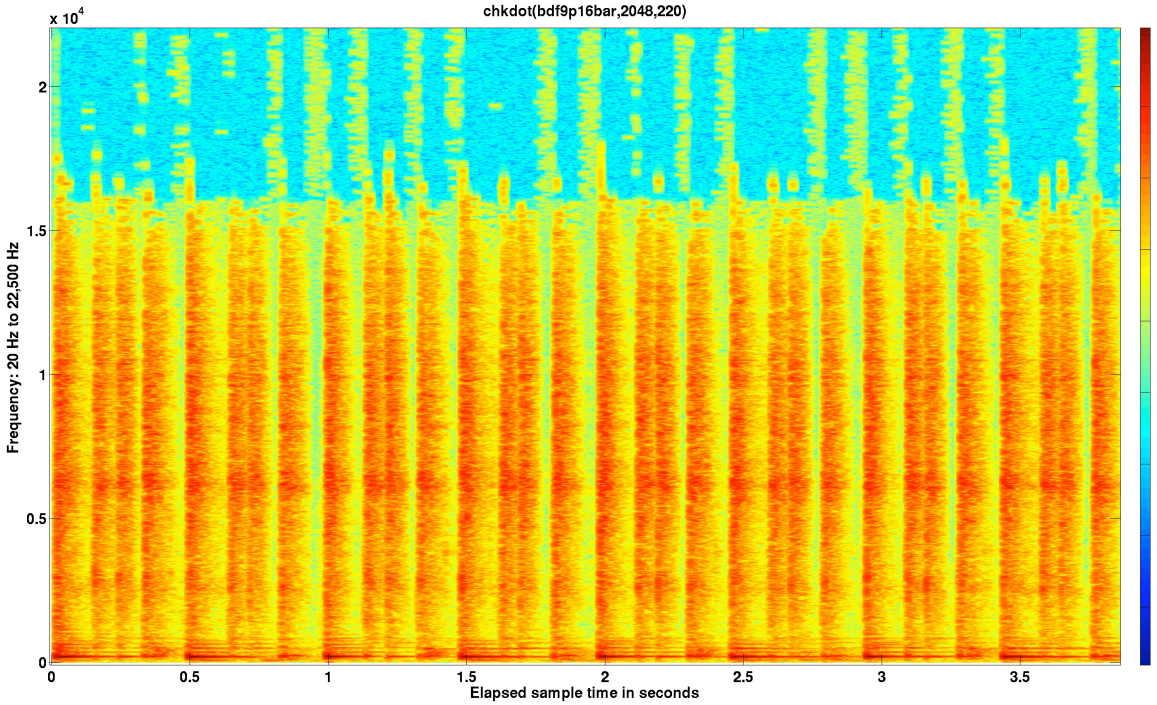

Figure 5.3.3.1 Specgram of Swingee Pandeiro Batida

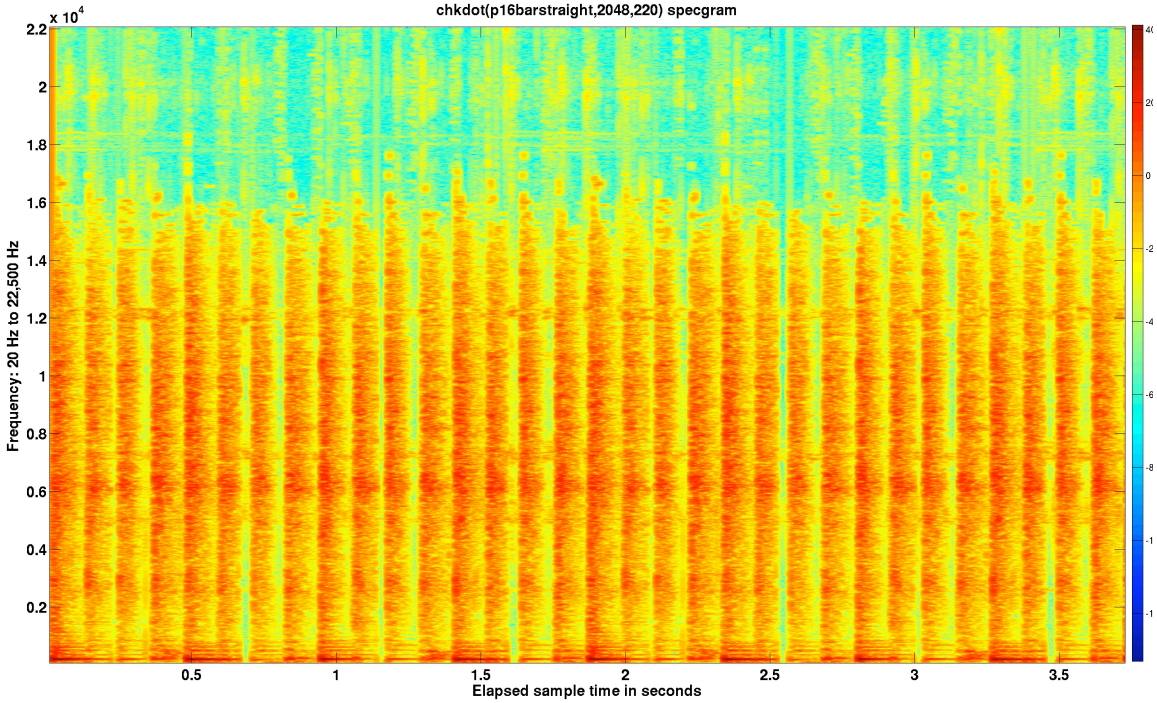

Figure 5.3.3.2 Specgram of Straight Pandeiro Batida

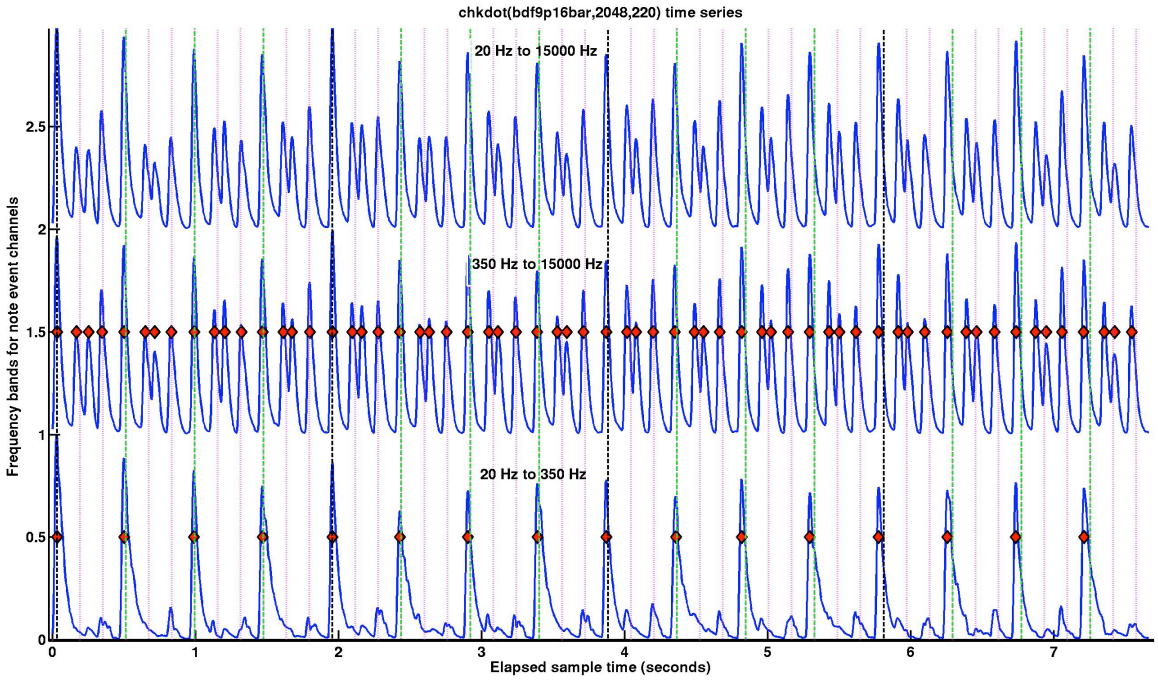

Figure 5.3.3.3 Time Series Plot for Events in Swingee Pandeiro Batida

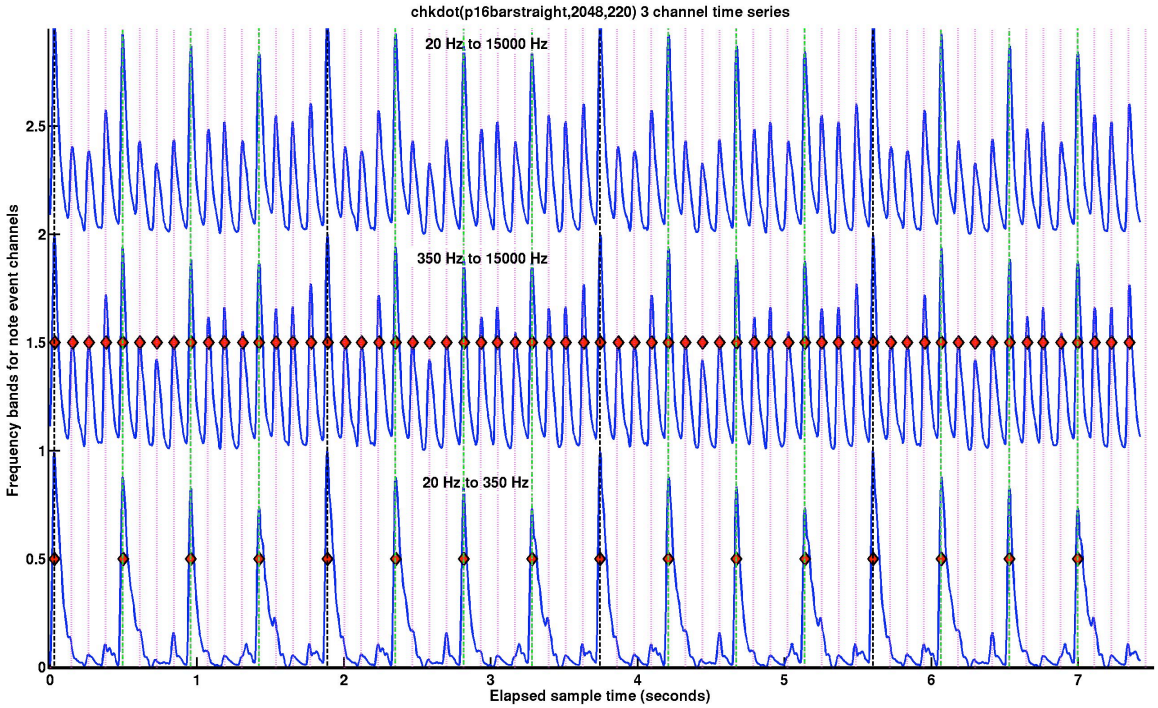

Figure 5.3.3.4 Time Series Plot for Events in Straight Pandeiro Batida

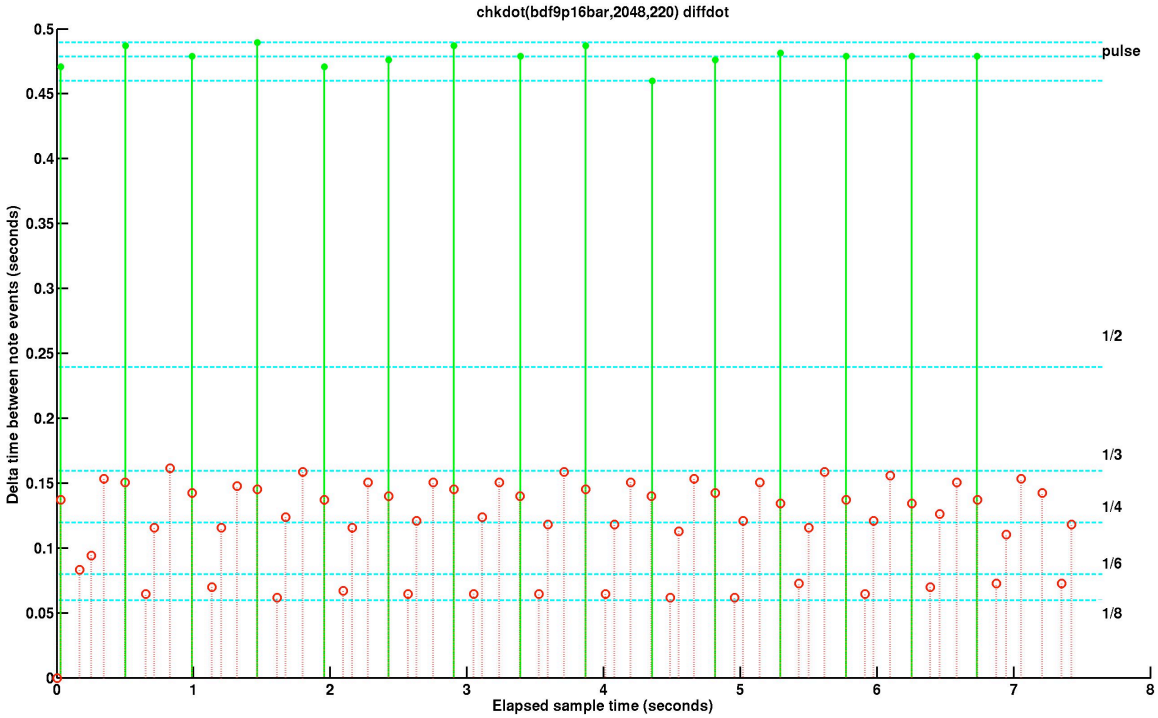

Figure 5.3.3.5 Note Timing Chart for Events in Swingee Pandeiro Batida

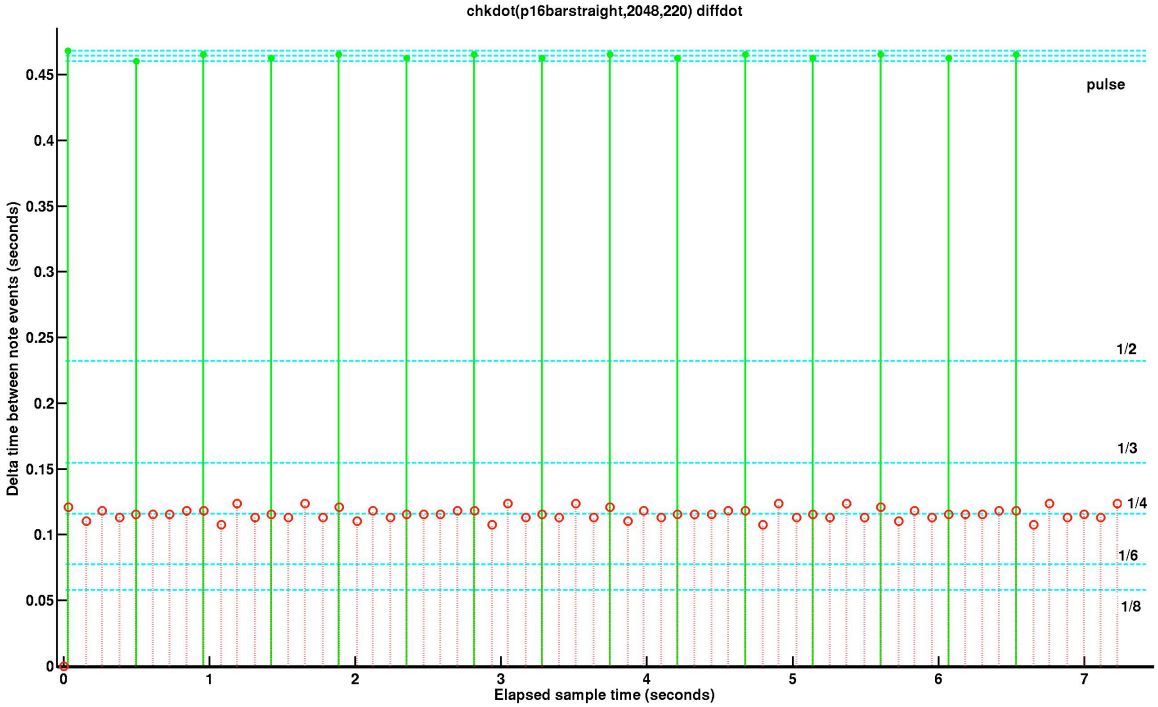

Figure 5.3.3.6 Note Timing Chart for Events in Straight Pandeiro Batida

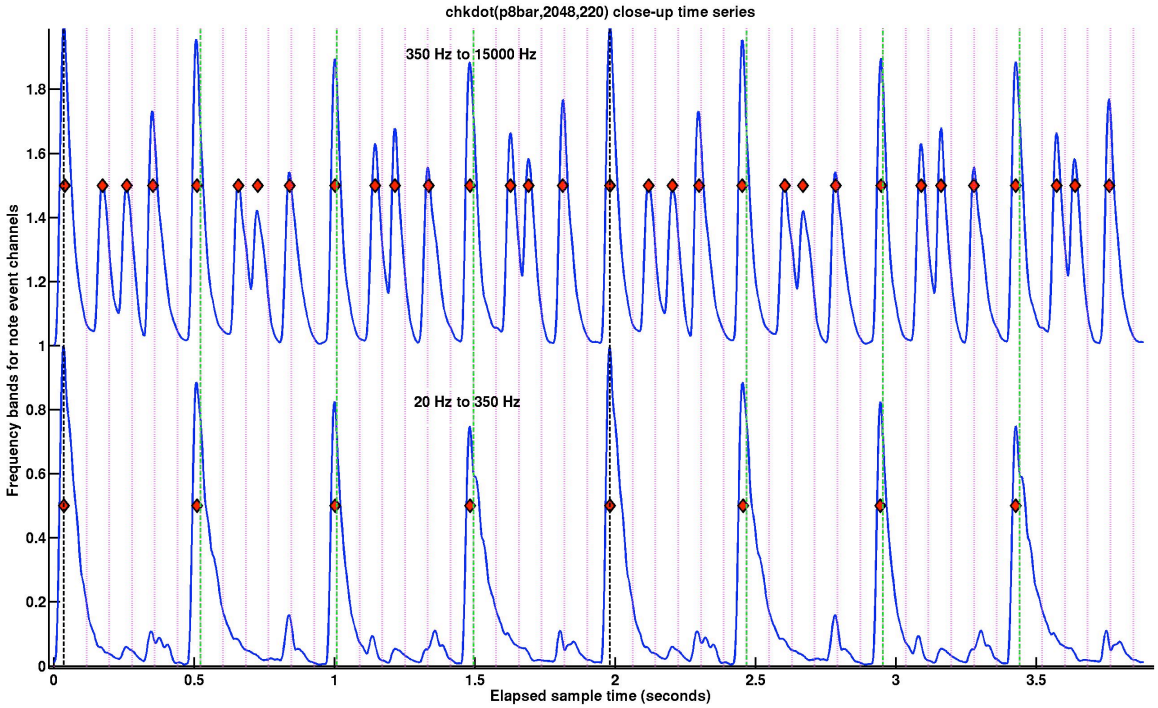

Figure 5.3.3.7 Close-up of Events in Swingee Pandeiro Batida

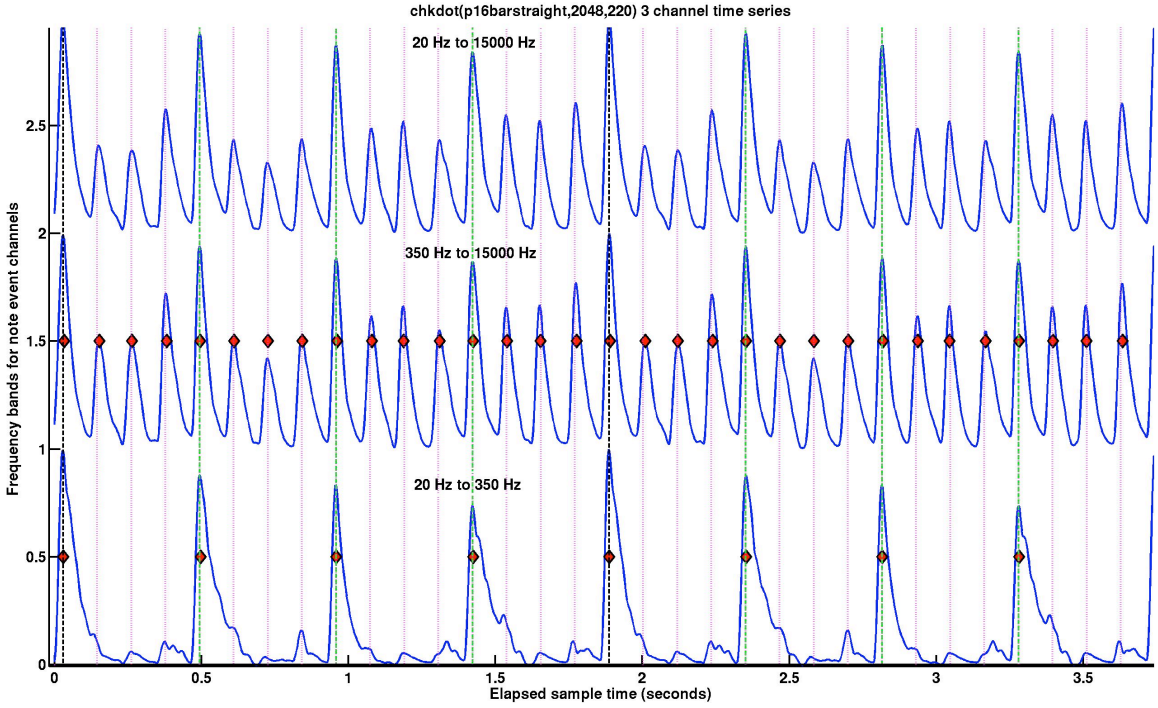

Figure 5.3.3.8 Close-up of Events in Straight Pandeiro Batida

5.3.4 *It Don't Mean a Thing if it Ain't Got that Swing*

Figure 5.3.4.1 is a time series plot for the beginning of Duke Ellington and Louis Armstrong's 1962 performance of ***It Don't Mean a Thing if it Ain't Got that Swing***. The upper event track shows the hi-hat cymbal sound as the drummer fades himself into the mix by playing slightly louder with each beat -- no overdubs or mixer board fading here. You can see how hi-hat note events start the phrase slightly off from the MB time locations and then home in on the exact time location of the triplet pickup to the beat.

Figure 5.3.4.2 shows the `diffdot` plot. The pulse timing shows some variance, but the red events (hi-hat) are tightly clustered on the 1/2 and 1/3 subdivisions, with a third cluster midway between the 1/6 and 1/8 subdivision. Figure 5.3.4.3 shows a larger view. The trumpet note events are visible in frequency bands 3,4 and 5. Figures 5.3.4.4-6 are specgrams showing both the rhythm section and Louis Armstrong's trumpet solo.

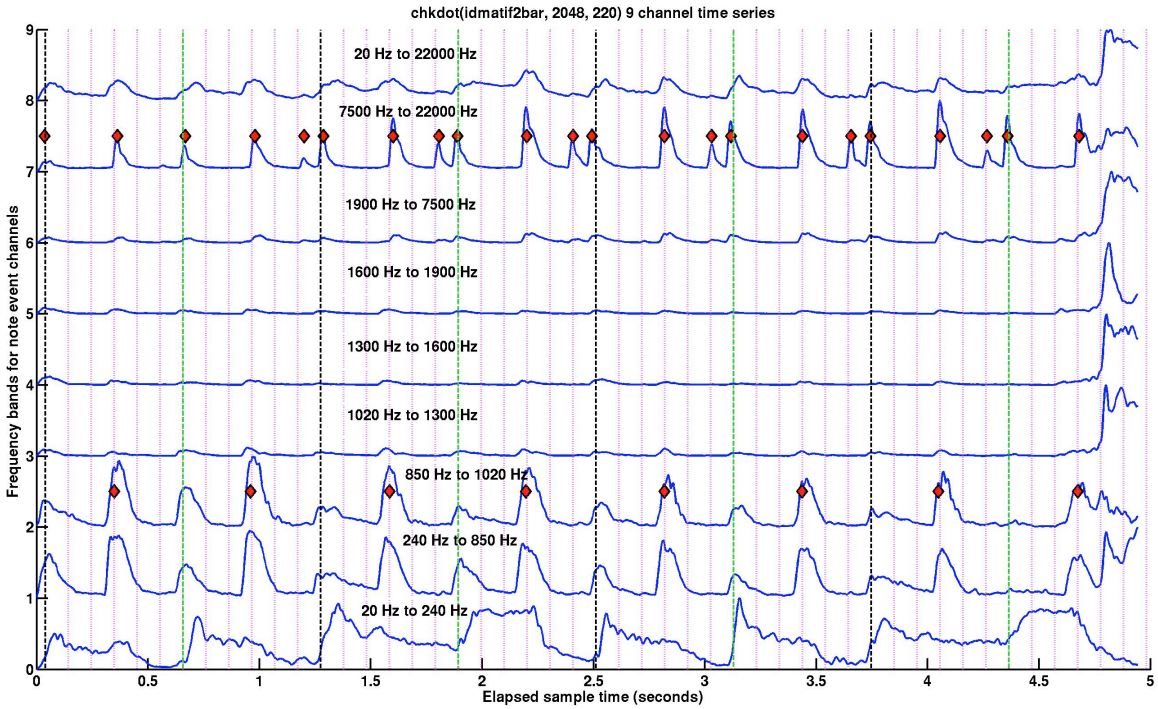

Figure 5.3.4.1 Events for ***It Don't Mean a Thing if it Ain't Got that Swing***
Hi-hat cymbal events marked in upper row, piano/bass in lower rows.

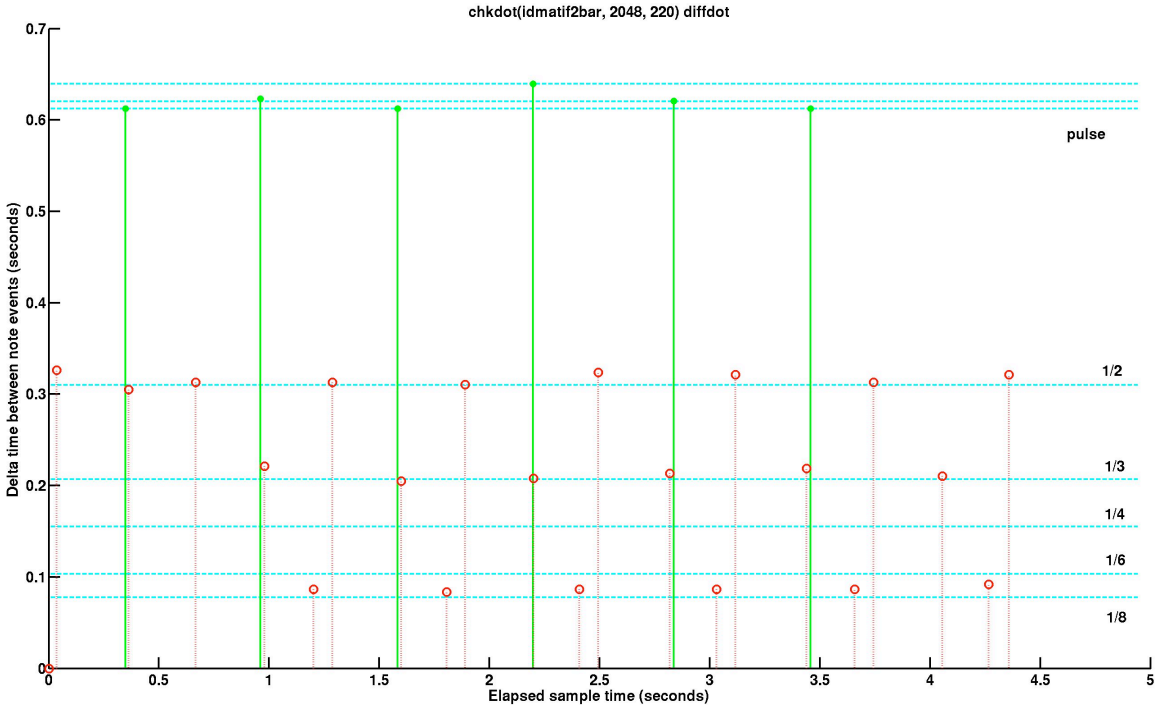

Figure 5.3.4.2 Event Times: ***It Don't Mean a Thing if it Ain't Got that Swing***

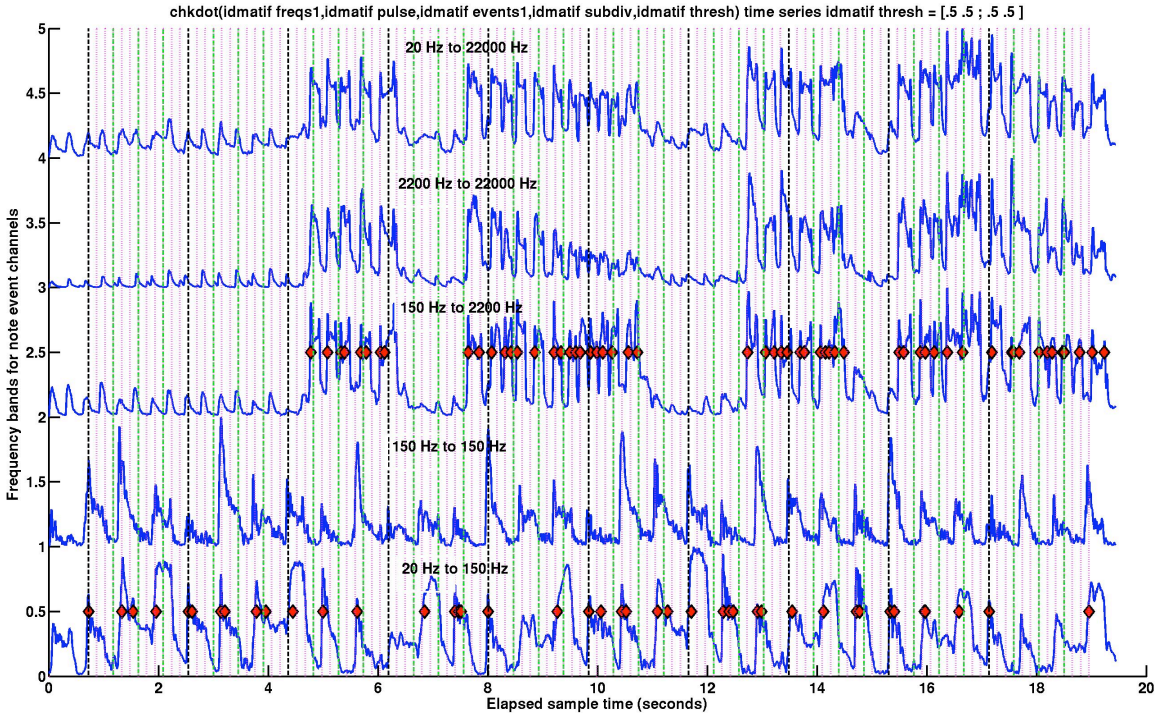

Figure 5.3.4.3 Time Series Plot Showing Rhythm and Trumpet Events

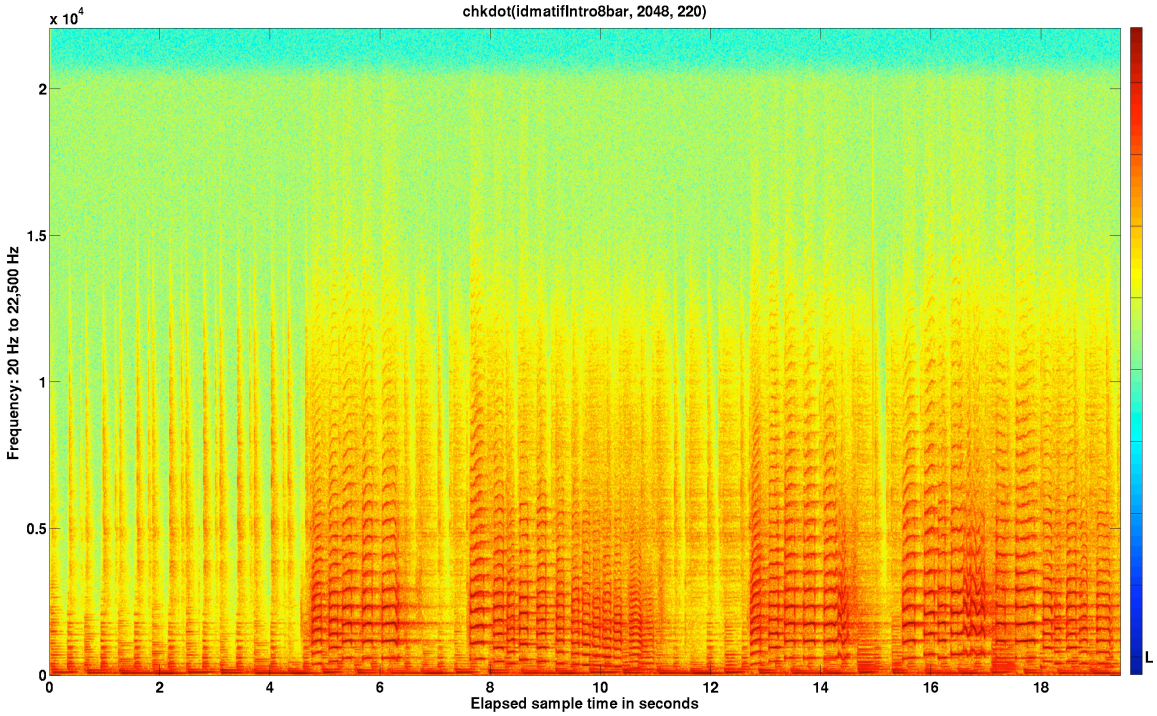

Figure 5.3.4.4 Specgram of Intro for ***It Don't Mean a Thing if it Ain't Got that Swing***

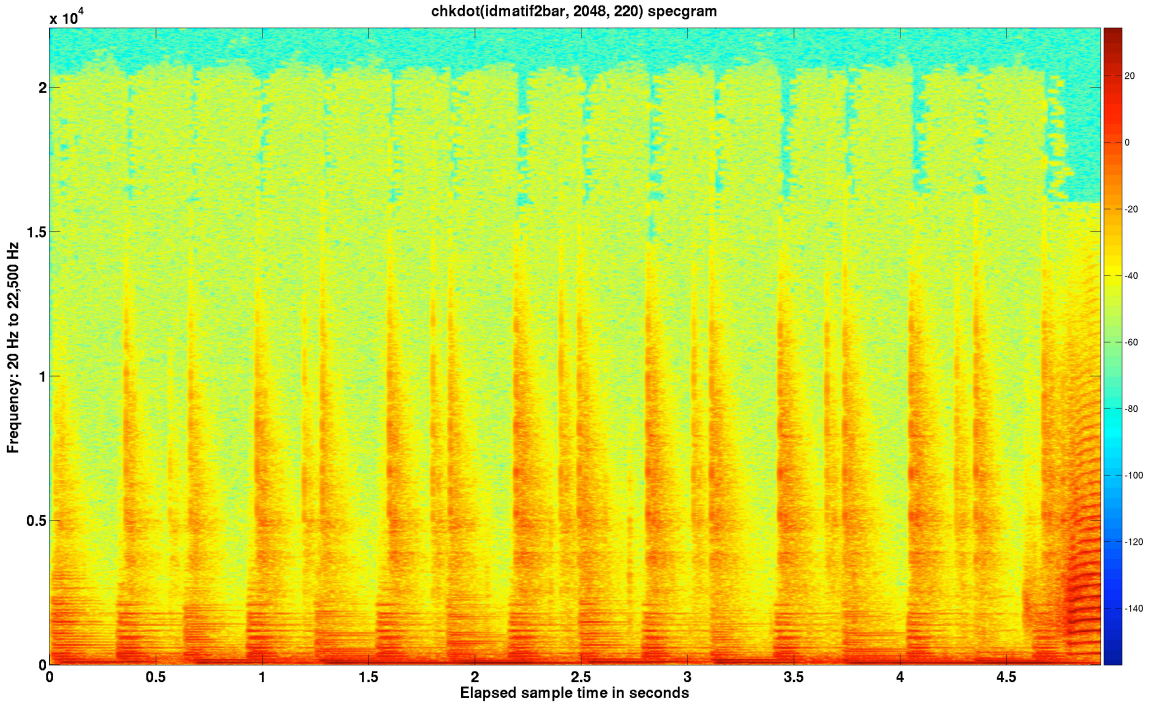

Figure 5.3.4.5 Close-up of Specgram of Intro Showing Piano and Drums

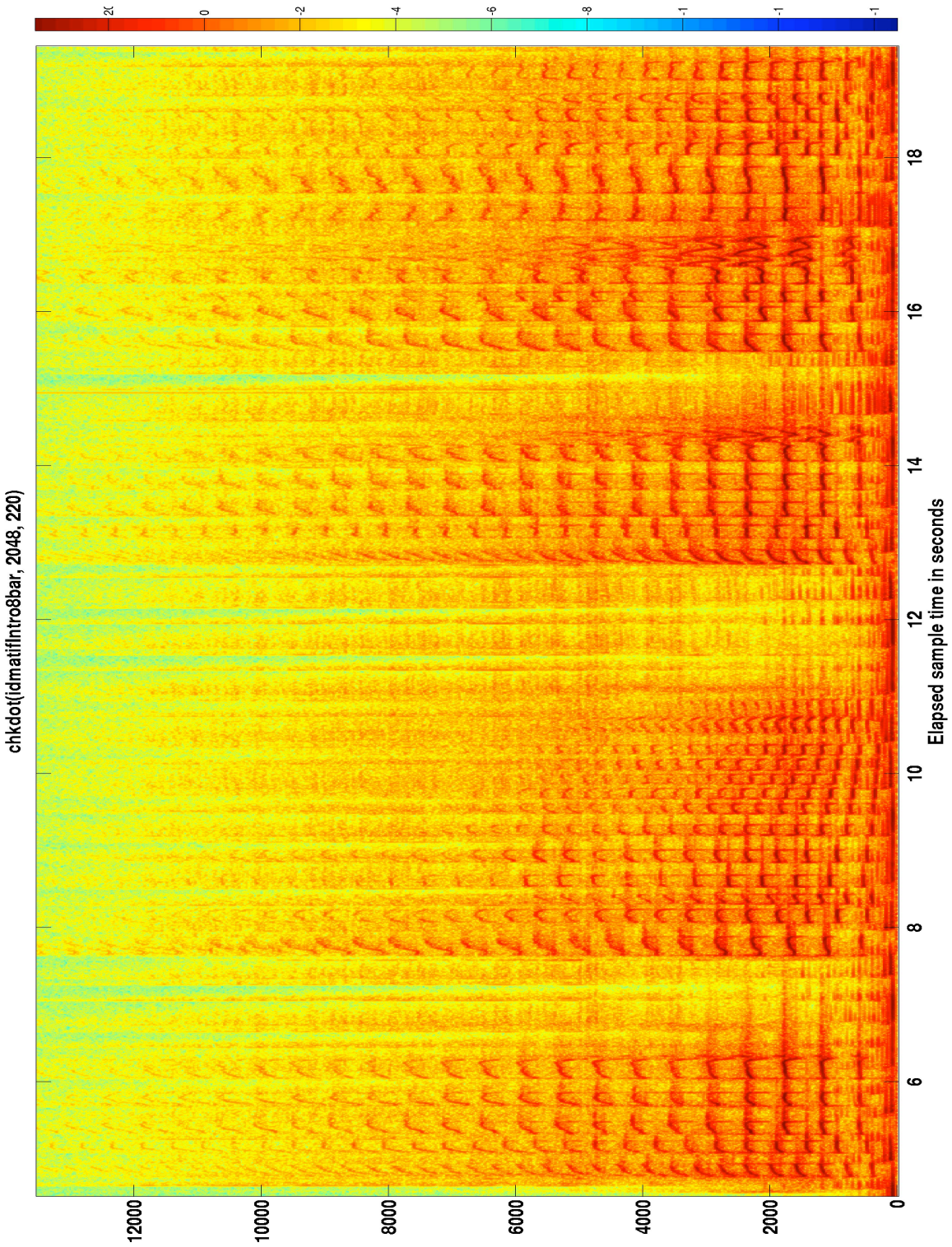

Figure 5.3.4.6 Close-up of Specgram Showing Trumpet Note Events

5.3.5 Tamborim Batida: Playing Around the Beat

Figure 5.3.5.1 shows slight timing variations in the performance by a tamborim player. We discovered these during extreme close inspection. In this sample, the pandeiro plays the principle beats (downbeats and offbeats) in the lower plot. The other notes in the pandeiro batida are not shown but are the same as in the previous analysis of the pandeiro batida. Note that the principle beats are not all exactly on MB subdivisions. This intentional and quite precise looseness is part of the swingee style. Both the tamborim and the pandeiro play some notes exactly on the MB subdivisions and some notes slightly off, generally ahead of the beat. These variations are typically between about 20 milliseconds and 50 milliseconds.

In the upper plot when the tamborim starts playing, it is not at the standard beginning of the batida. Instead the drummer plays a variation on a portion of the second half of the entire tamborim phrase, which leads into the downbeat. The downbeat is indicated by the green marker at time location 1700, except there is a further variation -- it is not the primary downbeat but the offbeat, so the tamborim is playing on the opposite side from the pandeiro. This is not however, the *wrong* side. It is very common in Brasilian music for some two phrase batidas to be played with the two phrases swapped. This is analogous to the 3-2 clave and 2-3 clave style in Cuban music. Swapping the sides gives a different feel, usually more syncopated if the unfamiliar variant is played.

The tamborim batida is very syncopated even when played straight. The “standard” place to start the basic tamborim batida is at note event #6 in figure 5.3.5.1 at temporal location 1700, very slightly ahead of the beat. Many batidas have beats that are played ahead of the MB subdivision beat, and/or also slightly ahead of or behind the note events of other instruments. In this example, at this temporal location, the pandeiro plays about 30 milliseconds ahead of the MB subdivision downbeat, and the tamborim plays about 15 milliseconds ahead of the pandeiro. This is not accidental but is used to give a push to the feeling of the rhythm by both instruments. A few beats on either side of the

1700 point, both instruments have notes that are played exactly on an MB subdivision. The feeling of this pattern is quite consistently the same throughout the entire sample which is several minutes long.

Looking at the two sets of three evenly spaced notes starting at 1700 and 2000, note that the first and third beats are slightly ahead of where they would be if played exactly according to *some* even MB subdivision, however complex the subdivision might be. To reiterate, these beats push the rhythm slightly and give a somewhat more energetic feeling to the music than if they are played "straight". This is what we referred to previously as parallel time shift of non sequential but related note events. In this case, these two tamborim note events are also accented, further emphasizing the push to the rhythm at these two time points. The combination of time push and accent are caused by the tamborim player putting a little extra "juice" into the rhythm for these note events. (Waadelund, 2004) has studied the relation between this type of "body english" and the rhythms played by drummers on drum kits. The investigation of the relation between motion and rhythm started in the early 20th century. (Seashore, 1938) and (Gabrielsson, 1987) both include a variety of reports, insights and opinions about this phenomenon.

In our example, the tamborim plays the first beat right on top of the pandeiro on the "real" downbeat, instead of playing at the "standard" temporal location for the note. This portion of the batida starts its repetition at the ninth event location (time 2000, triplet pickup to downbeat), just before the main downbeat, marked by the black line at time 2050. You can see that the first beat ordinarily is on the triplet pickup to the downbeat, and the next two beats are *almost* exactly evenly spaced on the subsequent triplet time points. The slight variations from playing exactly on temporal locations that correspond to an MB subdivision are part of the swingee style. While there is some looseness similar to the ***Graceland*** example, generally Brasilians play these slight temporal variations quite precisely, consistently and intentionally.

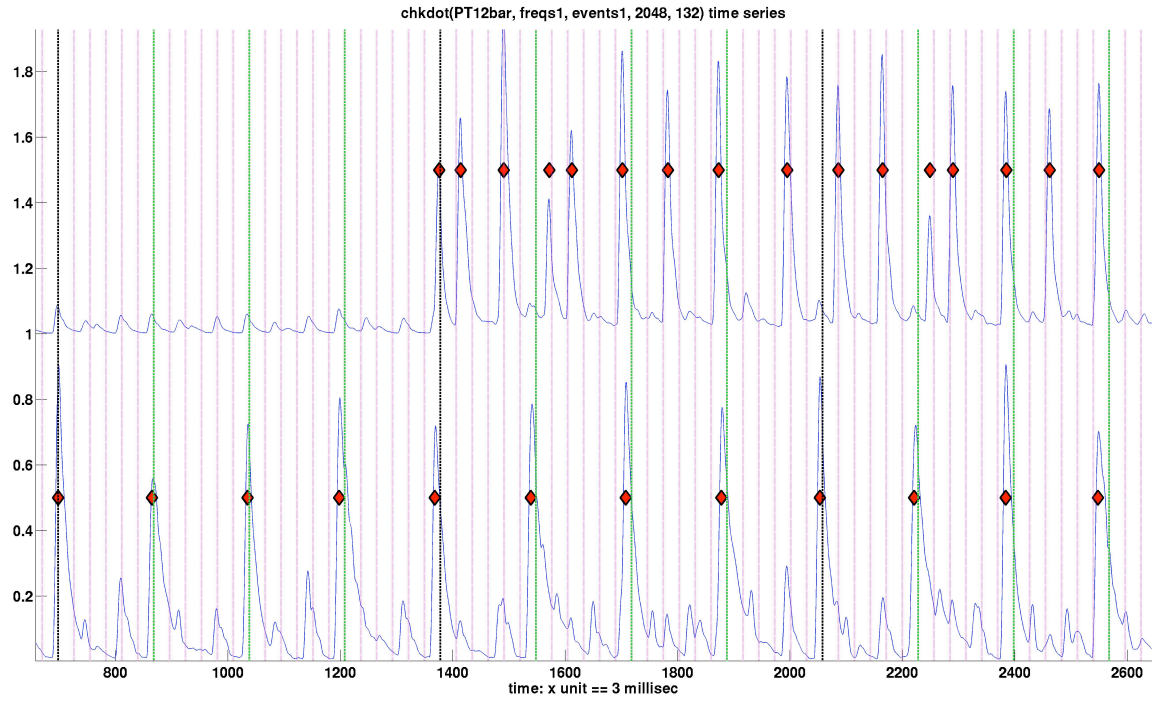

Figure 5.3.5.1 ***Tamborim Batida***: Playing Around the Beat

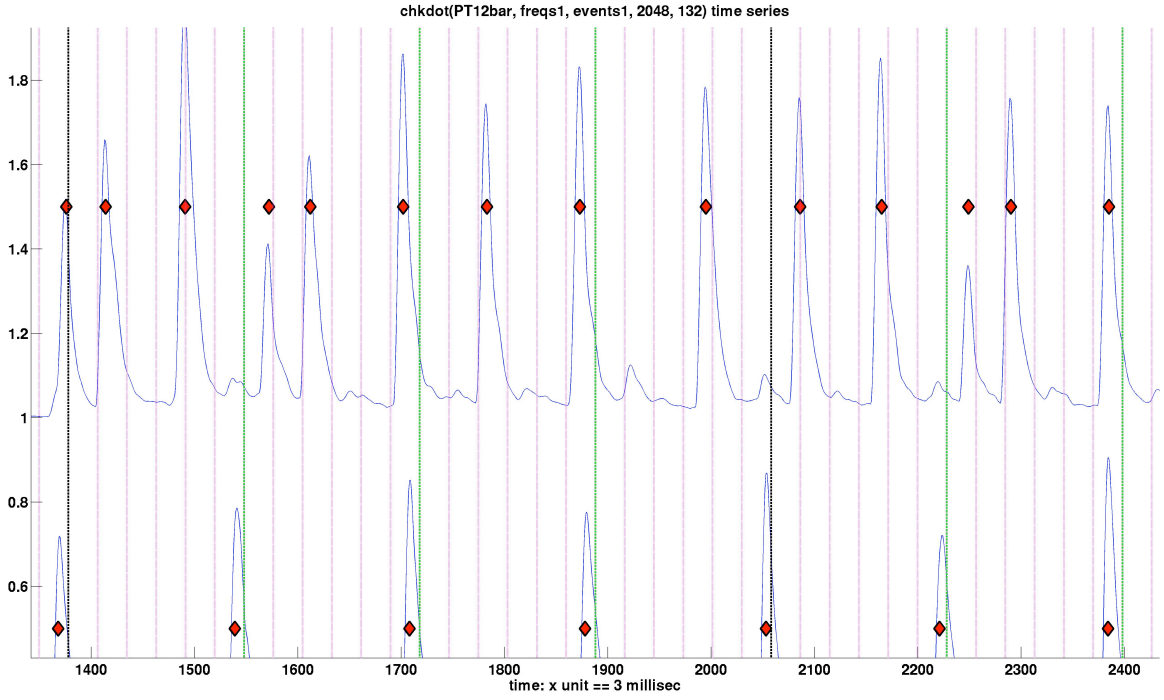

Figure 5.3.5.2 ***Tamborim Batida***: Playing Around the Beat (close up)

5.3.6 Shuffle (Surdo and Afoxe)

We use the term shuffle to describe a wide range of swing rhythms. A shuffle has a temporally less exact sound than typical percussion note events. Shakers, brushes on a snare drum or hi-hat cymbal, caixa, afoxe, guiro are all examples of shuffle instruments. Single events can be identified, but overall there is a feeling of blurring and blending of each note event into the next. The meter of the rhythm is defined by the loudness peaks or other identifiable but somewhat temporally ambiguous events. Shuffle is an odd combination of vagueness and precision, difficult to describe with language.

Note ID is more difficult for these less precise musical events, and marking the onset time locations precisely can be subject to interpretation of how the rhythm feels. The standard Brasilian ganza (shaker) rhythm usually has a noticeable snap that leads the downbeat, but the remaining notes are more blurry. The snap gives a precise anchor to the rhythm which makes the blurry parts sound well integrated to the ensemble swing, rather than being played carelessly. It is easy to see in the `diffdot` plots how the swingee and straight versions of this sample have quite different timing variations in both the pulse and secondary events tracks. Even the straight version has a substantial amount of temporal variations, similar to ***Graceland.***

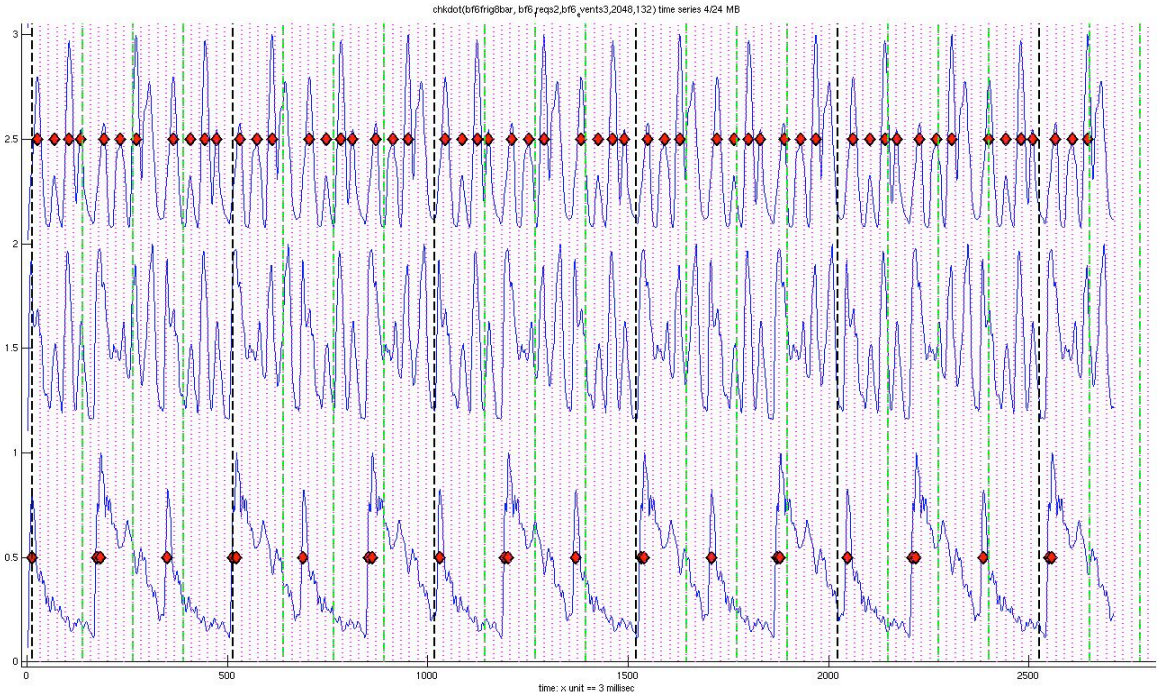

Figure 5.3.6.1 Time Series Plot for Swingee Shuffle Batida

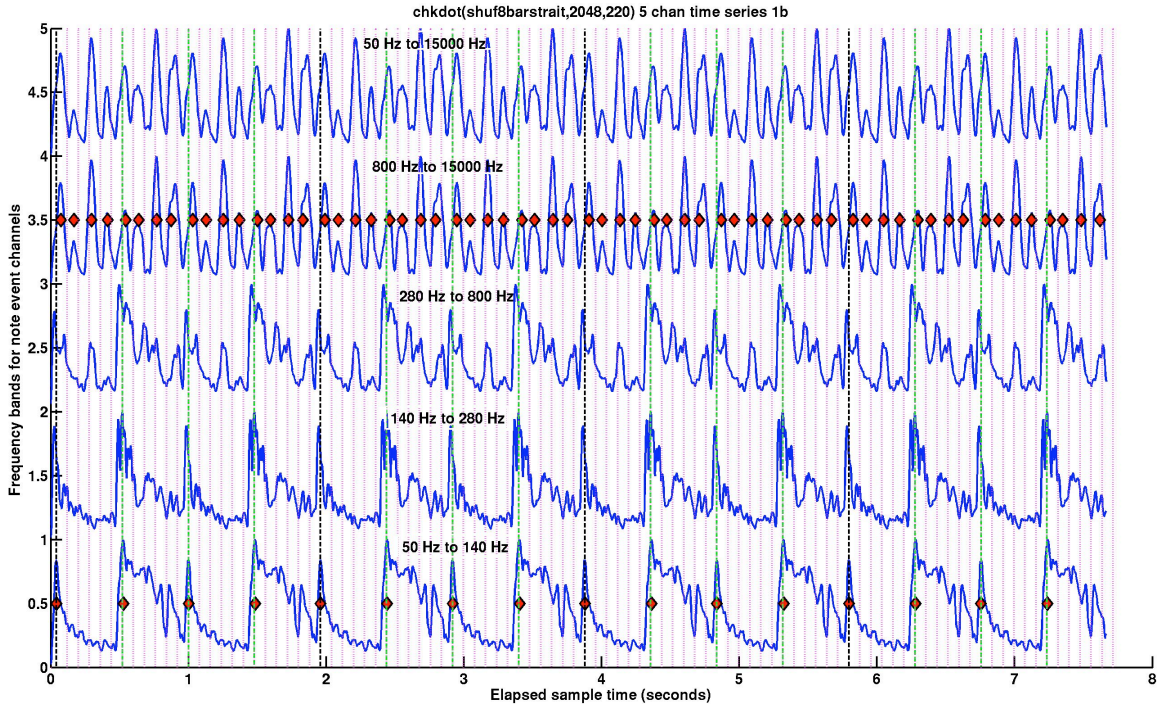

Figure 5.3.6.2 Time Series Plot for Straightened Shuffle Batida

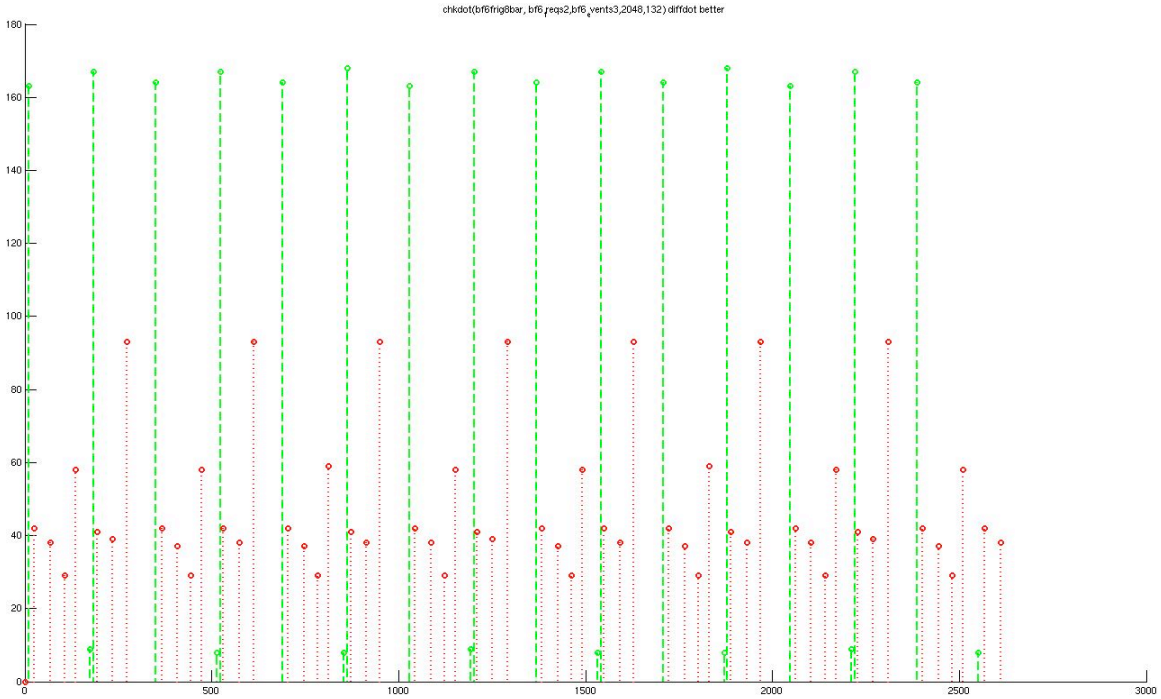

Figure 5.3.6.3 Note Timing Chart for Swingee Shuffle Batida

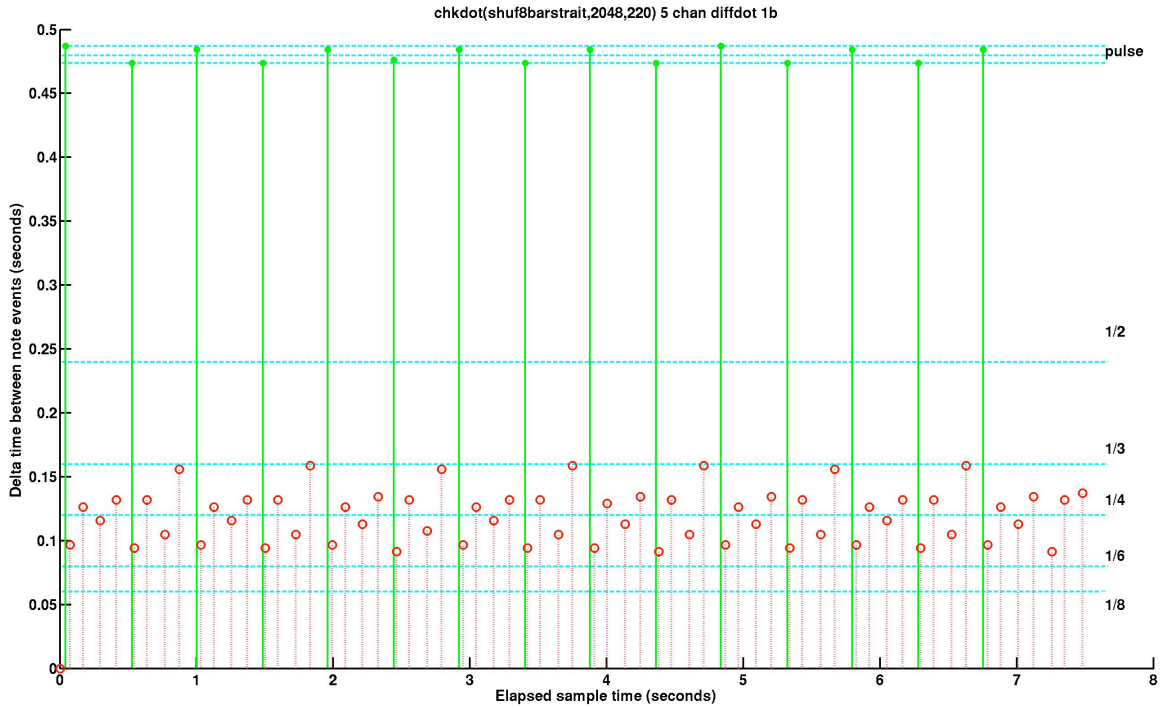

Figure 5.3.6.4 Note Timing Chart for Straightened Shuffle Batida

5.3.7 Reggae by Bob Marley

Reggae music from Jamaica is characterized by complex rhythms, many of which explicitly use the *absence* of a note event as a rhythmic anchor. In American music, note events on the downbeat or backbeat typically anchor the rhythm at the 1 or 3 beat of a 4 beat measure. In Reggae, the downbeat is often not played by any instrument, and other canonical MB beat locations may be demarcated by silence, perhaps followed by several very quick drum beats in a complex rhythm which may *end* on the next canonical MB beat. Detection of rhythms which have an empty note event as an important feature of their pattern is a challenging task, both for a musician or a computer algorithm. In addition, the counting scheme in our algorithm was developed for more conventional rhythms, and is quite inadequate for satisfactory extraction of the rhythmic structure of Reggae. Nonetheless, we had some success and show these results in this section.

Stir it up (1973) is one of the best love songs ever written (in my opinion, and my fiancee's). It begins with a very tight and clipped *kip* played on the backbeat of the rhythm by Bob Marley on the electric guitar. As the other instruments join in, a sparse and relatively simple sounding gestalt emerges, and the kip is revealed as a backbeat, whereas played by itself, it could be interpreted as the downbeat. I find it impossible not to dance to this tune (making it difficult to write this section sitting down).

Figure 5.3.7.1 shows the specgram for the intro to ***Stir it up.*** The six double short vertical red lines at the left are the kip. Later in the song this double beat is sometimes played as a triple or quadruple set of beats, maintaining the same tight rhythm. One outstanding feature of the specgram for this song is the presence of the row of pyramids in the lower part of the figure. This is caused by the sound of the keyboard as its notes roll smoothly up and down in frequency. Given the large number of Biblical references in Bob Marley's lyrics, I suspect he would like this revelation. Indeed, he might even claim that it is an intentional accident.

The pulse and bass drum events are shown in figures 5.3.7.2 and 5.3.7.3, which are subdivided in thirds, even though we have not yet found any real evidence for triplets in this song. We made plots using other subdivisions, but the figures we present seem the clearest. The pulse beats are either on the MB beat location lines, or exactly between two of the triplet lines, indicating a very straight and tight quarter note subdivision.

Figures 5.3.7.4 and 5.3.7.5 show close-ups of the pulse, and drum break. Both are very exactly subdivided in a multiple of two, 1/8th or 16th notes, depending on how one chooses to count and subdivide a measure. Figures 5.3.7.6 and 5.3.7.7 show specgrams of Bob Marley singing. These are quite beyond our current analytical approach, and would require both finer resolution in the spectral decomposition and much more sophisticated pattern recognition than we now use. We include them, like the pyramids, for their peculiar and somewhat mysterious beauty. Figures 5.3.7.8 shows a tempo change of the pulse.

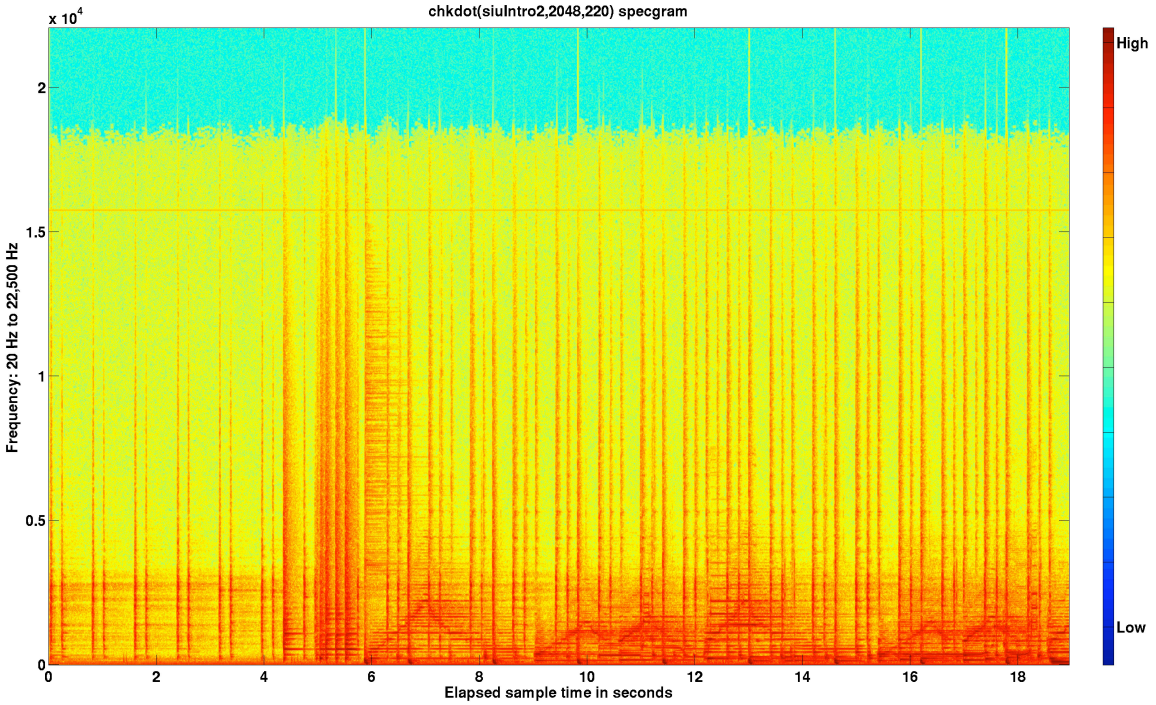

Figure 5.3.7.1 Specgram for Intro of ***Stir it up***

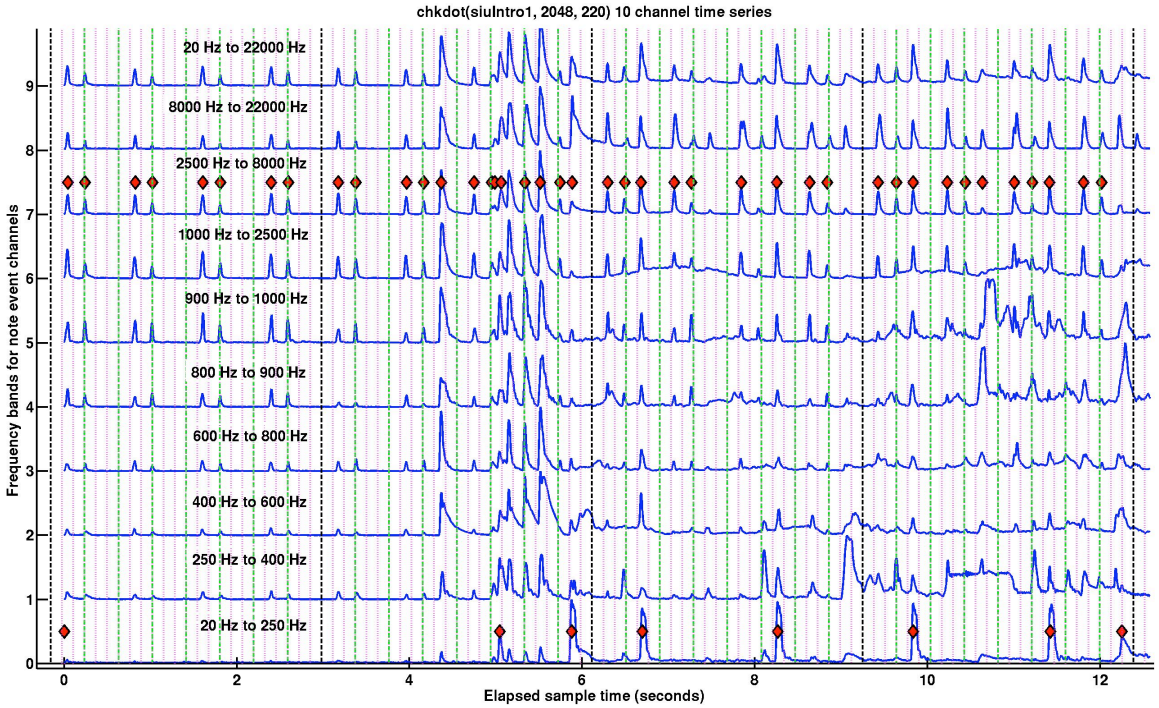

Figure 5.3.7.2 Ten Channel Events Time Series for ***Stir it up***

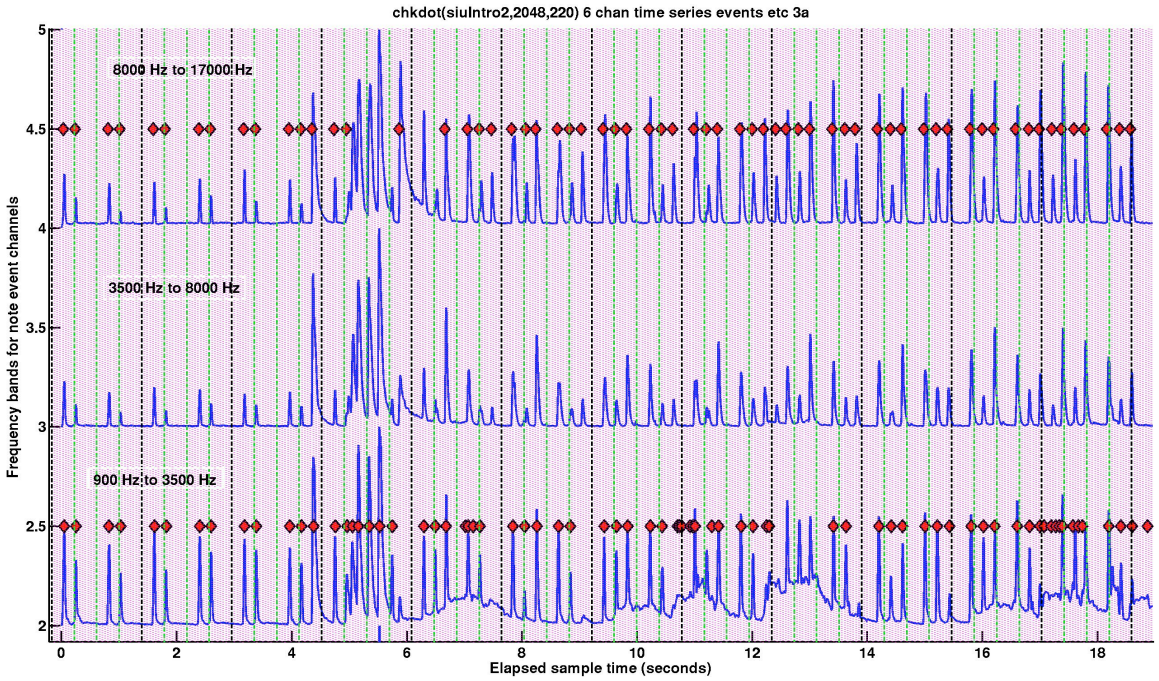

Figure 5.3.7.3 Close-up of Pulse and Drum Channels for ***Stir it up***

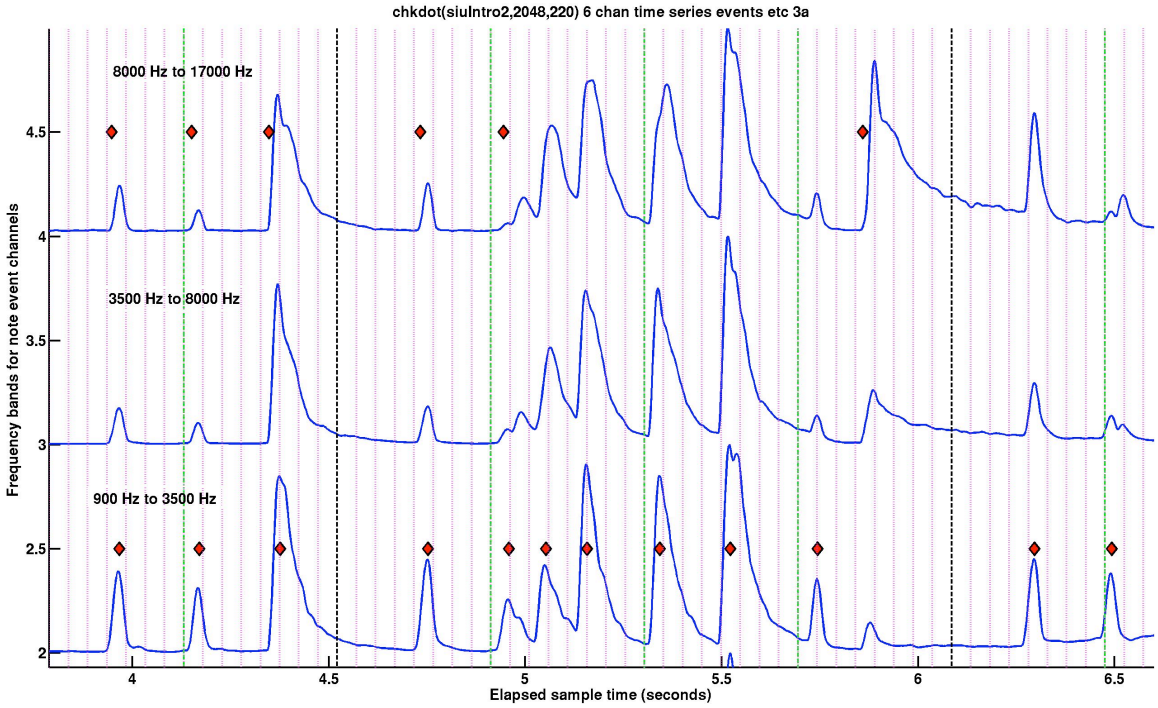

Figure 5.3.7.4 Close-up of Pulse and Drum Break for ***Stir it up***

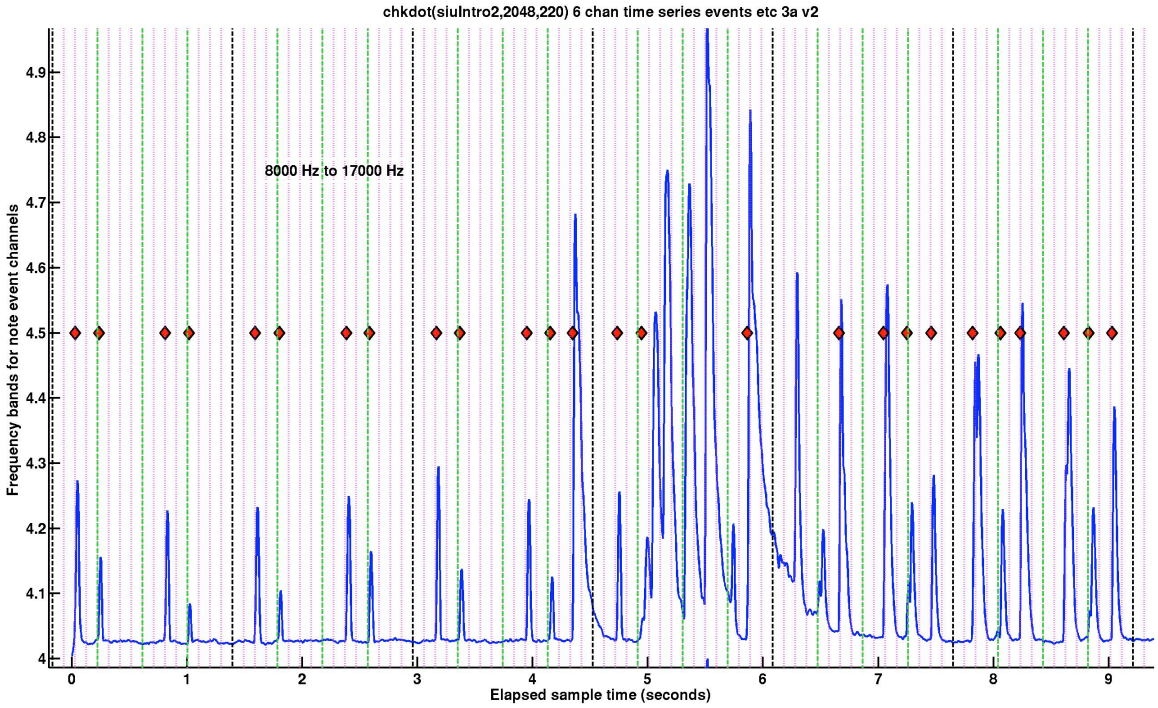

Figure 5.3.7.5 Close-up of Pulse for ***Stir it up***

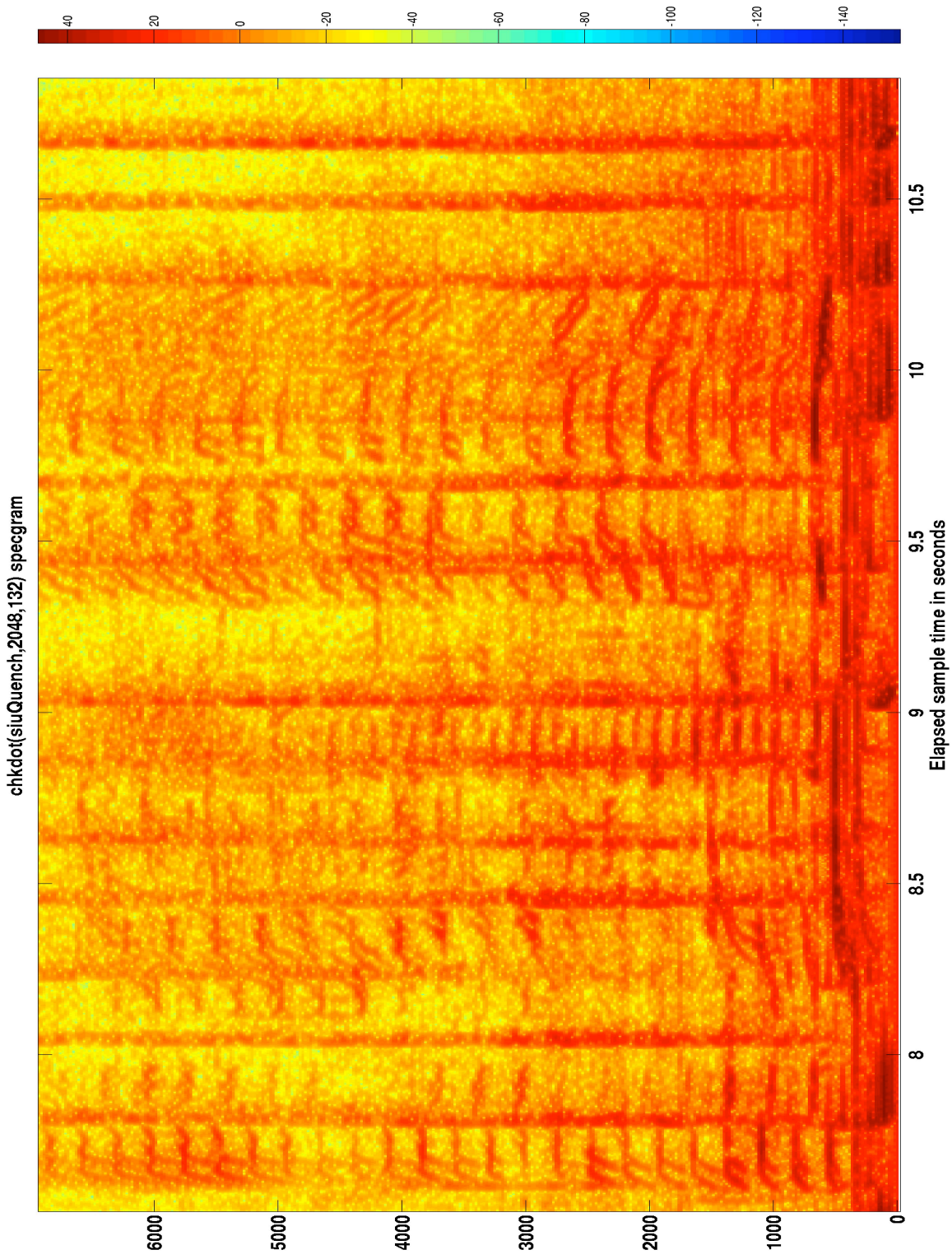

Figure 5.3.7.6 Specgram of Vocal for ***Stir it up: "C'mon cool me down baby ..."***

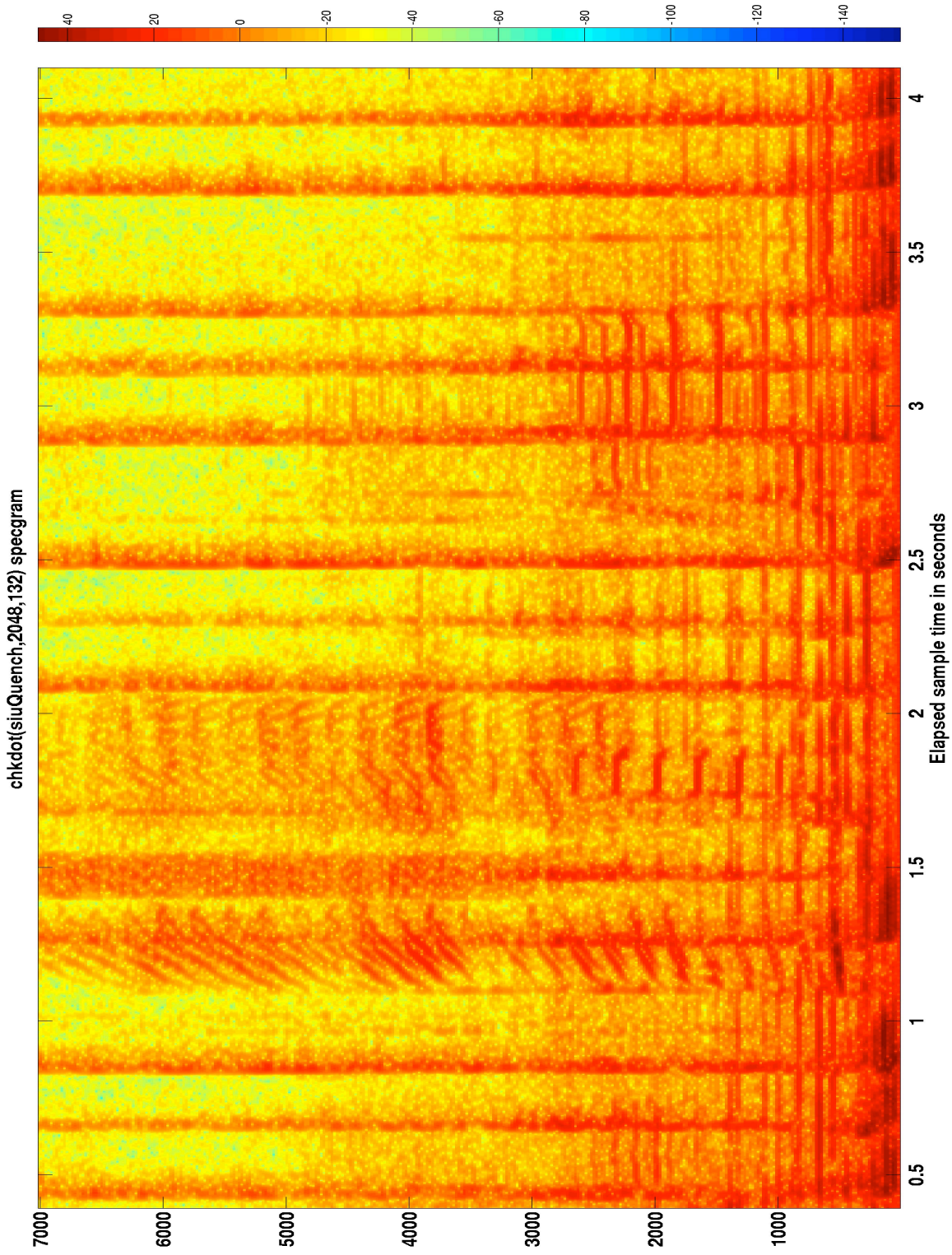

Figure 5.3.7.7 Specgram of Vocal for ***Stir it up: "When I'm thirsty"***

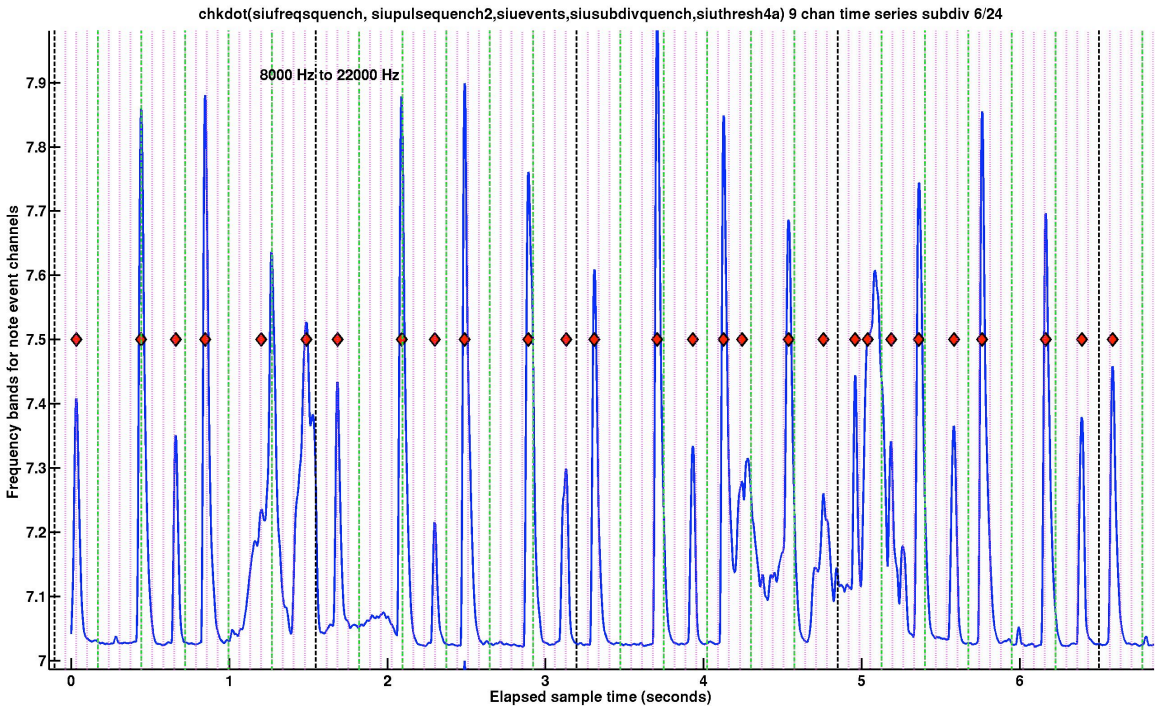

Figure 5.3.7.8 Tempo change in ***Stir it up***

Bob Marley had a very distinctive singing voice. While it might not be considered particularly "good" by some metrics, it is a very expressive and soulful voice. The specgrams showing the singing of lyrics show a great deal more complexity and subtlety than the vocals shown earlier for Natalie Cole singing ***Fever.*** My opinion is that the inter-twining waveforms visible in the specgram are the technical correlate of the emotional expressiveness that is clear when listening to Bob Marley singing. The actual data and information representations presented here are at the limits of what I could achieve with standard Fourier spectral analysis at this time. Techniques that allow finer resolution in the time and frequency domains are needed in order to produce clearer representations of the subtleties of the singing voice. We are addressing some of these technical issues in a separate project, "Optimization strategies for FFT use in musical audio analysis."

5.4 Swingee Notation Music Format

We have devised a novel form of notation that is intended to be a more informative form than standard MB notation. The idea is to make a simple visual rendering, in the context of standard notation, of the types of timing details we have investigated. Figures 5.4.1 and 5.4.2 show this idea using tablature for a pandeiro batida as an example. The pandeiro batida is rendered as straight quarter notes, which is how it is usually taught. Note events which should be played ahead of the MB beat are shown with a red leading edge. The amount of red indicates the amount of temporal variation from the MB beat. Since the triplet pickup to a downbeat, backbeat or offbeat (all canonical MB beat locations) is an important special case, we indicate a perfect triplet subdivision by including a "3" as a footnote to the note glyph in the tablature. Additionally, we color the triplet blue to indicate its special status.

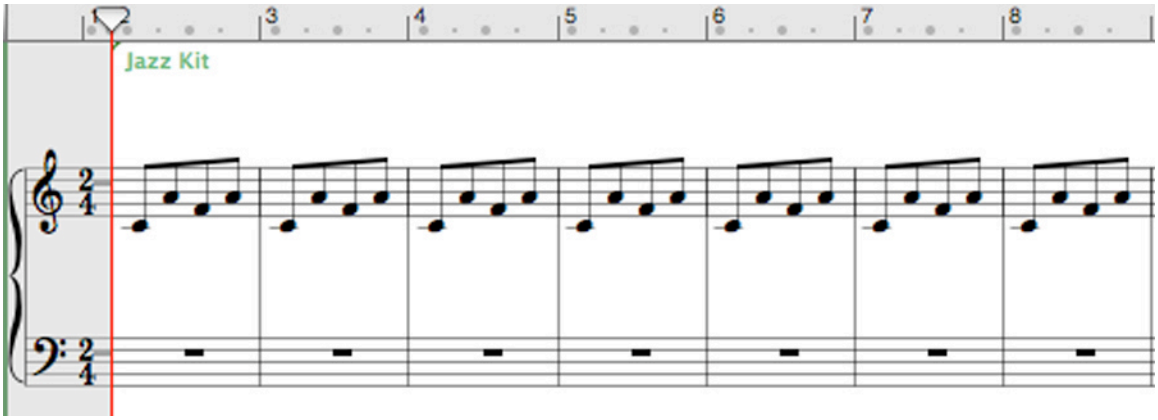

Figure 5.4.1 Standard Notation for Pandeiro Batida

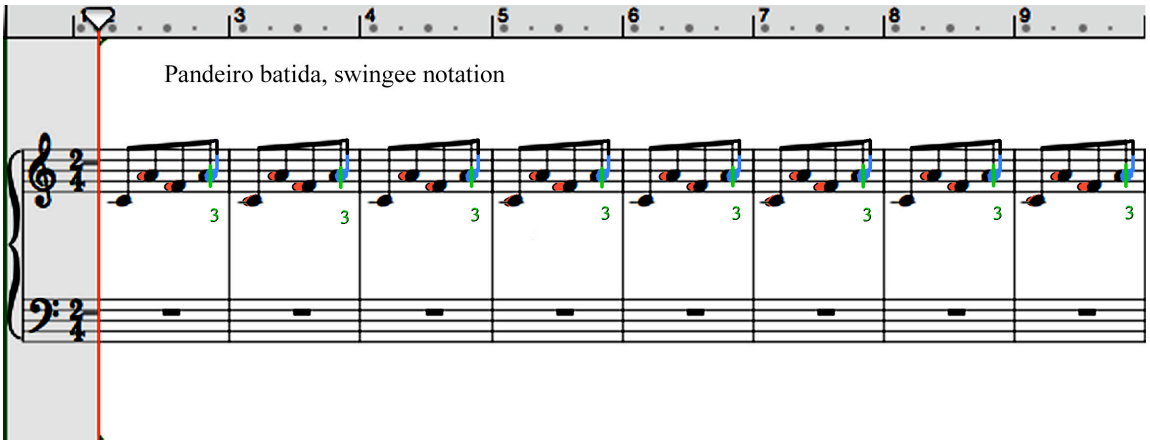

Figure 5.4.2 Swingee Notation for Pandeiro Batida

6. CONCLUSIONS AND FUTURE WORK

6.1 Assessment of Our Results

Prior technical research into swing style in music has shown that a major feature of swing is in variations of the timing between notes which are evenly spaced as written in a musical score. We entered into our research work wholly ignorant of such results, but with a reasonable conviction that timing variations are an important element in swing, based on our experience playing and listening to this type of music. Our results indicate that swing can result by shifting certain note timing as little as 50 milliseconds, typically note events that precede a major MB beat location such as a downbeat or backbeat. Brasilian swingee is a more complex style than American Swing, and timing differences as short as 10 to 15 milliseconds can change the feel of the music in ways that are perceptible even if not easily analyzed by a purely perceptual approach. Our algorithms allow easy scrutiny of the details of such timing variations.

An important model of swing as presented by Friburg, Gouyon and others is the swing ratio, which is a simple arithmetic ratio of the short and long time intervals between the swung notes. While this is an important metaphor, in our research we show it is not adequate to describe Brasilian swing, and in our musical experience we are led to believe that this deficiency is generally valid for rhythms that are not rooted in European classical music and the Mozart-Bach notation of straight time. We make a distinction, as do many researchers, between swing and the more general case of *rhythmic expression*.

Rhythmic expression can be found in all human music, and is largely absent from sequencer generated computer music. This more general form of expression, like swing, also derives from patterns of temporal variation in playing a piece of music, rather than playing the music as it is literally written in the score. The temporal variations can be de-

scribed by motion models: vehicles, animal and human circumambulation, ebb and flow of ocean tides (in certain types of classical music, see (Gabrielsson, 1987)) and other real world motion models. In a later section we explore the idea of using mathematical models of dynamical systems to generate timing patterns suitable for algorithmic generation of swing rhythms, and possibly also for more general cases of rhythmic expression. Like rich, complex audio tones and timbres, these complex timing patterns somehow reach into the human mind and connect with the non-symbolic emotional elements of the human psyche. We believe that music is a very useful data source for technical research into the neuro-physiology of emotion. While copious research has been done into the emotional aspect of music, we have not investigated this topic formally. As musicians, we state our opinions about music and emotion drawn from our own experience.

The complexity of rhythms with African roots is well known to musicians. Much of Brasilian, Cuban and Caribbean music is strongly influenced by African rhythms. We have given examples of complex timing variations that exist in the performance of even the simplest of Brasilian rhythms. The relationships among more than two rhythms played together give rise to much more complex systems of temporal variations, often called *ensemble swing*. These are known to musicians and perceivable by a trained attentive listener, although our pattern recognition techniques are not adequate for good technical analysis of such musical samples at this time. More sophisticated note identification algorithms would allow us to extend our approach to these more complex examples.

6.2 Neural Networks

Computational neural networks are a software approach to pattern recognition which have been successfully used for note identification in music (Kahn et al., 2004). The two neural net metaphors we plan to investigate for note identification are adaptive learning, and classification based on feature vectors. Adaptive learning in multi-layer feedforward perceptron networks (the “classic” neural net) would be a practical approach to automated identification of different types of note events. Its development is a time

consuming process that can give good results, once the system is properly trained, and is extensible to new data examples (note events). Feature vector classification is less flexible but quicker to develop, and for percussion music probably quite practical. In feature vector classification the identification of useful features, and collecting these into *a priori* sets of vectors, is done by the human researcher. This human activity substitutes for the computational learning process of adaptive neural nets. The feature vectors are used as coordinates into a multi-dimensional state space, and clusters of data points in this state space represent different types of notes. Learning vector quantization (LVQ) and self organizing maps (SOM) are the two classification approaches we believe would be most useful for extending our pattern recognition in the area of note identification. For modeling temporal aspects of rhythm, Hebbian learning and recurrent neural nets look like two useful techniques to explore. (Haykin, 2002)

6.3 Parsing Musical Audio into MIDI Events

The techniques we have presented are suitable for generating MIDI event files based on the extracted note events correlated with their temporal locations in the musical audio stream. We plan to develop this as part of the strategy to move this research out of the laboratory and into practical user application software.

6.4 Interactive Swingee Notation Software

Swingee notation (section 5.4) could be implemented as interactive software, and used to augment the notation features in music production applications like GarageBand or Logic. If the analysis and identification techniques which we have implemented in Matlab could run in realtime (which is quite likely), then this could be combined with swingee notation to produce a software tutor that can help a student learn to play a particular style of swing, given audio samples of the rhythm.

6.5. Improvements to Fourier Analysis

Fourier Analysis has been used for about 200 years to translate temporal data into spectral data. Since its popularization by (Cooley & Tukey, 1965), the Fast Fourier Trans-

form (FFT) has proven to be a practical algorithm for general signal processing. However, the FFT and Fourier Analysis have a substantial shortcoming: they both decompose a time based signal into *equally spaced* frequencies in the spectrum. As can be seen in the specgrams in chapter 5, most of the useful information in an audio signal is in the frequencies less than about 5000 Hz, or even lower. The higher frequencies are greatly over specified by the data in the frequency vector which is the result of the FFT. This result can be thought of as a mathematical vector that determines a location in a high dimensional space. Our examples using 2048 sample length FFTs correspond to a 1025 dimensional space. At most only about 20% of this information is truly useful, and in the samples we analyzed, a much smaller number of frequencies could be used to represent the spectrum above 5000 Hz than the approximately 800 that the FFT produces.

Compare and contrast the linearly spaced FFT and Fourier Series with the exponential nature of the frequency distribution analysis that is performed by the human cochlea. Both give useful amounts of information from the same data stream, but the cochlea has much finer frequency discrimination in the lower frequency part of the spectrum than does the FFT. The frequency relationships of tones which we hear are well described by the octave system of frequency spacing, at least below 5000 Hz. This octave system is designed with exponential spacing of the frequencies.

There do exist alternatives to the FFT and Fourier Analysis. Wavelets are a popular technique for extracting time and frequency information from a time based data stream, but as we noted in chapter 3, this is a very deep mathematical subject. (Young, 2001) presents the subject of non-harmonic Fourier Analysis, which develops the idea of using Fourier series that are sets of sine and cosines waves not related by integer multiples (i.e., cos(x), cos(2x), cos(3x) ...), but by non-integer real or complex numbers. Mathematically, nonharmonic Fourier series are closely related to wavelets. The basic idea of the math for wavelets and nonharmonic Fourier series is to create a method to accurately describe a Hilbert space of mathematical functions (i.e. audio waveforms). Both

wavelet analysis and nonharmonic Fourier series seek to generate sets of basis vectors which span the N-dimensional frequency space, just as the classical Fourier series does. Since there are infinitely many spanning sets of basis vectors, it seems likely that a class of basis vectors could be constructed that is equivalent to a Fourier series but which is exponentially spaced in the frequency domain. Initial discussions with my math advisors gave a positive view of this idea, and complete uncertainty about whether such a frequency analysis strategy has ever been attempted.

Whether this alternative to standard Fourier series is practical depends on whether an algorithm similar to Cooley-Tukey could be devised, which takes advantage of symmetries in the transformation matrix to reduce the amount of computation required for accurate spectral analysis of audio data. It seems likely that such a matrix factoring scheme could be found.

6.6. Improvements to the Cooley-Tukey FFT

6.6.1 Outline of Efficiency Concerns and Opportunities

The STFT as we use it in our algorithm includes a substantial amount of overlap in the time intervals between the set of component FFTs that make up the STFT. There is possible redundancy here and an opportunity for efficiency improvement if a variation of the Cooley-Tukey algorithm can be designed that lets the STFT reuse some data from the subdivision steps of one FFT transform with neighboring FFTs.

(Press, et al., 2002), (Elliot & Rao, 1982) and (Brigham, 1974) show details in the FFT decomposition logic that strongly suggest this is a practical, possibly relatively simple, optimization, but the idea came to us late in this research, and we haven't fully explored its possibilities. It would be a good topic for a PhD dissertation.

The basic strategy of the FFT is to transform the original time domain data set into a frequency domain data set by progressively factoring the matrix which represents the overall Fourier transform. The straightforward approach to the transform would use

an NxN matrix for an N-point FFT. Each factor in the matrix represents an exponent for the complex exponential function, which is defined as $e^{x+iy} = cos(x) + i*sin(y)$. These matrix factors determine the coefficients of the sine and cosine functions which are the components of the FFT frequency vector. The sines and cosines are harmonically related by integer multiples of *x: cos(x), cos(2x), cos(3x) ...* . The Cooley-Tukey decomposition factors the entries in the transform matrix by the strategy that most integers are multiples or sums of other integers, e.g. *8 = 6 + 2, 16 = 2 * 8 etc*. The output data of the FFT is built up by hierarchical factoring and composition of integer multiples inside the sine and cosine functions which represent the harmonically related frequencies. A very large number of intermediate small data sets are generated.

The matrix coefficients could of course be calculated directly by numerical integration of products of the Fourier basis functions with the audio data set, but this strategy is very costly: $\boldsymbol{O(N^2).}$ By comparison, the matrix factoring scheme of the Cooley-Tukey transform creates a number of intermediate matrices that are very sparse -- i.e. in each intermediate NxN matrix, there are only N non-zero entries, and so each step only needs approximately N multiplications and additions. Since there are ***log(N)*** intermediate matrices, the overall compute cost of Cooley-Tukey is ***O(N*log(N)).*** In the Cooley-Tukey algorithm, the decomposition commonly is done by powers of two, taking further advantage of binary arithmetic and logical operations available as hardware operations in CPUs and DSP chips.

6.6.2 Reusing Overlapping Data Windows

It would be very advantageous when performing the STFT if the subdivided data sets of one FFT could be reused for computing part of the neighboring FFTs. An overlap of 128 data points would correspond to using the fourth subdivision step of an FFT for a 2048 data point sample ($128 * 2^4 = 2048$). Thus rather than computing this subdivided data set for each FFT, one would reuse it for any FFT which overlapped the time interval

of the original data that corresponds to this subdivided data set. Of course the set of decomposed integer factors in the transform matrix must also match. If a factoring scheme were constructed along these lines, the STFT compute cost would be greatly reduced.

6.7 Instantaneous Frequency Techniques

Our dissatisfaction with the low frequency resolution which can be obtained by the FFT led us to want an algorithm or piece of hardware which returns information about instantaneous frequencies that are present in the audio waveform. Instantaneous (or very short time slice) frequency extraction is a large part of the data collection strategy of the human ear, as noted in Appendix E. An idea from our work in numerical analysis suggested that a Taylor series could be constructed using the mathematical derivatives of the waveform to fit a spline interpolation of the input data, giving information about the dynamics of the signal on a much shorter time scale than is practical using an FFT. Splines can be time shifted and overlapped as is done in the STFT. Changes in the interpolation function between time slices can be a source for information features much like we use spectral changes between FFTs in the current algorithm. Proving correlations between the changing derivatives and instantaneous frequency as measured by the cochlea (or other high resolution device) would be a mathematically challenging task beyond the current scope. Finding useful patterns in the derivatives and spline interpolations might be a simpler task. If we plot the information in a visually cogent manner, as we have done for the STFT and time series waveforms, the human biocomputer can tell us at a quick glance if there are useful patterns present which can be extracted. The next step would be to create an extraction algorithm. This is a well proven research technique.

The derivatives of the audio waveform could give a more precise view of rapid changes of the audio than spectral analysis does. We have not analyzed the compute cost of the Taylor series approach. If it was similar to the cost of an FFT, this algorithmic approach would be practical. If instantaneous frequency processing in software is too ex-

pensive, then specialized DSP hardware would be needed. (Schwartz et al., 1999) report on work done to construct a silicon cochlea using VLSI chip fabrication technology.

A very late breaking discovery in the Matlab documentation also gives us hope that the Hilbert transform could be used to obtain instantaneous frequency information.

6.8 Swingee Maker

Based on visual inspection of the `diffdot` plots of rhythms in the analyzed samples, we believe that Fourier series can be used to generate the time variations which correspond to swing feel in music. Cases like the electric guitar in ***Graceland*** or the pulse and secondary events in the pandeiro batida clearly show a set of data points which could be closely approximated by a sum of sine and cosine waveforms.

Fourier series solution is a standard method for solving the systems of differential equations that would be used for physically accurate mathematical models of real world nonlinear dynamical systems like a train or streetcar, or the rhythmic motion of the human body. Due to the complexity and subtleties of swing rhythms' timing variations, an algorithm for automated generation of such patterns needs to accurately mimic the real world systems that give rise to swing as played by human musicians. If the algorithm used Fourier series to generate waveforms for producing these timing differences, the currently discovered swing waveforms (chapter 5) could be closely approximated. A further refinement would be to use the Fourier series waveforms to drive coupled nonlinear dynamical systems of partial differential equations. This would greatly reduce any mechanical repetition which might result from using the Fourier series by themselves. As previously noted, the human perceptual system is very astute at distinguishing natural and artificial patterns, and successful generation of high quality swing rhythm timing variations would probably require such an approach as we describe.

APPENDICES

A1. Interviews and Other Field Work

There's no substitute for direct interaction with music teachers if one wishes to gain insight into the mechanics and meaning of music. I have studied Brasilian percussion for about ten years with teachers from Brasil, and some teachers from the USA and other countries. I've had exposure to other drumming traditions, but my primary knowledge is rooted in Brasilian culture and music.

Throughout this learning curve, there have been several struggles. Primary is the difficulty of perceiving in real time what the rhythmic patterns are. Second is trying to perceive and understand the relationships between different batidas being played simultaneously. Finally, after getting the basic data (batida) correct and locating the rhythm in correct temporal relationship to one or more primary features in the other rhythms (e.g. the downbeat and offbeat as played by the surdo), then there is the difficulty of playing the batida with the right swingee feel. Note that "correctness" here may not be limited to a single answer. Musical performance is not like database information retrieval.

A1.1 Kim Atkinson's Thoughts on 4/4, 6/8 and Other Conundra

Kim Atkinson is a professional drummer and teacher in the San Francisco Bay area. Several conversations with Kim at California Brasil Camp (CBC) provided ideas and insights for this thesis project.

Kim gave me the original idea of comparing swingee with straight time during conversations about swing feel, and the limitations of music software. In particular Kim criticized how most if not all music production software tries to cram all rhythmic subtleties into MB notation: e.g. "... and then it [the software] put in about a million rest symbols because I didn't play the note exactly where it thought the beat should be." Kim and

John Santos both cautioned me in my ignorance about the differences between Cuban swing and other swing styles as compared to American Swing and swingee. Insights like these are difficult to garner from books. John is also a professional musician in the San Francisco Bay area, and has been nominated for two Grammy awards.

Kim also told a story about some West African drummers he knows, and their attempts to use MB form to notate some complex rhythms from their tribal tradition. The rhythms have elements of 4/4 and 6/8 counting, and possibly additional counting tricks and subtleties -- I wasn't entirely clear on this extra aspect. The drummers transcribed the rhythms into both 4/4 and 6/8 and then the rhythmic data was played by a computer sequencer. Neither the 4/4 nor the 6/8 meter adequately captured the authentic count or feeling of the rhythms. The quandary is that the rhythms really are both and neither 4/4 and 6/8 at the same time -- a fairly common motif in West African drumming and its Brasilian and Cuban descendants. This aspect is part of what Shawn Moore meant in his opinion that swing comes from a 6 against 4 rhythm (see introduction).

A1.2 Learning an Ile Aye Caixa Batida, and the Perception of Timing

At CBC 2005, I played under the direction of Marcio, a master drummer from the Ile Aye group in Salvador, Brasil. The caixa batida that Marcio taught is deceptively simple, but I found it very difficult to play cleanly. Also, I experienced a unique perceptual phenomenon while learning this rhythm. On the first day of class, playing the rhythm itself and hearing the beats in time with the other drums was difficult, and was made more so because there was a temporal mismatch between my hearing a beat event, and watching the same event played by a fellow student who knew the rhythm and played it well. Every time I heard the beat, the other student's drumstick was seen to be at the ***top*** of the swing rather than being at the drum head. The sound of the beat, of course, is generated at the point in time and space when the stick hits the drum head. Over the course of several days, my perception of these two sources of observing the beat events slowly became

synchronized as I learned the rhythm better and became accustomed to the enormous sound volume of the *bateria*. For those unfamiliar with Brasilian batucada (Samba music played by an army of drummers), it is at least as loud as standing between two train tracks with freight trains rushing by at high speed a few feet from your ears. Of course it is a good idea to wear earplugs. Eventually the visual and auditory inputs were closely synchronized in my perception.

I believe part of the cause of the perceptual synchronization mismatch was due to a physiological response in the hearing system. The front end audio processing neurons develop noise canceling internal signals that reduce the apparent sound volume in the audio cortex. This noise canceling effect is perhaps related to the tinnitus condition: e.g., I have a chronic high frequency internal audio signal which is matched to the horizontal scan frequency of television: a legacy of watching too much TV too closely on a noisy set as a child. A well known similar effect in the visual system produces a 3D effect from a single image (e.g. photograph) when one eye is covered with a dark lens while the other is not. This can be done by popping out one lens from a pair of sunglasses.[1] The image entering the visual cortex through the dark lens is slightly delayed in the neural processing circuity compared to the clear eye, and the temporal mismatch tricks the visual system into interpreting the stimulus as three dimensional information.

(Schulze, et al., 1999) have investigated the learning process with respect to the beat or periodicity perceptual phenomenon, as we allude to in Appendix E. Their research into the auditory response of the Mongolian gerbil showed two different neural encoding mechanisms for low vs high frequency sounds. They suggest that different learning rates for the two different neural coding mechanisms may be caused by the presence of beat pattern detection for low frequencies, whereas beat pattern detection does not happen for sounds at higher frequencies than about 1 or 2 KHz. This is an area for further research.

---

[1] Don Soloway, research scientist, NASA Ames Research Center. Mountain View, CA USA.

A1.3 California Brasil Camp

California Brasil Camp (CBC) is an annual workshop held in the Redwood forests of Northern California at the Cazadero Performing Arts Camp. It is a full time immersive experience in Brasilian music and dance. Up to six class sessions are held each day for a week, taught by professional Brasilian musicians and dancers. For those who are interested in Brasilian music, dance and culture, this is a very high quality experience, and each time I go there my skills and knowledge improve substantially. More information can be found at www.mameluco.com/cbc . Jovino Santos Neto, who teaches jazz composition and ensemble performance at CBC has published several source books on his website (www.jovisan.net) including (Neto, 2005), which contains many commonly played Brasilian songs and rhythms.

In addition to the formal classes, students and teachers mingle constantly, Portuguese is spoken as commonly as English, and the evenings include much music performance by the teachers such as *pagode* and *forro.* These are folk music forms in Brasil that many people play or listen to several times a week, starting from childhood, at neighborhood gatherings. The early exposure gives Brasilians a natural knowledge of the music patterns and feelings without resorting to technical learning approaches like counting the beat. Indeed, many very excellent Brasilian musicians don't read music[2], and some have difficulty counting time in the MB style. I remember Mestre Beiçola trying to teach some of us *samba de roda* which has a tricky 3 against 4 feeling. He plays it very well, but he learned by ear so couldn't count it very well. He ended up showing us and explaining "hit here [right hand], here [left hand], wait a little bit, here [right hand].'

A2. Brasilian Music and Culture

If you can't go to New Orleans for music, go to Brasil. If you go to Brasil, be sure to visit Salvador, Pernambuco and Rio de Janeiro at the least, because these three places

---

[2] e.g., Airto Moreira, a professional musician from southern Brasil who is a major innovator in American Jazz. Airto has played extensively with Miles Davis and other well known Jazz and Latin musicians.

are historical and current sources of a great deal of Brasilian music. Every town and city has its own styles and traditions. In addition to professional music and dance performance, it is important to connect with casual musical gatherings to experience these too. Brasilians are very open and friendly and such connections are easy to make.

A2.1 Musical Instruments and Style

The basic Brasilian percussion instrument is the pandeiro, analyzed in detail in Chapter 5. Other instruments include surdo (bass drum), caixa (*kai-shah,* a Brasilian snare drum), ganza and caixixi (*kai-shee-shee,* shakers), agogo (bell, usually 2 tones), tamborim (a small hand drum played rapidly with a very lightweight stick), conga and its traditional counterpart from the Afro-Brasilian tradition, the atabaque (*ah-tah-bah-key*). Another crucial and very peculiar instrument is the cuica (*quee-kah*), which sounds a bit like a cartoon monkey singing samba. ***Could You be Loved*** by Bob Marley includes a cuica throughout the entire song. String and wind instruments are also played in many of the Brasilian styles. More information can be found at

http://brazilianpercussion.com/english

www.brazmus.com

Brazildrums.com (http://65.254.62.162/~brazildr/main/)

www.espiritodesamba.com

www.espiritodrums.com

www.casasamba.com

A2.2 The Culture of Enjoying Life

There's something of a mystery as to why Brasilians are generally pretty happy and low stress people, even though there are many difficulties living in Brasil. Part of the reason is surely the fact that so many Brasilians either listen to or play live music several times a week. There is a strong folk tradition of *pagode* where people gather in some

one's back yard or public place to enjoy barbecue and singing and playing songs that everyone grows up with and knows by heart. The only cultural event in the USA which remotely resembles pagode would be singing folk songs around a campfire, but the level of musical quality and sophistication of such gatherings is very naive and weak compared to the musical ability of many Brasilians. The comment has been made that random people on the street in Brasil often play music better than many professional musicians in the USA. These random people (the ones I've met) usually are quite modest and insist they don't play very well.

B. Other Swing Style Music Software

We encountered two applications that derive their rhythms from samples played by professional musicians of the following styles: ***Darbuka*** is based on Middle East music, and ***Latigo*** is based on Latin American music, played by members of *Miami Sound Machine*. These are available from Wizoo Sound Design in Germany, and easily found using Google. Due to budget limitations, and lack of availability of free demo versions of the software, our experience of this software is limited to reading the manuals and talking to drummers who have used them. Based on this information, we believe that this software would be a good source for analyzing the swing in these two genres of music, and that these swing styles are different from the styles we have analyzed.

Another interesting software application for music production is ***MetaSynth*** from UIsoftware.com . A demo version of this software is available for free, and we found that many of the features are unique and interesting, including a feature to enhance rhythmic timing.

C. Code Listing

In this appendix we present our Matlab code used for processing the music samples in this thesis, as well as examples of code use. We include an example of a script which loads musical audio data and the specific parameter sets we used for processing the data: frequencies, choice of event bands, subdivision strategy and thresholds.

C1. Example Script for Loading Musical Audio Data

```
%  loadsiu.m  loads some samples from Bob Marley Stir it up

% revised version for chkdot rev2.

% Bob Marley, Stir it up
cd ~/Academics/SOU/thesis/final.delivs/data/audio/analyzed.samples/stir.it.up

siuIntro1 = wavread('Situp.intro.1a.m.mix.wav');
siuIntro2 = wavread('Situp.intro.12bar.mono.mix.wav');
siuQuench = wavread('Situp.quench.me.m.mix.wav');
siuBridge1 = wavread('Situp.bridge.32bar.m.mix.wav');
siuBridge2 = wavread('Situp.bridge2.16bar.m.mix.wav');
% for bridge3 use freqs = [ 20 240 2000 4500 17000 ]
siuBridge3 = wavread('Situp.bridge3.16bar.m.mix.wav');
siuBridge4 = wavread('Situp.bridge4.16bar.m.mix.wav');


siu_freqs = [20  250 400 600 800 900 1000 1200 1400 1500 1700 2500 8000 22000]
siu_freqs2 = [20 120 250 400 600 800 900 1000  3500 8000 22000]
siu_freqs2a = [20 120  400 600 800 900 1000  3500 8000 22000]
siu_freqs3 = [20   400   900   3500 8000 17000]
siu_freqs4 = [20 250  400  8000 17000]
siu_freqsbr3 = [ 20 240 2000 4500 17000 ]
siu_freqsquench = [ 20 200 250 370 800 2500 4500 8000 22000 ]
siu_pulse = [-13 8]
siu_pulse2 = [-9 4]
siu_pulse2a = [-8 8]
siu_pulse3 = [-3 4]
siu_pulse3a = [-5 4]
siu_pulse4 = [4 3]
siu_pulse4a = [4 11]
siu_pulsebr3 = [4 4]
siu_pulsequench = [-8 10]
siu_pulsequench2 = [-8 6]

siu_events = [1 ]
siu_events2 = [7 ]
siu_events3 = [3 ]
siu_eventsbr3 = [2 ]
siu_subdiv = [24  ]  % 1/6 subdiv
siu_subdiv2 = [32  ]  % 1/32 subdiv
siu_subdiv3 = [32  ]  % 1/32 subdiv
```

```
siu_subdiv3a = [16  ]  % 1/16 subdiv
siu_subdiv4  = [16  ]  % 1/16 subdiv
siu_subdivquench  = [60  ]  % 1/16 subdiv
siu_thresh = [.1, .1 ; .2, .35]
siu_thresh2 = [.1, .1 ; .2, .2]
siu_thresh2a = [.05, .05 ; .5, .5]
siu_thresh3 = [.05, .05 ; .15, .15]
siu_thresh4 = [.05, .05 ; .5, .4]
siu_thresh4a = [.2, .2 ; .4, .25]
siu_threshbr3a = [.2, .2 ; .1, .1]

cd ../../../gfx  % save all figs and pix to this dir
```

The resultant Matlab variables look like this:

```
siuQuench          575022x1                  4600176  double array
siu_events              1x1                        8  double array
siu_freqs               1x15                     120  double array
siu_freqsquench         1x9                       72  double array
siu_pulse               1x2                       16  double array
siu_pulsequench         1x2                       16  double array
siu_pulsequench2        1x2                       16  double array
siu_subdiv              1x1                        8  double array
siu_subdivquench        1x1                        8  double array
siu_thresh              2x2                       32  double array
siu_thresh2             2x2                       32  double array
```

C2. Example Matlab Function Calls

First call the `chkdot` script with a particular audio sample, and FFT parameters. Here we process the "Quench me darling ..." verse from ***Stir it up*** by Bob Marley and perform a 2 Ks FFT in the STFT. The FFT window slides 132 audio samples (3 milliseconds) between subsequent FFT processing. The size of this audio sample is `575022` elements. The number of FFTs in the STFT is given by

```
(samp_num_elems - FFT_size )/FFT_overlap,
```

in this case we would expect 4340 FFT tiles in the STFT.

```
matlab > chkdot(siuQuench, 2048, 132)
```

The first call to `chkdot` produces a specgram plot of the music sample based on the FFT parameters, and retains the spectral data as an internal Matlab matrix which is

used in subsequent calls to `chkdot`. This allows the specgram data to be reused with different frequency, event band, subdivision and threshold parameters. This is a practical strategy since extracting useful information from the specgram is generally an interactive process of discovery and refinement, rather than a single step that yields optimal results on the first attempt. Subsequent calls to `chkdot` look like this:

```
matlab > chkdot(siu_freqs, siu_pulse, siu_events, siu_subdiv, siu_thresh)
```

C3. Main Audio Processing Matlab Script

```
function  chkdot(param1, param2, param3, param4, param5)
%
%  parse an audio or other signal with specgram STFTs. and plot the result(s)
%
%  indata is the signal data,
%  freq_vec is a list of frequencies for sub-band
%  fft_len is the size of the STFT FFTs. use default FFT window (Hanning):
%  overlap_delta is the shift in sample count between STFTs in specgram
%
%  API redesign 24apr06: 5 calling options ==
%      { no params, 1 param, 2 params, 3 params, 5 params }
%        none  --> default example of pandeiro
%        one   --> audio data, 2 Ks FFTs, 440 samples overlap (10 msecs
%        two   --> ( 'save', 'dataname'   OR   'load', 'dataname' )
%        three --> ( indata, FFTlength, FFToverlap )
%              canonical first call. sets up data space for this sample
%              only plots the specgram, keeps persistent spectral data
%              another call like this wipes the old data, sets up new
%        five  --> ( freqs, pulse, events, subdivs, thresholds )
%              subsequent calls on persistent spectral data
%              freqs = ( f1, f2 [, f3 [, f4 [, f5 ... ]]] )
%              pulse = ( chan #, # of events in pattern [, initial skip ] )
%              events = ( chan1 [, chan2 [, chan3 ...]] )
%              subdivs = ( chan1 [, chan2 [, chan3 ...]] )
%              thresholds = ( func low, func high [, df/dt low, df/dt high
%                             [, d2f/dt2 low, d2f/dt2 high ]] )

tic % start timer to measure script execution time
argc = nargin
sample_rate = 44100 % CD sampling rate

% retain these: computed in the first pass, and used in all subsequent passes
persistent samp_spec samp_freqs samp_times overlap samplength sampletime

if argc == 0 % load a default sample
  indata =
wavread('~/Academics/SOU/thesis/data/audio/batucada.samples/pandeiro.samples/pa
ndeiro4barmono1.wav');
  fft_len = 1024
  overlap = 132  % 3 msec
  win_len = fft_len - 8
  event_loc = [1 4] % lowest freq band, 4 events/cycle
  freq_vec =  [ 20 750 15000 ]
```

```
  samp_name = 'p4bar'
  firstpass = true
end
if argc == 1
  indata = param1;
  fft_len = 2048
  overlap = 440  % 10 msec
  win_len = round(fft_len * 0.95)
  firstpass = true
end
if argc == 2
  % load and save functions. not yet implemented
  return
end
if argc == 3 % first pass of real processing work
  indata = param1;
  fft_len = param2
  overlap = param3
  win_len = round(fft_len * 0.95)
  if win_len <= fft_len - overlap
      win_len = fft_len - overlap + 1
  end
  firstpass = true
end
if argc == 5
  freq_vec  = param1  % print out values and ID the freq_vec
  pulse_vec = param2
  events_vec = param3
  subdiv_vec = param4
  thresholds = param5
  firstpass = false
end

if   firstpass == true
  % do a specgram on the indata, get/retain power, freqs & time info
    % handle mono or stereo indata -- don't mix, just use L chan
    samp_chans  = length(indata(1,:))
    if(samp_chans > 1)
        samp_data = indata(:,1);
    end
    if(samp_chans == 1)
        samp_data = indata;
    end
    % get mean value of sample power to use in offsets
    samp_data_pow = samp_data .* samp_data;
    samp_data_abs = abs(samp_data);
    samp_data_mean = mean(samp_data_pow);
    % pad the beginning to get a clean start of signal for FFT
    tenz = ones(3 * fft_len / 4, 1);
    %  don't want bogus data, so set pad value to the sample mean
    tenz = tenz * samp_data_mean;
    samp_data = [tenz ; samp_data];
    samplength = length(samp_data)
    sampletime = samplength/sample_rate

    % do the STFT and plot the result
    [ samp_spec samp_freqs samp_times ] = ...
        specgram(samp_data, fft_len, sample_rate, win_len, fft_len - overlap);
```

```
    figure  % newplot does **not** do the right thing, as advert'd
    imagesc([0 sampletime],samp_freqs,20*log10(abs(samp_spec)+eps));
    axis xy;
    xlabel('Elapsed sample time in seconds')
    ylabel('Frequency: 20 Hz to 22,500 Hz')
    colormap jet;
    colorbar('YTickLabel', {'Low','','','','','','','','','High'})
    whos
    toc
    return % done with 1st pass
end

% 2nd pass
% extract sub-band info from MxN STFT matrix, and sum the bands for each
%    time slice. rows are freqs, columns are time slices.
% pre-allocate array to hold the sums, rows = number of freqs (minus 1)
%    columns (time slices) is second dim of STFT matrix
num_freq_slices = length( freq_vec ) % how many freq sums to do
    %  num_time_slices = length( samp_spec(1,:) )
num_time_slices = length( samp_times )
    % make an extra slot for the grand total sum
subfreqsum = zeros( num_freq_slices , num_time_slices);
num_freqs = length(samp_freqs)
freq_count = 1;
sum_freq_ndxs = zeros(1,num_freq_slices);
% find the freq slice breakpoint ndxs in the specgram MxN matrix
for freq_ndx = 1:1:num_freqs
  if samp_freqs(freq_ndx) > freq_vec(freq_count)
      sum_freq_ndxs(freq_count) = freq_ndx
      freq_count = freq_count + 1
      if freq_count >= num_freq_slices + 1
          break
      end
  end
end

% compute the power in each freq slice.
%   then total all the slices into the last row of subfreqsim matrix
for freq_ndx = 1:1:num_freq_slices - 1
  subfreqsum(freq_ndx, :) = ...
    sum(abs(samp_spec(sum_freq_ndxs(freq_ndx):sum_freq_ndxs(freq_ndx+1),:)));
  % sum the grand total
  subfreqsum(num_freq_slices, :) = ...
    subfreqsum(freq_ndx, :)  + subfreqsum(num_freq_slices, :);
end

% norm all slices to one
for freq_ndx = 1:1:num_freq_slices
  % find max for each sub-band
  freqmax = max(subfreqsum(freq_ndx,:))
  % norm each sub-band by its max/min  to range [0, 1].
  subfreqsum(freq_ndx, :) = (subfreqsum(freq_ndx, :) / freqmax);
end

% find timeslice ndx's of events using event_loc vector(s)
pulse_band = pulse_vec(1)   % sub-band used for basic beat pulse
pulse_events = pulse_vec(2) % how many events in a cycle in pulse sub-band
pulse_downbeat = true       % pulse on the downbeat ?
if pulse_band < 0
```

```
  pulse_band = - pulse_band % negative ndx == pulse on backbeat
  pulse_downbeat = false
end

%  events setup logic
secondevents_flag = false;
eventbands_count = length( events_vec )
if eventbands_count > 0
  second_band = events_vec(1)
  second_events = subdiv_vec(1)
  secondevents_flag = true
end

  % get first  & second finite diff for each slice and norm to 0 < x < 1 .
  %   average value for most of the samples will be subfreqmean on
  %   a quiet channel, which is needed for pulse detection. first arm the
  %   event detector flag event_up when diff(n) > threshold(10. then when next
  %   diff(n) < threshold(2), set event_down flag for event loc, & get time ndx

event_count = [ 0 ; 0 ; 0 ] % anticipate having 3 event bands
event_up = false;
event_down = false;
local_max = 0.1;  % zero gives false event detects, so use small num > 0
% start with a minimum number for expected events
event_ndx = zeros(pulse_events, 3)
for freq_ndx = 1:1:num_freq_slices
  % find diff for each sub-band == first time derivative of signal
  subfreqsumdiff(freq_ndx,:) = diff(subfreqsum(freq_ndx,:));
  % raise sample min to zero by subtracting min. norm it to [0,1]
  freqmin = min(subfreqsumdiff(freq_ndx,:))
  subfreqsumdiff(freq_ndx,:) = subfreqsumdiff(freq_ndx,:) - freqmin;
  freqmax = max(subfreqsumdiff(freq_ndx,:))
  % norm each sub-band by its max
  subfreqsumdiff(freq_ndx, :) = ( subfreqsumdiff(freq_ndx, :) / freqmax );
  run_len = length( subfreqsumdiff(freq_ndx, :) ) % for pulse & other events
  % just using the height of function WF for event trigger logic.
  subfreqmean = mean( subfreqsum(freq_ndx, : ) )

  if freq_ndx == pulse_band
    skip = false;
    for time_ndx = 1:1:run_len
      % keep track of local max. reset after event up/down logic switch off
      if  subfreqsum(freq_ndx, time_ndx) > local_max
        local_max = subfreqsum(freq_ndx, time_ndx);
      end
      if subfreqsum(freq_ndx, time_ndx) > thresholds(1,1)
         % standard for many samples. OLD keep for reference 25apr06
         %  if subfreqsumdiff(freq_ndx, time_ndx) > (freq_ndx + 0.6 )
         % if subfreqsum(freq_ndx, time_ndx) > (freq_ndx + 2 * subfreqmean )
        event_up = true;
        event_down = false;
      end
      if event_up == true
        if ~skip
          if subfreqsum(freq_ndx, time_ndx) < local_max
            event_down = true;
            event_up = false;
            skip = true;
          end
```

```
        end
      end
      % if subfreqsumdiff(freq_ndx, time_ndx) < ( freq_ndx + 0.35 )
      if subfreqsum(freq_ndx, time_ndx) < thresholds(1,2)
          local_max = 0.01;
          event_up = false;
          skip = false;
      end
      if event_down == true
        event_count(1) = event_count(1) + 1;
        event_ndx( event_count(1), 1 ) = time_ndx - 1;
        event_up = false;
        event_down = false;
      end
    end
  end
  if secondevents_flag == true
    if freq_ndx == second_band
      skip = false;
      for time_ndx = 1:1:run_len
        if  subfreqsum(freq_ndx, time_ndx) > local_max
          local_max = subfreqsum(freq_ndx, time_ndx);
        end
%      if subfreqsumdiff(freq_ndx, time_ndx) > (freq_ndx + 0.6 )
        if subfreqsum(freq_ndx, time_ndx) > thresholds(2,1)
           event_up = true;
          event_down = false;
        end
        if event_up == true
           if ~skip
             if subfreqsum(freq_ndx, time_ndx) < local_max
               event_down = true;
               event_up = false;
               skip = true;
             end
           end
        end
%       if subfreqsumdiff(freq_ndx, time_ndx) < ( freq_ndx + 0.45 )
        if subfreqsum(freq_ndx, time_ndx) < thresholds(2,2)
          local_max = 0.01;
          event_up = false;
          skip = false;
        end
        if event_down == true
          event_count(2) = event_count(2) + 1;
          % detecting this means peak occurred one sample ago
          event_ndx( event_count(2), 2 ) = time_ndx - 1;
          event_up = false;
          event_down = false;
        end
      end
    end
  end
  freqmin = min(subfreqsumdiff(freq_ndx,:))
  freqmax = max(subfreqsumdiff(freq_ndx,:))
  event_ndx = event_ndx  % print out event ndx's
  event_count = event_count
end
```

```
% this needs to be updated for multiple event lists. 20feb06
pulse_events_count = length(event_ndx(:, 1));
% get the ndx of last non zero entry
for t_ndx = 1:1:pulse_events_count
    if event_ndx(t_ndx, 1) == 0
        break
    end
end
pulse_events_count = t_ndx - 1
% preallocate matrices for ndx'ing the beat lines
% vectors with event ndx metric
note_event_x = zeros(pulse_events_count * 2 + 2, 1);
note_event_y = zeros(pulse_events_count * 2 + 2, 1);
beat_line_x = zeros(pulse_events_count * 2 + 2, 1);
beat_line_y = zeros(pulse_events_count * 2 + 2, 1);

% when add 3rd channel, need to get length of non-zero part of 2nd like 1st
%   this current logic assumes more events in 2nd chan than in pulse chan
second_events_count = length(event_ndx(:, 2))

 % get MB beat locs for beat lines, not actual note event locs
 %   this whole biz is calc'd w/ndx of events. later, convert these
 %      (working algo) sets into time event metric, not event ndx.
 %      do this by dividing the working data sets by the right factor based
 %      on the gyrations needed for convert CD audio times to FFT times to
 %      chkdot events time ndx's to these MB ndx's.
 %   do this based on this logic:
 %       xtime = linspace( 0, sampletime, num_time_slices );  and
 %       %  plot(subfreqsum(freq_ndx, :)  + freq_ndx - 1 , xtime)
 %     which are currently buggy.

m1_beat1 = event_ndx(1, 1) % loc of first MB downbeat
m2_beat1 = event_ndx(1 + pulse_events, 1) % loc of 2nd MB downbeat
mb_delta = m2_beat1 - m1_beat1
subdiv_count = pulse_events - 1
subdiv2_count = second_events - 1
subdiv_delta = mb_delta / pulse_events
subdiv2_delta = mb_delta / second_events
% time shift if pulse is on the offbeat instead of downbeat
mb_offbeat_ndx = ( m2_beat1 - m1_beat1 )/ (2 *  pulse_events)
for beat_ndx = 0:1:pulse_events_count / pulse_events - 1
  mb_count_ndx = beat_ndx * mb_delta; % get delta ndx
  if pulse_downbeat == true  % pulse is on the downbeat
    beat_line_x(2 * beat_ndx + 1) = m1_beat1 + mb_count_ndx;
    beat_line_x(2 * beat_ndx + 2) = m1_beat1 + mb_count_ndx;
    % subdivision markers
    for subdiv_beat_ndx = 1:1:subdiv_count
      subdiv_line_x(subdiv_beat_ndx + (subdiv_count * beat_ndx + 1)) = ...
        beat_line_x(2 * beat_ndx + 1) + subdiv_delta * subdiv_beat_ndx;
      subdiv_line_y(subdiv_beat_ndx + (subdiv_count * beat_ndx + 1)) = ...
        num_freq_slices;
    end
    if secondevents_flag == true
      for second_beat_ndx = 1:1:second_events
        subdiv2_line_x(second_beat_ndx + (subdiv2_count * beat_ndx + 1)) = ...
          beat_line_x(2 * beat_ndx + 1) + subdiv2_delta * second_beat_ndx;
        subdiv2_line_y(second_beat_ndx + (subdiv2_count * beat_ndx + 1)) = ...
          num_freq_slices;
      end
```

```
    end
  else  % pulse is on the backbeat
    beat_line_x(2 * beat_ndx + 1) = m1_beat1 + mb_count_ndx - mb_offbeat_ndx;
    beat_line_x(2 * beat_ndx + 2) = m1_beat1 + mb_count_ndx - mb_offbeat_ndx;
    % subdivision markers. already have pulse, so only mark
    %   count(pulse_events - 1) for subdivs
    for subdiv_beat_ndx = 1:1:subdiv_count
      subdiv_line_x(subdiv_beat_ndx + (subdiv_count * beat_ndx + 1)) = ...
        beat_line_x(2 * beat_ndx + 1) +  subdiv_delta * subdiv_beat_ndx;
      subdiv_line_y(subdiv_beat_ndx + (subdiv_count * beat_ndx + 1)) = ...
        num_freq_slices;
    end
    if secondevents_flag == true
      for second_beat_ndx = 1:1:second_events
        subdiv2_line_x(second_beat_ndx + (subdiv2_count * beat_ndx + 1)) = ...
          beat_line_x(2 * beat_ndx + 1) + subdiv2_delta * second_beat_ndx;
        subdiv2_line_y(second_beat_ndx + (subdiv2_count * beat_ndx + 1)) = ...
          num_freq_slices;
      end
    end
  end
  beat_line_y(2 * beat_ndx + 1) = 0;  % mark the beats with vert lines
  beat_line_y(2 * beat_ndx + 2) = num_freq_slices;
end

%  get ndx's for note events
for note_ndx = 1:1:pulse_events_count
  note_event_x(note_ndx, 1) = event_ndx(note_ndx, 1);
end
for note_ndx = 1:1:second_events_count
  note_event_x(note_ndx, 2) = event_ndx(note_ndx, 2);
end

%  do the time series plot: line graphs showing events in diff freq sub-bands
% for plotting the diff WFs  add the
% freq ndx to get vertical offset same as timeseries data vert offset.
figure   % chkdot
hold on
% adjust x axis for elapsed sample time rather than ndx of time series
xtime = linspace( 0, sampletime, num_time_slices );
xlabel1 = 'Elapsed sample time (seconds)';
ylabel1 = 'Frequency bands for note event channels';
ylabel2 = 'Delta time between note events (seconds)';
% compute the equivalent factor to divide the beatline etc vectors by.
ndx_to_sampletime = num_time_slices / sampletime

% now use this factor to divide the sets of subdiv lines and other data
%     sets that are calc'd for event ndx's above.
%   e.g. note_event_x, note_event_y, beat_line_x, beat_line_y,
%        subfreqsum, subfreqsumdiff, subdiv2_line_x, subdiv2_line_y,
%        diff_pulse_times, diff_second_times   << maybe others >>

% vectors with elapsed sample time metric
note_event_xtime = note_event_x/ndx_to_sampletime ;
note_event_ytime = note_event_y/ndx_to_sampletime ;
beat_line_xtime = beat_line_x/ndx_to_sampletime ;
beat_line_ytime = beat_line_y/ndx_to_sampletime ;

for freq_ndx = 1:1:num_freq_slices
```

```
  plot(xtime, subfreqsum(freq_ndx, :)  + freq_ndx - 1)  % vert offset by ndx
% plot the first derivative. visually too cluttered for general use
%   plot(subfreqsumdiff(freq_ndx, :), 'g')
 %  plot(subfreqsumdiff(freq_ndx, :), 'k*', 'MarkerSize', 2)
   % mark the note events with red diamonds
  if freq_ndx == pulse_band
    for note_ndx = 1:1:pulse_events_count
      plot(note_event_xtime(note_ndx, 1), freq_ndx - 0.5, 'rd', ...
        'MarkerSize', 12, ...
        'LineWidth',2, 'MarkerEdgeColor','k', 'MarkerFaceColor','r')
    end
  end
  if secondevents_flag == true
    if freq_ndx == second_band
      for note_ndx = 1:1:second_events_count
        plot(note_event_xtime(note_ndx, 2), freq_ndx - 0.5, 'rd', ...
          'MarkerSize', 12, ...
          'LineWidth',2, 'MarkerEdgeColor','k', 'MarkerFaceColor','r')
      end
    end
  end
end
% plot the MB beat locs, not actual note event locs
for beat_ndx = 1:2:pulse_events_count * 2  / pulse_events - 1
  plot(beat_line_xtime(beat_ndx:beat_ndx + 1), ...
       beat_line_y(beat_ndx:beat_ndx + 1), 'k--', 'LineWidth',2)
end

% stem plot the meter subdiv markers
stem(subdiv_line_x/ndx_to_sampletime, subdiv_line_y, ...
    'g--', 'MarkerSize', 0, 'LineWidth', 2)
if secondevents_flag == true
  stem(subdiv2_line_x/ndx_to_sampletime, subdiv2_line_y, ...
      'm:','MarkerSize', 0, 'LineWidth', 2)
end
x_delta = overlap / 44.1;  % in msec. fix/verify this time increment 24apr06
x_delta_int = round(x_delta);
% get closest int if delta is small enough
if abs(x_delta_int - x_delta) < .05
    x_delta = x_delta_int;
end
xlabel(xlabel1)
ylabel(ylabel1)

% label the frequency bands
subfreqstring = zeros(num_freq_slices,1);
for freq_ndx = 1:1:num_freq_slices - 1
  str = sprintf('%d Hz to %d Hz',  freq_vec(freq_ndx), freq_vec(freq_ndx + 1))
  annotation(gcf, 'textbox','Position', [.4 freq_ndx/(1.2 * num_freq_slices)
             .05 .05 ], 'String', str, 'EdgeColor', [1 1 1])
end
num_freq_slices/(1.2 * num_freq_slices)
str = sprintf('%d Hz to %d Hz',  freq_vec(1), freq_vec(num_freq_slices))
annotation(gcf, 'textbox','Position', [.4 num_freq_slices/(1.2 * ...
           num_freq_slices) .05 .05 ],  'String', str, 'EdgeColor', [1 1 1])

% plot the note event diffs:
figure  % diffdot
hold on
```

```
% plot the delta time values
% first get the deltas for pulse band
%   mimic this: diff_pulse_times = diff(event_ndx(:, 1));
%   but get rid of neg value and trailing zeroes
diff_pulse_times(1) = 0;
pulse_event_x(1) = 0;
diff_pulse_times(2) = 0;
pulse_event_x(2) = 0;
for note_ndx = 1:1:pulse_events_count
  if event_ndx(note_ndx + 1, 1) > 0 % is next entry real?
    diff_pulse_times(note_ndx + 2) = event_ndx(note_ndx + 1, 1)
                                     - event_ndx(note_ndx, 1);
    pulse_event_x(note_ndx + 2) = note_event_x(note_ndx,1);
  else
    break % bail out after finding all real entries
  end
end

% get the secondary events lined up like ducks
% this deals with the events time diff vec, not the events vec
for note_ndx = 1:1:second_events_count - 1
  diff_second_times(note_ndx + 1) = event_ndx(note_ndx + 1, 2) ...
                                    - event_ndx(note_ndx, 2);
  second_event_x(note_ndx + 1) = note_event_x(note_ndx,2);
end

pulse_event_xtimes = pulse_event_x(3:length(pulse_event_x))/ndx_to_sampletime;
scaled_diff_pulse_times =
diff_pulse_times(3:length(diff_pulse_times))/ndx_to_sampletime;
max_sdpt = max(scaled_diff_pulse_times)
min_sdpt = min(scaled_diff_pulse_times)
mean_sdpt = mean(scaled_diff_pulse_times)
ddx = [0, sampletime];
ddymaxp = [max_sdpt , max_sdpt];
ddyminp = [ min_sdpt , min_sdpt];
ddymeanp = [ mean_sdpt , mean_sdpt];
ddhalf = .5 * mean_sdpt;
ddqtr = .25 * mean_sdpt;
ddeighth = .125 * mean_sdpt;
ddthird = .33333 * mean_sdpt;
ddsixth = .16667 * mean_sdpt;
ddyhalf = [ddhalf, ddhalf];
ddyqtr = [ddqtr, ddqtr];
ddythird = [ddthird, ddthird];
ddysixth = [ddsixth, ddsixth];
ddyeighth = [ddeighth, ddeighth];
plot(ddx, ddymaxp, 'c--', 'LineWidth',2)
plot(ddx, ddyminp, 'c--', 'LineWidth',2)
plot(ddx, ddymeanp, 'c--', 'LineWidth',2)

plot(ddx, ddyhalf, 'c--', 'LineWidth',2)
plot(ddx, ddyqtr, 'c--', 'LineWidth',2)
plot(ddx, ddythird, 'c--', 'LineWidth',2)
plot(ddx, ddysixth, 'c--', 'LineWidth',2)
plot(ddx, ddyeighth, 'c--', 'LineWidth',2)
for freq_ndx = 1:1:num_freq_slices
  if freq_ndx == pulse_band
    stem(pulse_event_xtimes, scaled_diff_pulse_times, ...
        'go',  'fill', 'LineWidth',2)
```

```
  end

  second_event_xtimes = second_event_x/ndx_to_sampletime;
  scaled_diff_second_times = diff_second_times/ndx_to_sampletime;
  if secondevents_flag == true
    if freq_ndx == second_band
      stem(second_event_xtimes, scaled_diff_second_times, ...
          'r:o', 'MarkerSize', 9, 'LineWidth',2)
    end
  end
end
xlabel(xlabel1)
ylabel(ylabel2)
%annotation('textbox',[1 ddyhalf .1 .05, ], 'String', '1/2',...
    'EdgeColor', [1 1 1])
annotation(gcf, 'textbox','Position', [.95 .5 .05 .05 ], ...
    'String', '1/2', 'EdgeColor', [1 1 1])
annotation(gcf, 'textbox','Position', [.95 .333 .05 .05 ], ...
    'String', '1/3', 'EdgeColor', [1 1 1])
annotation(gcf, 'textbox','Position', [.95 .26 .05 .05 ], ...
    'String', '1/4', 'EdgeColor', [1 1 1])
annotation(gcf, 'textbox','Position', [.95 .18 .05 .05 ], ...
    'String', '1/6', 'EdgeColor', [1 1 1])
annotation(gcf, 'textbox','Position', [.95 .11 .05 .05 ], ...
    'String', '1/8', 'EdgeColor', [1 1 1])
annotation(gcf, 'textbox','Position', [.95 .9 .05 .05 ], ...
    'String', 'pulse', 'EdgeColor', [1 1 1])
% annotation(gcf,'textbox', 'Position',[0.38 0.96 0.45 0.026])

whos % print out the variables
toc   % print out elapsed time
```

D. Discography

Artist or Compilation Name ***Album Name*** (Date)

Group or Artist Name ***Song Title***

Paul Simon ***Graceland*** (1986)

Paul Simon ***Graceland***

Paul Simon ***Rhythm of the Saints.*** (1990)

Paul Simon ***Obvious Child***

Bob Marley ***Legend*** (1984)

Bob Marley ***Stir it up*** (1973)

Bob Marley ***Could you be loved?*** (1980)

Putumayo Presents ***Carnival*** (2001)

Martinho da Vila ***Canta, Canta minha Gente*** (1974)

Putumayo Presents ***Swing Around the World*** (2002)

Ka'au Crater Boys ***Opihi Man***

Duke Heitger and his Swing Band ***Swing Pan Alley***

Louis Armstrong & Duke Ellington ***Louis Armstrong meets Duke Ellington*** (1962)

It don't mean a thing if it ain't got that Swing

Ray Charles ***Genius Loves Company*** (2004)

Ray Charles & Natalie Cole ***Fever***

Grupo Batuque ***Samba de Futebol*** (2004)

Various examples of batucada, pandeiro, tamborim etc.

Various Artists ***Batucada por Favor*** (1998)

Bob Azzam ***Batucada por Favor***

Os Ritmistas Brasileiros ***Batucada Fantastica*** (1963/1998)

Luciano Perrone e Nilo Sergio various tracks.

Virginia Rodrigues ***Sol Negro*** (1997)

Virginia Rodrigues ***Adeus Batucada***

Other Brasilian groups which we plan to analyze in the future:

Dudu Tucci, Jorge Aragao, Olodum, Ile Aye, Martinho da Vila, Zeco Pagodinho.

E. Physiology and Psychophysics of the Human Auditory System

The most sophisticated information system on Earth is arguably the human mind. While there may exist similarly complex systems, we are ignorant of them. Most of human information is analyzed in terms of symbol systems, primarily language and mathematics. Indeed some people consider that there is no information, thought or meaning without language. These people apparently don't understand music.

While music can be represented as a symbol system using MB or other notation, this static form written on a page is not "music" but merely a guide to the performer for playing the music. By rendering the notation into sounds in the real world, the performer(s) create the reality of music from the thin sketch of information contained in the notation. For many people, music is something that they can only fully appreciate if they hear it. While trained musicians may be able to create music in their head by looking at notation, this is a difficult or impossible task for most people. In many cultures of the world, written notation is not used at all. The music created by these people is often sophisticated and complex, with deep informational and emotional content.

E.1 Human Auditory System

I do not claim to be an expert in audiology and psychophysics. This section was originally planned to be only a couple of pages, but the subject area proved fascinating (and immense). One thing led to another and this piece expanded greatly. Late in the game, Dr. Dean Ayers gave his feedback as a domain expert, pointing out both the flaws of my interpretations and analyses, as well as his opinions regarding the current views of mainstream audiological science. In short, research since (Bekesy, 1960) and (Harwood & Dowling, 1986) has shown that these early researchers' theories were not entirely correct. This is no surprise in scientific research of course. Rather than attempting an exhaustive re-write to fix things, I have cleaned up the egregious errors and left some of my

reports about Bekesy, Harwood & Dowling and others as they were, since for all we know, research in the near future may show that the theories may be more correct (or less) than the current "wisdom" indicates. Every theory in science will eventually be considered erroneous. It is useful to be aware of our history, and humble about our own theories and accomplishments.

In this section we present details of how the human ear transforms sound vibrations into patterns of nerve impulses in the auditory cortex. Human beings can generally hear sound frequencies from about 20 cycles per second (Hz) to 20,000 Hz at the lower and upper limits. Many people have a restricted range of frequency perception, so this is considered a best case scenario. Below 20 Hz and above 20,000 Hz perception of audio signals is possible, but the mechanism is different from what "hearing" is usually considered to be. Frequency of sound waves as measured by laboratory instruments corresponds to the human perception of pitch or tone: low frequencies are heard as low tones, high frequencies as high tones. However, there is more complexity in the human concept of tone than merely a short list of frequencies measured in the audio input signal. First, one frequency may not always be perceived as the same tone. Loudness of the sound can change its apparent pitch under some conditions, and in some frequency ranges. Second, combinations of distinct frequencies can create the perception of frequencies which do not "officially" exist in the sound source. Church organs take advantage of this effect (and have for hundreds of years) to create extremely low notes. The length of an organ pipe determines its fundamental (lowest) frequency and very long pipes are needed for very low notes. Alternatively, several pipes of shorter length can be used in combination, and by setting up a consistent set of frequency *differences* between pipes, the extremely low tone can be generated. This low tone only exists in the human ear and mind. Its "frequency" is not physically present in the external sound field (Plomp, 2002). There are other situations where frequency and tone do not map directly onto each other. This is a vast area of research that includes psychophysics, neuroscience and applied psychology.

Figure E.1.1 shows an overview of the human auditory data collection system, although it looks like Mr. Spock's ear, so maybe it's really Vulcan. Hearing begins with sound vibrations from the air striking the ear drum. Movements of the eardrum are transferred to the inner ear by three bones in the middle ear called the hammer, anvil and stirrup. These convert the physical scale of vibrations in gaseous air to a scale suitable for the liquid environment in the inner ear whose main component is the cochlea. The cochlea is a tapered tube rolled up in a spiral. Dividing the cochlea along its length is the basilar membrane and the Organ of Corti which contains neural vibration sensors, including small hair-like cells that are frequency sensitive. Different cells sense different frequencies depending on the cell's location along the length of the cochlea. The signals from these sensors are encoded into the nerve trunk and transmitted to the audio cortex. Information features are extracted starting with the initial actions of the Organ of Corti: the amplitude of the audio signal at various frequencies, and the timing relationships between frequencies. Frequency relationships such as the detection of correlated harmonics amongst the many frequencies present in the audio signal, or phase relationships between these component waveforms, may be partially encoded by the cochlea, but these more subtle distinctions may be perceived further up the processing chain in the audio cortex or brain. Generally, audiology research has failed to show evidence of the use of phase relationships by the ear (or the brain). My opinion is that this failure is at least partly due to the research techniques used. My own perception indicates subtleties which I attribute to phase discrimination by my ear/brain. Using the metaphor of sets of specific frequencies in an audio signal is useful, but is only a mathematical model of the data present in the signal. The real world sound is a complex three dimensional system of physically coupled motion patterns of air molecules. The decomposition of this system into a specific set of frequencies and phases is a convenient approximation, but can be misleading if interpreted literally for all situations. The activity of the audio cortex is far more sophisticated than that of any currently used computerized DSP and pattern matching techniques. For

example, phase information is known to be used at low frequencies for binaural detection of directional information from the external sound field. I have not read of any similar techniques used by researchers in computer music analysis.

Figure E.1.2 shows the cochlea by itself, from several viewpoints and a cross section. The basilar membrane including Organ of Corti can be seen at several turns of the spiral, dividing the two tunnel like chambers of the cochlea.

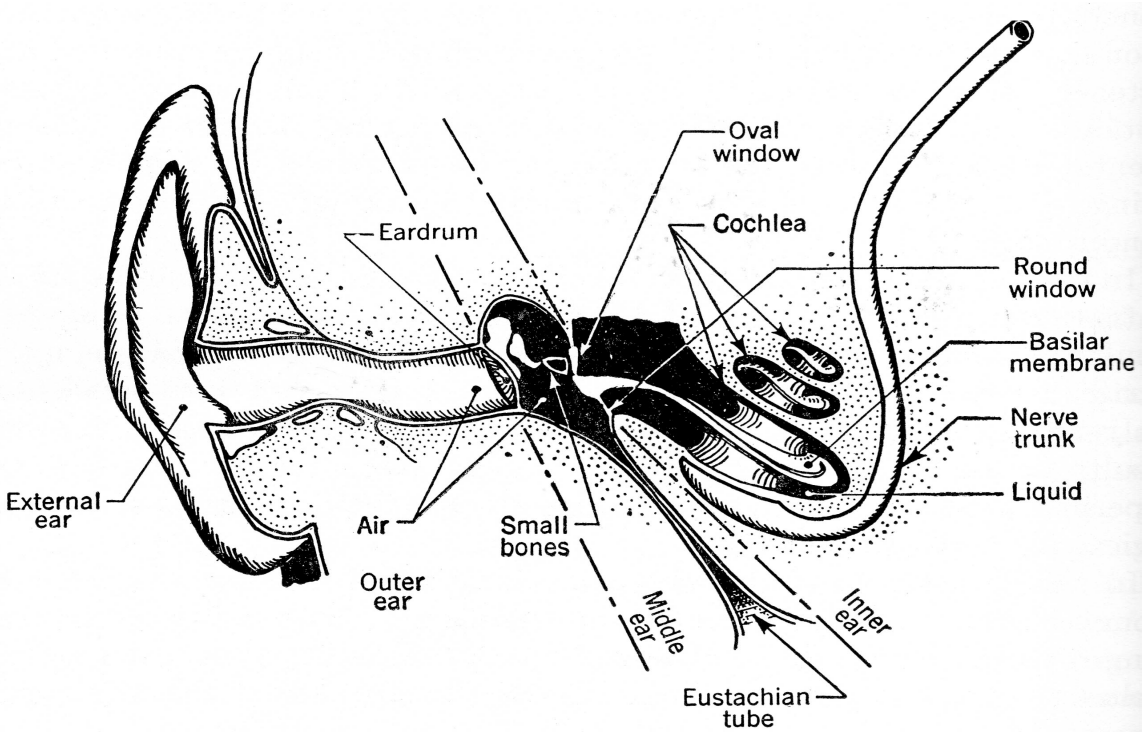

Figure E.1.1 Overview of Human Auditory Data Collection System
From (Beranek, 1954)

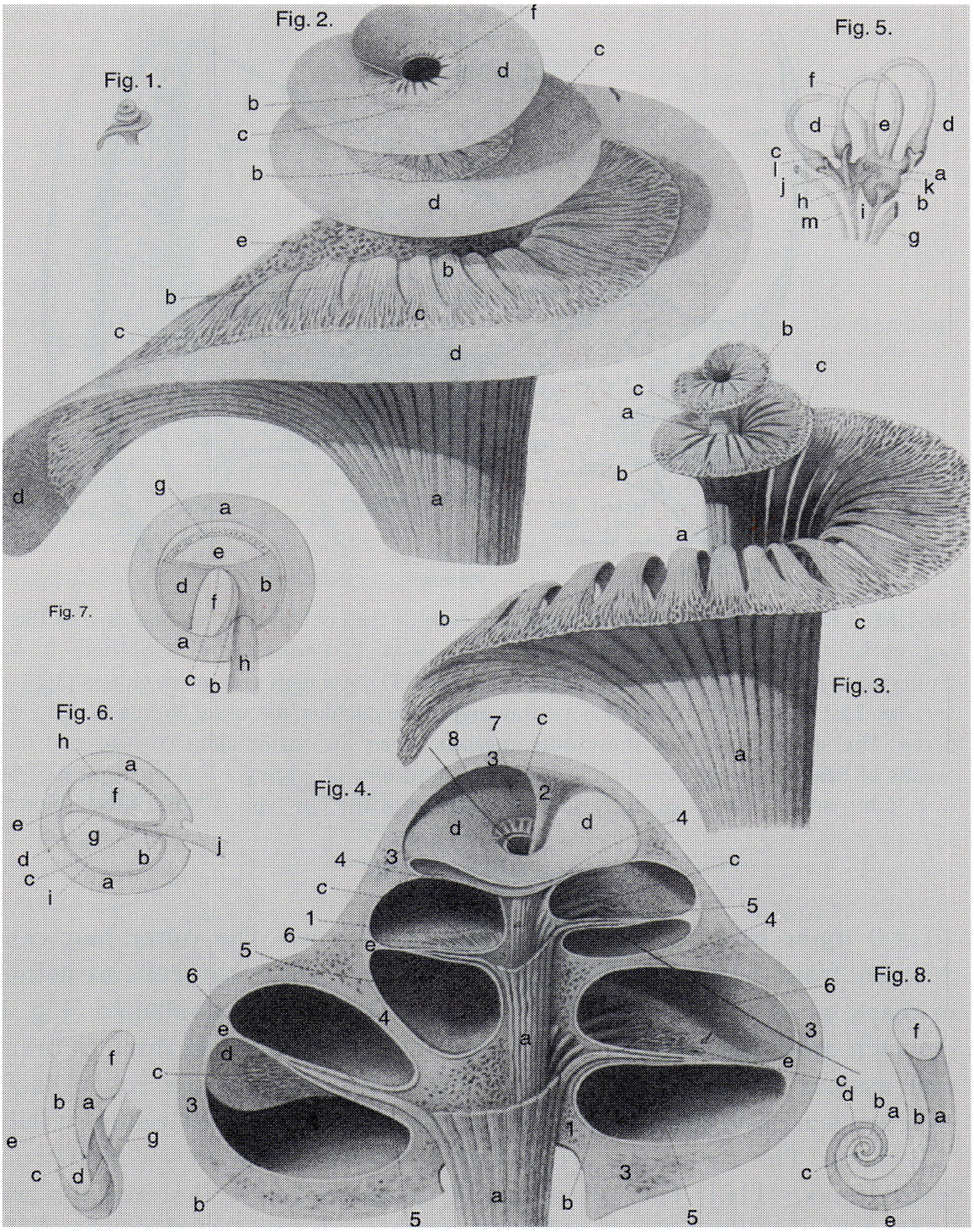

Figure E.1.2 The Human Cochlea

From (Møller, 2002). Original drawings by Brescher.

Figure E.1.3 shows a schematic cross section of the cochlea with details including the basilar membrane and Organ of Corti. This is a close-up of one of the turns in Figure E.1.2, showing most of the upper chamber and part of the lower chamber of the cochlea.

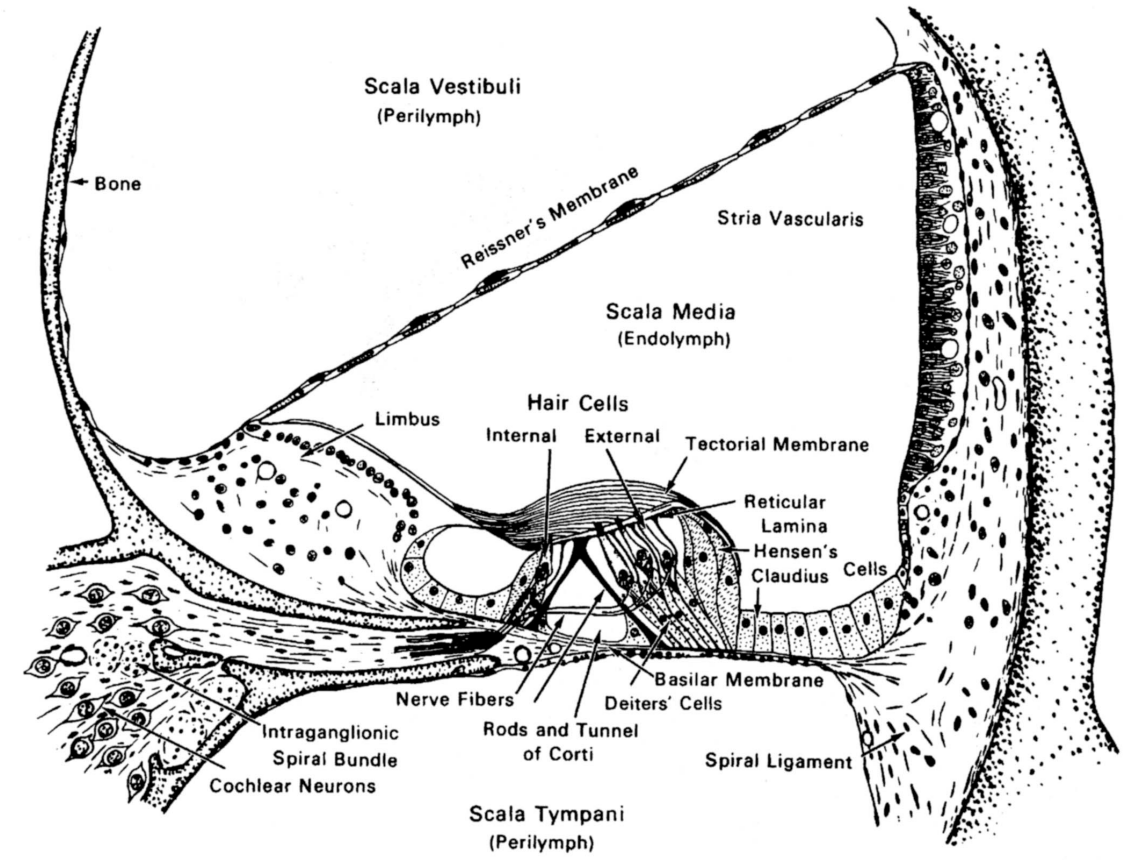

Figure E.1.3 Cross Section of Cochlea

From (Møller, 2002). Originally published by (Davis, et al., 1953) in *J. of Acoust. Soc. Am.* ***25****: 1180-1189.*

Figure E.1.4 shows an extreme close-up of a small section of the Organ of Corti and its frequency sensing hair cells, taken by a scanning electron microscope. Figure E.1.5 shows an even closer view of one of the several dozen hair tufts which are visible in Figure E.1.4. The hair tufts are colored yellow in Figure E.1.4, and orange in Figure E.1.5. These hairs move from the vibrations of the basilar membrane and the fluid in the space enclosed by the tectorial membrane, and then transmit their data to the nerve cells

colored pink in Figure E.1.4. It is known from (Bekesy, 1960) that fluid motion exists in the other chambers of the cochlea, but current theories hold that only the fluid motion in the tectorial chamber actually stimulates the hair cells.

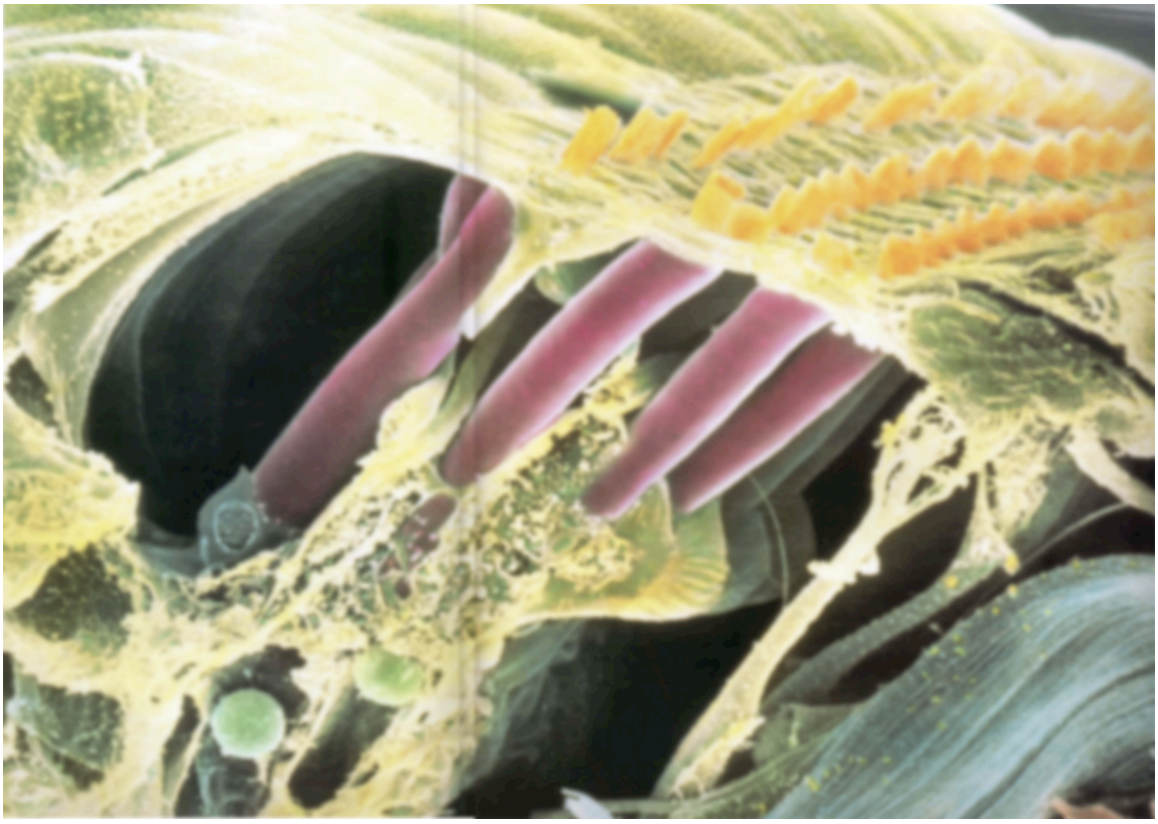

Figure E.1.4 Extreme Close-up of a Section of the Organ of Corti
From (Firefly, 2002)

These images are presented because they illustrate the complexity of the human audio data collection hardware. Whereas a computerized audio system uses two CD quality channels (44,100 samples/second, 16 bits/sample), the ear has millions of individual transducers, each collecting time based data at different locations, with patterns of time delays, phase and frequency values amongst these information channels being correlated by the neural networks in the audio cortex and brain. All of these information pathways essentially represent continuous functions in real time, while the computerized data form has very coarse granularity in both time and frequency. The action of the nervous system

is encoded as discrete nerve impulses rather than a continuous function in the mathematical sense, but the enormous number of different nerve impulses and pathways can be seen to approximate a true continuous function to a very fine granularity in both time and frequency. (Bekesy, 1960) reports detecting eddy currents in the cochlear fluids caused by sound vibrations. The hair cells in Figures E.1.4 and Figure E.1.5 respond to these fluid movements (in the tectorial chamber) as well as responding to vibrations in the basilar membrane. Neuroscience is currently charting these pathways.

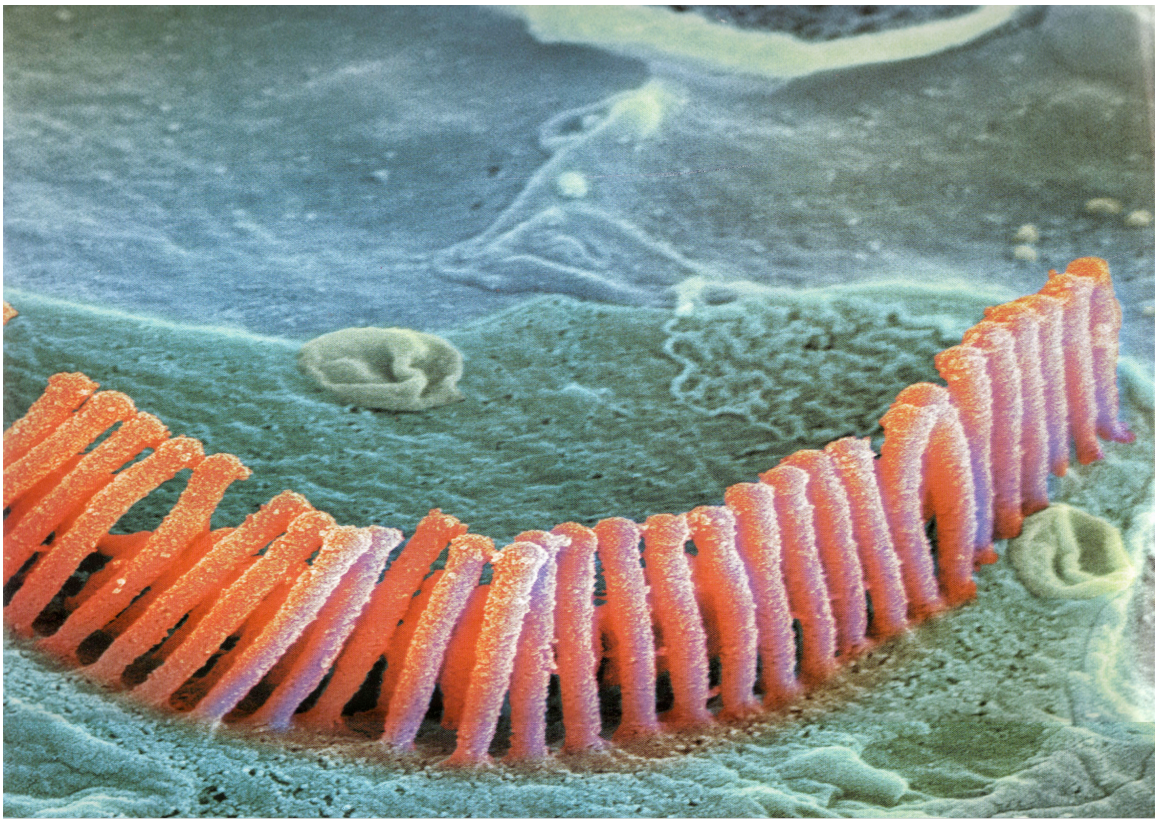

Figure E.1.5 Extreme Close-up of Frequency Sensing Cells

From (Firefly, 2002)

The transformation of audio signals into neural patterns and hence cognitive perceptions is complex and not completely understood. The main *result* of the process is the extraction of frequency and direction information from the incoming audio signal, and fusing the information into a continuous three dimensional perceptual reality. This is an

area for further research. Plomp gives a history of such scientific investigations, and cautions strongly to avoid being trapped by *a priori* thinking like reductionism (he calls it *microscopic view).* Apparently many researchers in the 150 years or so of acoustic science have believed that hearing is somehow a relatively simple process, much as early microbiologists thought that the interior world of a living cell is "formless protoplasm".

Helmholtz in the 19th century started the study of the physics of hearing, and proposed the "tuning fork", or "piano strings" model, which sees the cochlea as a fancy Fourier series analyzer that measures exact frequency components of the audio and passes them to the brain which extracts information and patterns. (Bekesy, 1960) showed that the cochlear response to frequency is more complex and subtle than merely being a row of finely spaced tuning forks as Helmholtz envisioned.

Audio vibrations enter the cochlea by the vibration of the stirrup bone on the oval window at the front of the cochlea. The signal is transmitted into the cochlea not as vibrations, but rather as a series of traveling wavefronts corresponding to the inward push on the eardrum by each incoming audio wave. These are transmitted by the bones of the inner ear (hammer, anvil, stirrup) to the oval window of the cochlea where they launch a wave disturbance in the upper chamber of the cochlea. Figure E.1.6 shows the eardrum and bones of the middle ear. The time for a wavefront to travel along the basilar membrane ranges from less than 0.1 millisecond for high frequencies to about 10 milliseconds for the lower limit of 20 Hz (Bekesy, 1960). Due to damping, the high frequencies only excite vibrations for a short distance, while low frequencies power curves peak at the far end of the cochlea.

The backward action of the incoming vibrations apparently has no effect on the frequency sensing hairs, which are only activated by the wavefront in the forward direction (Dowling & Harwood, 1986). These wavefronts are complex curves representing the instantaneous sum of many frequencies which are present in the audio signal, and the ear

extracts instantaneous frequency information from them. With each wavefront, different frequency information is induced as vibrations on the basilar membrane. The tuning of the basilar membrane creates power curves (traveling waves) from the incoming wavefronts, rather than vibrating at a particular frequency *per se.* Figures E.1.7 through E.1.11 show several aspects of the traveling wave phenomenon from (Bekesy, 1960) who mapped the frequency response of the basilar membrane. Note that the wave shapes of the traveling waves have power peaks at different locations along the length of the basilar membrane. Additionally, as the wavefront moves through the cochlear fluid, the basilar membrane shape flexes into unique shapes determined by the frequency and loudness information. The action of the basilar membrane short circuits the wavefront of particular frequencies at the location of the cochlea which is sensitive to that frequency. The energy from the wavefront is thus transmitted to the lower cochlear chamber by the basilar membrane, reducing or eliminating the energy at that frequency in the upper chamber beyond the sensitive location for that frequency. The entire collection of the various vibrational responses is probably encoded as a Gestalt as well as being decomposed by frequency (Plomp, 2002).

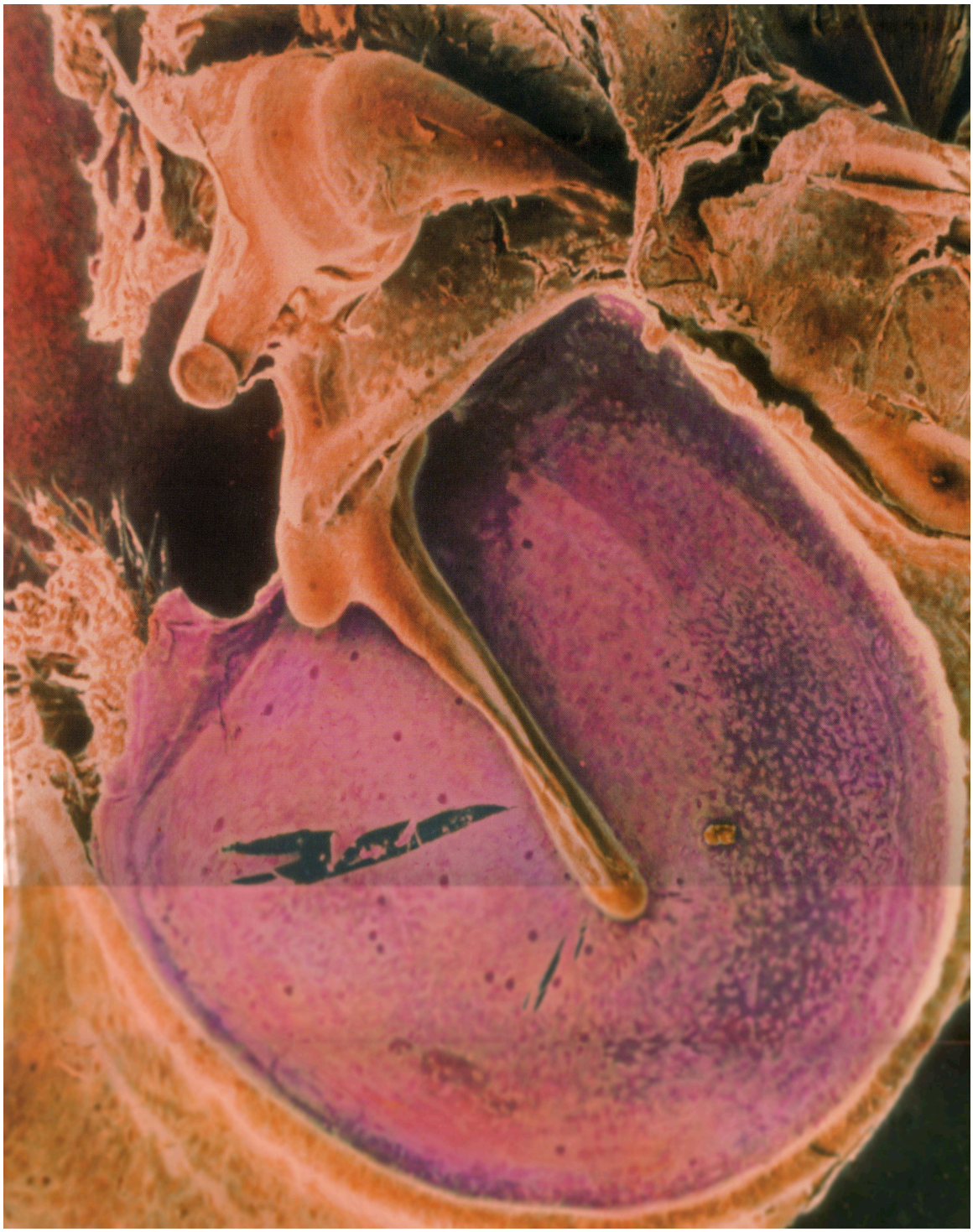

Figure E.1.6 Eardrum and Middle Ear Bones

From (Firefly, 2002)

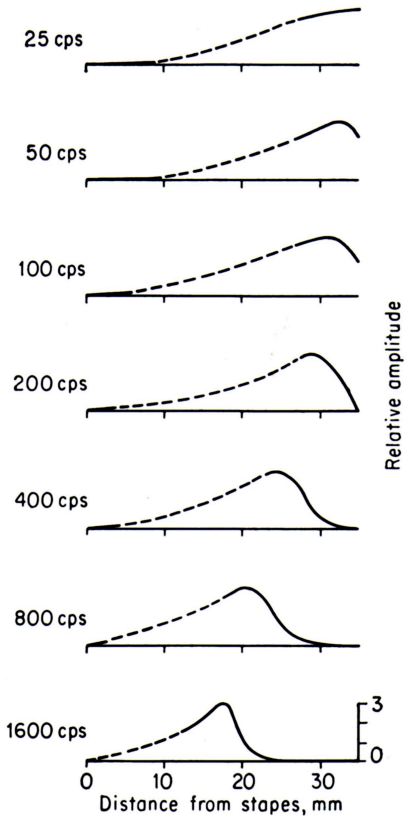

Figure E.1.7 Power vs Distance Curves in Cochlea for Several Frequencies

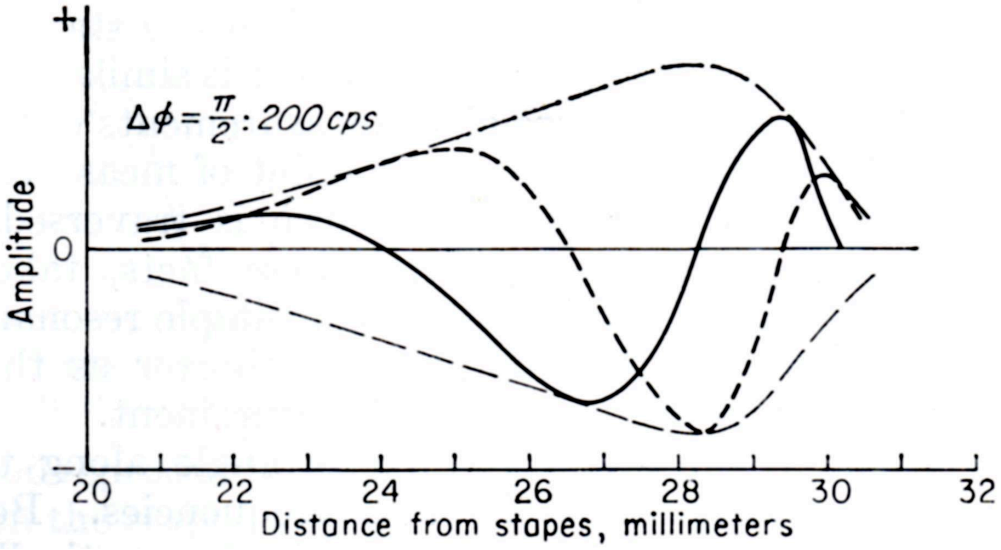

Figure E.1.8 Traveling Wave for 200 Hz at Several Moments in Time

From (Bekesy, 1960)

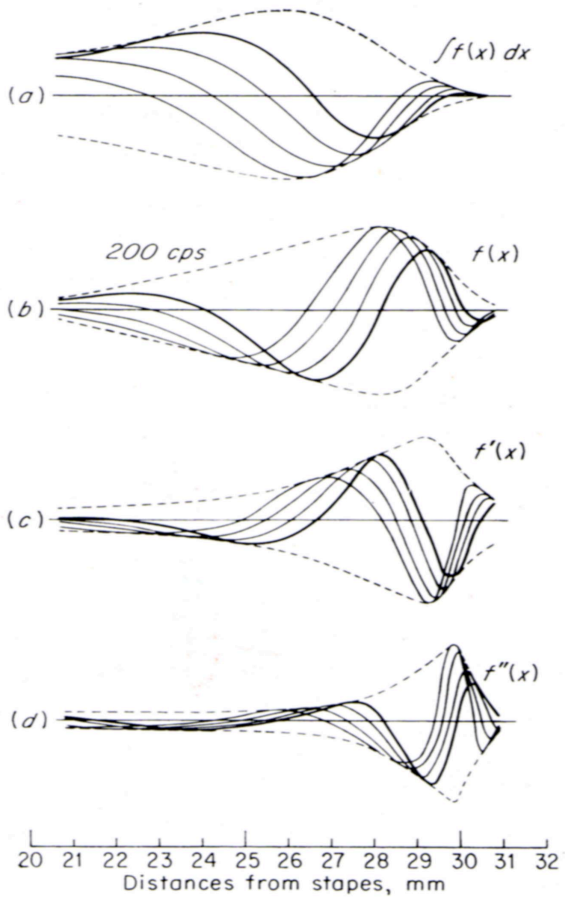

Figure E.1.9 Waveform, First, Second Derivatives, and Integral for 200 Hz

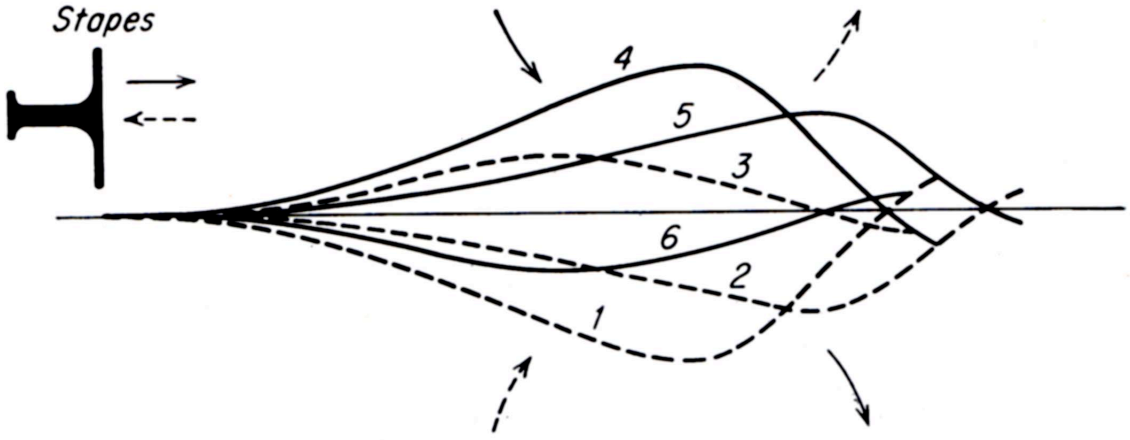

Figure E.1.10 Traveling Wave, Showing Generation of Eddy Currents

From (Bekesy, 1960)

At low frequencies, below about 100 Hz, the entire basilar membrane vibrates as a whole. Above about 200 Hz, the frequency sensors mentioned above are affected by the incoming vibrations. From approximately 200 to 2000 Hz there are two mechanisms for extracting information from the audio signal: the frequency sensors, and detection of beat patterns in the incoming waveform. The beat patterns are caused by constructive and destructive interference between waves present in the external three dimensional space, much as waves on the surface of a pond have peaks and troughs as they intersect each other. The results of both the beat and frequency detection mechanisms are interpreted as tonal information at higher cognitive levels in the audio cortex. The incoming beats directly trigger nerve impulses, while the frequency sensors respond to wave shapes of the incoming vibrations. Nerve cells have a limit to how fast they can fire, and so above approximately 2000 Hz (0.5 milliseconds), only the frequency sensors extract information. The beat pattern extraction mechanism operates down to extremely low frequencies, well below the 20 Hz "limit" of human hearing. At very low frequencies, the beat patterns are perceived as individual events rather than tones (Dowling & Harwood, 1986). NB: the beat phenomenon as reported by Dowling & Harwood may be erroneous. Events with fast onsets such as percussion sounds have time scales for the onsets in the frequency range of the beat mechanism. Thus there may be some recognition of such events in the front end of audio processing, as well as later in the cortex where neural processing stages extract timing information from the changing input stream. This phenomenon is an important consideration for the Ile Aye caixa experience described in Appendix A.

Each frequency sensing hair is broadly tuned, responding to a range of frequencies. Due to the traveling wave effect, the power distribution along the cochlea takes on different waveforms depending on the frequencies in the signal as described by (Bekesy,1960). The spatial variations of the power distribution is used for distinguishing different frequencies. The amplitude increases as the wavefront moves into the area of the cochlea that is tuned to the frequencies corresponding to the incoming wave shape. After

the wavefront passes through the tuned section, the amplitude begins to decrease. The detection of the peak is enhanced by lateral inhibition from nearby frequency sensors whose signal strength is less than the peak. The neurons in the audio cortex subtract this nearby data from the power distribution waveform, sharpening the peak relative to the broad waveform. This is one of the reasons we hear precise tones rather than a smeared combination of frequencies. (Dowling & Harwood, 1986). NB: Dowling & Harwood may be incorrect about their lateral inhibition report. While lateral inhibition is proven to have a significant role in visual perception, research in audiology has not produced clear evidence that lateral inhibition operates in hearing perception.

There is substantial evidence that the ear produces sounds by itself, called *oto-acoustic emissions.* These sounds have been detected by sensitive microphones placed in the outer ear. It is not entirely clear what role these sound emissions play in hearing, but theories that they assist in frequency discrimination are the most prevalent[3].

Mathematical models of the basilar membrane vibrations are commonly used in neuroscience research. Figure E.1.11 is an example which shows patterns of a normal frequency response, and one with a damaged basilar membrane. We include this because it provides two insights. First, it is a clear intuitive example of how loss of proper vibrational response of the cochlea contributes to hearing impairment. Second, it actively shows how the physiology of the ear helps to generate forms of information from the input sound vibrations, due to the nonlinear resonant action of the tissues and fluids. Artificial neural networks rely heavily on nonlinear response for teasing out subtleties in complex entanglements of signals.[4] DSP uses linear processing for the most part, and consequently is more limited in the availability of information than a nonlinear system.

---

[3] Dr. Dean Ayers, Southern Oregon University, Department of Physics

[4] Dr. Charles Jorgensen, Senior Research Scientist. Neuro-Engineering Lab, NASA Ames Research Center. Mountain View CA USA.

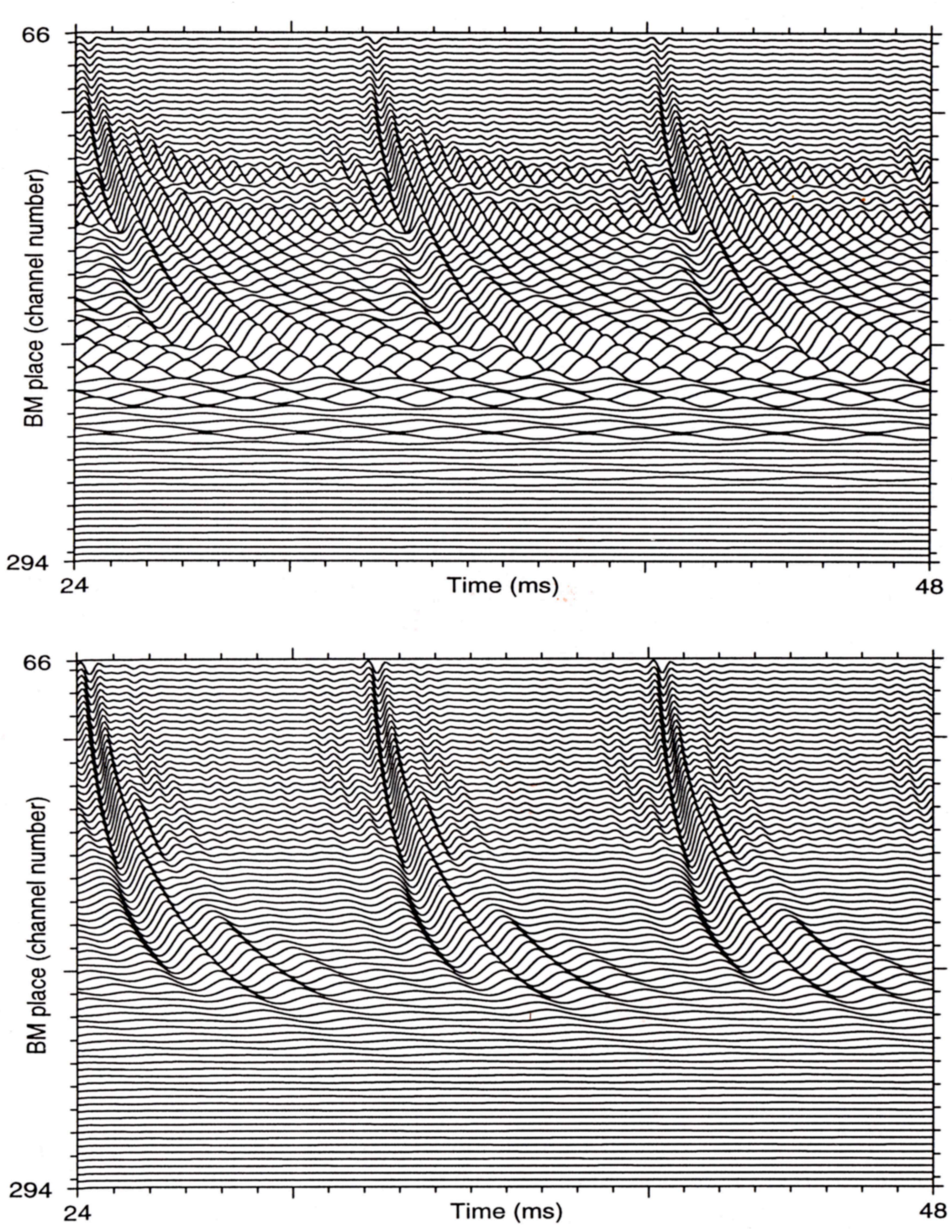

Figure E.1.11 Mathematical Model of Vibrations in the Basilar Membrane

From (Giguere & Smoorenburg, 1999) in (Dau et al., 1999)

Other mechanisms for transforming data into information features will no doubt be discovered by neuroscience in the coming years. We have scratched the surface with our readings in (Dau et al. 1999). We believe that percussion events are very useful in this type of research because the onset of events is very short, typically from 1 millisecond up to about 40 milliseconds. These quick events are easier to track in the audio cortex using EEG than are more complex sounds, such as speech or melodic instruments. Percussion events are more complex than the audio signals used in psychology research which are typically simple sine waves or square pulses. The complexity of the percussion sounds can be used to study the pattern recognition pathways in the audio cortex.

E.2 Psychological Studies of Human Perception

(Fraisse, 1982), (Deutsch, 1992) and others have done perceptual studies that identify certain time ranges as "natural" for human imitative tapping and rhythm. These are mostly in the tempo range of standard music pieces. If tempo increases or decreases beyond the natural range, most people shift to the next higher or lower synchronized pattern whose timing fits in the natural range.

This background information is relevant to the story in the appendix about learning the caixa batida from Ile Aye. We also reference human timing perceptual issues in our descriptions of creating seamless rhythmic loops. Our experience leads us to the conclusion that some of the standard psychological models of reaction time and human response time in general are inadequate. All of the studies we have read have tested subjects using isolated sequences of events. We have found that temporal context of patterns of events is an important part of a perceptual mechanism that is temporally more fine grained that the standard models of human time perception.

E.3 Human Emotions and the Meaning of Music

The connection between music and emotions is widely, perhaps universally, recognized. "Music hath charms to tame the savage beast" is an old folk saying, and music

has been used for therapeutic medical purposes. Recently, researchers have applied modern methods such as electro-encephalogram (EEG) to monitor a subject's physiological responses during music therapy (Fox, 2005). This is strong support for our contention that music, health and emotional well being are closely related. It also supports our purpose to facilitate the learning, teaching and playing of music by using technical approaches that help people understand non symbolic information and other subtleties which are fundamental to music.

BIBLIOGRAPHY

Anemüller, J. & Gramss, T. (1999). ***A Neural Network for Sound Source Separation.*** In (Dau, et al., 1999).

Bekesy, G. von. (1960). ***Experiments in Hearing.*** Translated by E.G. Wever. McGraw-Hill Book Co. New York.

Bengsston, I. (1987) ***Notation, Motion and Perception: Some Aspects of Musical Rhythm.*** in (Gabrielsson, 1987).

Beranek, L. (1954). ***Acoustics.*** McGraw-Hill Book Co. New York.

Birch, Alisdair MacRae. (2003) ***It Don't Mean a Things.*** Available online at www.alisdair.com .

Brigham, O. (1974). ***The Fast Fourier Transform.*** Englewood Cliffs, NJ. Prentice-Hall.

Cholakis, Ernest. (1995) ***Jazz Swing Drummers Groove Analysis.*** Numerical Sound. Available online at www.numericalsound.com .

Cooley, J. W. & Tukey, J. W. (1965) ***An Algorithm for the Machine Calculation of Complex Fourier Series.*** Math. of Computing. 19, 297-301.

Dau, T, Hohman, V & Kollmeister, B. (1999). ***Psychophysics, Physiology and Models of Hearing.*** World Scientific. Singapore, New Jersey.

Deutsch, D. (ed.) (1999). ***The Psychology of Music, 2nd edition.*** San Diego, CA. Academic Press Series in Cognition and Perception.

Dixon, S. (1999). ***A Beat Tracking System for Audio Signals.*** Proceedings of the Conference on Mathematical and Computational Methods in Music, pp. 101-110. Vienna, Austria.

Dowling & Harwood. (1986). ***Music Cognition***. New York. Academic Press.

Elliot, D. F. & Rao, K. R. (1982) ***Fast Transforms: Algorithms, Analyses and Applications.*** Academic Press. Orlando, FL.

Firefly Books. (2004). ***Photographic Atlas of the Human Body.*** Octopus Publishing Group, Ltd. & The Science Photo Library.

Fox, D. (2005 , 24 December.). ***Do the brainwave boogie-woogie.*** New Scientist Magazine issue 2531.

Fraisse, P. (1963). ***The Psychology of Time.*** New York. Harper and Row.

Friberg, A. & Sundstrom, J. (1999). ***Jazz Drummers' Swing Ratio in Relation to Tempo.*** Royal Institute of Technology. Stockholm, Sweden. Journal of the Acoustical Society of America ASA/EAA/DAGA conference, Berlin.

Friberg, A. & Sundstrom, J. (2002). ***Swing ratios and Ensemble Timing in Jazz Performance: Evidence for a Common Rhythmic Pattern.*** Music Perception 19(3), 333-349. http://www.speech.kth.se/music/performance/Texts/ensemble_swing.htm

Gabrielsson, A., ed. (1987). ***Action and Perception in Rhythm and Music. Papers Given at a Symposium in the Third International Conference on Event Perception and Action.*** Royal Swedish Academy of Music, #55. Stockholm.

Gabrielsson, A. (2000). ***Timing in Music Performance and its Relation to Music Experience.*** In (Sloboda ed., 2000).

Giguere, C. & Smoorenburg, G. F. (1999). ***Computational Modeling of Outer Hair Cell Damage: Implications for Hearing Aid Signal Processing.*** In (Dau, et al., 1999).

Goto, M. & Muraoka, Y. (1994). ***A Beat Tracking System for Acoustic Signals of Music*** San Francisco ACM Multimedia 0-89791 -686-7/94/0010..

Guoyon, F. (2005). ***A Computational Approach to Rhythm Description.*** PhD Thesis. University of Barcelona, Spain.

Hamer, M. (2000, 23 December.). ***It don't mean a thing if it ain't got that swing. But what is swing?*** New Scientist Magazine, 2270, p.48.

Hamming, R. W. (1983). ***Numerical Methods for Scientists and Engineers, 2nd ed.*** McGraw-Hill Book Company, New York.

Haykin, S.(1999). ***Neural Nets: A Comprehensive Introduction.*** Englewood Cliffs, NJ. Prentice-Hall.

Klapuri, A. (2004). ***Signal Processing Methods for the Automatic Transcription of Music*** PhD Thesis. Tampere University of Technology Publications 460. Tampere, Finland.

Møller, A. (2000). ***Hearing: Its Physiology and Pathophysiology.*** Academic Press. San Diego.

Neto, J.S. ((2005). ***BRJazzBook: 20 Brasilian Jazz Tunes.*** Retrieved from www.jovisan.net. October 2005.

Plomp, R. (2002). ***The Intelligent Ear: On the Nature of Sound Perception.*** Lawrence Erlbaum Associates, Publishers. Mahwah, New Jersey. London.

Press, W H., Teukolsky, S. A, Vetterling, W, T. & Flannery, B. P. (1988, 1992, 2002). ***Numerical Recipes in C++, 2nd Edition.*** Cambridge, UK. Cambridge University Press.

Sloboda, J. A.(ed.). ***Generative Process in Music.*** (2000). Oxford, UK. Clarendon Press.

Schwartz, A., Mertsching, B., Brucke, M., Nebel, W., Hansen, M. & Kollmeier, B. (1999) ***Silicon Cochlea: A Digital VLSI Implementation of a Psychoacoustical and Physiologically Motivated Speech Preprocessor.*** In (Dau, et al., 1999).

Schulze, H., Scheich, H. & Langner, G. (1999) ***Periodicity Coding in the Auditory Cortex: What Can We Learn From Learning Experiments?*** In (Dau, et al., 1999).

Seashore, C. E. (1938/1967) ***Psychology of Music.*** Dover Publications, Inc. New York.

Tzanetakis, G. & Cook, P. (2002) ***Human Perception and Computer Extraction of Musical Beat Strength.*** Proc. of the 5th Int. Conference on Digital Audio Effects (DAFx-02).

Young, R. M. (2001) ***An Introduction to Nonharmonic Fourier Series, revised 1st edition.*** Academic Press. San Diego, CA.

Waadelund, C. H. (2004) ***Spectral Properties of Rhythm Performance.*** Norwegian University of Science and Technology. Trondheim, Norway.