5-21-2004

Multipitch Analysis and Tracking for Automatic Music Transcription

Richard Baumgartner
University of New Orleans

MULTIPITCH ANALYSIS AND TRACKING FOR AUTOMATIC MUSIC TRANSCRIPTION

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
The Department of Electrical Engineering

by

Richard T. Baumgartner

B.S., University of New Orleans, 2002

May 2004

ACKNOWLEDGEMENTS

The work presented in this thesis results from the help, guidance, motivation, and enthusiasm of so many people. Much credit and many thanks goes to Dr. Dimitrios Charalampidis, my thesis advisor. His willingness to give of his time, wisdom, insight, and interest for the project made the endeavor a success. A special thanks is also in order for those who served on my thesis committee; Dr. Edit Bourgeois, Dr. Vesselin Jilkov, and Mr. Ken Lannes.

I would also like to thank Ryan Thiel, a fellow graduate student and good friend, for assisting in generating many of the test signals used in this project. Sincere appreciation goes to my fiancée, Adrienne Couret. Her patience and motivation helped me to accomplish all of my academic pursuits. I would also like to heartily thank my parents, Rick and Rhonda Baumgartner, for showing me the reward of hard work and the feeling of accomplishment from a job well done.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Music has always played a large role in human life. The technology behind the art has progressed and grown over time in many areas, for instance the instruments themselves, the recording equipment used in studios, and the reproduction through digital signal processing. One facet of music that has seen very little attention over time is the ability to transcribe audio files into musical notation. In this thesis, a method of multipitch analysis is used to track multiple simultaneous notes through time in an audio music file. The analysis method is based on autocorrelation and a specialized peak pruning method to identify only the fundamental frequencies present at any single moment in the sequence. A sliding Hamming window is used to step through the input sound file and track through time. Results show the tracking of nontrivial musical patterns over two octaves in range and varying tempos.

CHAPTER 1

Introduction

In general, sound files contain complex harmonic structures. There are actually very few cases that can be considered as having a simple harmonic structure, i.e. a single speaker or a single note from a single instrument, and even then the instrument would have to produce a very pure tone like that of a flute. In almost all cases, harmonics are generated as integer multiples of the fundamental frequency. The amplitudes of the harmonics generated are determined by the characteristics of the sound source. These characteristics color the sound to create unique representations for the same musical note, which is how the same note being played on both a guitar and a piano can be easily distinguished as coming from one or the other. In the real world both complex harmonics and noise exists. These aspects of audio processing complicate the determination of the fundamental frequency of a signal. A method of seeing through the noise and eliminating harmonics is required for multipitch determination.

The first methods introduced for multipitch determination were severely limited in their effectiveness to identify multiple pitches. Most of the earlier methods were only able to identify at most two pitches. Advances were made in the techniques used and more simultaneous pitches could be identified. These methods, however, became computationally cumbersome for reasons that will be discussed in a later section. Despite the expensive processing, multipitch analysis has many applications in today's audio processing.

The first application for multipitch analysis is for automatic transcription of polyphonic music. Music contains information from several instruments, each playing several notes

simultaneously. Each note has harmonics associated with it and all of the frequencies are mixed together in a single channel. The proposed automatic music transcription algorithm must identify each fundamental frequency present in order to musically annotate a note.

Artists wish to transcribe their music on paper, which typically requires proceeding note-by-note for each instrument manually writing out what is played. To aid in music transcription some instruments, like keyboards, are equipped with a Musical Instrument Digital Interface (MIDI), which allows direct connection to a computer equipped with transcription software. Since this interface is simply not available on most instruments, transcription is a long and laborious procedure. The ultimate goal would be to have a musician play a track of music and have the automatic transcription follow directly from the recording and require no manual input. This thesis provides a starting point for such a concept.

To implement the multipitch tracking algorithm, a two channel autocorrelation-based multipitch analysis method proposed by Tolonen and Karjalainen serves as a foundation. The proposed automatic transcription algorithm modifies the two channel method slightly and then applies it to the specific challenges present in multipitch analysis of music. The proposed algorithm was implemented in MATLAB and the results are given in a later chapter.

The chapters of this thesis are organized as follows: Chapter 2 provides some background information on the concept of automatic music transcription and applications for this service. Chapter 3 presents introductory material on MIDI and its ties to music transcription. Chapter 4 provides a great deal of background information on past research regarding human pitch perception and multipitch analysis methods in general. Chapter 5 offers a discussion on the proposed modifications to the two channel multipitch analysis model, which leads to the results of the thesis presented in Chapter 6. The results show the successful identification of the

fundamental frequencies present in the signals tested over two octaves and varying tempos. Chapter 7 concludes the thesis with a discussion and conclusions section, which offers additional topics for future research.

CHAPTER 2

Automatic Transcription of Music

Transcription of Music can be defined as the act of listening to a piece of music and of writing down musical notation for the notes that constitute the piece. This implies the extraction of specific features out of a musical acoustic signal, resulting in a symbolic representation that comprises notes, their pitches, timings, dynamics, and may include the identification of the instruments used [1].

Depending on the complexity of the piece to be transcribed, the process can be extremely time consuming and tedious. People with musical education usually have a much easier time to transcribe polyphonic music than does someone with no musical training. Polyphonic music describes a piece that contains multiple notes played concurrently. An understanding of musical styles, how different instruments sound, and a working knowledge of music theory can give musically educated listeners the ability to resolve even the most complex and rich polyphonies. The ability to interpret the music taking into account the music theory and experience rather than strictly analyzing a piece enables manual human music transcription to far outperform current automatic transcription systems. Monophonic (music that contains only a single note played at any one time) transcription solutions have developed over time and employ many accepted algorithms, including time-domain techniques based on zero-crossing and autocorrelation, as well as frequency-domain techniques, based on the discrete Fourier transform and the cepstrum. The cepstrum involves using the log function along with the FFT to warp the spectrum. These algorithms, when applied to monophonic music, demonstrated reliable operation and became

commercially applicable. The majority of these algorithms could be implemented in real-time, and received great acceptance in the area of speech processing. The complexities of polyphonic music, however, make transcription a more difficult endeavor and such algorithms have showed less relative success. The lack of success in the past opens many opportunities for research, however, the area of polyphonic music transcription has received far less research effort due to the more limited application when compared to speech processing [1].

Moorer [2] and Piszczalski and Galler [3] proved the first attempts towards automatic polyphonic music transcription back in the 1970s. At this time they coined the term *automatic music transcription*. Piszczalski and Galler's system operated on a frequency-based front-end, and tried to measure the fundamental directly from the resulting spectrogram. They limited the scope of their work to only single instrument analysis, and only on instruments with a relatively strong fundamental frequency. The fact that the instruments considered contained a strong fundamental was critical in their algorithm's success. Moorer's system was less restricted than the former efforts, however, the scope was limited to musical pieces with a maximum of two instruments. In addition, he restricted the inputs to limit the allowable simultaneous musical intervals [1]. The commercial application of such algorithms was impractical due to all the imposed constraints. Further attempts to make automatic polyphonic music transcription more reliable were made, but efforts could not produce a robust method.

Hawley presented a system in 1993 that he claimed could transcribe polyphonic piano performances based on differential spectrum analysis [4]. He implemented a short-time spectral analysis and spectral comb filtering to extract note spectra, and looked for note on-sets in the high-frequency power and with bilinear time-domain filtering. He boasted the system was fairly successful, although with narrow scope [1]. While Hawley's research did not produce a

comprehensive package, the effort to create an automatic music transcription system continues today because great possibilities exist with such a system.

2.1 Applications

Through music transcription information, a computer would be have the ability to perceive music in a way never possible thus far and from this perception stems many applications. Nevertheless, Martins and Ferreira [1] point to some fields of application for current and future music perception systems, such as:

- *Music transcription systems*. Mainly of interest to music composers and musicians, these systems could efficiently analyze compositions that exist only in the form of acoustic recordings. The symbolic representation results allow flexible and selective musical analysis, editing and mixing, otherwise difficult or impossible to perform.
- *Synthetic Performance Systems*. With the ability to hear music, systems for enabling musical collaboration between humans and machines could become more expressive and capable.
- *Algorithmic Composition.* Current computer-composition systems still need to have their outputs evaluated by humans so that they can be adjusted in order to create meaningful and interesting musical pieces. The use of human perception models may enable machines to evaluate and critique their own compositions, making it possible to attain a more truly automated composition process.
- *Visual Music Displays.* Real-time computer graphic routines can generate compelling multimedia experiences when synchronized with music. By enabling real-time music listening as a component of such a system, the display can be made to work interactively with any music input (i.e. musical improvisations).

- *Access to musical databases*. Using robust automated music perception solutions, systems could be built to augment Internet-based music recommendation services with the ability to make musical judgments. Musical databases could also benefit from the automatic music transcription and recognition as a way to improve indexation, classification, and retrieval of information (and so following the line of activities defined by the MPEG7 standard). Users of such databases could query the system using examples or higher order descriptions of the music pieces they were looking for.
- *Structured Audio Encoding*. The theory of musical listening could shed light on structural regularities in music that could be used in structured audio coding. This is not only in the case of music transcription for generating note lists, but in more subtle ways such as identifying perceptually equivalent sounds for coding. As *loudness* and *masking* models led to the development of low-bit rate perceptual audio coders, so can psychoacoustic music processing pave roads to new techniques for audio coding and compression.
- *Automatic Teaching Systems*. A new generation of assisted music teaching tools could include the capability of automatic music recognition, allowing the system to hear and evaluate the performance of a musical student. The system would be able to analyze the notes played, their pitch precision, their timings, dynamics, and even the expressiveness of the notes and then criticize and make suggestions to the student's performance.

CHAPTER 3

Musical Instrument Digital Interface (MIDI)

MIDI is a communications protocol that allows digital instruments to interact with each other and with computers. MIDI has become the primary digital production tool for musicians since its invention in 1983. A MIDI file contains no sounds, just instructions describing the notes played in a performance and related information [5].

The original intent for the protocol was to control digital keyboards. This purpose alone was quickly overshadowed as soon as computers entered the studio. Computers were more frequently connected to the MIDI chain as software became available for recording, printing, and editing musical symbols. Graphically-based design programs were introduced to edit sound files by composing directly to musical notation. This drag-and-drop editing provided an excellent user interface, but the MIDI file was only the first step in creating musical output.

Research and development concentrated on creating new methods of generating sounds. In the 1980's keyboard synthesizer technology evolved and became very popular. Use of the synthesizer grew until it finally joined the list of the widely used musical instruments. Similarly to the concept of the one-man-band, where a single individual controls several instruments simultaneously, synthesizers allowed new sounds to be created by combining the timbres of more than one instrument. This process of stacking sounds on top of each other lead the industry to the term 'layering.' The obvious problem is allowing MIDI devices from different manufacturers to communicate. A small group of synthesizer design technicians from different manufacturers met in 1983 to discuss a communications protocol to control a number of

synthesizers from competing manufacturers via external cables that allowed either instrument to control the other. They called it the Musical Instrument Digital Interface, or MIDI [5].

3.1 MIDI Protocol

The concept of the protocol was to design a system where two synthesizers could communicate using MIDI in the same way that two computers can communicate over modems. The musical performance was now dictated only by the data flow between MIDI devices. The MIDI information contains the desired note to be played and the duration of that note. This is passed in the form of byte sequence that supplies the value of the note, a start command, and finally an end command. In addition, supplemental information is transferred to indicate the velocity of a keystroke and the volume of a note. The MIDI information also contains the desired instrument, or patch, that should be used by the device to create the audio output [5].

3.2 Software Applications

There are many software applications available, which serve a variety of functions using the MIDI interface. Possibly the most common is the sequencer. The concept of a sequencer is to transform a computer into a complete multitrack recording studio for MIDI tracks. These sequencers allow the computer to perform many functions such as recording, storing, editing, and replaying MIDI data for all of the different tracks. The MIDI data is then stored in the Standard MIDI File (SMF) format as a song project. From this standardized file format, any sound card or synthesizer can realize the performance. Most sequencers provide extensive editing capabilities [5].

Another application is music notation programs. These programs represent the MIDI data as a musical score on the monitor. This musical notation can be edited using drag-and-drop

functionality, which eliminates the need for an external MIDI device like a keyboard or synthesizer.

3.3 Computers in MIDI Chains

The functionality of a computer in the MIDI chain is no different than any other MIDI device in the chain employing a three port network. The typical assignment for this three port interface is the following: *IN, OUT, THRU*. A computer can serve as both a MIDI controller and as a recorder depending on the external connections made. The computer can send signals to external MIDI devices instructing them to play specific notes using specific instrument sounds, or it can receive information sent from other controllers in the loop and record the MIDI information for future editing on the computer [5].

The latter property is what is of interest for future work on this thesis because it is supposed that once the algorithm is successfully implemented, the information gathered (notes being played, note start time, and note end time) can be imported into a MIDI editor and the transcription will be complete. Not only will transcription be accomplished, but also the data can be edited to instruct other sequencers to play the same notes using a different instrument. This could be of great use to a musician who has the knowledge of one instrument, but would like to include in a recording sounds from an instrument the musician is unfamiliar with playing.

CHAPTER 4

Pitch Perception Models

Many pitch detection algorithms employ some form of a perceptual model to aid in determining pitches present in a signal. Creating such a model for human hearing has been the topic of study and debate for many centuries.

4.1 A Historical Review

4.1.1 Interval and ratio, pitch and frequency

The first psychophysical model was attributed to Pythagoras (6th century BC) for relating *musical intervals* to ratios of string length on a monochord [6]. A board was used with two bridges on the extreme ends and between which a string was stretched. The string was then divided by a third bridge placed in between the extreme bridges. Pythagoras learned that intervals of unison, octave, fifth, and fourth arise for length ratios of 1:1, 1:2, 2:3, 3:4, respectively. This experiment is considered one of the first psychophysical models. The model yields insight into the link between human perception of musical interval and a ratio of physical qualities as expressed by length. Two centuries later Aristoxenos disagreed with the Pythagoreans that these length ratio numbers are related to music. He instead argued that musical scales should be defined based on what one *hears* rather than physical phenomenon. In 1581, Vincenzo Galilei also confronted the role of numbers used in music. His argument stemmed from a slightly different perspective than Aristoxenos. He conducted his own experiment to test the relation between numbers and musical qualities. He used weights to

change the *tension* of a string, and found that the previous intervals arise for ratios 1:1, 1:4, 4:9, and 9:16 respectively [7]. These ratios are considerably different and more complicated than the results from Pythagoras.

The notion of *pitch* was introduced by the Greeks. This can be defined as "a quality by which sounds can be ordered from grave to acute."[7] Galileo Galilei further recognized the relation between ratios of string length and ratios of *vibration frequency*. Mersenne modified Galileo's findings slightly by using strings long enough to actually view the vibrations. From this he was able to determine the true frequencies of each note in a scale [7]. These findings proved beyond any doubt that the perception of musical pitch is directly tied physical numbers and physical properties. These physical properties would be further investigated yielding great insight into the basics of signal analysis.

4.1.2 Superposition Methods

Mersenne was the first to document that within the sound of a string, or a voice, that he could hear up to five pitches. These additional tones corresponded to the fundamental, the octave, the octave plus fifth, etc. He found it difficult to believe that a string could simultaneously vibrate at more than one frequency to cause the upper frequencies that he experienced. This concept was more clearly defined in 1701 when Sauveur introduced the terms "fundamental" and "harmonic" to describe the phenomenon that troubled Mersenne. In the 18th century, several physicists including: Taylor, Daniel, Bernoulli, Lagrange, dAlembert, and Euler attempted to mathematically describe the existence of the fundamental and harmonics. In particular, Euler's contribution of the concept of *linear superposition* made it easy to understand the compound simultaneous vibrations of a string [7].

While Mersenne and Galileo were correct to assume periodicity in the vibrations, the oversight they made was not observing the waveform, or shape, of the vibration. The fact that the vibration was not a pure sinusoidal form meant that multiple frequencies of vibrations were present. The logical extension is then made to any *sum of sinusoids* using Euler's principle, assuming a linear system. This method is quite general and the fact that *any* shape can be obtained in this way was proven in 1820 by Fourier. More specifically, any periodic waveform can be expressed as the *superposition* (or sum) of sinusoids with frequencies that are integer multiples of the fundamental frequency [7].

4.1.3 Pattern Matching

Everyday we are confronted with incomplete audio patterns. Given this partial data our brain is well adapted to "reconstruct" perceptually the pieces that are missing. Models describing the pattern matching phenomenon assume that this is how pitch is perceived when the fundamental partial is the necessary correlate of pitch. The fundamental partial can be missing and provided that other parts of the pattern, harmonics often associated with it, are present the perceived pitch is the same [7].

The best-known models for pattern matching are those of Goldstein [8], Wightman [9], and Terhardt [10], but de Boer introduced an early form of this concept in 1956 in his thesis. Each model has unique insights, but all of these are closely related. The model proposed by Goldstein is probabilistic and performs optimum processing of a set of estimates of partial frequencies (obtained by a process that is not defined). Wightman proposed a limited-resolution profile of stimulated activity across the cochlea. This profile is then fed to a hypothetical internal Fourier transformer to yield a pattern similar to the autocorrelation function. Terhardt suggests that each partial can be considered as creating its own sensation of *spectral pitch*. From

the spectral pitch an internal template derives a *virtual pitch* that matches that of the (possibility missing) fundamental [7]. The internal template is learned and expands over time as an individual encounters more virtual pitches.

A real correlation is present between pattern-matching models (those described above as well as others) and spectrum-based signal-processing methods for *fundamental frequency estimation*. A few of these estimation techniques include: subharmonic summation, harmonic sieve, autocorrelation, and cepstrum. The latter two are receptive to the regular pattern of harmonics because of the fact that the Fourier transform, applied to a spectrum (power spectrum for autocorrelation, log spectrum for cepstrum), is also sensitive to the harmonic pattern [7].

The model proposed by Terhardt is unique in that internal templates are learned through time by exposure to harmonically rich stimuli. This is an interesting notion that humans build a database of possible virtual pitches in their mind, but this concept is also limiting. The constraint lies in the implication that missing fundamentals can only be reconciled if the virtual pitch has been established based on past experience and exposure. In showing that templates might be learned by exposure to noise, Shamma and Klein strengthened the argument against such a learned template. This assertion reinforces the concept that the internal template used for pattern matching is merely a mathematical function that is discovered rather than something learned [7]. This is a far more reasonable assertion that not every possible virtual pitch must be learned before it can be recognized. Examples of objects or processes that display pattern matching properties include both the autocorrelation function and the vibrating string. An example of this phenomenon in the autocorrelation function will be shown in the results section using the musical chord experiment. The string embodies this property in that it will be sympathetically stimulated by harmonics of its tuned fundamental.

4.1.4 Temporal Models

Democritus and Epicurus (5th and 4th century BC respectively) introduced the concept that sound propagates as a body (any thing that produces a sound) emits *atoms*. This idea is closely associated to the concept that as a string vibrates it "hits" the air repeatedly, and that the pitch represents the rate at which sound pulses hit the ear. If this is the case, it would be far simpler to observe the time interval between two consecutive atoms or pulses, instead of waiting for a train of pulses to build up sympathetic vibration in a resonator [7].

The early temporal models assumed that patterns of pulses are only handled by the brain. These tended to be less elaborate than resonance models. An apparent contradiction existed between contemporaries who disagreed as to where the pitch perception actually occurred. Compare, for example, Anaxagoras (5th century BC) for whom hearing involved *penetration of sound to the brain*, and Alcmaeon of Crotona (5th century BC) for whom *hearing is by means of the ears, because within them is an empty space, and this empty space resounds* [7]. The second assertion is more descriptive in that it ties the physical construction of the ear with known resonance theories.

Rutherford is one of several scholars that contested the supposition that sound was perceived and processed in the brain. He conducted experiments and observed the maximum nerve conduction rates in frog or rabbit nerves to be 352 per second. These rates were insufficient to carry the full range of sounds (up to 4-5 kHz for musical pitch). Wever and Bray reduced the need for high firing rates in 1930 by introducing the "volley theory". Subsequent measurements from the auditory nerve confirmed that the volley principle is essentially valid (in a stochastic form), in that synchrony to temporal features is measurable up to 4-5 kHz in the auditory nerve. Synchrony is also observed at more central neural relays, but the upper frequency limit decreases as one proceeds [7].

The concepts of temporal and resonance models differ in the amount time necessary to make a frequency measurement. Resonance models require more time than temporal models because the energy by accumulation of successive waves must build up a measurement can be taken. This requires time that varies inversely with *frequency resolution* [7].

In contrast, time-domain models require only enough time to measure the interval between two consecutive events. This time is augmented slightly by the time necessary to make sure that each event is an event, plus time to ensure that they are not both part of a larger pattern [7]. Two periods of the lowest expected frequency is approximately equal to the time required for time-domain models to take a measurement. The accuracy of these models is only limited by noise or imperfection in the implementation.

The necessity to identify events is a weakness of temporal models, as described so far. Whether the signal is to be analyzed by a model or perceived in the ear, events need to be extracted from the waveform. In the case of simple waveforms, identifying events is trivial with several possible approaches, two of which are an analysis of peaks or zero-crossings. When waveforms become more complicated and contain multiple simultaneous pitches, the problem becomes more complex. The information contained in the peaks and zero crossings no longer defines events because of the complex interaction between the different frequencies present. The task of identifying events becomes nearly impossible as these cases become more and more complex.

Early models clearly displayed this weakness in the *phase sensitivity* of the models. A method that simply measures the distance between peaks will be unable to determine the correct pitch if multiple peaks occur within one period. Pitch is often invariant for such phase variations, thus a method is required that is not phase sensitive [7]. Moreover, a method is

required which is both phase insensitive and can determine a periodicity measure of complex waveforms.

4.1.5 Autocorrelation

The response to the weaknesses present in temporal models is the *autocorrelation* (AC) model. For this model each sample of the waveform is used as an event. Each sample is then compared to every other sample, and the inter-event interval that gives the best match (on average) indicates the period [7]. More specifically, the comparison is accomplished by multiplying samples and summing the products over a sliding time window. This method is closely related to the convolution theorems where the sliding function is not reversed in time. If samples are approximately equal their products will be relatively large, and so the autocorrelation function (ACF) has a peak at the period (and its multiples). The fact that repeating peaks occur will be topic for a more detailed discussion in a later section. All peaks in the ACF become potential identifiers of pitch [7].

Licklider introduced the original AC model. The ACF in this model was calculated within the auditory nervous system, for each channel of the auditory filter bank. A more detailed discussion of this model will be provided in a later section. Meddis and Hewitt, which will also be discussed in more detail a later section, reformulated the Licklider model. Moore asserted a similar model based on first order interspike interval statistics. Another variation on this model was proposed by Patterson and several colleagues called the *strobed temporal integration* model. This model described how patterns are cross-correlated with a *strobe* function consisting of one pulse per period. A simpler predictive model was proposed by Yost which was based on waveform autocorrelation. Still other autocorrelation models exist in which the ACF was

produced by an internal “Fourier transformer”. This transformer acted on a specific outline coming from the cochlea [7].

The *Weiner-Khintchine theorem* is an important theorem as it states that ACF and power spectrum are Fourier transforms one of the other. With this in mind, the ACF can be seen as a combination of two concepts both *spectral analysis* and *pattern matching*. The ACF being a close derivative of two competing schools of thought is a very interesting observation. The two concepts obviously differ in actual implementation in the auditory nervous system, in characteristics such as frequency versus temporal resolution, and lastly in the way they can be modified to respond to complex combinations of tones [7].

An interesting exercise is to relate the autocorrelation process to the previous string experiment. The execution of autocorrelation as defined requires a delay. Associated with the delay is a multiplying factor. The delayed patterns are multiplied with time-stationary patterns to produce the autocorrelation function. The concept of delay for the string acts as a self-stimulating effect. This effect is described by the process where delayed patterns are *added* to undelayed patterns, and the result is delayed once more. This identifies both the basic similarity and difference between the string experiment and AC. The definition of AC allows for one delay at most. In the string, however, many delays are realistic. Moreover, these multiple delays are essential for the accumulation of resonance energy that allows the string to be highly selective [7]. From this historical overview of pitch perception and modeling, more specific methods will be introduced and detailed in the next sections.

4.2 Human Auditory System Biological Overview

The outer and middle ear serves as a channel or funnel for the sound and directs it to the inner ear. The inner ear, or cochlea, encodes the information from the acoustic signal into a

multi-channel representation that can be considered as instantaneous nerve firing probabilities [11]. A model for this encoding can be modeled in the following way:

- The model for the sound propagating along the Basilar Membrane (BM) is a cascade of second-order filters. As the BM responds to input frequencies, that movement is sensed by the inner hair cells.
- The inner hair cells physically respond to movement of the BM in only one direction. This can be implemented in the model as Half-Wave Rectifiers (HWRs) that sense the output of each second-order filter. The HWR introduces nonlinearity and serves to convert the motion of the BM at each point along the cochlea into a signal. That represents both the envelope and fine time structure because of the nonlinearities.
- The final stage of modeling involves the compression that occurs to the signal before it can be carried on the auditory nerve. This is modeled using Automatic Gain Control (AGC) to compress the dynamic range of the input to simulate the ear's adaptation to loudness.

The second stage of processing produces a two-dimensional image in which each row is the running short-time autocorrelation of the corresponding cochlea channel. Finally, a pitch detector combines the information in all the channels of the two-dimensional representation to identify a single pitch [11].

The previous was only an introduction to biological human modeling. The next section will detail the theory behind one of the original proposed autocorrelation models for pitch perception.

4.3 Licklider's "Duplex Theory"

The "Duplex Theory" of pitch perception accurately models how humans perceive pitch [12]. This accuracy comes at the cost of expensive processing power and time as will be described later. In general, this model for pitch detection combines a cochlear model with a bank of periodicity detectors computing autocorrelation. Since the autocorrelation function is performed independently for each channel, this detection method remains mostly insensitive to phase shifts across channels. The information in the correlogram is filtered, nonlinearly enhanced, and summed across all channels yielding an enhanced summary autocorrelation function. From this function, peaks are identified and a pitch is then proposed that is consistent with those time lags found [11].

In contrast to methods that determine pitch based on relatively simple representations of the signal such as waveform or spectrum, this method attempts to incorporate the human perception system to determine pitch. The representation that is used by the pitch detector, as proposed by Licklider, is the correlogram. The correlogram is distinctive in its concise representation by showing the frequency information and time structure of a sound on independent axes. An algorithm then analyzes the data contained in the correlogram and determines a single pitch. Humans have highly complex natural methods to formulate such decisions [11]. An example can be seen in music melodies, as these melodies represent abrupt changes in pitch. These frequency jumps cannot be modeled or predicted because there are so many 'next note' possibilities.

Licklider's "Duplex Theory" was first published in 1951, but his goal was more to model human pitch perception rather than to develop a method to be used for pitch detection and identification. Using the model comes at great computational expense because of the complexity of the cochlear model and the large bank of autocorrelators [11]. The image of a correlogram

was not published until 1984 specifically because the cost of the process served as such a deterrent.

The correlogram is determined by computing the short-time, windowed autocorrelation of the output of each cochlear frequency channel. The correlogram is an animated representation of the sound that displays spectral information along the vertical axis and time structure along the horizontal axis. The theory of operation stems from the assertion that if the input sound is periodic, the autocorrelation function results for all cochlear channels will have a high peak in time that corresponds to a time lag equal to the period, or similarly the inverse of the frequency [11]. The Patterson's Pulse Ribbon Model [13] is closely related to the correlogram and is also used for pitch perception. This method searches for common firings across cochlear channels after delaying the outputs of individual neurons. The correlogram is more robust since it sums up the firings from multiple channels and across time, in contrast to the Patterson method that does not necessarily consider all channels. Moreover, an autocorrelation is insensitive to the phase between channels since it is only concerned with time differences.

The algorithm to implement this pitch detector consists of only four steps. The first step is used to enhance the peaks through preprocessing. The second step is to sum all values across all frequencies at each time lag in the enhanced correlogram. The peaks in this summary function indicate a high probability of periodicity at that time lag. The third step incorporates evidence from the subharmonics of each potential pitch to make the estimate more robust. The forth and final step is to pick the largest peak. The frequency of this note is then identified as the reciprocal of this time lag [11].

4.4 Meddis and O'Mard Unitary Pitch Perception Model

The figure below shows a block diagram of the Meddis-O'Mard unitary pitch perception model. A filterbank is used to imitate the frequency resolution of perceptual hearing. The filterbank is most often implemented using bandpass gammatone filters which provide critical audio band selectivity and display the property of the bandwidth of the filter is a function of the center frequency. The filterbank divides the signal into many channels and each channel is half-wave rectified and lowpass filtered (about 1kHz) in order to emulate the reaction of the hair cells. Next a periodicity measurement is calculated in each channel by calculating its autocorrelation function (ACF) or a similar periodicity measure. Finally, the all of the ACFs are summed from each channel to yield a summary autocorrelation function (SACF) that shows the overall periodicity properties of the incoming signal [14].

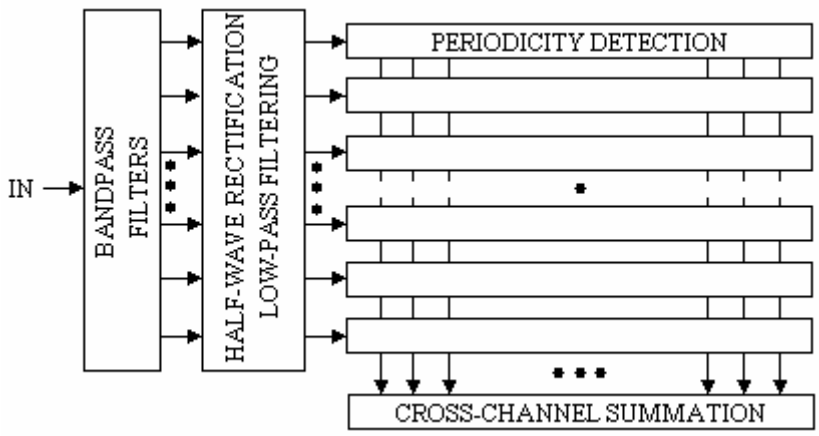

Figure 1 - A block diagram of Meddis-O'Mard's unitary pitch perception model

The performance of the Meddis and O'Mard unitary pitch perception model is capable of simulating several important cases of perception when compared with results from psychoacoustical experiments. The major limitation with using the unitary model, from a practical application point of view, is that the filterbank may have anywhere from 40-120

channels which makes filtering and computing the channel autocorrelations a computationally heavy task [14]. The next major advancement in pitch perception modeling was the advent of shaping a signal to more closely represent how it would stimulate the human ear. This work will be discussed in the following section.

4.5 Warped Linear Prediction (WLP)

In general, Warped Linear Prediction (WLP) involves the linear prediction (LP) process being applied to frequency warped signals. Orthonormal FAM functions are used to realize the warping. FAM functions are those in the class of Frequency-Amplitude Modulated complex exponentials.

Many methods have been proposed in the field of speech and audio processing to compress the signals according to the human auditory system response to a stimulus. The earliest methods used for speech coding implemented a filterbank with gradually increasing channel bandwidths. This further advanced when speech recognizer algorithms employed this principle of a nonuniform resolution preprocessing. Most recently and actively studied are perfect reconstruction orthonormal filterbanks and wavelet-based techniques. Oppenheim, Johnson, and Steiglitz first introduced the concept of non-uniform resolution FFT and the main purpose of their study was to produce the warped spectra by using a network of cascaded first-order allpass sections for frequency warping of the signal and then to apply standard FFT. The concept of a frequency-warped FFT was taken one step further when Strube applied the idea to WLP. Strube introduced a cascaded, first-order, allpass network to yield the frequency warping corresponding to the auditory Bark scale. The frequency warped signal is then used to calculate the autocorrelation function for LP. The LP process, with the frequency warping included as a

preprocessing stage, directly results in representation for the signal corresponding to audio perception [15].

The frequency resolution of the human auditory system is often expressed using the Bark scale. The critical bandwidth for resolving signal components is broader for higher frequencies, thus less resolution is needed for high frequencies. Employing the WLP can significantly reduce the prediction filter order over LP. There are many approximations for the Bark scale, but the following shows the general shape of the curve. The approximation used is shown below.

$$B = 13 \cdot \tan^{-1}\left(0.76 \cdot f/1000\right) + 3.5 \cdot \tan^{-1}\left(\left(f/7500\right)^2\right) \qquad (1)$$

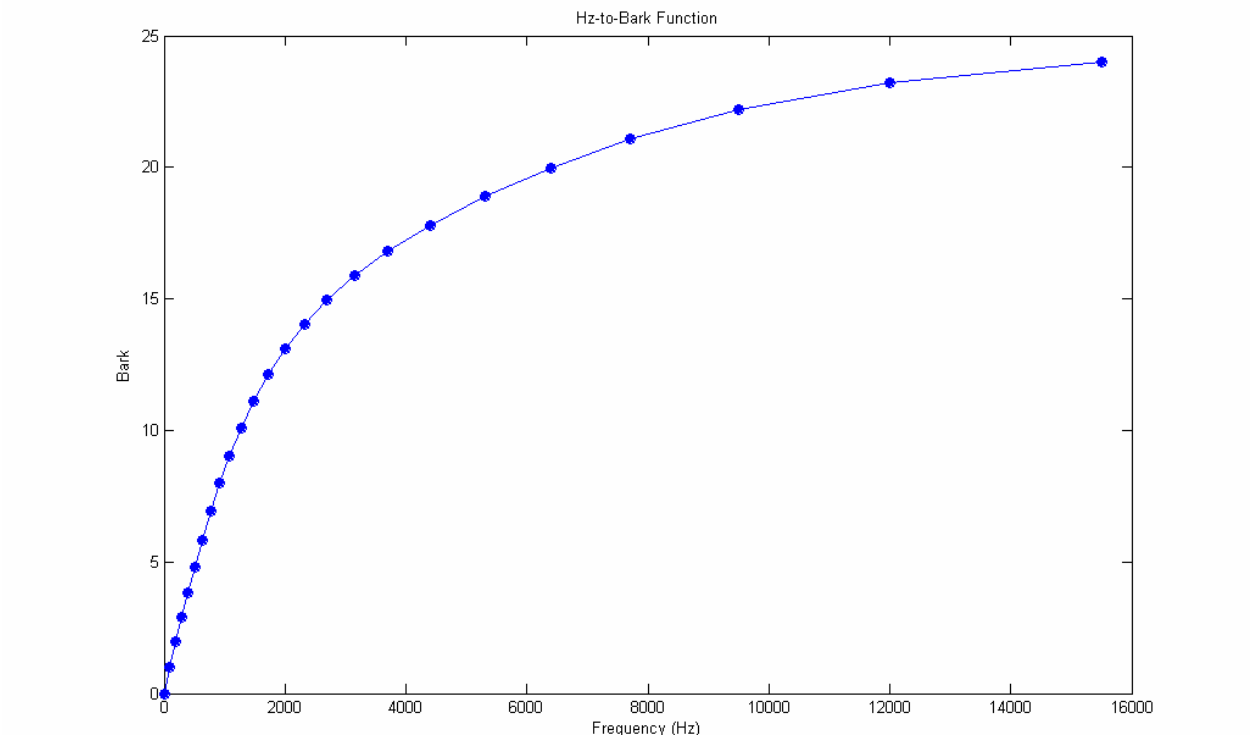

Figure 2 - Hz-to-Bark Warping Function

New spectral representations with variable (frequency dependent) spectral resolution can be produced from implementing FAM transforms. One example of the use of this method of transformation is to produce auditory spectra and spectrograms in which the linear frequency scale is mapped (warped) to a psychoacoustic frequency scale like Bark. The method is summarized as follows [15]:

- First define the warping function from Hz to Bark, i.e., $vj = v(f_k) = v(k)$, where f_k denotes a discrete frequency point and k the corresponding index in the linear frequency scale (Hz).
- Then construct the set of orthonormal FAM functions. This set of functions produces the FAM transform matrix $\mathbf{\Phi}v$, which is then used to map the Fourier spectrum $S(k)$ of the signal $s(n)$ to the frequency warped signal $s(a)$.
- Finally, use the Fourier transform to produce the auditory spectrum (spectrum on the new, warped v-scale).

Linear prediction (LP) yields the optimized coefficients for and FIR type predictor based on the statistics of the signal to be predicted. Assume the filter is of p order. The optimization strategy used is to minimize the average squared prediction error (squared difference between the actual and the predicted value). The predictor has a p-sample-long buffer, and once it is full the predictor will output a predicted value for the next sample. This predicted value is the result a linear combination of p past samples and the weights of the filter coefficients [15]. This process can be implemented in many ways including using autocorrelation, covariance, and lattice formulations.

Probably the most logical application of frequency domain warping is the representation of the auditory system. WLP can both successfully and efficiently be used to implement Bark-

scaled representation and coding of speech and audio signals [15]. Yet another application is to use WLP as a preprocessor to speech recognition algorithms.

The capability of spectral modeling similar to loudness density spectrum (auditory spectrum) estimation, and the fact that information in the (inverse-filtered) WLP residual resembles the overall information in the auditory nerve firing are all interesting properties of the WLP. A WLP front end for residual computation may be followed by a filterbank to separate the signal into critical bands if the application requires such separation [16].

Many speech and audio applications already take full advantage of the auditory properties of WLP processing. The principle has been applied in speech processing, for example, for speech synthesis where the main advantage is to reduce synthesis filter order in the source-filter modeling of speech signals, which helps in generating the control parameters of the synthesis filter. Another application is as a preprocessing method for speech recognition [16]. Moreover, WLP can be used general in speech coding and in the following discussion for multipitch analysis.

4.6 Using WLP for Multipitch Analysis Model

The unitary pitch analysis model proposed by Meddis and O'Mard is one of the best known recent models of time-domain pitch analysis. The unitary model has been tested and has yielded qualitatively good results when compared to human perception in many listening tasks such as missing fundamental, musical chords, etc. Despite the straightforward formulation of the model, a practical problem with implementation is the model's computational expense. This is due to so many filters in the filterbank and each of these requires its own autocorrelation function [17]. Techniques can be employed to take advantage of the concepts presented in these models, while decreasing the computational complexity.

The filterbank operates to emulate the response of the cochlea, and is often considered the principle element in any perception model [18]. The filterbank breaks out a sound signal into many channels that have bandwidths corresponding to the frequency resolution of the cochlea. The input signal can be divided into as many as 40-128 channels, where the number of channels depends on the application [19] [20]. The signal in each channel is half-wave rectified and lowpass filtered to simulate hair activation to movement in only one direction of the BM. Mathematically, this step loosely corresponds to the detection of the envelope of the signal in each channel. The envelope signals are fed into a periodicity measure of some form or other, such as the autocorrelation function (ACF), and is computed within each channel. Finally, the ACFs are summed across the channels to yield a summary autocorrelation function (SACF) that is used in pitch analysis [17]. Some implementations also include an extra transfer function to simulate the function of the middle ear, but this is often neglected.

Several approaches exist to yield a periodicity measure for each of the individual channels. A common choice is the time domain approach. The discrete Fourier transform (DFT) based autocorrelation computation is often used for pitch analysis for computational speed.

Multipitch analysis systems have not been fully deployed in the filterbank form because of the computing time required to accomplish the task. Ideally, such a system would have real-time capability which is simply not possible with a large filterbank. The computational requirements are mostly determined by the number of channels used in the filterbank [17]. Tolonen and Karjalainen proposed a two-channel model based on the Meddis and O'Mard model, but using only two channels it is streamlined for computational efficiency. This computational efficiency comes at a small price of output accuracy, but with other methods of

improving the algorithm the results for the two-channel pitch analysis model are quite good for stationary music files.

4.6.1 Two-Channel Pitch Analysis

A block diagram for the Tolonen and Karjalainen two-channel pitch analysis model is shown below. The first block is a pre-whitening filter that removes the short-time correlation of the signal. A warped linear prediction (WLP) is used as the whitening filter, as previously described. This filter serves two purposes; the first is to perform critical audio band resolution enhancement, and also to reduce the filter order when compared to linear prediction. Their study implemented a WLP filter of 12th order with a sampling rate of 22 kHz, Hamming windowing, frame size of 23.3 ms, and hop size of 10.0 ms. The pre-whitened signal results from inverse filtering with the WLP model. Basically, the whitening filter may be thought of as representing the normalization of the hair cell activity level toward spectral flattening due to the adaptation and saturation effects [17].

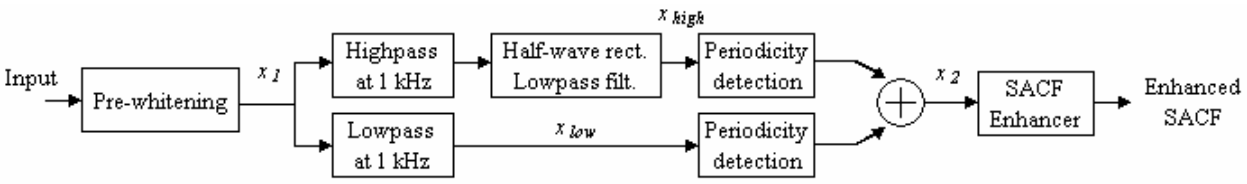

Figure 3 - A block diagram of the Tolonen and Karjalainen Model

The middle part of the block diagram corresponds to the Meddis and O'Mard unitary pitch perception model. The input signal is first divided into two separate channels, with the cutoff threshold set at 1 kHz. Filters that have 12 dB/octave attenuation in the stop-band are used to perform this channel separation. The lowpass signal is then highpassed with a similar 12

db/octave attenuation for frequencies below 70 Hz. The highpass channel undergoes half-wave rectification and highpass filtering at 70 Hz as was done to the lowpass channel [17].

The method of periodicity detection is performed using autocorrelation calculated using the DFT. The signal x_2 in the figure above corresponds to the SACF of Figure 1 and is obtained as:

$$x_2 = \text{IDFT}(|\text{DFT}(x_{low})|^k) + \text{IDFT}(|\text{DFT}(x_{high})|^k)$$
$$= \text{IDFT}(|\text{DFT}(x_{low})|^k + |\text{DFT}(x_{high})|^k) \qquad (2)$$

The parameters x_{low} and x_{high} are the signals in the low and high channel respectively before the periodicity detection blocks in Figure 3. For normal autocorrelation $k = 2$, but it is advantageous to use a value smaller than 2 [17]. The parameter k determines the frequency domain compression [21]. The parameter k allows for nonlinear frequency domain computation such as compression or the application of natural logarithm resulting in the cepstrum. This nonlinear computation is not directly available in straight time domain analysis. To improve computation speed both the fast Fourier transform (FFT) and its inverse (IFFT) are used. The final block in Figure 3 represents the enhancement process that the SACF undergoes to identify only those peaks from true fundamental frequencies [17].

The peaks in the SACF curve provide high probabilities of possible fundamental frequencies. These peaks are often not perfect and contain redundant information. This redundant information could result in poor pitch identification if left uncorrected. The topic of repeating peaks will be further illustrated in the following section through a series of figures.

There are also cases, for instance the case of musical chords, where the root tone often appears very strong although in many cases it should not be considered as the fundamental period of any source sound. To be more selective, a peak pruning technique is used in the model.

The technique defined in the block diagram as the enhancer is the following. The SACF curve is first half-wave rectified to only positive quantities and then expanded in time by two and subtracted from the original clipped SACF. This result is then clipped to have positive values only. This process removes repetitive peaks with twice the time lag. It also eliminates the near-zero time lag part of the SACF curve. This operation should be repeated for time scaling with factors of three, four, etc., as far as desired, in order to remove higher multiples of each peak. The resulting function is called the enhanced summary autocorrelation function (ESACF) [17].

The subsequent chapters will detail the proposed modifications to the two-channel multipitch analysis model, present analysis considerations for implementation, show results, and offer discussions and conclusions.

CHAPTER 5

Multipitch Analysis and Tracking

The preceding chapter provided the basics for multipitch perception and modeling. This chapter will detail the proposed approach to tracking multiple musical pitches that vary through time for the future use of automatic music transcription. The method to accomplish this task builds on the foundation laid by the two-channel algorithm proposed by Tolonen and Karjalainen with a few modifications. The modifications are the necessary to improve results when applying the two-channel multipitch model to tracking musical melodies, which was not tested by Tolonen and Karjalainen. The first of these modifications stems from different analysis goals and expectations between their implementation and that of automatic music transcription. Their method attempts to provide a means of verifying that a two-channel model can be used as a computationally efficient alternative to the more elaborate filterbanks used in the past for multipitch analysis. The proposed method for automatic music transcription is focuses less on how human pitch perception modeling, although the method is rooted and benefits from the principles set forth by pitch perception research. Rather, the focus is on efficiency and more importantly accuracy in tracking multiple pitches through time.

The first modification to the Tolonen and Karjalainen algorithm was to eliminate the pre-whitening stage. Results justifying the removal of this stage will be shown in the results chapter. The pre-whitening stage using WLP is needed in their model to nonlinearly transform the input signal's frequency content to simulate the frequency response of the human ear. For automatic

music transcription such detailed modeling is not required since the result is not how the human ear perceives the tones, but rather what fundamental frequencies are present in the signal.

The second modification was to extend the length of the Hamming window in time to 100 ms. This was done for frequency resolution purposes, however, the lengthening of the window in time has an adverse effect on temporal resolution. Consider the 23.3 ms Hamming window as proposed by Tolonen and Karjalainen, it has a frequency response as shown in the following figure.

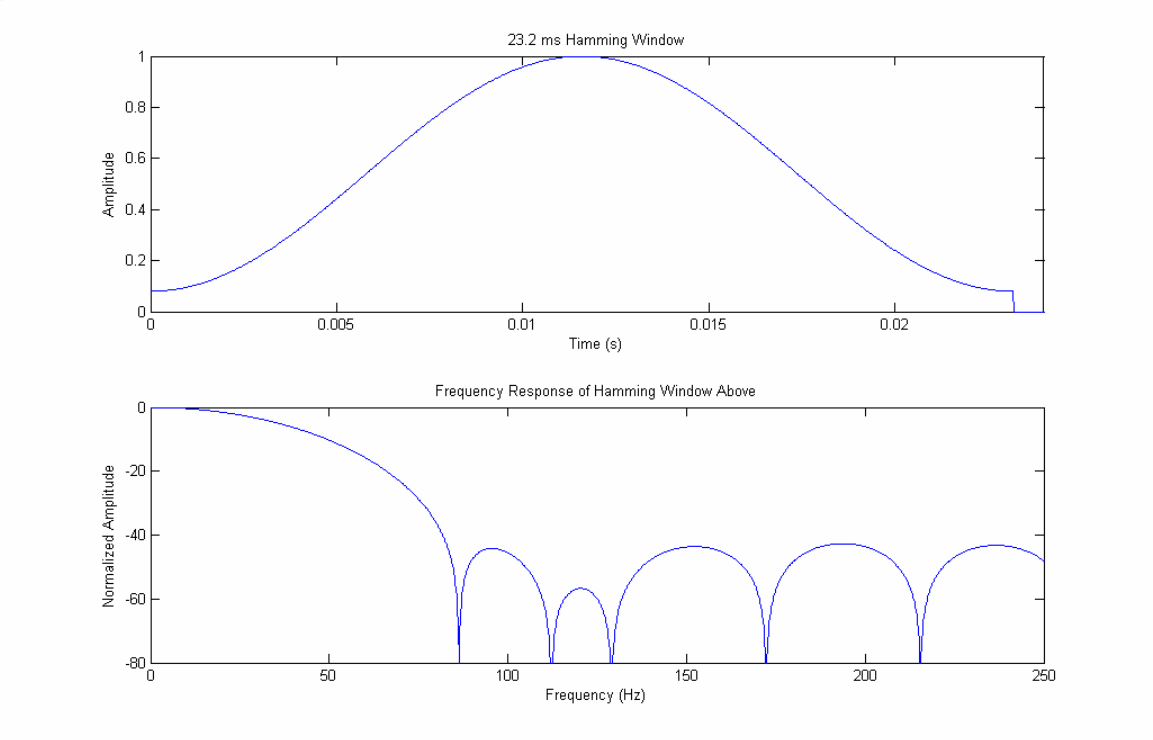

Figure 4 - 23.2 ms Hamming Window

The wide main lobe of the frequency response will act to blur the frequencies present in the windowed input signal. This is because of the property where multiplication (windowing) in the time domain is equivalent to convolution in the frequency domain. The convolution process will spread out the frequency content present in the original input. By enlarging the window in time to 100 ms, the main lobe shrinks in frequency as shown in the following figure. Since the

main lobe is not as wide in frequency the blurring of the frequency content is less, but it can never be ideal. The only way to eliminate the blurring would be to consider the signal as a whole, which in effect is equivalent to multiplying the input by an infinitely long time window. The obvious problem with that method is the temporal resolution decreases as the window increases in length, so by considering the input as a whole, no pitch tracking can take place.

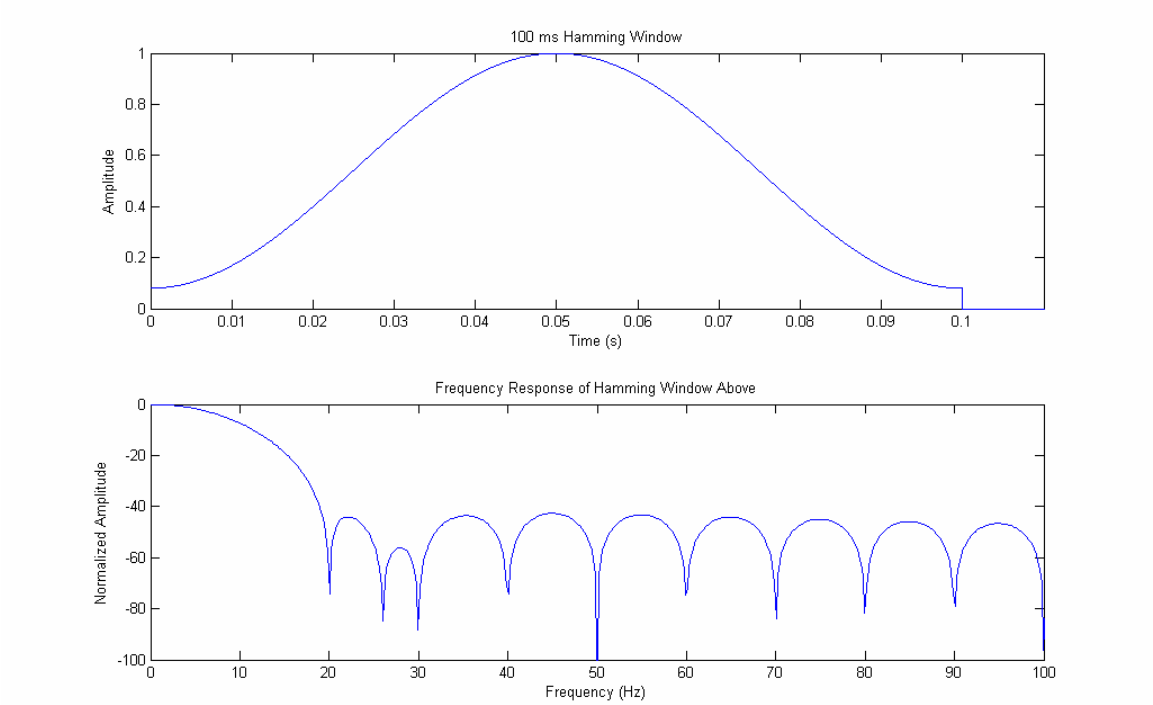

Figure 5 - 100 ms Hamming Window

The lengthening of the Hamming window did improve the fundamental frequency identification results, while its extra length did not adversely affect the temporal resolution. Again, the results to justify these claims can be found in the results chapter.

The following figures demonstrate the peak pruning method used to create the ESACF from the SACF as defined in section 4.5.1. The method will be detailed through the figures. The first figure shows a test signal generated to contain a fundamental frequency at 200 Hz and the

first two harmonics with relative amplitudes of 1, 0.37, and 0.25 respectively. The test signal and the SACF are shown. Take note that the SACF contains many repetitions of the main desired peak. This shows the need for the peak pruning algorithm to create the ESACF by which only the main lag peak remains.

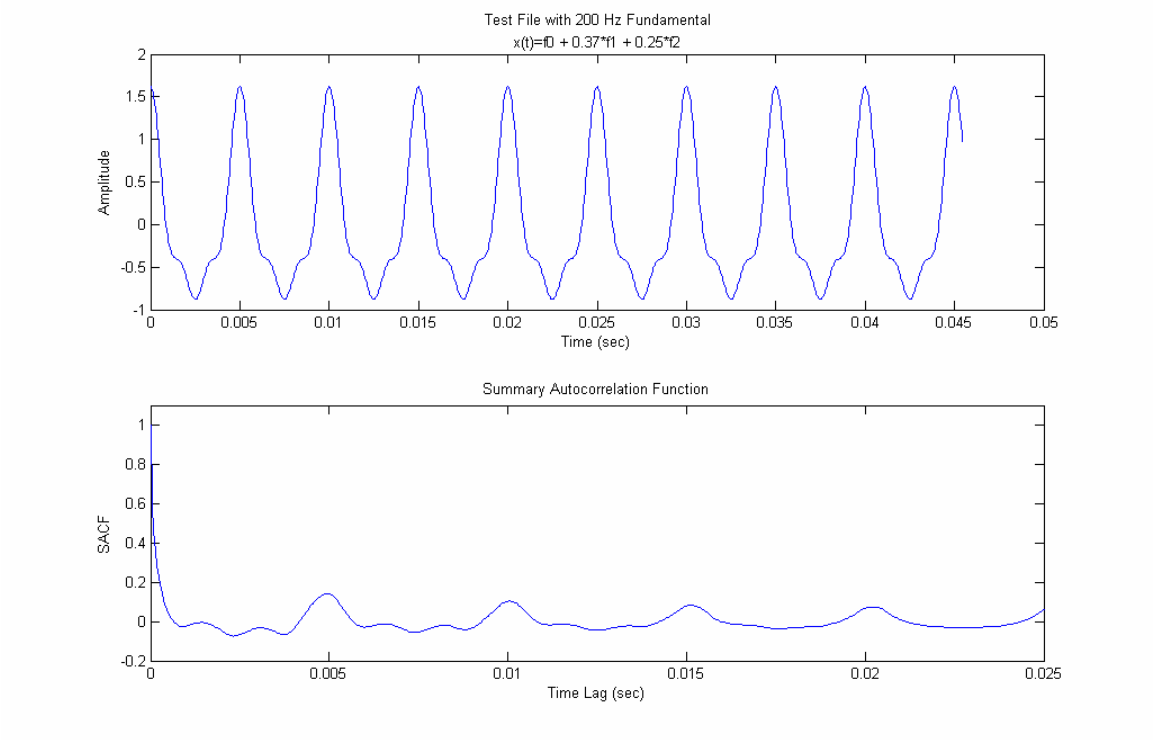

Figure 6 - Test Signal and SACF

The next figure shows the clipped SACF and the first iteration of expanding the clipped SACF. The figure shows that when the two are subtracted that the repetitive peaks begin to be cancelled out, and the peak of interest is reinforced. The result of the first iteration of peak pruning is also shown. This signal will then go through subsequent iterations of pruning until the fundamental peak alone remains.

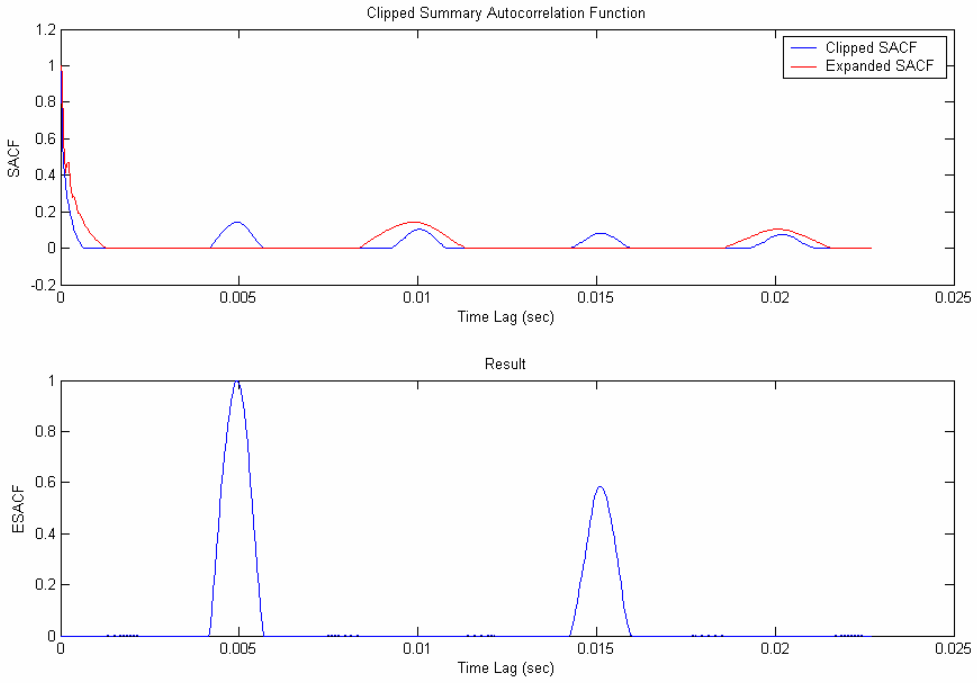

Figure 7 - First Iteration of Peak Pruning

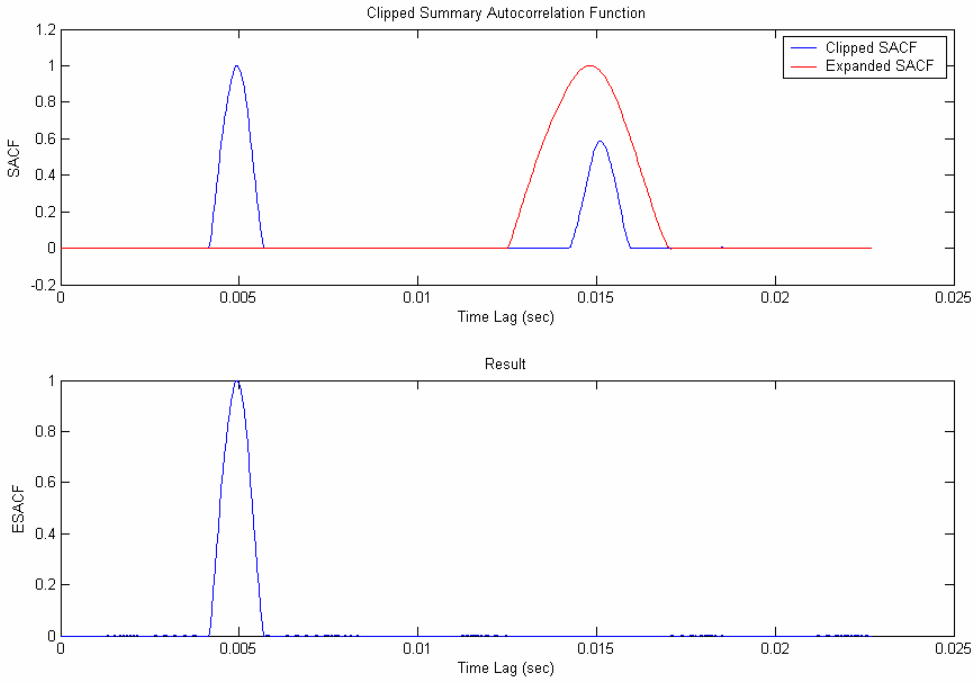

Figure 8 - Second Iteration of Peak Pruning

The figure above shows how the method can be repeated as many times as desired to yield the one peak of interest. In the previous example the main peak of interest has the greatest amplitude when compared to the repeated peaks. This is not always the case and will lead to the final modification resulting in the proposed algorithm.

The final modification to the Tolonen and Karjalainen algorithm was the need to further strengthen the peak pruning algorithm. The proposed peak pruning algorithm follows the groundwork laid by Tolonen and Karjalainen, however, rather than performing a straight subtraction between the reference autocorrelation function and the lengthened autocorrelation function, the latter is multiplied by a weight to reinforce the peak pruning process. The reinforcement further suppresses repeating peaks found in the SACF that can at times be higher stronger than the actual peak of interest.

The table below summarizes the model parameters used in the algorithm as were discussed earlier.

k	0.67
Hop Size	10 ms
Hamming Window Length	100 ms
ESACF Weighting	1.5

Table 1 - Model Parameters

The following table provides a reference to connect a few music notes with the actual fundamental frequency of that note. Take notice the musical convention of each repeating note (C to the next highest C) is one octave, and therefore a doubling of frequency. Also, C_1 represents the musical middle-C.

Musical Notes	Frequency (Hz)
C	524
B	496
A	440
G	392
F#	370
F	350
E	330
D	294
C_1	262
B	248
A	220
G	196
F#	185
F	175
E	165
D	147
C	131

Table 2 - Musical Notes and Frequency

The following figure is a screen shot from program and algorithm development stages to show the information provided as the program analyzes the input sequence. In the upper, left plot the entire time sequence is shown, while just beneath that is the current windowed version of that signal. Below the windowed time sequence is the short-time Fourier analysis of the windowed time sequence. Take special note of the harmonic complexities present in this signal. In fact, the relative amplitude of the fundamental frequencies are shown as less than that of the first few harmonics. In the upper, right the summary autocorrelation function is shown. This output is then enhanced to yield the plot just beneath it as the enhanced summary autocorrelation function. Also, part of the screen for the developing process was a pop-up showing the identified frequencies for the windowed input sequence as defined as the inverse of the peaks in the ESACF.

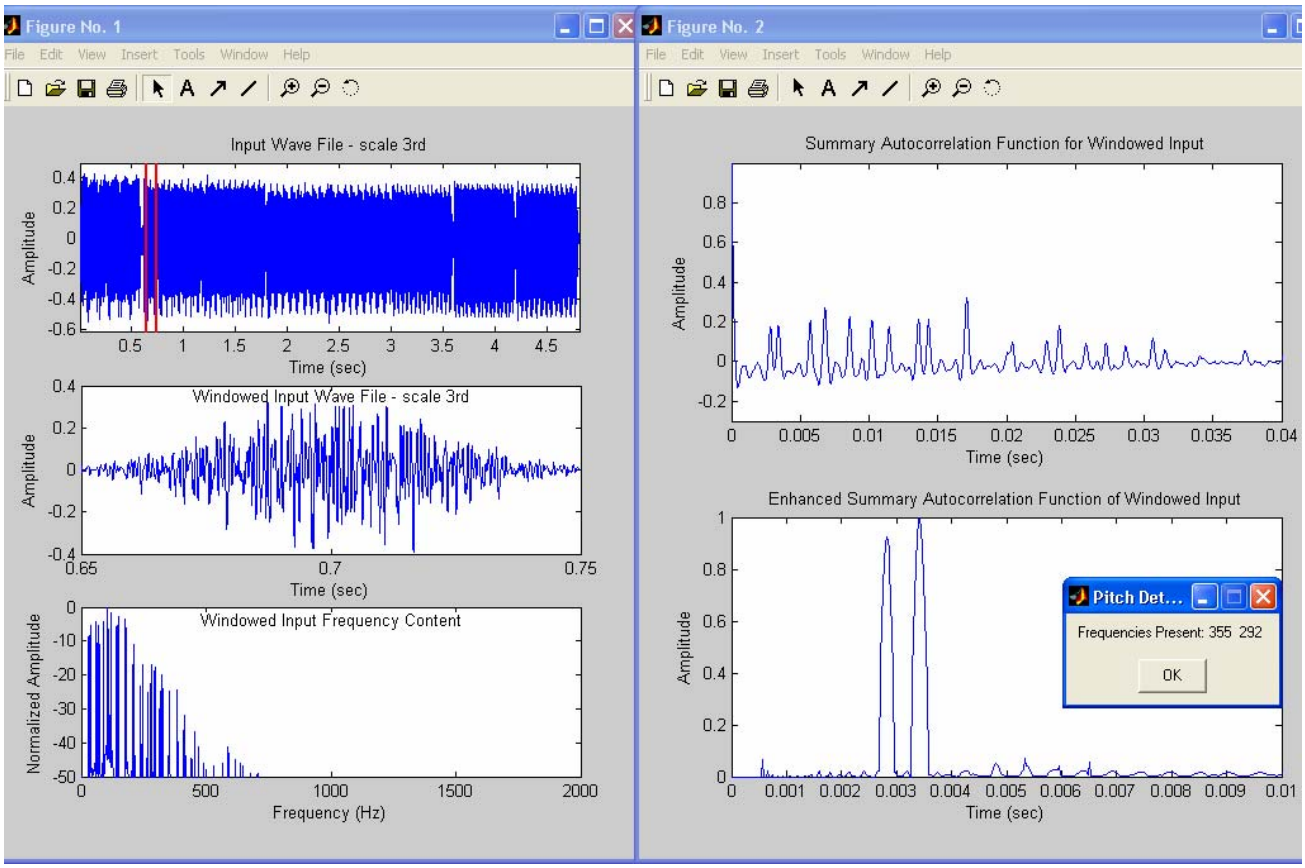

Figure 9 - MATLAB Developing Window Screen Shot

The next chapter will present the results of the proposed algorithm.

CHAPTER 6

Results

The objective of the first stage of the project is to establish comparable results to Tolonen and Karjalainen with the modifications proposed in this paper to verify that the modifications would produce valid and satisfactory results. The first test file used to verify these modifications is the 'musical chord' experiment. More specifically, this test shows the method's insensitivity to the removing the pre-filtering WLP. The results of this experiment are shown in the following figure.

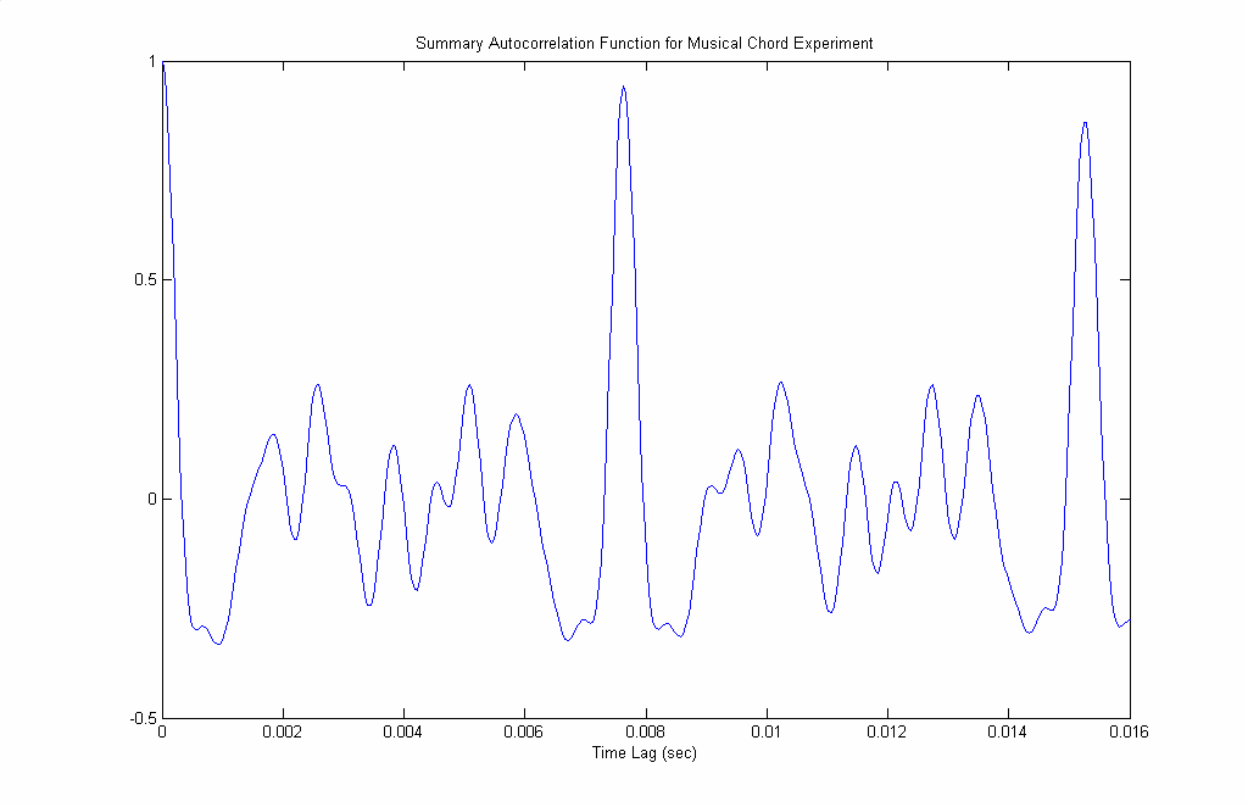

Figure 10 - Summary Autocorrelation for Musical Chord

Just as Tolonen and Karjalainen report, the result of a chord consisting of three fundamental frequencies 392, 523.2, and 659.2 Hz corresponding to tones G, C, and E respectively has a strong contribution at a lag time of 7.7 ms. This corresponds to a frequency of 130 Hz (C below middle-C), which is the root tone of the chord.

The next validation of the proposed modifications to the Tolonen and Karjalainen algorithm was to test the multipitch analysis and peak pruning algorithms with a test signal used in one of their experiments. The test signal consisted of three tones from a clarinet with fundamental frequencies 147, 185, and 220 Hz, which correspond to a D, F#, and A notes to comprise a full D chord. The result from the proposed algorithm from this input signal is shown in the following figure.

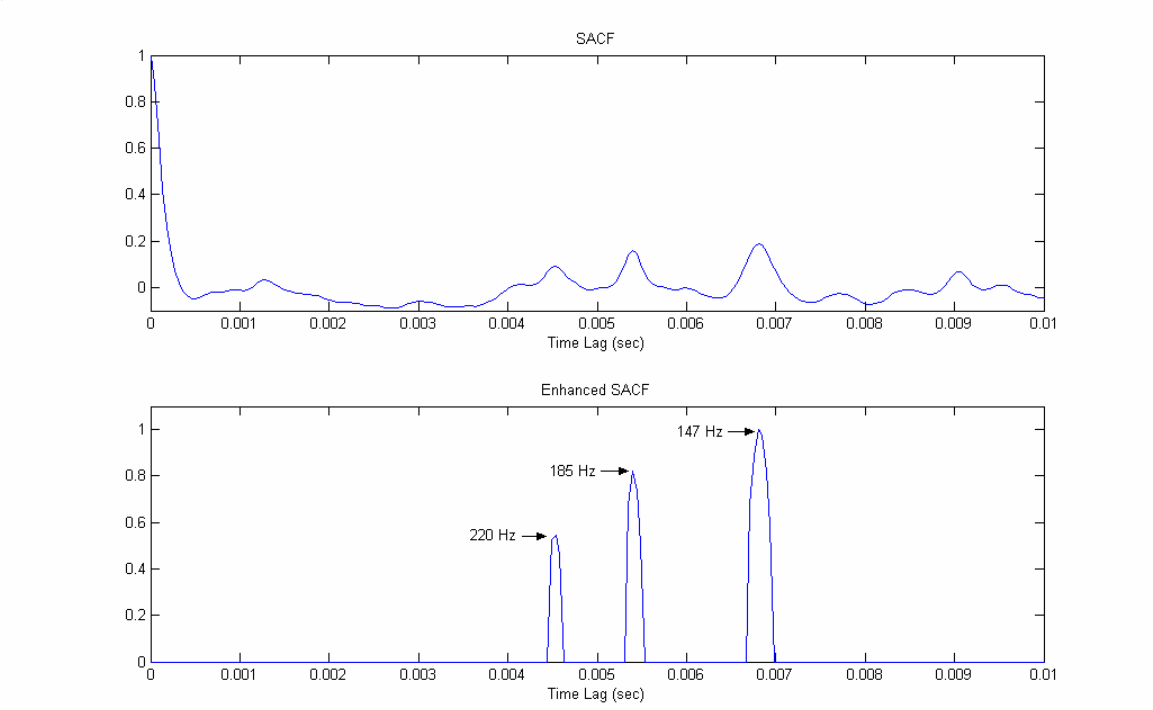

Figure 11 - 'D' Chord Experiment Results

The algorithm successfully detected the frequencies present in the summary autocorrelation function and successfully isolated those fundamental frequencies using the proposed modified peak pruning technique.

The next task was to build test files to test the pitch tracking through time. The test files had to be generated since no time-varying musical files were used in the Tolonen and Karjalainen paper. Test files were created using the following musical patterns.

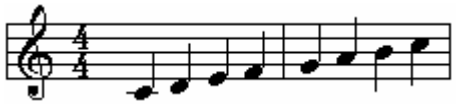

Figure 12 - Single Octave Musical Scale

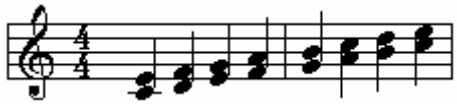

Figure 13 - Single Octave Musical Thirds

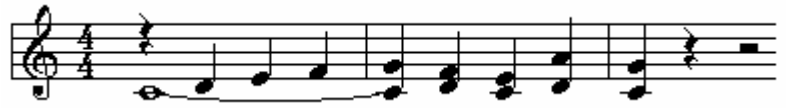

Figure 14 - Musical Example

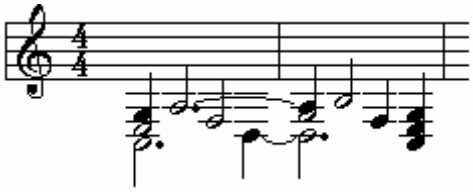

Figure 15 - Musical Example 2

The patterns above were used to create test files of varying tempos and octaves. The following table summarizes all of the files generated and tested.

File Name (*.wav)	Beats Per Minute (bpm)	Starting 'C' Frequency (Hz)	Pattern (from above)
scale_down	100	131	scale
scale_down_3rd	100	131	thirds
scale_down_file1	100	131	example
scale_down_150	150	131	scale
scale_down_3rd_150	150	131	thirds
scale_down_file1_150	150	131	example
fast_scale	16th notes at 150	131	scale
fast_scale_3rd	16th notes at 150	131	thirds
fast_scale_file1	16th notes at 150	131	example
scale	100	262	scale
scale_3rd	100	262	thirds
scale_file1	100	262	example
scale_150	150	262	scale
scale_3rd_150	150	262	thirds
scale _file1_150	150	262	example
file2	125	131	example 2

Table 3 - Test Files Generated

These files were then used to test the algorithm. The files that contain only a single-note scale test the time tracking portion, while the files containing patterns of thirds and the musical example tested both the time tracking and the multipitch analysis. The first musical example is fairly complex in that it contains notes that are held for a longer duration than others and it certainly does not follow a structured pattern like the scale of thirds does. The second musical example tests the algorithm's ability to detect more than two simultaneous notes. To test the temporal resolution, the file containing 16th notes at 150 bpm was created. This file tests the algorithm's ability to recognize and detect notes that occur in an extremely short time period. This is also meant to test the proposed modification to expand the Hamming window size to 100 ms.

The following figure displays the results of the proposed algorithm to test file 'scale_down.wav'. This is computationally the easiest of the tests because it does not require

multipitch analysis and the notes are changing at a very slow rate (100 bpm). The results show very steady pitch identification and sharp transition from note to note. The highest note (C_1) does vary slightly in its identification, but since the variation is on the order of 2-3 Hz and the separation between the next adjacent note is approximately 14 Hz, the variation is acceptable.

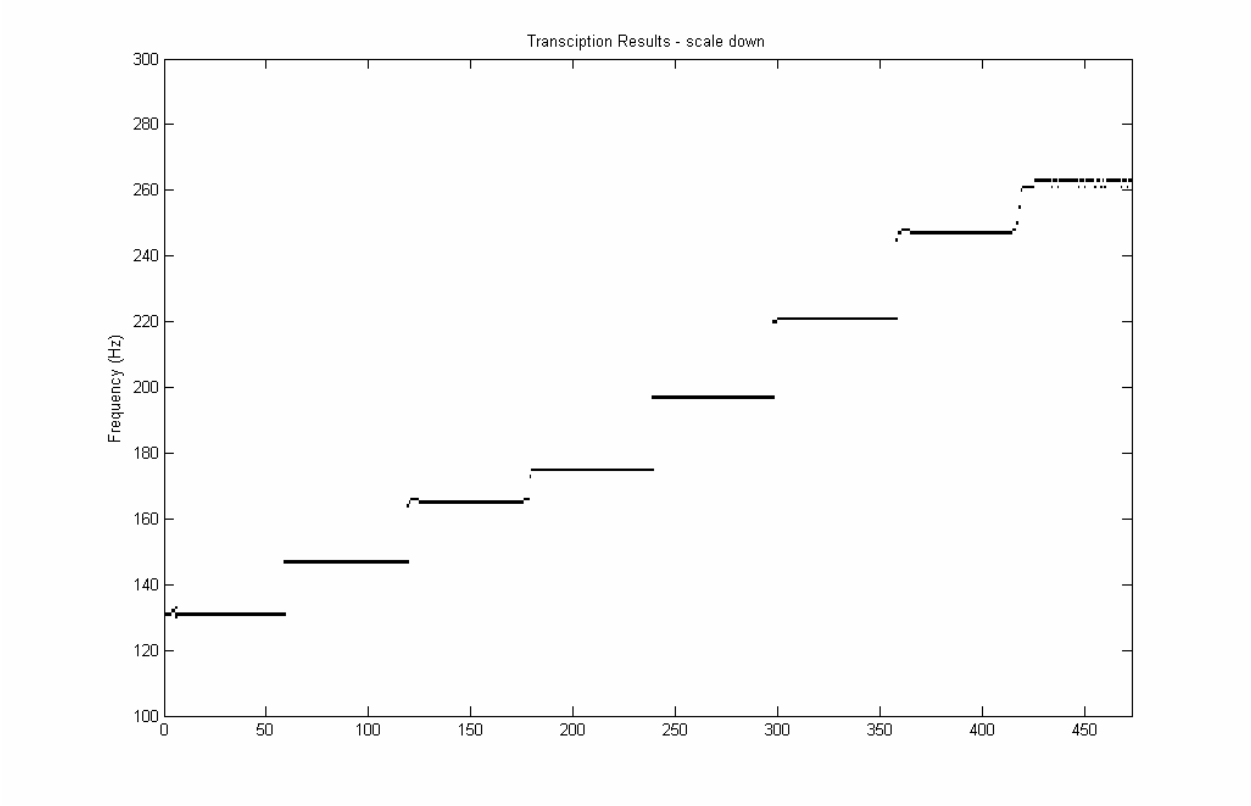

Figure 16 - Results for scale_down

The next figure begins to test both the pitch tracking and multipitch characteristics of the algorithm simultaneously. This test signal, 'scale_down_3rd.wav', is rather simple compared to the subsequent tests because the notes vary at only 100 bpm. The results do show slightly more frequency variation in the pitch identification, however, they are still only on the order of 2-3 Hz and deemed acceptable. The transitions from note to note are crisp, with only a few transitional ghosts present.

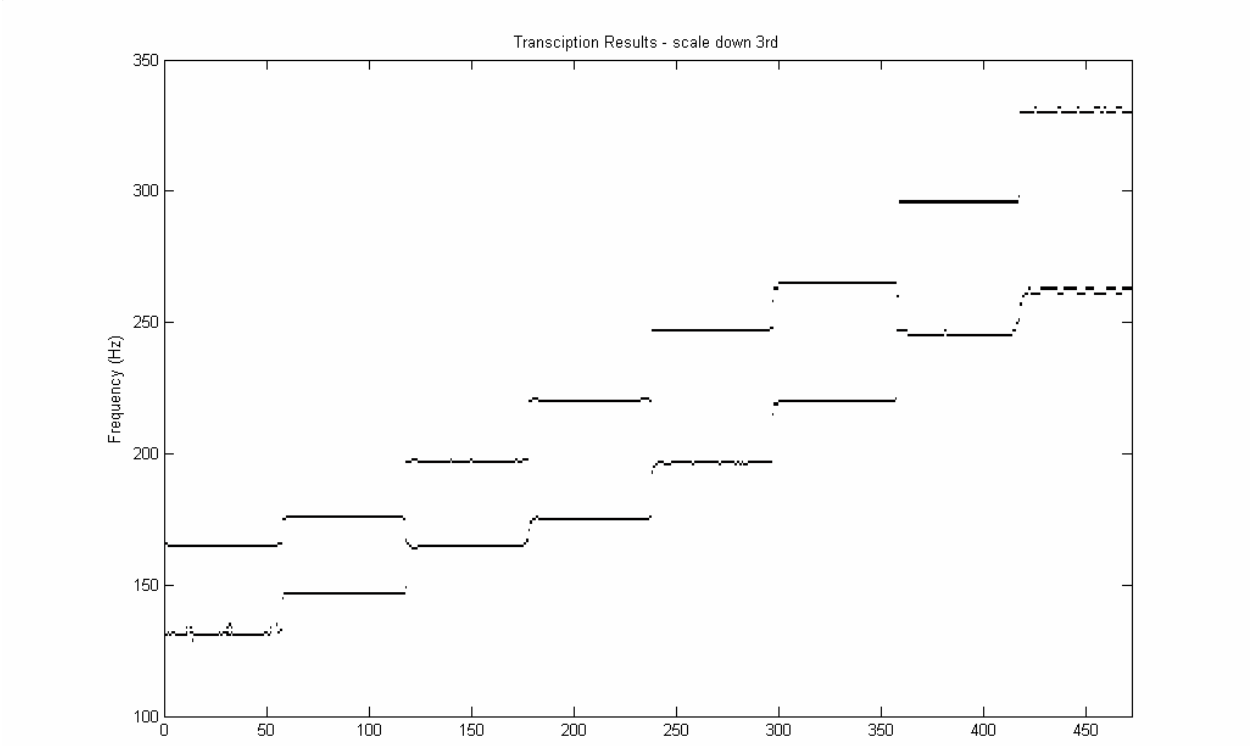

Figure 17 - Results for scale_down_3rd

The next figure shows the results for test file ‘scale_down_file1.wav’. These are the first results considered from the musical example pattern as defined earlier. This pattern truly tests the ability of the algorithm to perform multipitch analysis in that the first note is held over while others are varying, and the typical musical intervals are present as well (thirds and fifths). Overall, the results are excellent. There are a few instances of the identified frequency varying slightly, but still limited to 4-6 Hz. This remains within the tolerance since two adjacent notes in this octave range are separated by at least 16 Hz.

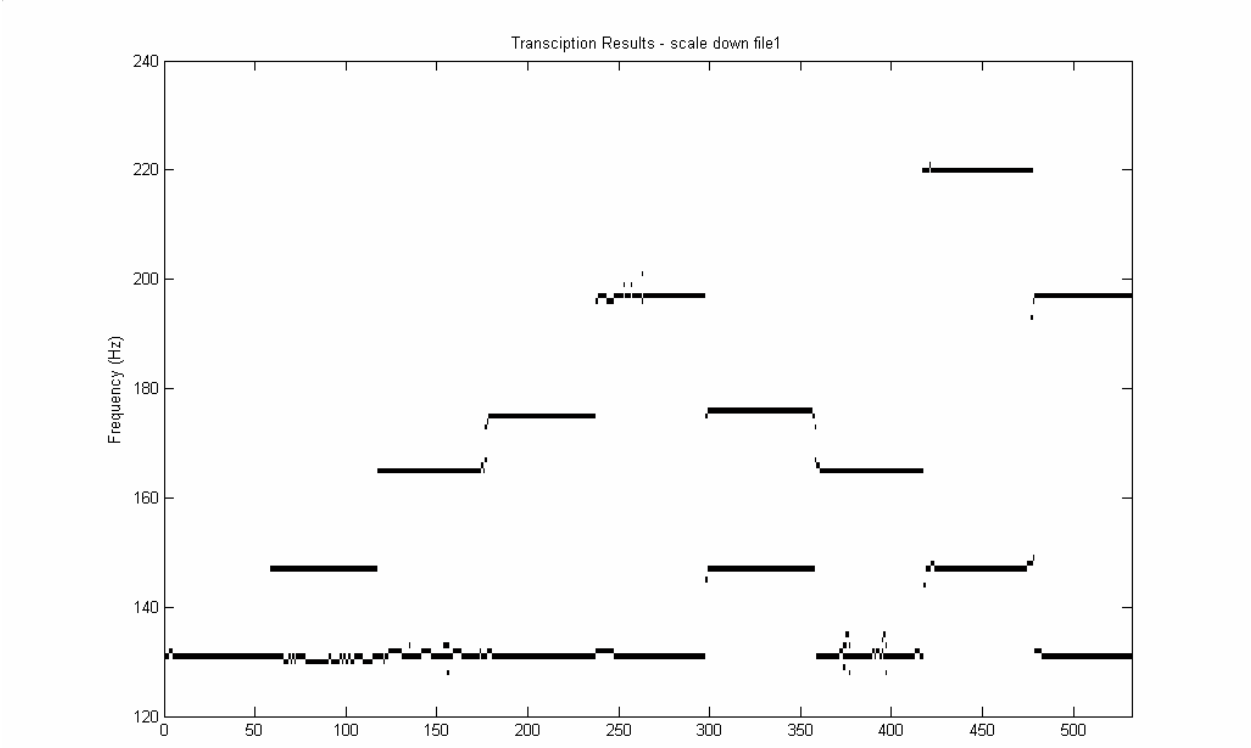

Figure 18 - Results for scale_down_file1

The previous figures demonstrated good performance at a fairly slow rate of change in the notes. The next set of figures show the results for the same patterns, however, at an increased rate of change to 150 bpm. The first test file is 'scale_down_150.wav' and the results are shown in the following figure. The results are nearly identical to the previous single-note scale test file, except that the number of iterations required to step through the entire input sequence is reduced by a factor of 1.5 since the file contains the same number of notes, but they are changing faster (lasting for a shorter time).

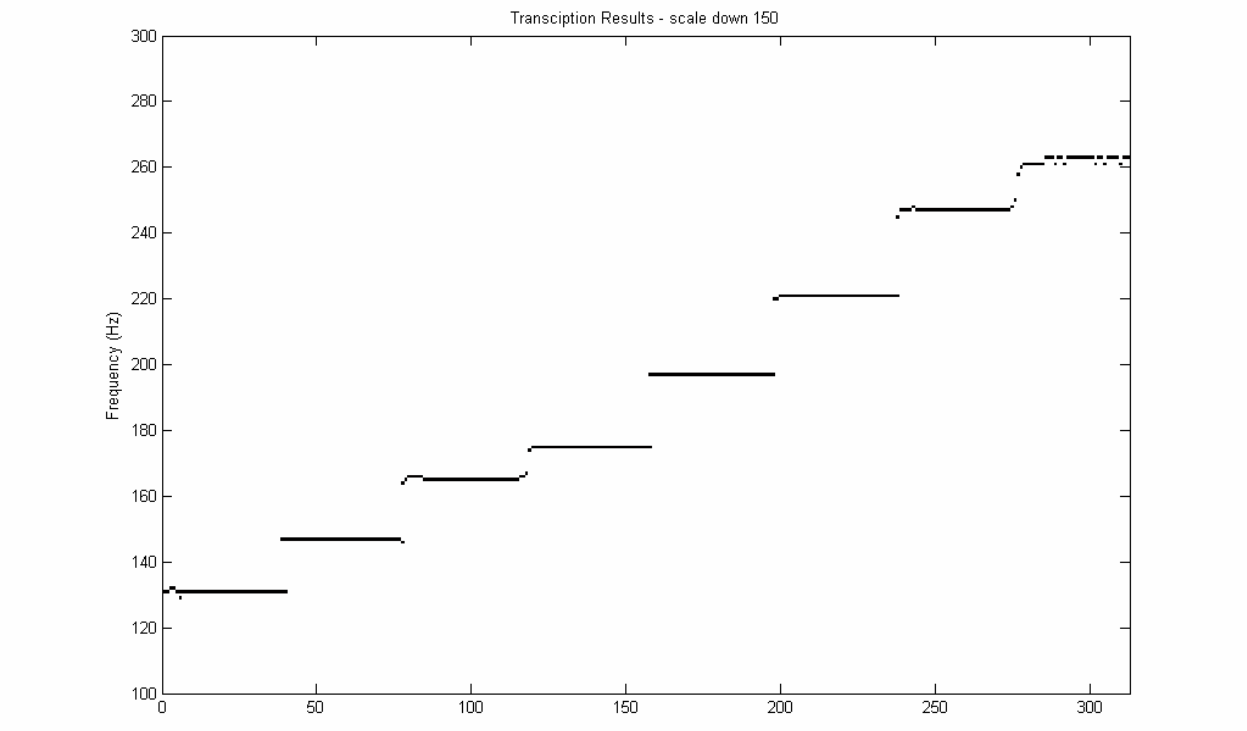

Figure 19 - Results for scale_down_150

The figure above closely matches the results obtained from the same pattern at 100 bpm, which indicates that the algorithm is insensitive to slight changes in tempo. The following two figures will demonstrate this same quality by using the other two patterns generated at the faster speed of 150 bpm. The following figure is associated with test file 'scale_down_3rd_150.wav'.

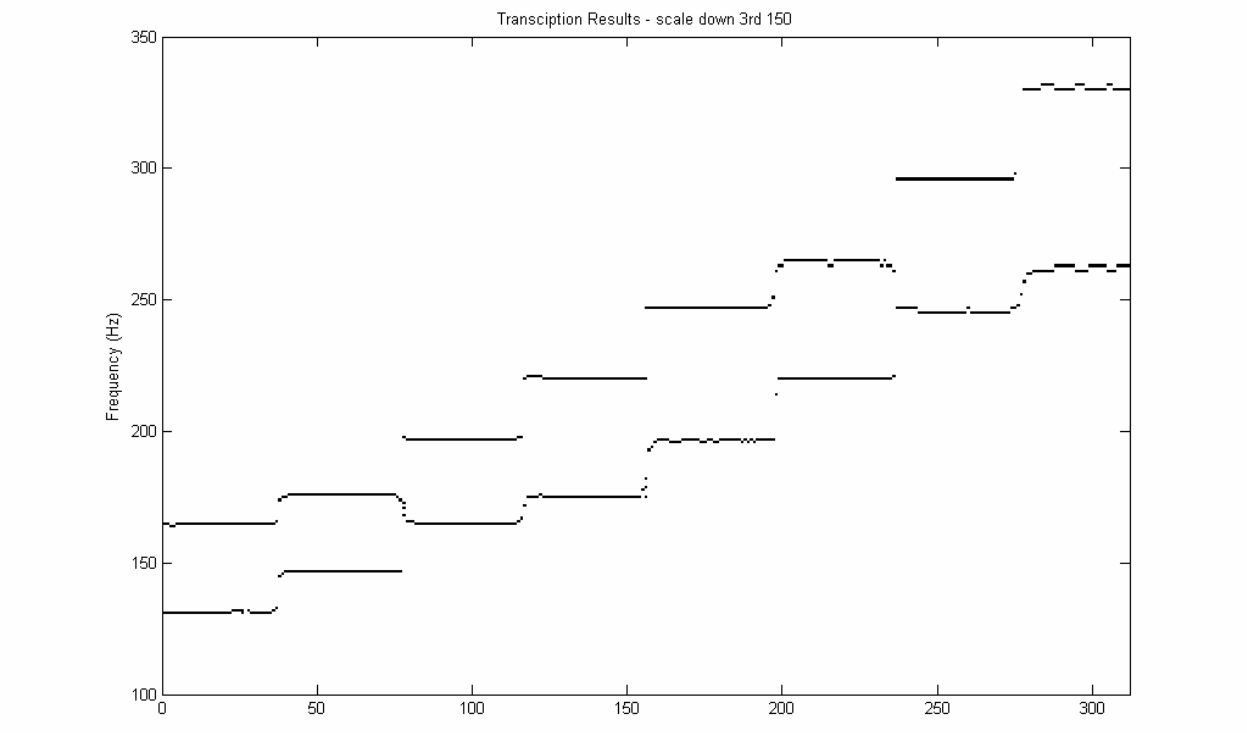

Figure 20 - Results for scale_down_3rd_150

Again, the results for the thirds at 150 bpm closely match the results from the slower file of the same pattern. The following figure demonstrates similarly good results for the musical example pattern generated at 150 bpm using the file 'scale_down_file1_150.wav'.

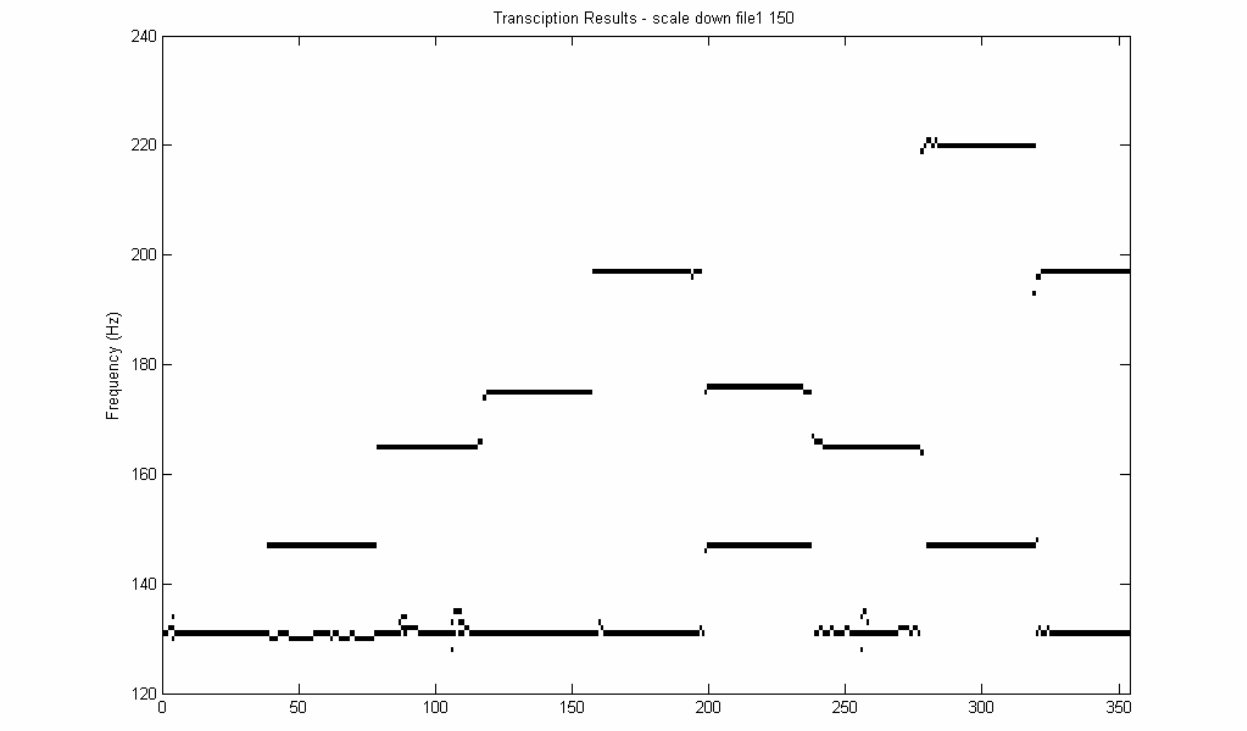

Figure 21 - Results for scale_down_file1_150

The remainder of the testing of the multipitch tracking algorithm was aimed to push the limits in regards to the temporal resolution. The following results originate from files at the same tempo (150 bpm) but instead of quarter notes (one note per beat) sixteenth notes were used (4 notes per beat). The following figure is the result of analysis of the file 'fast_scale.wav'. The shortened duration of the first note is not from an error in the proposed algorithm, but rather the MIDI sequencer did not hold the note for the same time duration when the file was created and can be seen if that input file is plotted. Despite the high velocity of the notes, the algorithm is able to identify all of the proper fundamental frequencies. At this speed, however, the algorithm does begin to show the window overlapping two consecutive notes and identifying them as being simultaneous even though they are not. Also take note that while previous scale patterns took approximately 320-460 iterations, this file only required approximately 70 iterations.

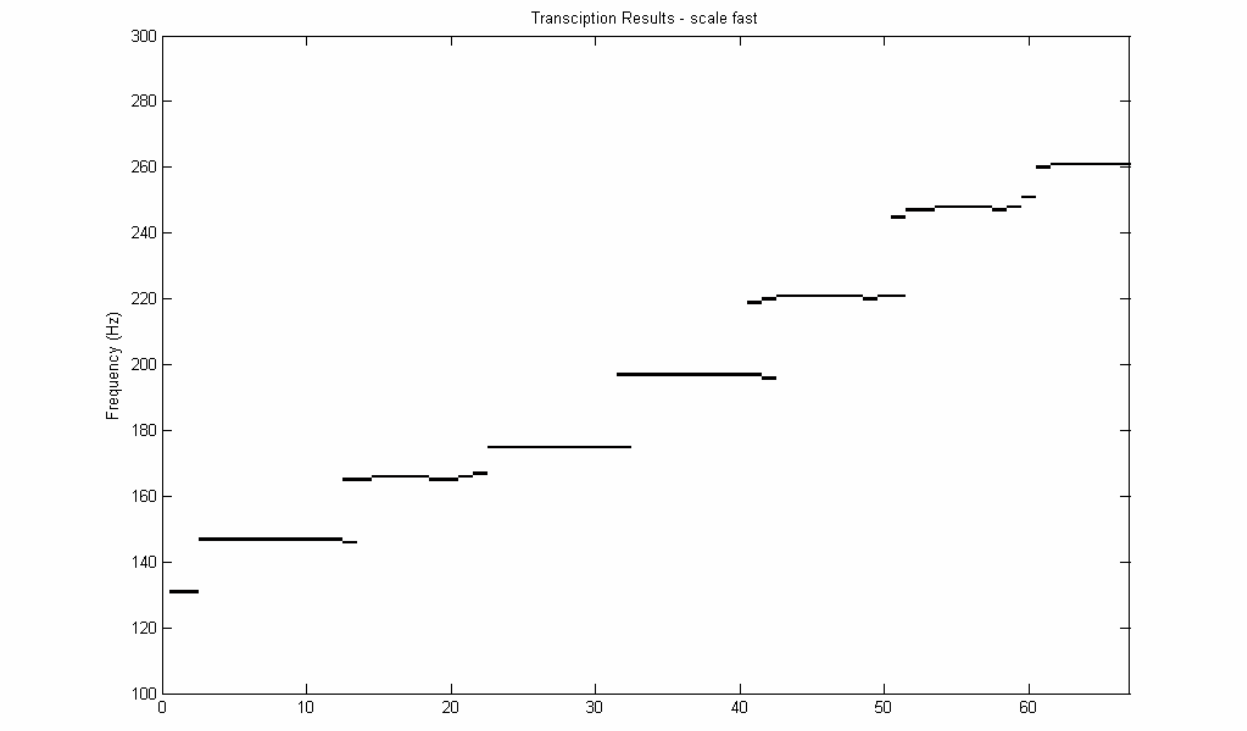

Figure 22 - Results for scale_fast

The following figure shows the results for the file ‘fast_scale_3rd.wav’. Again, the algorithm is able to identify all of the correct fundamental frequencies even at this high velocity. In the transition regions there begins to display a smoothing effect from one note to the next, but it quickly resolves itself and settles to the correct frequencies.

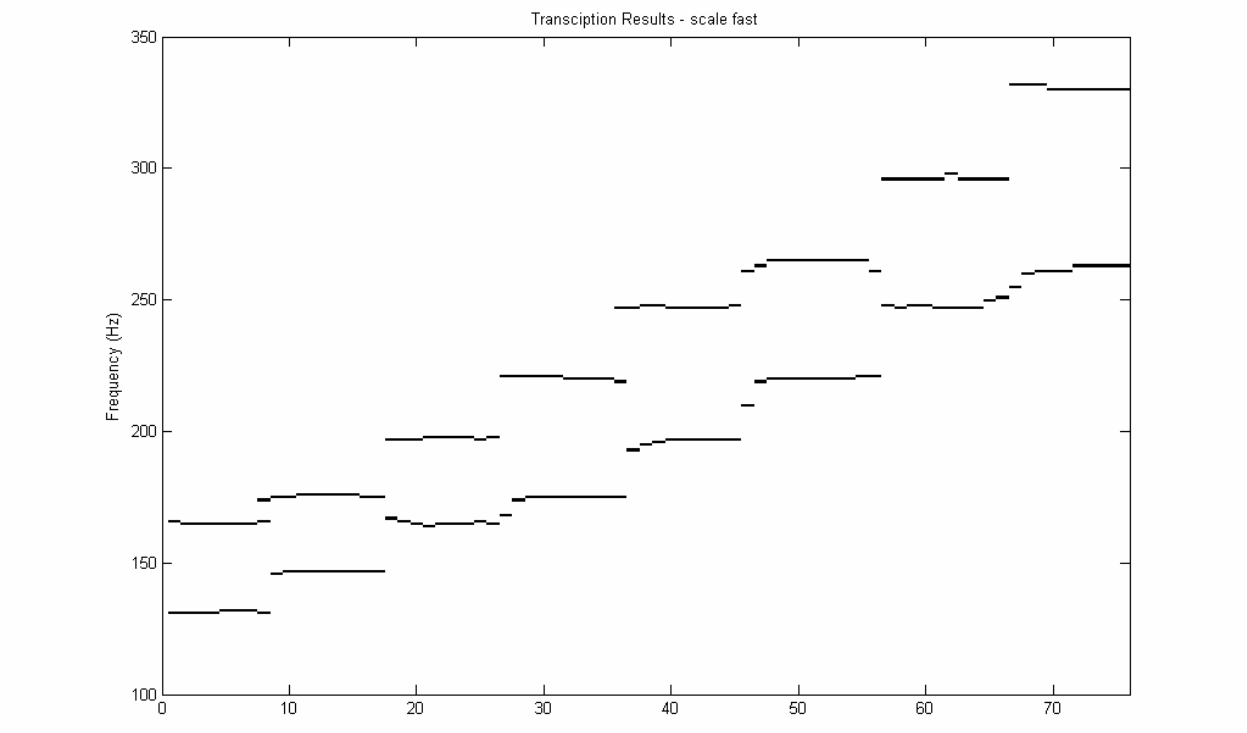

Figure 23 - Results scale_fast_3rd

The figure below shows the results of the file ‘fast_scale_file1.wav’. The results presented here truly display both the temporal resolution and multipitch analysis capabilities of the proposed algorithm. The transitions between notes are relatively sharp and very little smoothing has occurred between notes.

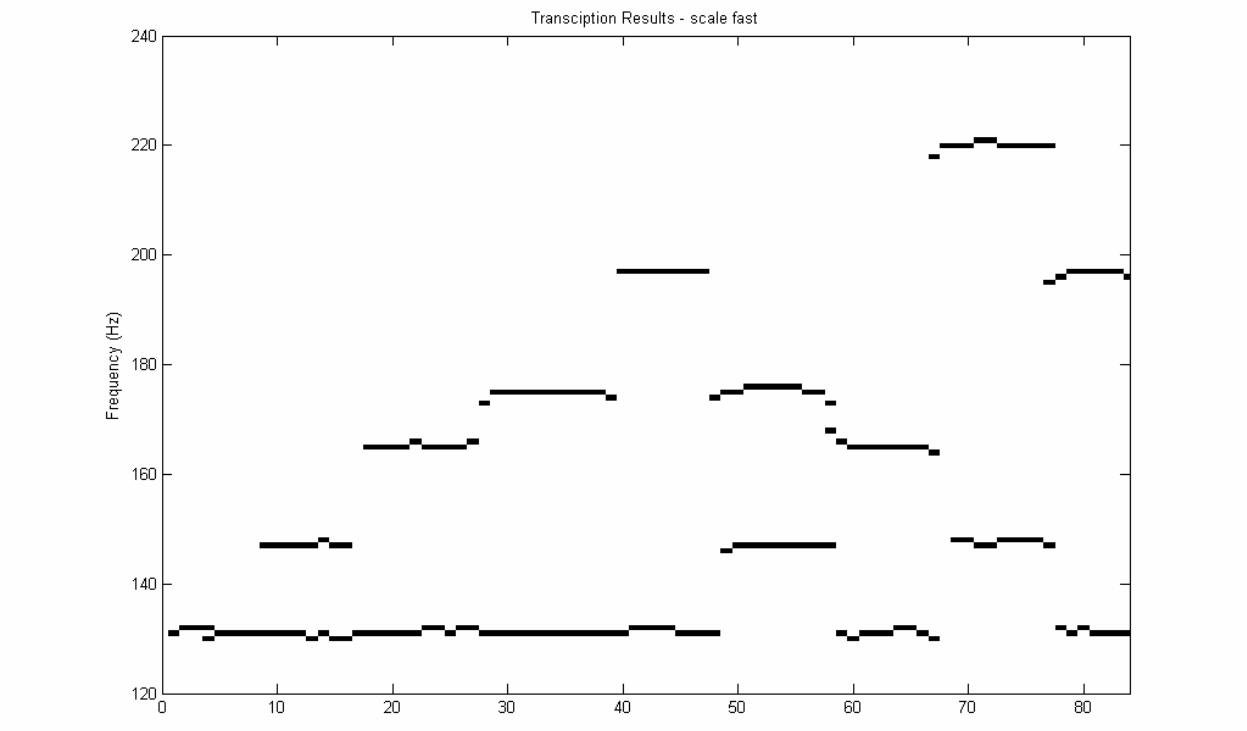

Figure 24 - Results for scale_fast_file1

The patterns previously tested were again tested at 100 bpm and 150 bpm (quarter notes only) except the entire scales were shifted up one octave. The results for all of these tests will not be presented, but the following will show the results for the thirds pattern at 150 bpm from the file 'scale_file1_150.wav'. This file is representative of the other results from this octave range.

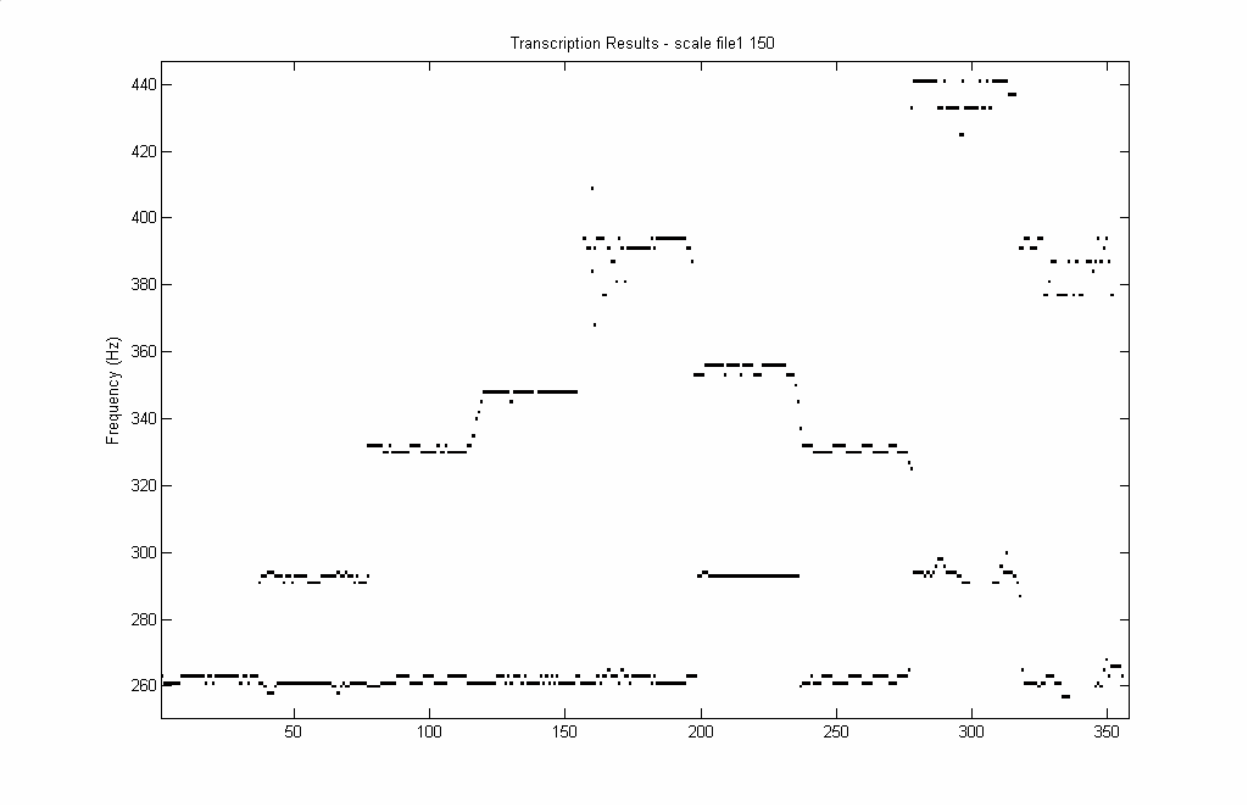

Figure 25 – Results for scale_file1_150

It is obvious that the algorithm does not perform as well in this octave range. The deviations in the identified frequencies vary considerably more than in the lower octave. Also, the transition regions are not as well defined and sharp. This could be due to the rather arbitrary separation between low and high channels of the summary autocorrelation function set at 1 kHz. Perhaps if this filter cutoff was set higher, this octave range would display similarly good results as the lower register. The final test will determine if the algorithm is able to detect three simultaneous notes, and the results are shown in the following figure.

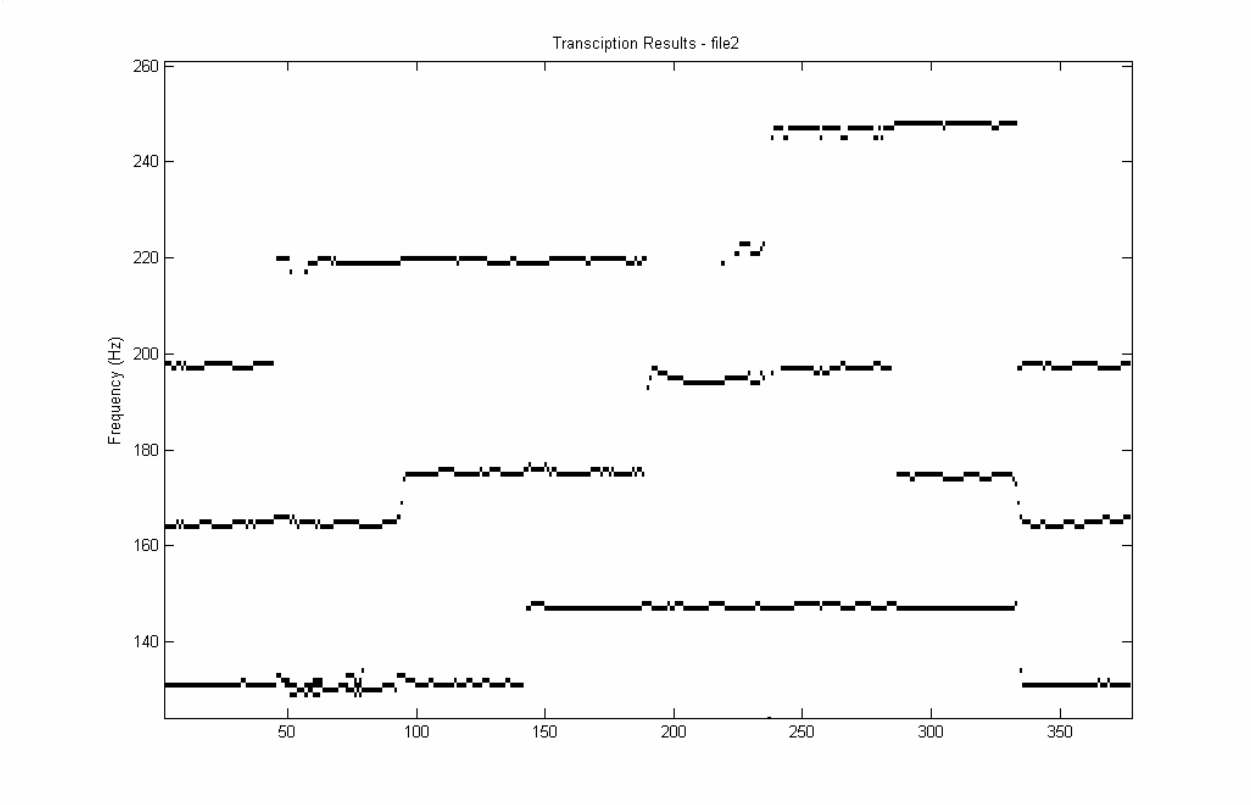

Figure 26 - Results for file2

The algorithm is able to identify all three simultaneous pitches in the music sequence. There is very little deviation in the frequency output and the note durations are representative of the expected results. The expected results are shown in the following figure for comparison.

C								
B								
A								
G								
F								
E								
D								
C								

Figure 27 - Expected Results for file2

The next and final chapter brings the thesis to a close, makes suggestions as to future research in this area, and lists the current limitations on the proposed algorithm.

CHAPTER 7

Discussion and Conclusions

The proposed algorithm modifications to the autocorrelation-based multipitch analysis were verified using test samples gathered by Tolonen and Karjalainen and compared to their results. Upon satisfactory completion of this verification, the algorithm was tested using input signals that contained varying patterns of musical notes with two notes occurring simultaneously. The major advantages of this algorithm are its relative ease of implementation, computational efficiency, and good results for those signals tested. The true robustness of the algorithm cannot fully be determined by those few tests performed here.

There are many topics for future research using and expanding this algorithm. Some of these could include the following:

- Identification of the instrument being analyzed by looking at the harmonic structure.
- Simultaneous multi-octave pitch analysis. The current algorithm which employs the rather simple peak pruning method will not return satisfactory results if applied to a signal that contains multi-octave information.
- Synthesis of a single note as it is being played. Once the fundamental pitch has been identified, it can be isolated as well as its harmonics to recreate the time signal for that one note during that time interval. This could be useful for separating different sound sources.

The proposed algorithm does have one major shortcoming in that it will not analyze multi-octave signals. As mentioned before, the peak pruning method will eliminate all repetitive

peaks and will be unaware as to whether the repeated peak in autocorrelation corresponds to a true repeated peak or one generated by a legitimate new fundamental frequency. Another shortcoming is that the algorithm was not tested using different instruments. Since the algorithm is based on autocorrelation and the shortest time lag should associate with the true fundamental, the algorithm should be immune to changes in the harmonic structure but this hypothesis is left untested.

In conclusion, the algorithm did display excellent multipitch results for those files tested. Also, the temporal resolution was excellent by detecting the sixteenth notes at 150 bpm. This algorithm does provide an excellent starting point for developing a fully automated music transcription package.

REFERENCES

[1] Luís Gustavo P.M. Martins and Aníbal J.S. Ferreira, "PCM to MIDI Transposition", *Presented at the 112th convention of the Audio Engineering Society,* Munich, Germany, May, 2002.

[2] James A. Moorer, "On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer", *Ph.D. thesis*, Department of Music, Stanford University, Stanford, CA, May 1975.

[3] M. Piszczalski and B.A. Galler, "Automatic Music Transcription", *Computer Music Journal*, vol. 1, sec. 4, pp. 24-31.

[4] Michael Hawley, "Structure out of Sound", *Ph.D. thesis*, MIT Media Laboratory, 1993.

[5] Brad Hansen, "Musical Instrument Digital Interface (MIDI)", pp. 604-615, 2003.

[6] F.V. Hunt, "Origins in Acoustics", *Journal of the Acoustical Society of America*, Woodbury, New York, 1992.

[7] Alain de Cheveigné, "Pitch Perception Models – A Historical Review", *CNRS – Ircam*, Paris, France.

[8] J.L. Goldstein, "An Optimum Processor Theory for the Central Formation of the Pitch of Complex Tones", *Journal of the Acoustical Society of America*, vol. 54, pp. 1496-1516, 1973.

[9] F.L. Wightman, "The Pattern-Transformation Model of Pitch", *Journal of the Acoustical Society of America*, vol. 54, pp. 407-416, 1973.

[10] E. Terhardt, "Pitch, Consonance and Harmony", *Journal of the Acoustical Society of America*, vol. 55, pp. 1061-1069, 1974.

[11] Malcolm Slaney and Richard F. Lyon, "A Perceptual Pitch Detector", *Presented at the 1990 International Conference on Acoustics Speech and Signal Processing*, vol. 1, pp. 357-360, 1990.

[12] J.C.R. Licklider, "A Duplex Theory of Pitch Perception" in *Psychological Acoustics*, E.D. Schubert (ed.), Dowden, Hutchinson and Ross, Inc., Stroudsburg, PA, 1979.

[13] Roy D. Patterson, "A Pulse Ribbon Model of Monaural Phase Perception", *Journal of the Acoustical Society of America*, vol. 82 (5), pp. 1560-1586, November 1987.

[14] Matti Karjalainen and Tero Tolonen, “Multi-pitch and Periodicity Analysis Model For Sound Separation and Auditory Scene Analysis”, *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 929-932, 1999.

[15] Unto K. Laine, Matti Karjalainen, and Toomas Altosaar, “Warped Linear Prediction (WLP) In Speech And Audio Processing”, *Proc. IEEE ICASSP-94*, Adelaide, Australia, 1994.

[16] Matti Karjalainen, “Auditory Interpretation and Application of Warped Linear Prediction”, *Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing*, Finland, 2000.

[17] Tero Tolonen and Matti Karjalainen, “A Computationally Efficient Multipitch Analysis Model”, *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 708-716, November 2000.

[18] R.D. Patterson, “The sound of the Sinusoid: Spectral Models”, *Journal of the Acoustical Society of America*, vol. 96, pp. 1409-1418, September 1994.

[19] R. Meddis and L. O’Mard, “A Unitary Model for Pitch Perception”, *Journal of the Acoustical Society of America*, vol. 102, pp. 1811-1820, September 1997.

[20] R. Meddis and M. Hewitt, “Virtual Pitch and Phase Sensitivity of a Computer Model of the Auditory Periphery – I: Pitch Identification”, *Journal of the Acoustical Society of America*, vol. 89, pp. 2866-2882, June 1991.

[21] H. Indefrey, W. Hess, and G. Seeser, “Design and Evaluation of Double-Transform Pitch Determination Algorithms with Nonlinear Distortion in the Frequency Domain – Preliminary Results”, in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 11.11.1-11.11.4, 1985.

APPENDIX

Matlab Codes:

Functions developed for computational efficiency and code maintainability:

- *high_low* – this function takes the windowed input sound file and returns two files to be used to determine the summary autocorrelation.
- *esacf* – this function takes the summary autocorrelation result and performs the peak pruning function to eliminate repetitive peaks in autocorrelation.

VITA

Richard Baumgartner was born in 1980 in New Orleans, Louisiana. He graduated from the Saint Paul School in Covington, Louisiana in 1998. In the fall of that year he began studies in electrical engineering at the University of New Orleans. He earned a Bachelor of Science in Engineering degree in May 2002. In the fall of that year he began graduate studies in electrical engineering at the University of New Orleans. During that time he also worked as a teaching assistant in the department of electrical engineering. During the summer of 2003 he worked as an intern for a consulting firm in New Orleans.