MEANINGFUL MUSIC VISUALIZATIONS

by

Kathryn (Lupacchino) Schmidt

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

School of Engineering Technology

West Lafayette, Indiana

December 2018

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. R. Mark French, Chair

Department of Mechanical Engineering Technology

Prof. Nancy L. Denton

Department of Mechanical Engineering Technology

Prof. J. Mike Jacob

Department of Electrical Engineering Technology

Prof. Davin H. Huston

Department of Electrical Engineering Technology

Approved by:

Dr. Duane D. Dunlap

Head of the Graduate Program

Dedicated to my family, and in loving memory of my grandparents.

ACKNOWLEDGMENTS

Above all, I thank my God for giving me the strength and capacity to accomplish this substantial undertaking, along with an unending desire to learn. I wish to gratefully acknowledge my thesis committee for their patience, insight, help, and support throughout this thesis process. Thank you for not only challenging and pushing me, but also reining me back when I needed it. I would like to thank everyone who has helped and supported me throughout the process of this thesis, whether it was technical guidance, grammatical or formatting advice, or emotional and moral support. I specifically want to thank:

- Dr. Mark French
- Professor Davin Huston
- Professor Mike Jacob
- Professor Nancy Denton
- Dr. Daniel Leon-Salas
- Niedra McLeland

I would also like to acknowledge the unending love and support of my family over the course of this journey. I especially want to thank my husband and best friend, Joel. I could not have realized this goal without his guidance, encouragement, and love.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

3-D	three-dimensional
ACF	Autocorrelation Function
AMDF	Average Magnitude Difference Function
CAMLS	Computer-Aided Music-Learning System
¢	cent(s)
D/HH	Deaf or Hard of Hearing
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
HPS	Harmonic Product Spectrum
Hz	hertz
kHz	kilohertz
MAM	Music Animation Machine
MIDI	Musical Instrument Digital Interface
MPM	Motion Pixels of Music
µs	microsecond
ms	millisecond
PDE	*Processing*© Development Environment
SDF	Square Difference Function
USB	Universal Serial Bus

GLOSSARY

Unless otherwise noted, all definitions are taken from the Merriam-Webster online dictionary (Merriam-Webster.com, retrieved 2018).

Cent – a unit of pitch based on the equal tempered octave such that one equal tempered semitone is equal to 100 cents (Nave, n.d.).

Frequency – the number of repetitions of a periodic process in a unit of time: such as the number of complete oscillations per second of energy (such as sound or electromagnetic radiation) in the form of waves

Note – a written symbol used to indicate duration and pitch of a tone by its shape and position on the staff

Octave – a musical interval embracing eight diatonic degrees

Pitch – the property of a sound and especially a musical tone that is determined by the frequency of the waves producing it: highness or lowness of sound

Rhythm – the aspect of music comprising all the elements (such as accent, meter, and tempo) that relate to forward movement

Semitone – a difference in sound that is equal to 1/12 of an octave

Synesthesia – a subjective sensation or image of a sense (as of color) other than the one (as of sound) being stimulated

Tempo – the rate of speed of a musical piece or passage indicated by one of a series of directions (such as largo, presto, or allegro) and often by an exact metronome marking

Timbre – the “color” of the tone (Smith & Williams, 1997)

Visualization – using software to generate a visual representation of specific music properties (Fourney & Fels, 2009)

ABSTRACT

Author: Schmidt, Kathryn L. M.S.
Institution: Purdue University
Degree Received: December 2018
Title: Meaningful Music Visualizations
Major Professor: R. Mark French

Given the powerful influence that music carries in all cultures, it is ideal that everyone have the access and means to learn and understand music. Music visualizations are powerful tools which have proven abilities to help people learn and understand music, as well as enjoy the music. However, when used for educational purposes, it is imperative that the visualization be accurate. This thesis investigated the creation of a visualization which could take input from a live guitar and, in real time, accurately display the pitch within ±2¢, correctly display the note name, and transcribe the note onto the Western music staff. Research was conducted on the history of music visualizations, types of music visualizations and their uses, and commonly used methods of pitch detection.

Processing©*3* was selected as the development environment for creating the visualization. Autocorrelation was chosen as the method of pitch detection. Sine waves accurate to 0.01 Hz were generated in MATLAB and used to test the visualization. A *tc electronic*® PolyTune® 2 guitar tuning pedal was used to tune the guitar before input into the visualization. This served as a means to verify the accuracy of the visualization's output. A *Rocksmith*® 1/4-inch to USB cable was used to bring the live guitar signal into the visualization.

Processing©*3* served as a successful tool for creating the music visualization. The visualization correctly displayed the note name, transcribed the note onto the Western music staff, displayed the audio input, displayed the FFT of the audio input, and accurately displayed the pitch in real time. However, autocorrelation did not give the desired results for pitch accuracy.

The detected pitch was not consistently accurate to the desired ±2¢. But, the pitch was accurate to at least ±8¢ over the open string range (E_2 to E_4), and at least ±18¢ over the full range of the guitar (E_2 to D_6). Though *Processing*©*3* worked well to create the visualization, it may not be the best tool for processing the audio when high accuracy is desired in real time.

CHAPTER 1. INTRODUCTION

Chapter one introduces the problem statement, research question, scope, significance, assumptions, limitations, delimitations, definitions, and other background information relevant to this research project.

1.1 Statement of the problem

Music is all around us (Fourney & Fels, 2009); it can be heard in restaurants, stores, offices, gas stations, churches, homes, etc. Music is a vital part of every culture; it is a method of passing down stories and information as well as providing a source of expression, enjoyment and entertainment. When combined with pictures or visualizations, music becomes an even more powerful force. This musical impact is so powerful that many songs have become iconic, and the thoughts and emotions that they evoke have become integral parts of various cultures. For example, the American culture contains many exemplars of effectual/iconic songs, including "Jaws," "Jeopardy," "The Twilight Zone," "The Wedding March," and "The Star-Spangled Banner."

Given music's influence on society, many people of all ages and walks of life aspire to learn and play music. Many begin by learning a familiar instrument such as the piano or guitar. However, learning to play an instrument can be quite a daunting task, especially for those with hearing impairments. Learning music theory is also quite a challenge; grasping fingerings, remembering the notes, and understanding the structures and notations can be a barrier which causes some students to give up on their musical education endeavors. DePrisco, Malandrino, Pirozzi, Zaccagnino, and Zaccagnino (2017) and Cantareira et al. (2016) articulated the difficulties of students to gain a true understanding of musical structures and properties. For people with hearing impairments, musical education can be even more arduous. The development and usage of hearing aids has bolstered the accessibility of music, but hearing aids are not necessarily the ideal solution. Chasin (2003) reported how hearing aids can really distort music, which certainly will not help the user. To truly make music more accessible to the deaf or hard of hearing (D/HH), the music must be conveyed using senses other than auditory.

Visualization is one approach that people have used to bridge this gap in musical accessibility, and also to help students of all levels/abilities learn and understand the technical aspects of music. However, the visualization must be reliable and accurate if it is to be trusted and used. The visualization should also be engaging to the users so that they find it both educational and enjoyable.

1.2 Research Questions

The research questions investigated in this study were:

1. Can an open-source programming environment that is geared towards the visual arts be used to identify and display an individual note's pitch value from a live instrument as accurately as a standard guitar tuning pedal?

2. Can an open-source programming environment that is geared towards the visual arts be used to transcribe the individual notes from a live instrument into a standard musical notation system?

1.3 Scope

This study focused on the creation of a visualization which could accurately identify and display an individual note's pitch value from a live instrument. The visualization also transcribed the individual note into a standard musical notation system. An open-source programming environment was used to both process the audio and create the visualization. The visualization was run in real-time. The number of channels in the visualization was limited to one in order to demonstrate the concept. Only one instrument was chosen for use with the visualization to demonstrate the concept. The visualization was not designed for detecting chords, only single notes. The detectable frequency range of the visualization was limited to the frequency range of standard guitar tuning, E_2 (82.4 Hz) to D_6 (1174.7 Hz), with the reference pitch of A_4 = 440.0 Hz.

1.4 Significance

Music is known and created in some form within every known culture around the world, and exists to pass on history, tales, and other information as well as for pure expression and entertainment. People play music for a myriad of reasons; to create a certain atmosphere, influence emotions, aid with healing from health problems, express feelings, or just simply for fun (DePrisco et al., 2017). Lee and Fathia (2016) state that "hearing and vision are closely tied in human sensitivity" [p. 1]. Musicians use music as a medium to convey their ideas, stories, and expressions. Many artists hold that music has a tremendous ability to touch the human soul (Brougher & Mattis, 2005, p. 31). Suttoni (1989) made the following statement in the introduction of his book translating Franz Liszt's writings:

> Music, in other words, is too ideal of itself to be truly understood by any but the select few. If it is to flourish, therefore, in the world of the Real, it should have a more comprehensible point of reference for the ideas and/or feelings it means to convey. One way of achieving this is to relate it to another form of art, visual or literary. [p. XXV]

Liszt's ideas here apply to both the enjoyment of music and the education of music. Musical ideas can sometimes be difficult to understand; and the concepts, theories, and structures can be particularly difficult to learn. By using another medium as an aid to the music, true understanding can be more easily achieved.

So, why choose visualization? Griscom (2014) stated that "our full experience of the world relies on all of our sensory capabilities...sensory systems can maximize the usefulness of these various streams of information for perception by combining input from multiple modalities into a single, unified representation..." [p. 2]. Brougher and Mattis (2005) talked about how music can be translated to other sensory organs, specifically the eye. Visualization has become one of the most popular ways of relating to music through the use of another sense. Bain (2008) noted that "visualizing music provides an alternate medium that can capture a brief snapshot or history of a music performance" [p. 1].

Visualizations can also be used to supplement and enhance the educational learning experience for students who study music. Cantareira et al. (2016) described the struggles for students to understand the structures as well as gaining a deeper comprehension of the associated musical properties. If students could visualize these structures and properties, and understand what the sound "looks" like, it could greatly aid in the development of musicians. DePrisco et al. (2017) conveyed some challenges that lie in visualizing the structures of music and gaining a true understanding of those structures. The two major challenges in visualizing musical structures identified by DePrisco et al. (2017) included preprocessing the musical input data and creating a "meaningful and intuitive graphical representation of music" [p. 140]. The focus of this thesis centered in on the preprocessing challenge.

The D/HH community is no exception when it comes to enjoying, learning, and understanding music. Fourney and Fels (2009) express that something should be done to help the D/HH community so that its members are more equipped and able to fully enjoy and understand music. Visualization is a natural choice of medium to enhance the musical experience for the D/HH community.

1.5 Assumptions

The assumptions for this study included the following:

- The need existed for this music visualization tool.
- The programming software chosen was a viable tool for processing the audio and creating the visualizations.
- The programming environment chosen worked properly and as expected.
- The guitar tuning pedal used in the study was properly calibrated by the manufacturer and met the standards and accuracy specified by the manufacturer.
- The guitar string tension remained constant between tuning and input into the visualization.

- The guitar string bending stiffness was negligible.
- The guitar string was plucked in an analogous manner each time.

1.6 Limitations

The limitations for this study included the following:

- Only one note was analyzed and displayed at a time.
- Only one instrument was chosen for use in this study.
- Only one music notation system was chosen for transcription of the notes.
- The detectable frequency range of the visualizer was limited to the frequency range of standard guitar tuning with the reference pitch of A_4 = 440.0 Hz.

1.7 Delimitations

The delimitations for this study included the following:

- The system was not designed to analyze and/or display multiple notes simultaneously.
- This study did not explore or compare all possible visualization software options.
- This study did not seek to characterize all instruments.
- This study did not analyze all possible pitch detection algorithms.

1.8 Summary

Chapter one detailed the problem statement, research question, scope, significance, assumptions, limitations, delimitations, definitions, and other background information for the research project. The next chapter provides a review of the literature relevant to this research.

CHAPTER 2. REVIEW OF LITERATURE

Chapter two provides a review of the literature relevant to music visualizations, as well as the techniques and methodologies used to create them.

2.1 Why Visualize Music?

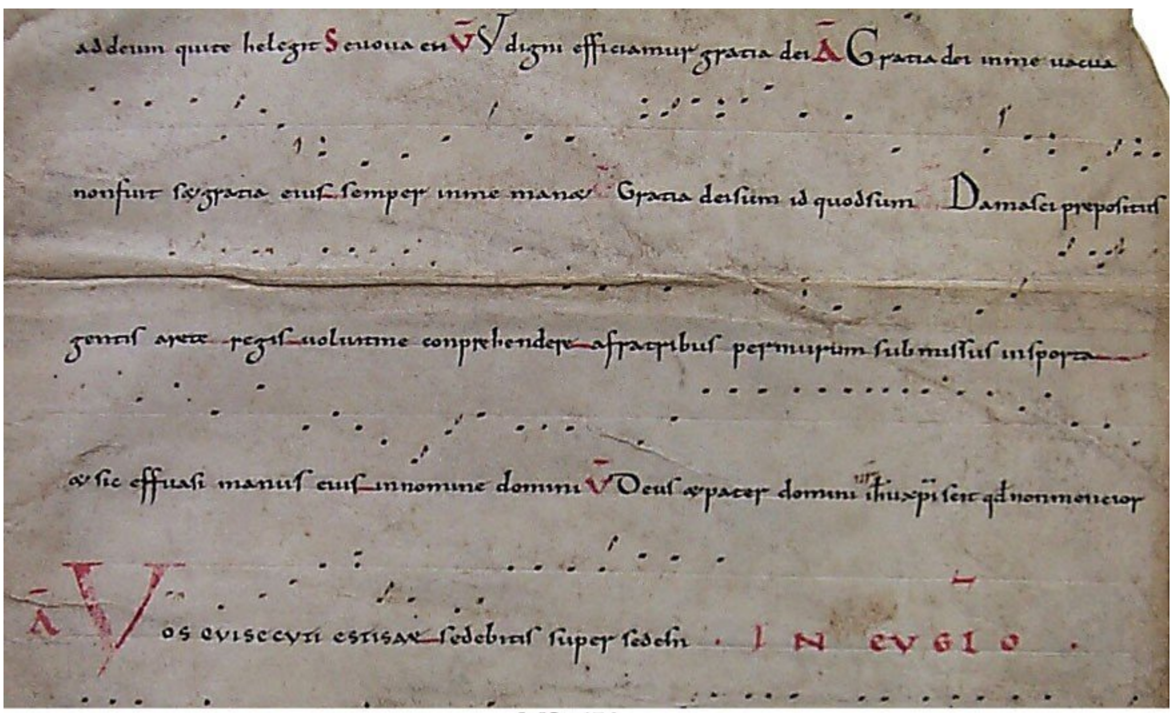
Visualization in some form plays an appreciable role in our everyday lives. When it comes to music, visualizations can be equally as important. Throughout history, both musicians and the people enjoying the music have used visualization as one means to give music a more permanent form (Pon, 2012). Bain (2008) writes that "Visualizing music provides an alternate medium that can capture a brief snapshot or history of a music performance" [p. 1]. Combining music with visualizations allows listeners to gain a deeper connection to the artist and the emotions that the artist seeks to convey through the music. Visualization also plays a vital role in the successful comprehension of music (Thibeault, 2011), both for the enjoyment and the education. In his book titled *The study of ethnomusicology: Thirty-one issues and concepts*, Nettl (2005) conveys how an understanding of music requires some sort of visual aid. Chapter seven, titled "I can't say a thing until I've seen the score" [p. 74], specifically discusses use of the Western music notation format.

2.2 A Brief History of Music Visualizations

Musical notations and their history is a very substantial subject, and this thesis does not seek to give a comprehensive history of musical notation. A brief overview of this history, however, is beneficial to the understanding of musical visualizations and the key roles that they fill in human lives.

2.2.1 Musical Notations

The two primary purposes for using music notation are to guide the performer and to serve as a written record (Pon, 2012). The origins of the Western conventional music notation can be traced back to mnemonic devices known as neumes – dots and squiggles (Pon, 2012)– which were used in plainchant as a method of indicating changes in the pitch or melody. These neumes are thought to be rooted in cheironomic hand gestures (Gerson-Kiwi & Hiley, n.d.), as the Latin meaning of *neuma* signified “gesture, sign, movement of the hand” (Pon, 2012, p. 15). After the 10th century, neumes came to represent more; they could represent both single pitches as well as a series of pitches simultaneously. However, specific pitches were not indicated (Bent et al., n.d.). Figures 2.1 and 2.2 show examples of neumatic notation found on historical manuscripts (*The Schøyen Collection Series*, n.d.).



MS 658
Aquitanian neumes on 1-line F staff. Limoges, France, ca. 1030

Figure 2.1. Neumatic notation - Aquitanian neumes (*The Schøyen Collection Series*, n.d.).

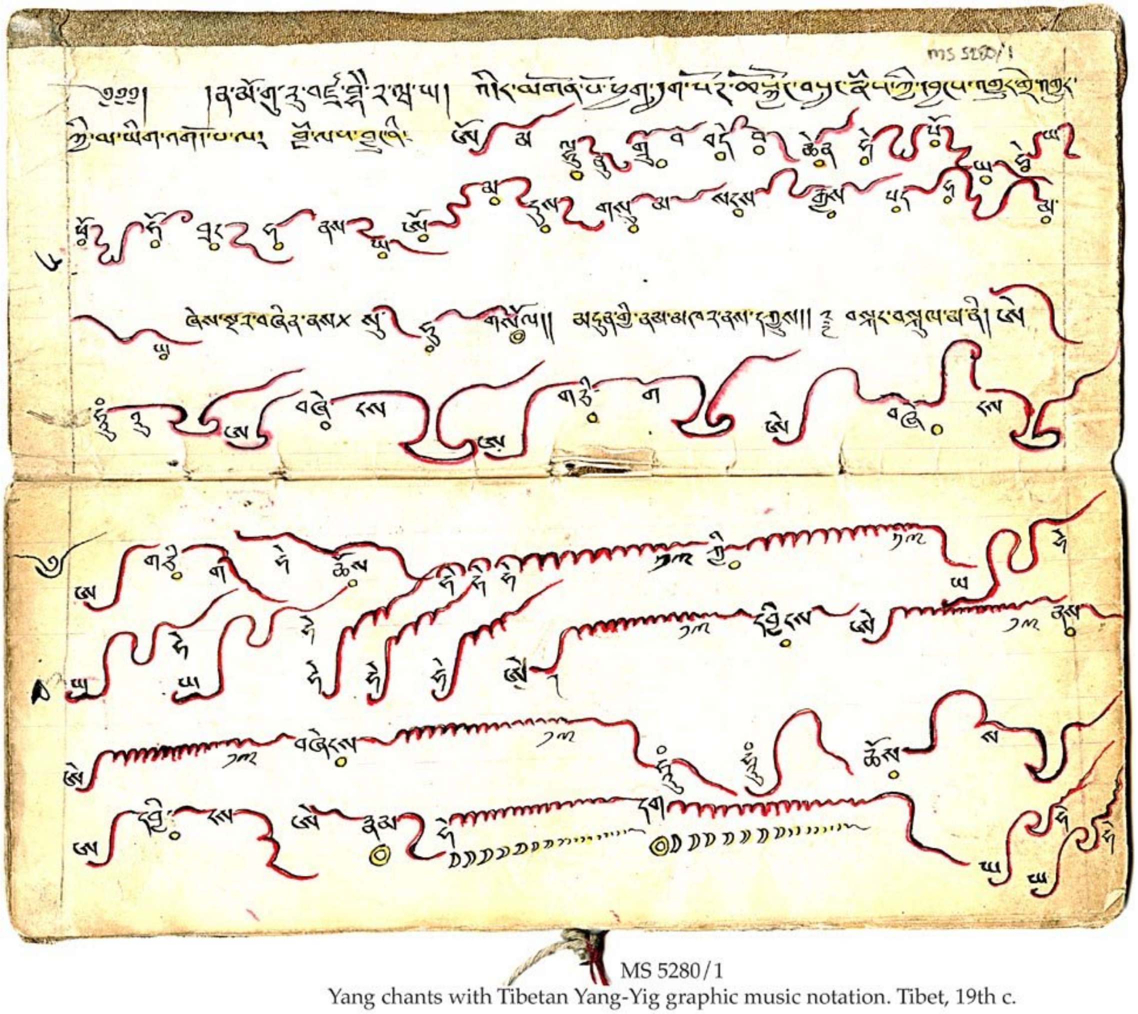


Figure 2.2. Neumatic notation - Yang chants with Tibetan Yang-Yig graphic music notation (*The Schøyen Collection Series*, n.d.).

Musical notations continued to evolve over time, accommodating for more complex music. Modern Western staff music notation has become the ubiquitous standard system for the notation of music. This system now includes both symbols (ie. fermatas and accidentals) and graphics which resemble the sounds created (ie. crescendos and decrescendos) (Pon, 2012). See figure 2.3 below for a sample of commonly used symbols and graphics in Western music notation.

Figure 2.3. Commonly used symbols and graphics in Western Music (*Musical Symbol 2228728*, 2016).

However, one large problem with Western staff notation is that it does not necessarily translate well across other cultures - these notations may not correctly represent another musical culture (Thibeault, 2011). Nettl (2005) states that "...there is no inter-culturally valid conceptualization or definition of music. Very few societies have a concept (and a term) parallel to the European 'music.' They may instead have taxonomies whose borders cut across the universe of humanly organized sound in totally different ways from those of Western societies." [p.17]. Another problem with Western staff notation is that it requires training to understand; and to fully comprehend and use its capabilities requires many hours of extensive studying and training. This steep learning curve tends to cause music students to struggle, and for many it becomes a barrier to music. In order to address this struggle, new forms of notation systems have started to emerge, and many have the goal of making the process of learning how to read, write, and play music a more approachable and enjoyable experience. One such example is the *Music Notation Project*, developed by Morris, Dalley, Johnston, Keislar, and Keller (n.d.). This visualization project seeks to provide users with alternative forms of notations. The mission stated on the project's website is:

> To raise awareness of the disadvantages of traditional music notation, to explore alternative music notation systems, and to provide resources for the wider consideration and use of these alternatives [para. 1].

2.2.2 Music through Art

Music can be visualized through several mediums, including art and dance (Fourney & Fels, 2009). With the advancement of electronics and computers, a whole new form of music visualization was opened, as well as a new way to use art and dance to visualize music. One popular example of this is the film *Fantasia* developed by Disney in 1940. *Fantasia* is a film that was remarkably technologically advanced for its time, and each of the musical works chosen for the film was beautifully depicted through both abstract and literal art developed by Disney's team of artists (see figures 2.4 and 2.5). These artistic renderings not only successfully communicated some of the emotional, technical, and other aspects of the music, but was also highly entertaining. The film required an immense amount of work from hundreds of people; such films would never be practical to develop for mass amounts of music. (Fourney & Fels, 2009; Lee & Fathia, 2016).

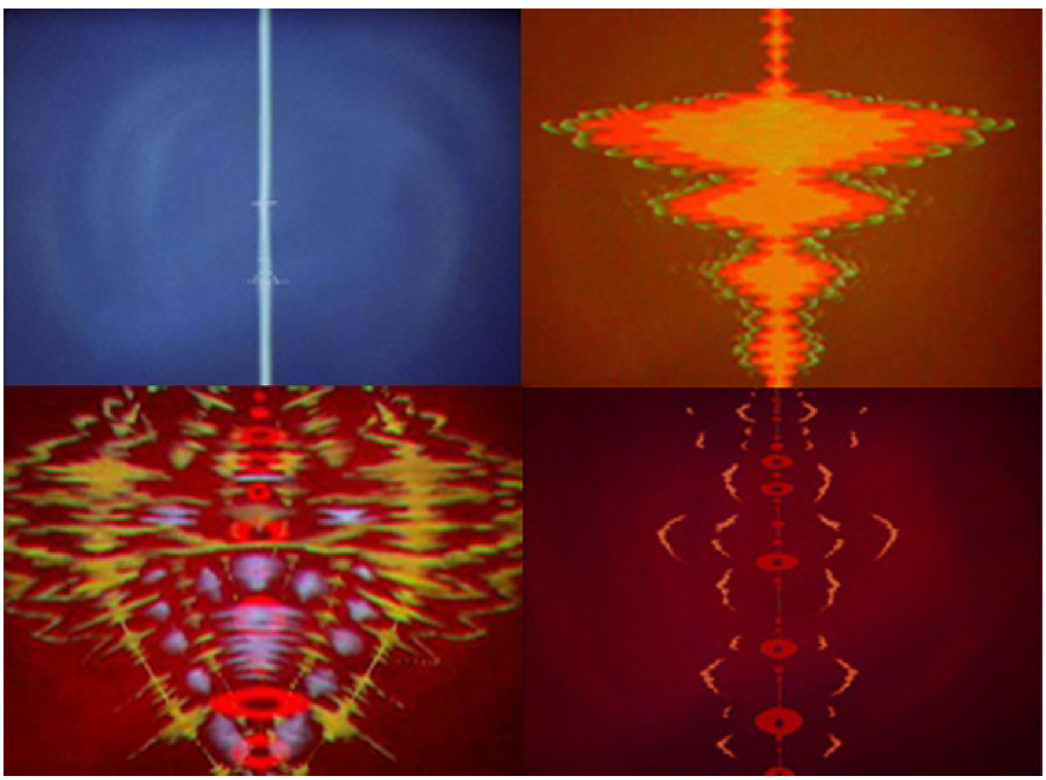


Figure 2.4. Walt Disney's *Fantasia* – the Soundtrack visualization was used to depict what an instrument's sound *looks* like (*Fantasia (1940)*, 2015).

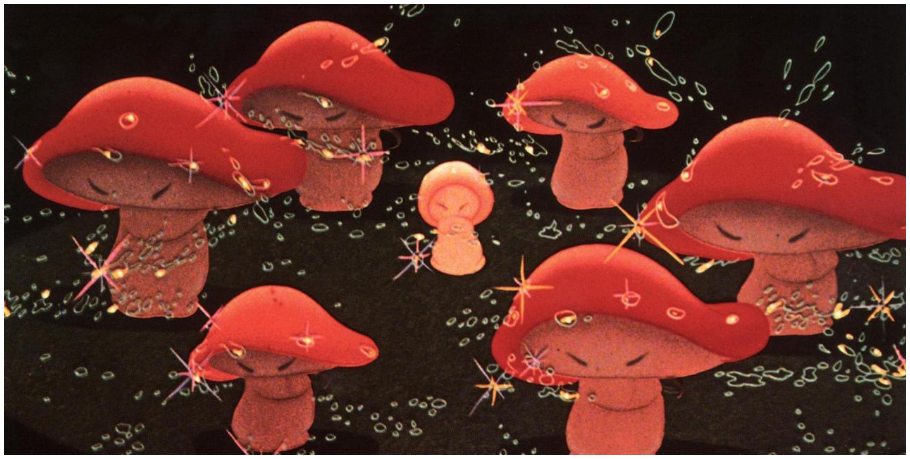


Figure 2.5. Walt Disney's *Fantasia – Chinese Dance* by Pytor Ilyich Tchaikovsky (Maltin, 2015).

Another early use of electronics to visualize music through art was the Atari Video Music system. Developed in 1976 by R.J. Brown, this program enabled users to create their own visual effects on a television using a series of buttons and knobs (see figure 2.6). The user could also place the system in an automatic mode where the video would automatically react to the music (Brown, 1976). This visualization was purely abstract, and created solely for the user's entertainment. While this system was very entertaining to use, it did not serve as a good means of communicating the technical information in the music.

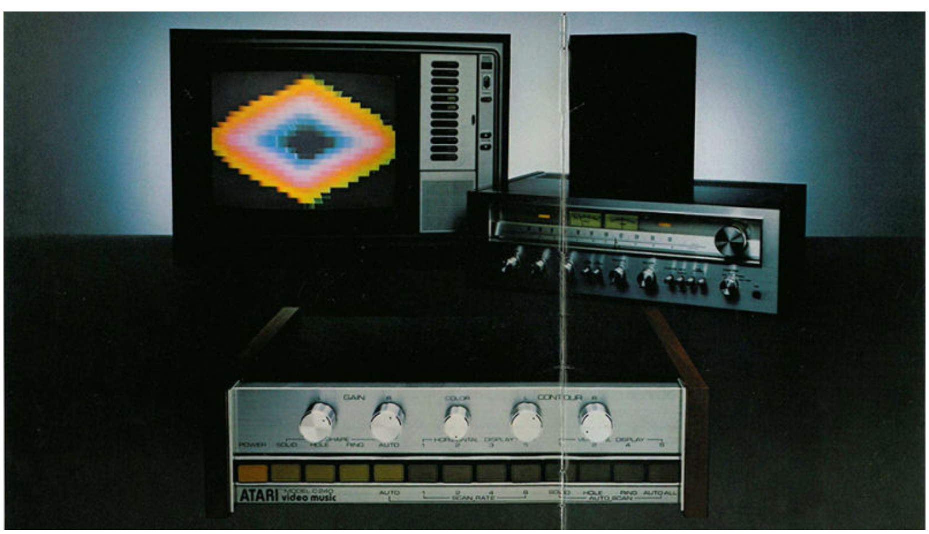


Figure 2.6. R.J. Brown's Atari Video Music (Blinddog, 2012).

Griscom studied the relationships people have between music and color, using the Emotional Mediation Hypothesis: “cross-modal mappings between music and color are actually mediated by shared emotional meaning of the two sets of stimuli” (Griscom, 2014, p. 2). He discovered evidence that semantic and emotional features may play a guiding role in the cross-modal responses. While he did not directly create a visualizer based on musical emotions, he did discover a link between the music and emotions which guide users towards particular shapes and colors. However, using colors to illustrate emotion in music inherently bears a large challenge. Different colors carry very different meanings across cultures, and some of those meanings lie at opposite extremes. For example, blue can represent relaxation and peacefulness, but it can also represent sadness and loneliness (ie. having “the blues”). Red is a happy and auspicious color in Asian countries, but in the Middle East it is associated with danger and evil (Cousins, 2012).

Exploring a more mathematical relationship between music and color led many artists and inventors such as Sir Isaac Newton, Castel, Rimington, and Helmholtz to devise color-mapping schemes which mapped sounds to colors of the rainbow. In 1704, Sir Isaac Newton conceptualized the chromatic mapping of the seven notes of the piano (A, B, C, D, E, F, G) to the seven colors of the rainbow (red, orange, yellow, green, blue, indigo, violet) (Bain, 2008; Collopy, 2004). Since then, many other color-note scales have been presented. See figure 2.12 below for an illustration of some of these color scales. Beyond the chromatic mapping, another way in which artists and inventors sought to map colors to notes was using the circle of fifths. Alexander Scriabin used the circle of fifths in 1911 (figure 2.8), and described his approach as one “which assumes an approximately spectral order if we commence with the note C and proceed in the circle of fifths’” (Jones, 1972, p. 104).

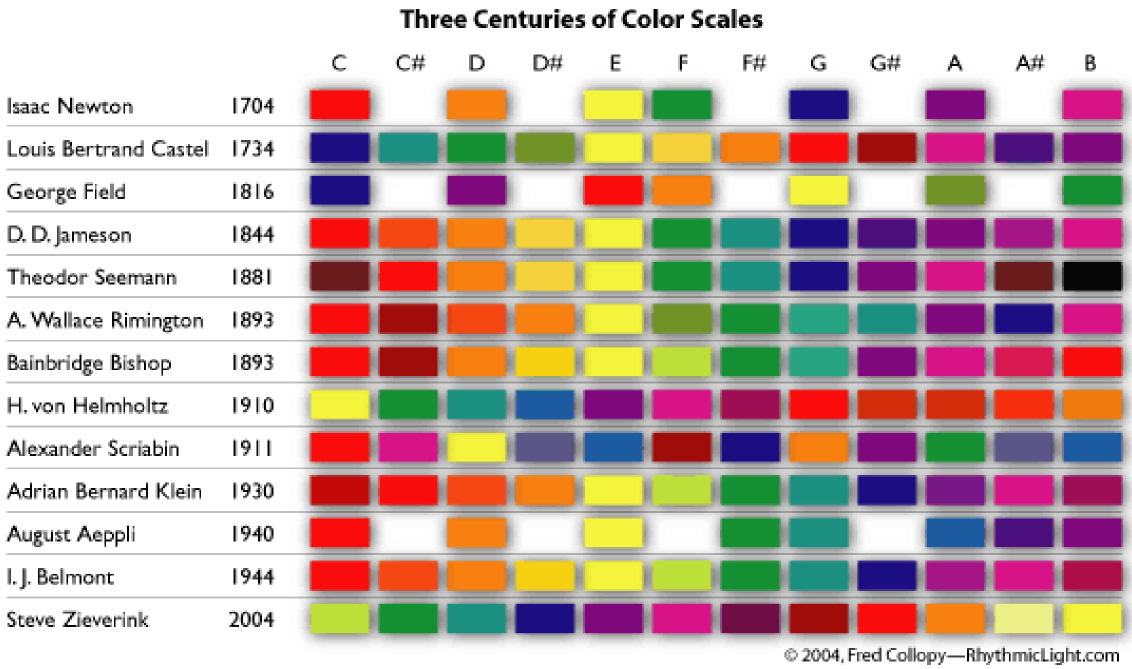


Figure 2.7. Three centuries of color scales (Collopy, 2004).

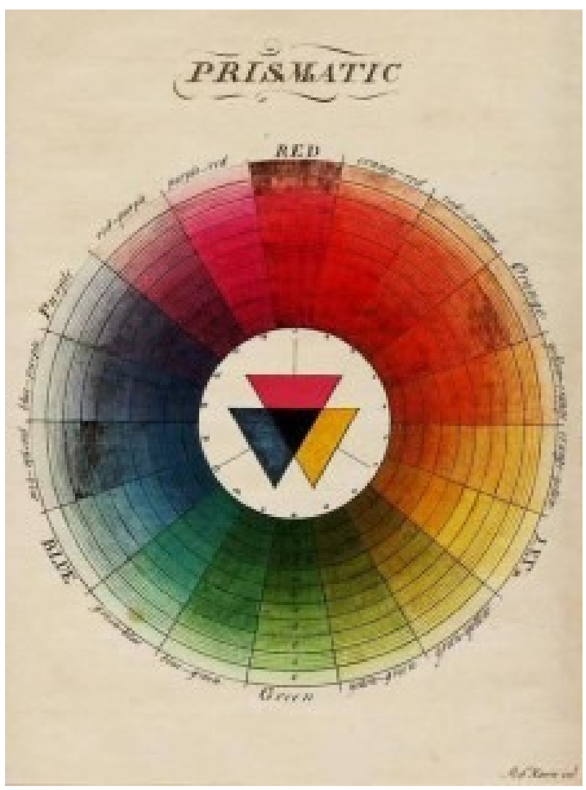


Figure 2.8. Scriabin's color wheel, based on the color-theory of Rudolf Steiner (Wolfman, 2016).

2.3 Visualization Applications and Techniques

The following sections will provide an overview of several visualization applications and specific programs, as well as the various techniques implemented and their effectiveness.

2.3.1 Visualizations for Information and Education

Visualizations in some form have been used as a tool for education in a wide variety of subjects for as long as humans have been educating themselves. Visualizations have numerous helpful applications when applied to the study of music. DePrisco, Malandrino, Pirozzi, Zaccagnino, and Zaccagnino (2017) presented a study which successfully concluded that visualization is an effective approach to aid in the understanding of the structures of musical compositions. Klemenc, Ciuha, and Solina (2011) also explored these educational possibilities, specifically using colour visualization to depict the fundamental harmonic relationships between musical tones.

Fourney and Fels (2009) found that most music visualization research is being conducted with an emphasis on musical analysis and education. This research ranges from new notation systems to the creation of various computer programs that allow the users to “see” the notes, instruments, sound waves, etc. Real-time visualizations are created for educational and training purposes, based on how musicians mentally visualize sounds. Many of these visualizers can effectively communicate much of the technical information within a piece of music. A sampling of such visualizers will be detailed in this section.

Cantareira, Nonato, and Paulovich (2016) developed a visualizer called *MoshViz* (Musical Overview, Stability and Harmony Visualizer) to help students to understand structures and harmonic patterns in music, as well as the qualities of various musical instruments such as harmony, stability, and complexity. *MoshViz* displays entire pieces of music, while also allowing the user to zoom in on specific sections to see more detail (see figure 2.9). Displaying information about individual notes, *MoshViz* can depict the relationships between note sequences as well as the musical properties which enables students to interpret and analyze musical pieces quickly.

While *MoshViz* is a great tool for musical education, it does not provide much in the way of user entertainment.

Figure 2.9. MoshViz prototype tool (Cantareira et al., 2016).

Also aimed at music learning, Hiraga, Watanabe, and Fujishiro (2002) created a music visualizer that displays an entire piece of music. The main goal of their visualizer was to help students improve their musical performances; but the visualizer also served to give users a better overall understanding the performance, as well as make the musical information of a piece accessible and usable. The emotional content of the music was communicated using expressive cues, displayed as human facial expressions, which related to various qualitative music terms. Users generally found this visualizer to be helpful when trying to understand the various aspects/qualities of the musical performance. While Hiraga et al. created a useful visualizer that also possessed the ability to convey some of the emotions in music, it has no user entertainment value.

Thibeault (2011) used spectrograms as a means to enhance the analysis and understanding of world music. Thibeault sought to find a way to visually explore various music cultures from around the world, because Western standard notation is not translatable to all other cultures. Spectrograms were chosen because they provide a visualization of the entire audio spectrum in a simple and easy-to-understand manner. Spectrograms are able to capture and display important elements of music, such as timbre and pitch, in a clean and combined view (see figure 2.10). The user is able to view the amount of time desired as well as the frequency range desired. Thibeault found that students were able understand and become comfortable with the spectrogram visualizations with as little as five minutes of training (Thibeault, 2011). While spectrograms were shown to be a highly useful tool for displaying the musical features and information, they do not provide a high entertainment value.

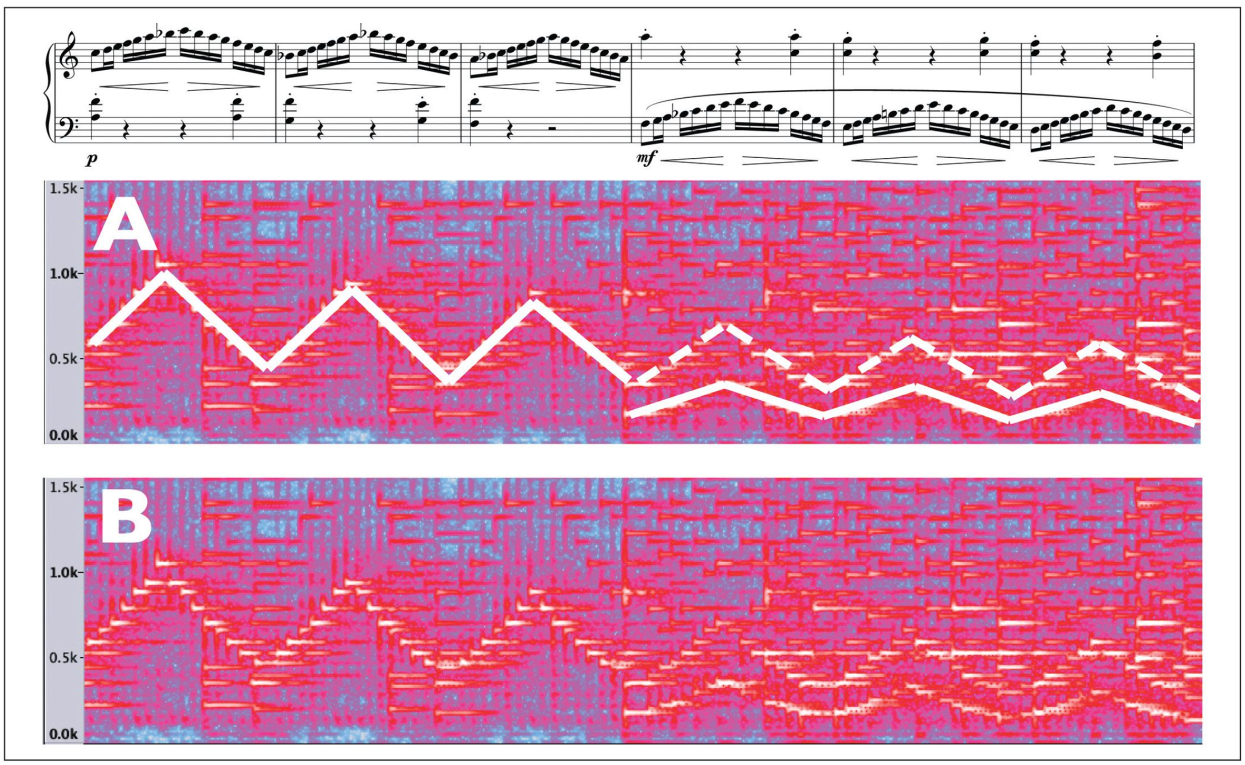


Figure 2.10. Thibeault's spectrogram. (A) Marked image showing fundamental pitch (solid line) and overtone (dotted line), (B) unmarked image (Thibeault, 2011).

Smith and Williams (1997) developed a new visualizer which used color and three-dimensional (3-D) space to represent the musical information. The program took musical instrument digital interface (MIDI) files and generated graphical portrayals of the information located in the file. Tones were represented with colored spheres, and the location, size, and color of the spheres were determined by three musical properties – pitch, volume, and tone. Pitch (or note value) determined each sphere's location along the y-axis. This axis was scaled proportionally to the maximum and minimum pitch values in the individual piece of music. Volume (or velocity) dictated the radius of the sphere. The radii values were also scaled proportionally to the maximum and minimum volume values in the piece of music. Timbre is a characteristic of a tone that can be described as the "color" of the tone. Ergo, timbre was chosen as the designator of the color of each sphere, and depicted the contrast between groups of instruments. Instruments were spaced evenly on specific values along the x-axis, and a global scale factor was applied so that everything scaled together nicely. No work was done to incorporate the emotion of the music into the visualizer.

Educational visualizations have also been developed to assist the D/HH. One such visualization was developed by Yang, Lay, Liou, Tsao, and Lin (2007) to teach D/HH junior high students various tones and how to play them on the flute. Yang et al. sought to understand how pitch response impacted those with hearing impairments, so they developed the computer-aided music-learning system (CAMLS). The CAMLS system served as a visualization tool which presented real-time feedback to the students through animations, so that students could *see* what they sounded like while learning to play the flute. While this was a very useful educational tool, it did not provide much entertainment value.

2.3.2 Visualizations for Entertainment

A plethora of visualizer programs are available on the Internet where users can both play and visualize their music. The majority of audio player software programs now have some sort of visualization(s) built in to the program. These audio programs are very popular, and are used by many people every day. A few of these programs with built-in visualizations include *iTunes®*, *Windows Media Player™*, *WinAmp™*, *VLC™*, *Plane9*, *Renderforest*, and *SoundSpectrum™*. See figures 2.11 and 2.12 below for examples of these visualizations. Visualizations such as these can be seen in public venues as well, including restaurants and concerts. While these types of visualizers can be very entertaining and maybe even artistic, and may convey some of the emotion behind the music, they are mostly abstract and provide no technical information about the music. They can also be misleading; the visual excitement of the images may convey an emotionally exciting song, but in reality the song may be very sad.

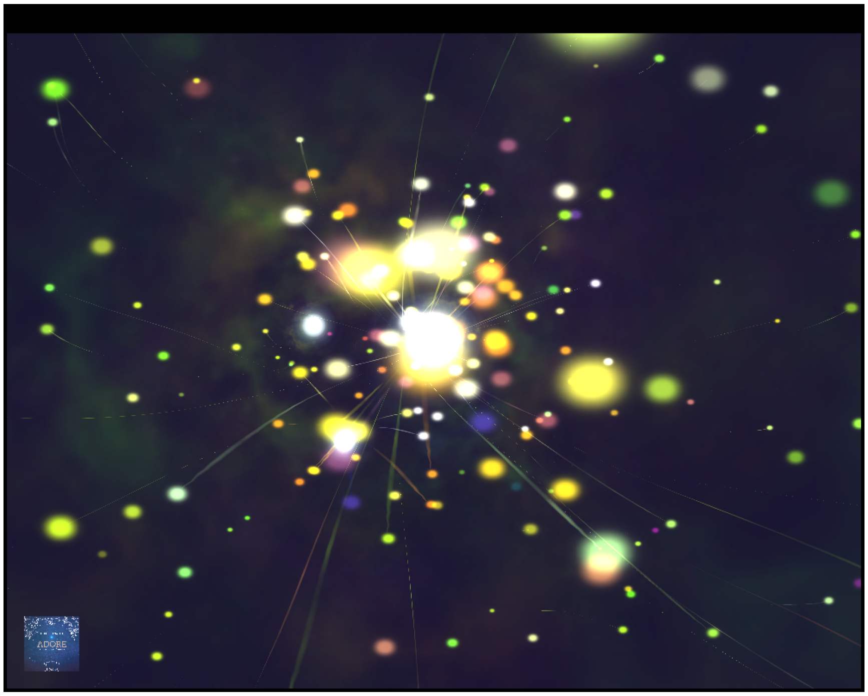


Figure 2.11. iTunes® visualization.

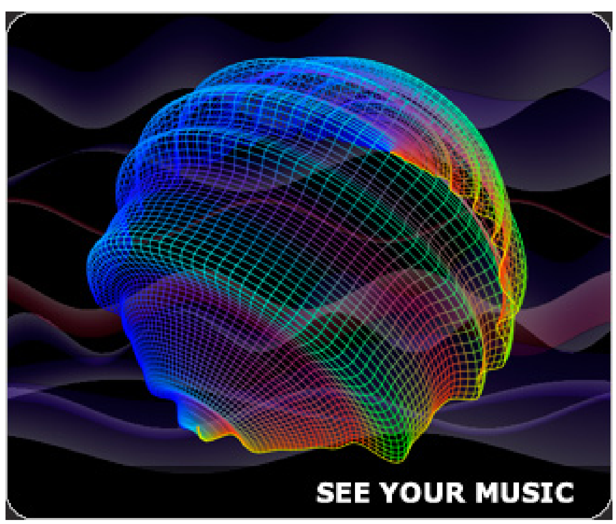


Figure 2.12. Whitecap by *SoundSpectrum™* (O'Meara et al., n.d.).

Outram (2016) created *Synesthesia*, an interactive music visualization which was based in a virtual environment. *Synesthesia* allowed the users to completely "immerse" themselves in the three-dimensional (3D) artistic representation of the music. The program was also used to visualize the user's own voices. Users could pause/resume the visualizations at their will, and also "orbit" around to observe the visualization from different angles. Figure 2.13 shows a screen shot from the visualizer. Outram reports that "colors are related to the sound frequency spectrum, the size of the particles are related to volume, and time is the forward spatial dimension. Different rows of sphere particles correspond to (from top) pads, FX, kicks, bass, percussion and synths" [p. 1]. *Synesthesia* sought to visualize the way the humans perceive sound and color while also providing an entertaining experience. While this approach produces a large value of entertainment, it does not provide the user with clear and easily understood technical information.

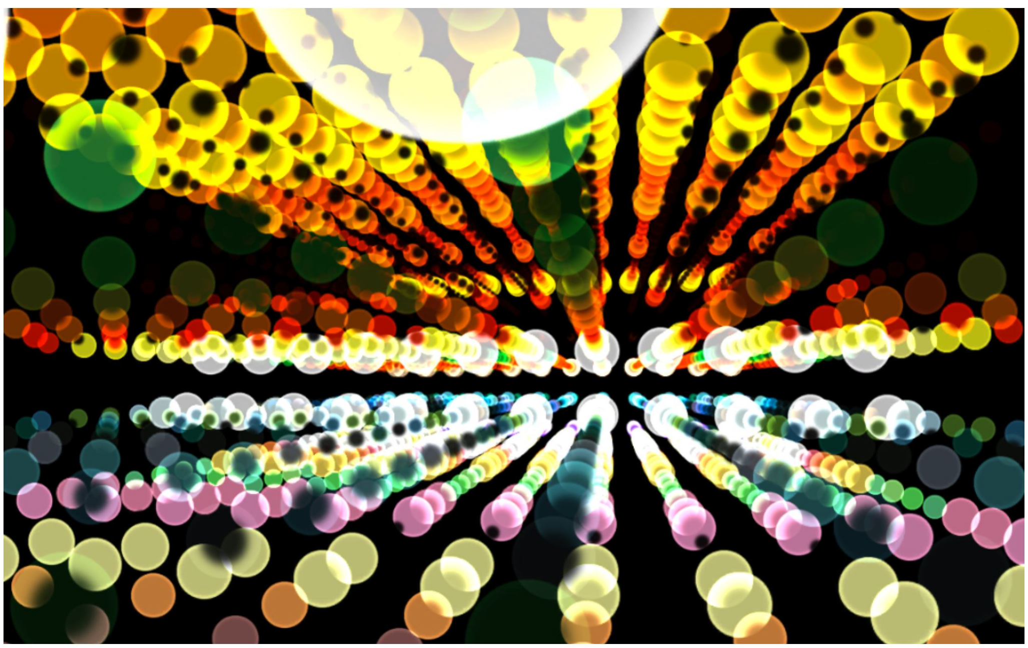


Figure 2.13. Outram's interactive visualizer *Synesthesia* (Outram, 2016).

Vusik is an interface developed by Pon (2012). *Vusik* enables the users to compose electronic music graphically, allowing them to visualize and experience the music while also directly interacting with it. Users are able to *see* their music come to life as they compose. The user "paints" gestures onto a large interactive screen using an electronic palette and brush (see figure 2.14). The sounds generated from the paintings can be heard in playback mode, but also in real-time as the user is composing, enabling the user to focus on the sounds coming from individual strokes of the brush. *Vusik* proved to be an engaging and useful tool for the creation of beautiful music visualizations. However, these visualizations are not necessarily geared towards entertainment or education; they are used more for the exploration and composition of music.

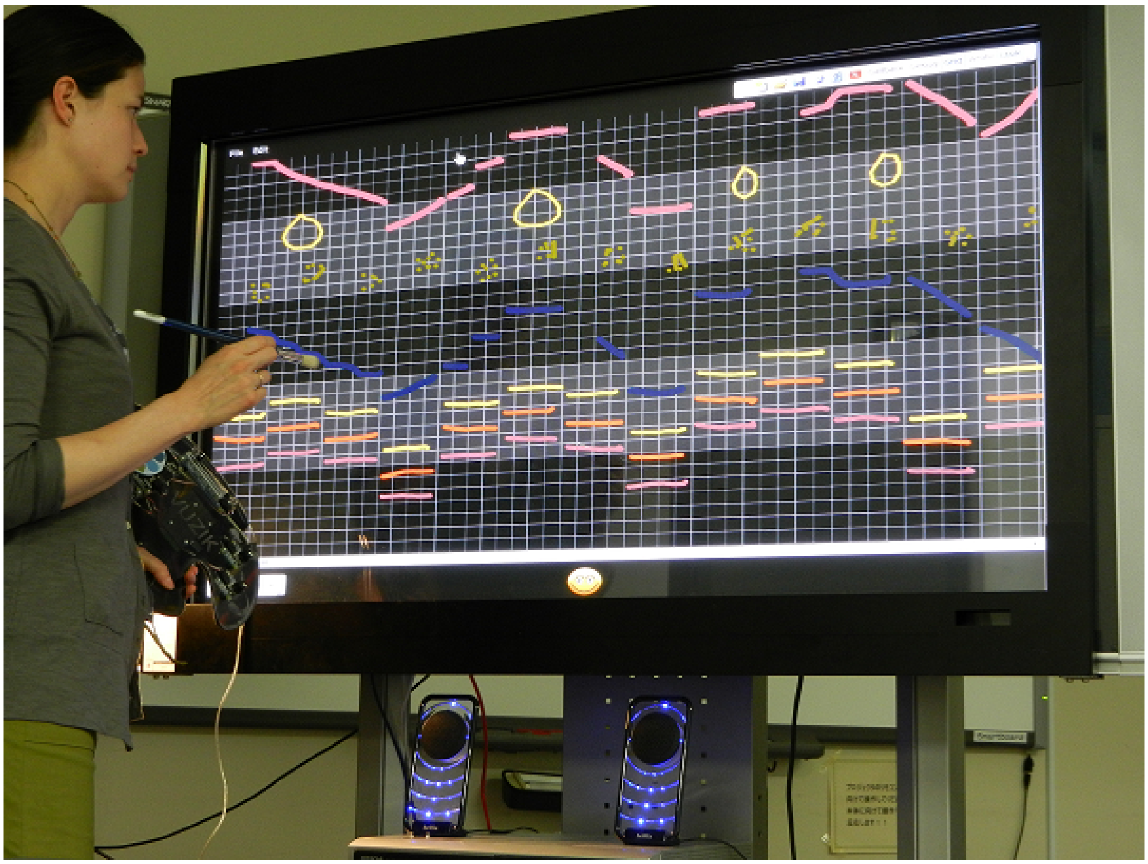


Figure 2.14. Pon demonstrating composition with Vuzik (Pon, 2012)

2.3.3 Visualizations for both Education and Entertainment

The majority of visualizers have a goal of either teaching or entertaining; very few are designed to do both. One fitting example of a visualizer that may be capable of both teaching and entertaining is the *Music Animation Machine* (MAM) developed by Stephen Malinowski. Malinowski stated that:

> Music moves, and can be understood just by listening. But a conventional musical score stands still, and can be understood only after years of training. The Music Animation Machine bridges this gap, with a score that moves – and can be understood just by watching (Popova, n.d., para. 2).

Malinowski developed the MAM in the 1970's to provide a more understandable and visual way of representing musical scores (Popova, n.d.). He used the computer to develop his MAM based on MIDI file inputs. The MAM uses several visualization techniques, including piano roll, part motion, compass, sonar, swarm, curves, and tessellation. MAM visualizations also consist of numerous different visual configurations using various shapes, colors, patterns, and backgrounds (Malinowski, n.d.). Examples of the MAM visualizations can be seen in figures 2.15 and 2.16. MAM visualizations are certainly dazzling and enthralling, but any musical information beyond the artistic representation of the written score is not necessarily clear. Technical information such as what instruments are being played, tempo, rhythm, playing techniques (ie. legato or pizzicato), and note names are a few things that, if their artistic portrayals were delineated, could improve this visualization's educational value.

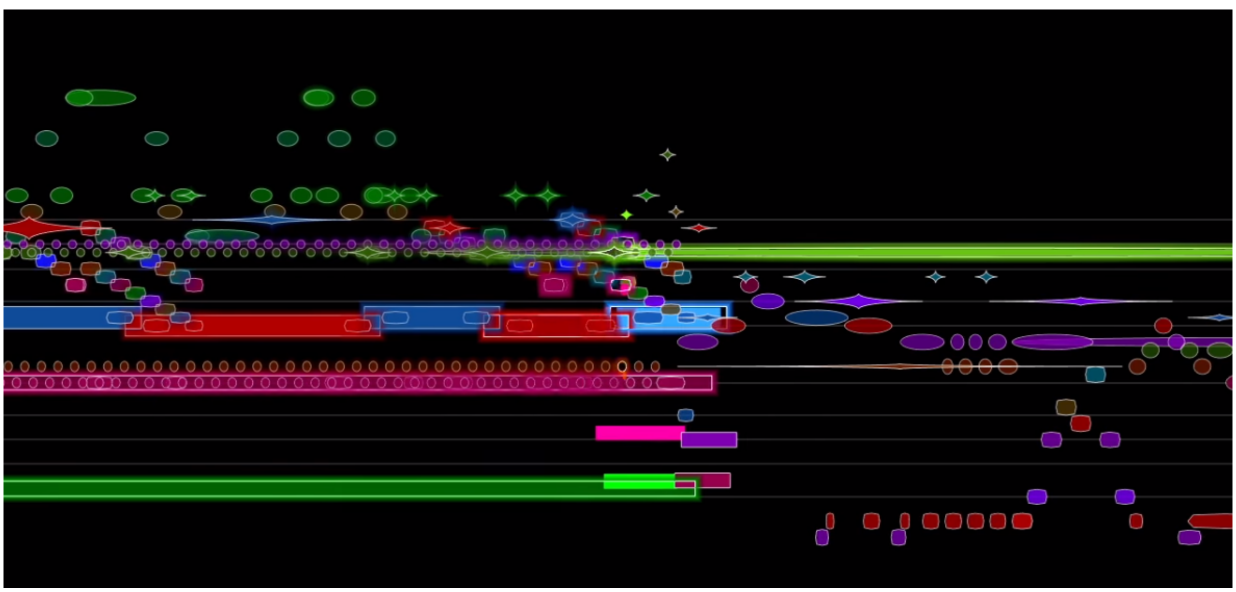


Figure 2.15. Malinowski's *Music Animation Machine* – piano roll technique (Malinowski, n.d.).

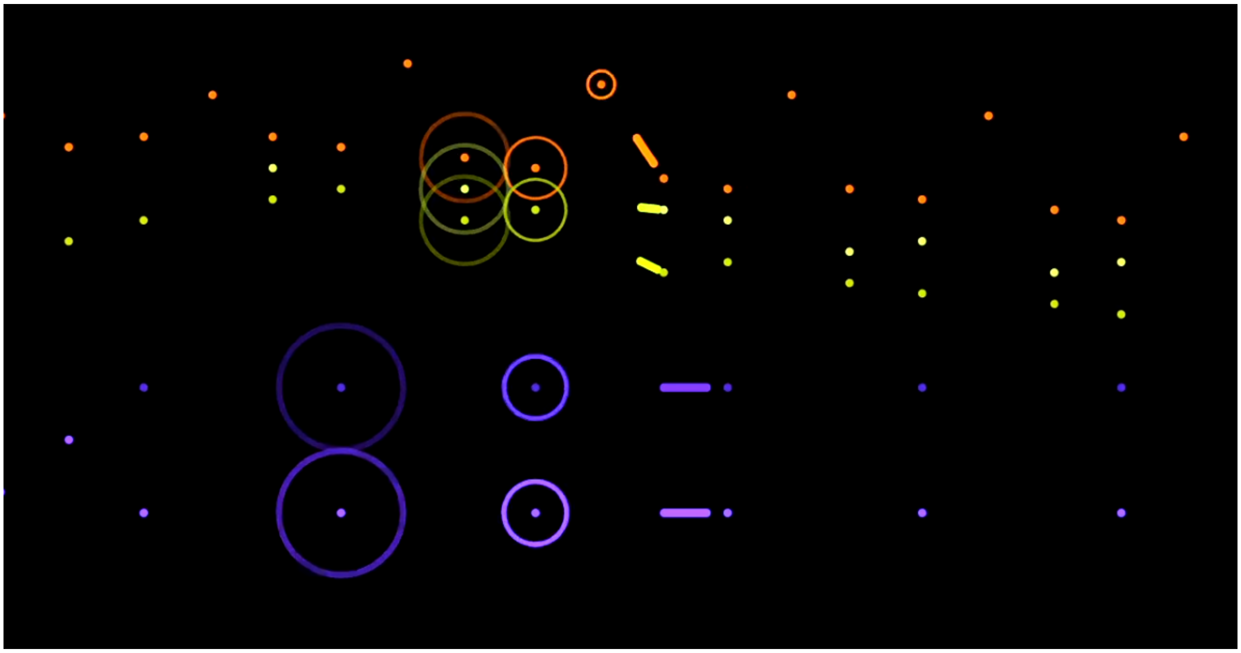


Figure 2.16. Malinowski's *Music Animation Machine* – radar technique (Malinowski, n.d.).

Fourney and Fels (2009) conducted a study on various music visualizations, and the participants in the study deemed Malinowski's MAM to be somewhat lacking. Participants indicated that MAM's part motion visualizer did the best job of indicating what was going on in the music, but they did not enjoy the MAM visualization as much as the *iTunes®* *Magnetosphere* visualization. The piano roll and tonal compass MAMs were both described as uninteresting. The *Motion Pixels of Music* (MPM) visualization was found to be the most informative, but "boring" and "too visually busy" [p. 943]. Other participant's comments included that the "music was not conveyed well", and that they "could not really connect" [p. 943] to the visualizations.

2.4 Musical Experiences for the Deaf and Hard of Hearing

When people think of music, the aural experience is generally the main thing that comes to mind. However, music contains much more than aural information; there are both tactile and visual sensations that can supplement the musical experience (Fourney & Fels, 2009). These tactile and visual sensations are best experienced at live performances – feeling the beat and/or watching performers on stage. Due to these extra sensations, the D/HH can really experience the music even if they cannot completely hear it, because they can *see* and *feel* it. When music is recorded, it loses some of the effectiveness of the tactile and visual elements of the live show, which makes it much more difficult for the D/HH.

Pool (1993), writer for the Los Angeles Times, described another method of bringing music to life for the D/HH through the use of sign language. Mindy Brown, an interpreter for the deaf, uses sign language to not only bring spoken words to life, but also concerts. She works to bring the tempo, tone, and sounds of the instruments to life. When interviewed, Mindy stated "There will be a story. The story will be in my mind. I'll think about the music with my heart and then paint a picture of it with my hands." [p. 1]. Mindy described how deaf people long to know what different instruments sound like, and also how she translates these instrument's various sounds through gestures and motions. She then suggested that a flute could be depicted as the wind, and cymbals could be depicted as thunder.

Pool (1993) also interviewed Rhondee Beriault, an actress who was born deaf, who attended a concert where Mindy Brown was interpreting. Rhondee recalled the many deaf children sitting around her at the concert, all of whom were very excited. Rhondee said "when they [children] saw Mindy translating it into pictures, they could understand." [p. 2]. Mindy was able to successfully translate the music into visual signs and images for the deaf attendees so that they could understand the music. An executive member of the Los Angeles Philharmonic Association, James Ruggirello, stated that "it's a matter of educating the hearing people: many deaf people do love music." [p. 1]. While the deaf may not be able to hear the music directly, they are still able to enjoy it by simply experiencing it in a slightly different way. Though tactile sensations can play a key role in helping the D/HH to experience music, visualizations play a crucial role in brining the technical aspects of music to life for the D/HH.

2.5 Introductory Music Theory Concepts

The subject of music theory is both broad and complex, and this thesis does not seek to provide an extensive survey of the subject. However, a brief overview and basic understanding of music theory is beneficial to the comprehension of this thesis.

In his book titled *Engineering the Guitar: Theory and Practice*, R.M. French (2009) wrote that "all music, no matter what kind, is made of the same building blocks – individual notes" [p. 9]. The term *note* is used to define a named pitch. The term *pitch* is defined as "a perceptive quality that describes the highness or lowness of sound" (McLeod, 2008), determined by the frequency of the sound waves. The sine wave is the simplest form of a pitched sound; its pitch is determined exclusively by the frequency of the sine wave. Likewise, if multiple sine waves are played concurrently, each wave will have a peak in the frequency spectrum which coincides with an individual note (Forsberg, 2009).

Music in the real world, however, naturally contains periodic components known as partials (or overtones). These partials vary in frequency, as well as phase and amplitude. The partials can be either harmonic or inharmonic. The harmonic partials align in what is known as a harmonic series. These harmonic frequencies are whole-integer multiples of the fundamental frequency (Forsberg, 2009), and can be seen in a Fast Fourier Transform (FFT).

In Western music, there are seven natural notes (A, B, C, D, E, F, and G) and five derived notes, known as sharps and flats, which each lie between two notes. On the piano, the natural notes are the white keys and the sharps and flats are the black keys. These notes are all related to each other via very specific ratios of frequencies, known as intervals. Intervals are described as "the distance between notes" (French, 2012, p. 53). The smallest interval is a half step (or a semi-tone), and all of the other intervals are comprised of various multiples of the half step. These musical intervals are typically expressed in units known as *cents*. In the equal tempered octave (Nave, n.d.), which is the largest interval, each tempered semitone has a value of 100¢. See table 2.1 below for a listing of the intervals and the number of half steps and cents that define them.

Table 2.1. Musical Intervals

Half Steps	Cents	Interval Name
1	100	Half Step
2	200	Whole Step
3	300	Minor Third
4	400	Major Third
5	500	Perfect Fourth
6	600	Augmented Fourth/Diminished Fifth
7	700	Perfect Fifth
8	800	Minor Sixth
9	900	Major Sixth
10	1000	Minor Seventh
11	1100	Major Seventh
12	1200	Octave

Any given note is able to be adjusted in frequency lower or higher than its ideal value without changing the note itself; this adjustment is known as tuning. Notes are tuned to a desired pitch, and the pitch is customarily correlated to a specific frequency (ie. A_4 = 440 Hz). Albert Frantz (n.d.) illustrated the difference between notes and pitches with the concept of a rainbow. Just as there are seven naturals (or primary notes) in Western music, there are, likewise, seven distinct colors in the rainbow: red, orange, yellow, green, blue, indigo, and violet. Just as a rainbow technically contains an infinite number of colors (covering the full color spectrum), the musical spectrum technically contains an infinite number of pitches. A red color can be a few shades away from pure red, but will still be perceived as red. In the same way, a C note can be a few cents off of its calculated pitch, but still be a C.

Though frequency values are typically expressed in hertz (Hz), when a note is off pitch, the distance between the current pitch and the ideal pitch is conveyed in cents (¢). Using the equal tempered octave where one octave equals 1200¢, there is a power of two relationship (Nave, n.d.) which yields the following formula:

$$\frac{f_2}{f_1} = 2^{\frac{c}{1200}} \tag{2.1}$$

2.6 Pitch Detection

In order to identify a note, a user must take the audio input and perform some signal processing to identify the pitch (or fundamental frequency), and then the note name can be determined based upon the frequency value. There are two main categories for pitch detection techniques: time-domain and frequency-domain.

2.6.1 Time-Domain Methods

Time-domain methods of pitch detection use the raw waveform data to determine the fundamental frequency.

2.6.1.1 Zero-Crossing

One of the most basic time-domain methods of detecting pitch is the zero-crossing method (Forsberg, 2009; McLeod, 2008). The zero-crossing method determines the fundamental frequency by simply looking at the features of the waveform, specifically the number of times that the wave crosses over the threshold between positive and negative (zero-crossing). The time between two consecutive zero crossings is known as the period of the waveform. The frequency of the waveform has an inverse relationship with the period, T, and can be calculated using the formula:

$$f = \frac{1}{T} \tag{2.2}$$

The zero-crossing method is computationally inexpensive. However, it is not the most suitable solution for complex or noisy signals, as harmonics can cause multiple zero-crossings to occur within one period (McLeod, 2008).

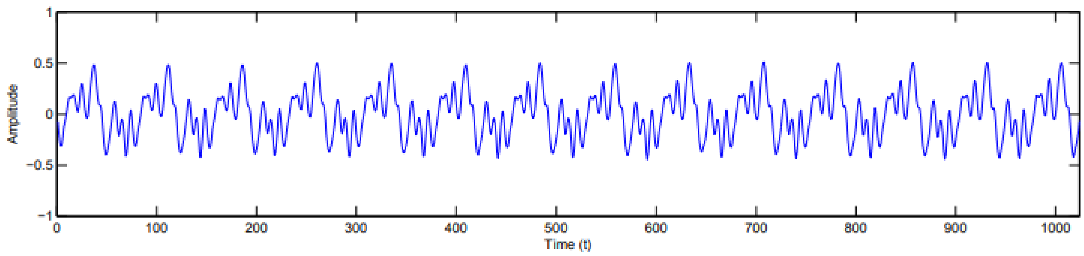
2.6.1.2 Autocorrelation

Autocorrelation is one of the popular time-domain pitch detection methods (Forsberg, 2009; McLeod, 2008; Singh & Kumar, 2014; VonDemKnesebeck & Zölzer, 2010). Autocorrelation works by taking an input waveform and comparing it with a shifted copy of itself, resulting in a cross-correlation. The autocorrelation function (ACF) creates peaks (or maxima) at integer multiples of the period T of the input signal. There are two equations for the ACF (McLeod, 2008): one is for use on a discrete window containing a stationary signal (eq. 2.3), and the other for use on non-stationary signals (eq. 2.4).

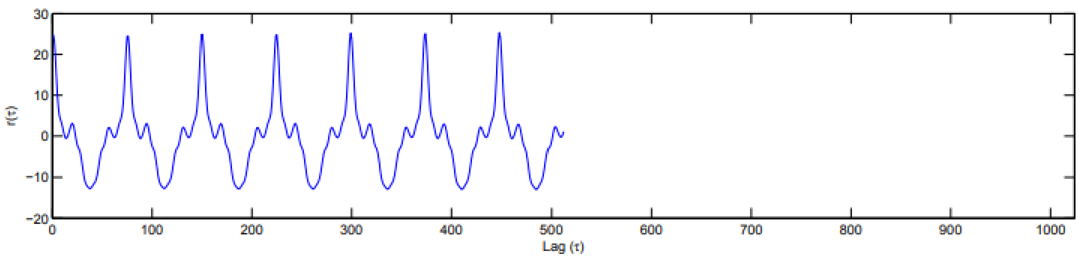
$$r(\tau) = \sum_{n=0}^{(N/2)-1} x(n) * x(n+\tau), \qquad 0 \leq \tau \leq \frac{N}{2} \tag{2.3}$$

$$r(\tau) = \sum_{n=0}^{N-1-\tau} x(n) * x(n+\tau), \qquad 0 \leq \tau \leq N \tag{2.4}$$

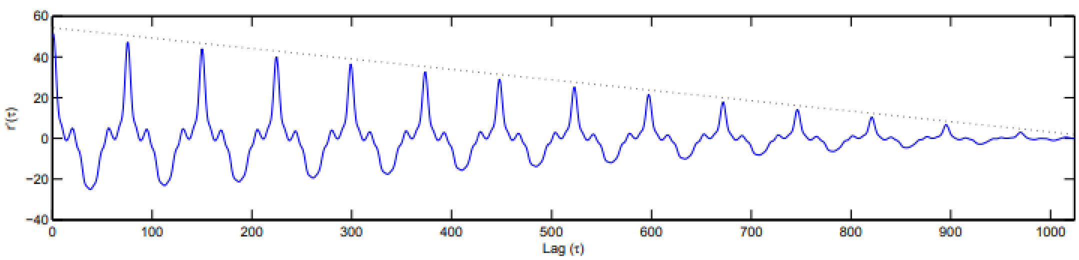
The second equation (eq. 2.4) creates a tapering effect on the autocorrelation result, so the values will move closer and closer to zero as the summation approaches the end of the window (length N). See figure 2.17 for a comparison of the two autocorrelation functions. Figure 2.17 part (c) shows the effects of the tapering on the autocorrelation result. This tapering effect is useful in finding the largest maxima after $\tau = 0$, which corresponds to the first period. The time at this maxima is then divided into the sampling rate, resulting in the fundamental frequency, or pitch. Before running the data through the ACF, a windowing function can be applied to facilitate smoother transitions as the window moves (Fourney & Fels, 2009). A commonly used window in autocorrelation is the Hamming window.



(a) A window of size $W = 1024$ taken from a violin recording



(b) An autocorrelation of type-I



(c) An autocorrelation of type-II

Figure 2.17. A comparison of two autocorrelation functions on an audio signal. (b) equation 2.3, (c) equation 2.4 (McLeod, 2008).

The autocorrelation method works well when signals are perfectly periodic; but, real instruments do not always have such nice periodicity (McLeod, 2008). Even still, the ACF is generally robust against signals with harmonic imperfections (Singh & Kumar, 2014). However, the ACF is highly sensitive to sampling rate. The shifted waveform is ultimately a shift in samples, the size of which is dictated by the sampling rate. Ergo, the lower the sampling rate the lower the resolution, and the higher the sampling rate the higher the resolution.

2.6.1.3 Square Difference Function

The Square Difference Function (SDF) is a close cousin to the autocorrelation function (deCheveigné & Kawahara, 2002; McLeod, 2008). However, instead of the algorithm coming to a maixma at the periods, the square difference function comes to a minima. The equation for the SDF is shown in equation 2.5 below.

$$r(\tau) = \sum_{n=0}^{N-1-\tau} (x(n) - x(n+\tau))^2 \tag{2.5}$$

2.6.1.4 Average Magnitude Difference Function

The Average Magnitude Difference Function (AMDF) is also very similar to autocorrelation. The equations are very close, but like the SDF, the AMDF calculates the difference between the original and shifted segments instead of their product (eq. 2.6). This makes the AMDF computationally faster than the ACF. AMDF is thought to be more accurate than autocorrelation, but autocorrelation has performed better when compared with AMDF (Singh & Kumar, 2014). Given AMDF's similarities with autocorrelation, it is also highly sensitive to the sampling rate.

$$r(\tau) = \sum_{n=0}^{N-1-\tau} |x(n) - x(n+\tau)| \tag{2.6}$$

There are numerous papers written about the ACF, SDF, AMDF, and various combinations and modifications of them (deCheveigné & Kawahara, 2002; Forsberg, 2009; McLeod, 2008; Muhammad, 2011; Singh & Kumar, 2014; VonDemKnesebeck & Zölzer, 2010). Some have been more successful attempts than others, but all of these time-domain equations have the same major limitation – the sampling frequency. Each of these methods is comparing the original signal with a shifted version of itself, and the shifting is done in increments of samples. Autocorrelation and its related methods are best suited for monophonic pitch detection in the mid-to-low frequency ranges (Forsberg, 2009).

2.6.2 Frequency-Domain Methods

The following sections will describe several major frequency-domain methods, including the Fourier Transform, Harmonic Product Spectrum, and Cepstrum.

2.6.2.1 The Fourier Transform

The well-known Fourier Transform takes a time-domain signal and transforms it into the frequency domain, resulting in what is known as a frequency spectrum. The Fast Fourier Transform (FFT) is an optimized version of the Discrete Fourier Transform (DFT). A Fourier Transform works by exploiting the symmetrical properties of the DFT calculation (equation 2.7) so that it can improve the speed of the calculation. In order to perform an FFT, the FFT length (or window size, N) must be a power of two.

$$X(k) = \sum_{n=0}^{N-1} x(n) * e^{\frac{-j2\pi nk}{N}} \tag{2.7}$$

The highest frequency in the FFT is known as the Nyquist frequency, and it is equal to half of the sampling frequency. For example, if the sampling rate is 44100 Hz, the Nyquist frequency would be equal to 22050 Hz (equation 2.8). This means that the spectrum will only contain values for frequencies less than 22050 Hz.

$$f_{nyquist} = \frac{44100\ Hz}{2} = 22050\ Hz \tag{2.8}$$

The Nyquist frequency shows that the sampling rate must be at least twice that of the highest frequency present in the waveform (equation 2.9).

$$f_{sampling} \geq 2 * f_{max} \tag{2.9}$$

DFTs and FFTs alike suffer from the phenomenon known as spectral leakage. Spectral leakage is a result of an input signal frequency not being an integer multiple of the fundamental frequency. When this happens, the input frequency will show up in the DFT or FFT result. In order to address this leakage issue, windowing functions can be applied to the input signal to taper the beginning and end of the signal. Figure 2.18 shows a spectrum with leakage before and after windowing. Some popular windowing methods include Hamming, Hanning, triangle, and Blackman.

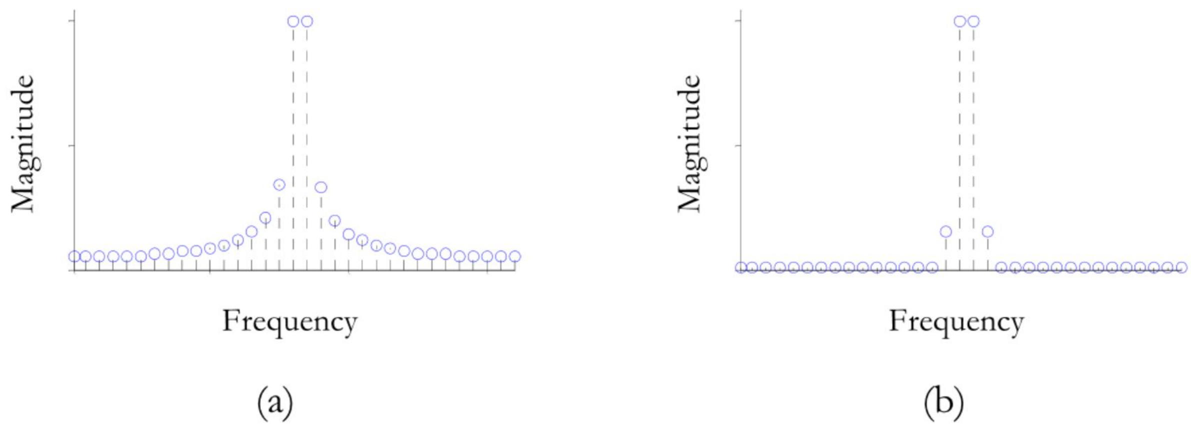


Figure 2.18. (a) A spectrum with leakage, (b) The same spectrum with a Hamming window applied (Forsberg, 2009).

In an FFT spectrum, the fundamental frequency (pitch) corresponds to the first peak of the spectrum. However, the first peak may not be the tallest peak. This makes detecting the fundamental tricky, and this becomes an even larger problem in cases where the fundamental is very small or missing (McLeod, 2008). While the FFT is a relatively simple and straightforward method, another major drawback of using the FFT as a means of pitch detection is that the resolution of the FFT is directly limited by both the sampling frequency and the FFT length. The resolution of the FFT is defined as:

$$\Delta f = \frac{f_{sampling}}{N} \tag{2.10}$$

In order to achieve an accuracy comparable to a guitar tuner, one would need a very high sampling frequency as well as a very long FFT window. This is highly impractical for most audio processing solutions, and would not work well in a visualization application where accuracy was a key focus.

2.6.2.2 The Harmonic Product Spectrum

Once an FFT has been performed on an audio signal, the Harmonic Product Spectrum (HPS) can be used to determine the pitch (Forsberg, 2009; McLeod, 2008). The HPS determines the fundamental frequency by downsampling the spectrum a certain number of times. Each time the spectrum is downsampled, a more compressed version of the spectrum is produced. Downsampling results in an alignment of the harmonics (see figure 2.19). These downsampled/compressed spectrums are then multiplied together, which amplifies the harmonics that are aligned and reveals the fundamental. Overall, HPS works well and is insensitive to noise (Forsberg, 2009). However, a major limiting factor is that HPS performs better with longer windows; windows with only two or three periods of the signal are not handled as well (McLeod, 2008). This becomes problematic for the frequencies at the lower end of the guitar range, and extending the window length increases the processing time.

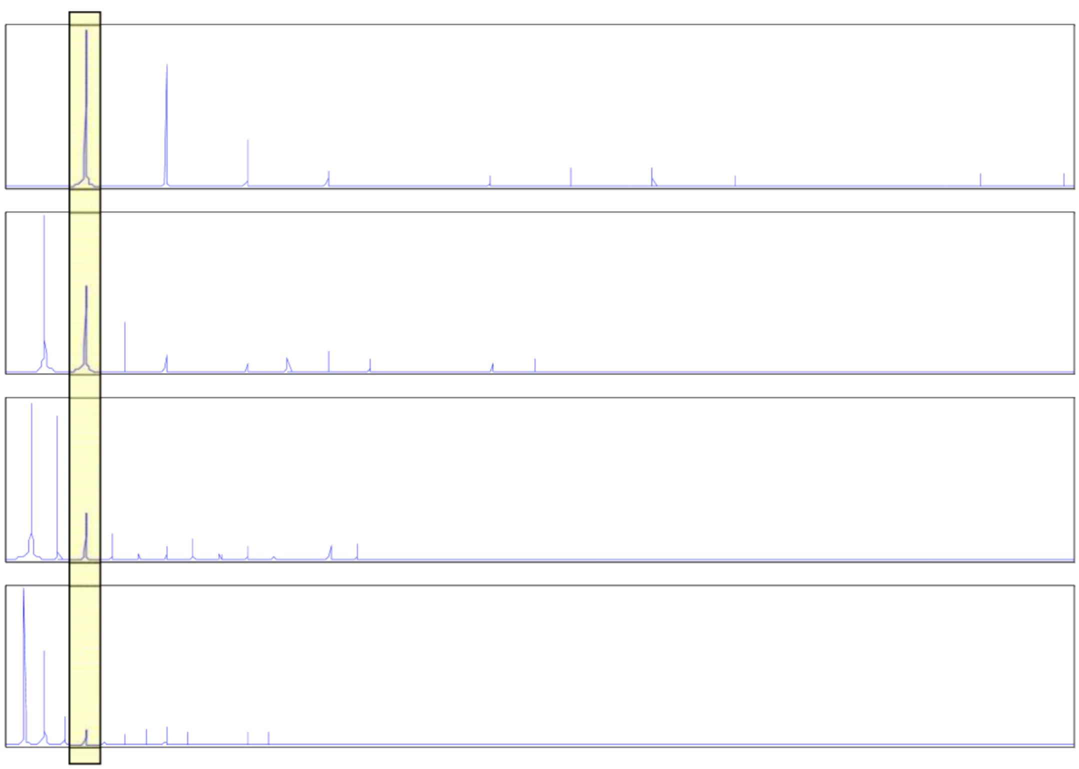


Figure 2.19. HPS – alignment of harmonics in downsampling (Forsberg, 2009).

2.6.2.3 Cepstrum

The term "cepstrum" is a clever play on the word "spectrum," as it is simply the reversal of the first four letters of the word "spectrum." As its name suggests, a cepstrum is a spectrum of a spectrum (McLeod, 2008; Singh & Kumar, 2014). The time domain input is converted to the frequency domain using an FFT, and then that spectrum is converted from a linear to a logarithmic scale. The inverse FFT is then applied, resulting in the power spectrum (and converting back to the time domain). The maxima in the cepstrum's quefrency domain is located, and the fundamental frequency is then calculated in the same manner as the ACF (Singh & Kumar, 2014).

2.7 Summary

Chapter two provided a review of the literature relevant to music visualizations, introductory music theory concepts, and an overview of a sampling of common pitch detection methods. The next chapter provides the framework and methodology to be used in the research project.

CHAPTER 3. RESEARCH METHODOLOGY

Chapter three provides the research framework and methodology used in the research study. It details the hypotheses, population and sample set, key variables, measures for success, threats to validity, and test methodology.

3.1 Study Design

This study was quantitative. The goal was to determine if music notes and their pitch can be accurately identified, transcribed, and visualized using an open-source programming environment that is geared towards the visual arts. The programming environment chosen was *Processing*©*3*. The PDE (*Processing*© Development Environment) was used to both process the input audio and create a visualization containing the pitch value, note name, and note transcription. The number of instruments used in the study was limited to one to demonstrate the concept, and a guitar was selected as the live instrument. The tuning pedal chosen for this study was the *tc electronic*® PolyTune® 2, which boasts an accuracy of ±0.1¢ (tc electronic, 2014). A *Rocksmith*® 1/4-inch to USB cable was used to input the guitar signal into *Processing*©*3*. Data collected included note names, pitch values, and note location. The note names and locations were judged on correctness, and pitch value data was analyzed for accuracy.

3.2 Unit & Sampling

The following sections discuss the hypotheses, population, sample, variables, and the measures for success.

3.2.1 Hypotheses

The hypotheses for this study were the following:

$H1_0$: The individual note's pitch was *not* correctly identified and displayed with an accuracy of ±2¢.

$H1_\alpha$: The individual note's pitch was correctly identified and displayed with an accuracy of ±2¢.

$H2_0$: The individual note name was *not* correctly displayed and transcribed into a standard music notation.

$H2_\alpha$: The individual note name was correctly displayed and transcribed into a standard music notation.

3.2.2 Population

The population was all notes within the standard guitar range. Using a reference pitch of A_4 = 440.0 Hz, this standard guitar range covered 82.4 Hz (E_2) to 1174.7 Hz (D_6), which resulted in a population of 47 notes.

3.2.3 Sample

Given the small population size of 47 notes, the sample size for testing with sine waves was all 47 notes in the standard guitar range. For testing with a live instrument, a subset of this population was chosen. The subset consisted of six notes, one for each open string on the guitar. The six open strings were chosen due to the fact that a guitar can only be tuned when the strings are open; the pitch is fine-tuned when the string tension is adjusted with the tuners, located on the headstock of the guitar. The six open strings were tuned to a specific pitch using the tuning pedal, and then input into the visualization at a known frequency.

3.2.4 Variables

The variables in this study were note, pitch, and accuracy. Accuracy was the dependent variable, as it was the variable of interest. Note and pitch were the independent variables, as they were the contributors of change against which accuracy was measured.

3.3 Measures for Success

Success in this study was considered achieved with the culmination of the following:

- The visualization could be created using an open-source programming environment.
- The visualization could be run in real time.
- The visualization could accurately identify and display single note pitch values from a live instrument within ±2¢.
- The visualization could correctly display single note names from a live instrument.
- The visualization could correctly transcribe single notes from a live instrument.

3.4 Threats to Validity

Threats to validity in this study included the following:

- Assumptions that string tensions are constant.
- Assumptions regarding the string bending stiffness to be negligible.
- Assumptions that the string was plucked in exactly the same manner and same place each time.

3.5 Visualization Design

The programming environment *Processing*©*3* was used to design and build the visualization, as well as process the audio signals. The visualization design was kept simple and clean, as the focus of this thesis was the accuracy of the visualization. The visualization displayed the following:

- note name and octave
- pitch (fundamental frequency)
- raw audio input waveform
- FFT of the audio input
- transcription of note on a Western music staff

Colors were incorporated into the visualization to add interest and to aid in the recognition of notes. The seven colors of the rainbow were mapped to the seven natural notes of the scale. Table 3.1 details the color mapping chosen for this visualization. In the case of sharps and flats, both notes were displayed with their respective natural note colors, and the FFT and audio waveform were mapped to the middle value in between the two colors of the corresponding two natural notes. Figure 3.1 shows the visualization displaying a sharp/flat note.

Table 3.1. Note to Color Mapping

Note Name	Color
C	Yellow
C♯ / D♭	Yellow–Green
D	Green
D♯ / E♭	Green–Blue
E	Blue
F	Indigo
F♯ / G♭	Indigo–Violet
G	Violet
G♯ / A♭	Violet–Red
A	Red
A♯ / B♭	Red–Orange
B	Orange

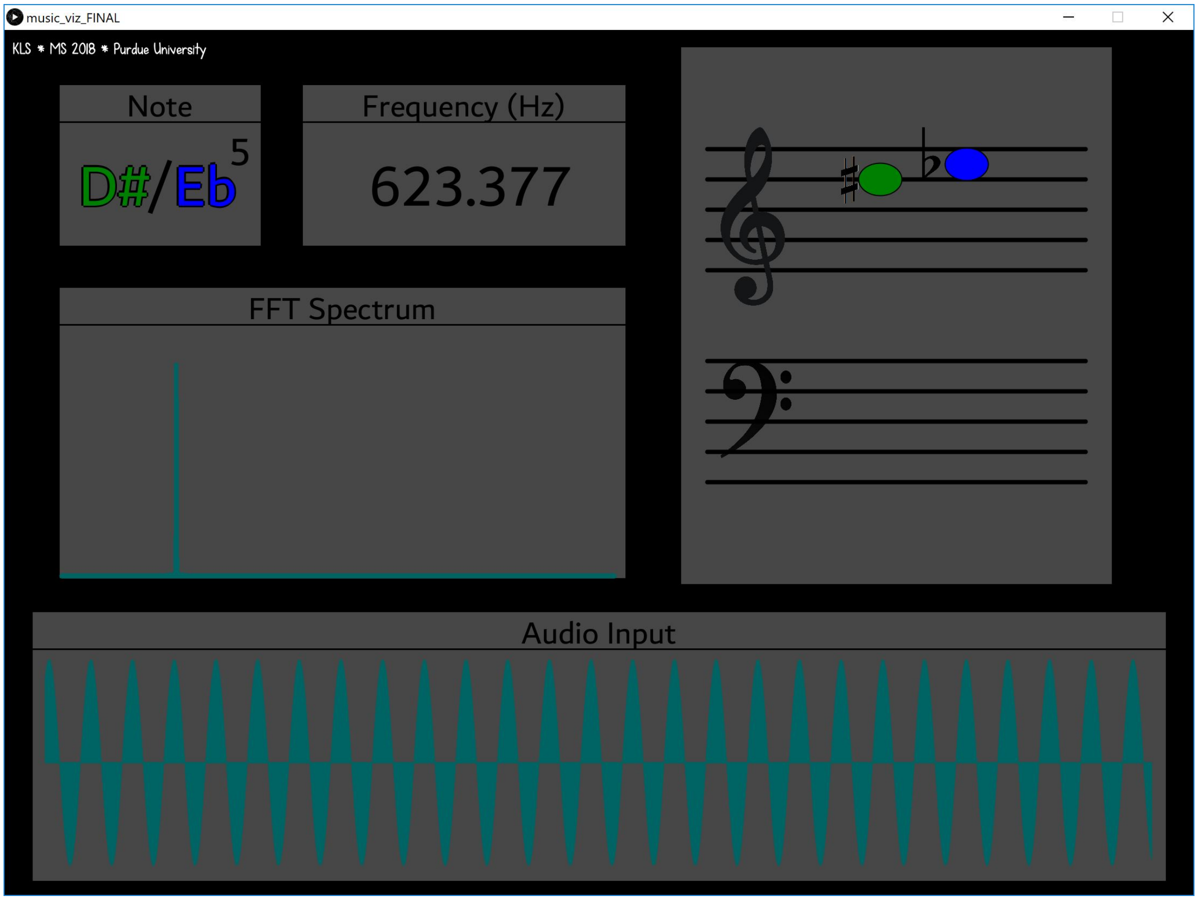


Figure 3.1. The created visualization with a sine wave input.

3.6 Pitch Detection Strategy

Autocorrelation was chosen as the method of pitch detection in this study. Singh and Kumar (2014) found the autocorrelation algorithm to be better than both AMDF and cepstrum methods. When choosing the sampling frequency and window size, consideration was given to the trade-off between resolution and time (McLeod, 2008). If the window was kept large, the accuracy would increase; but, so would the time it took to process the audio. If the window was kept small, the accuracy would decrease, along with the processing time. Given that this visualization was to be run in real time, the window length was kept small.

A table of reference pitches for each note was generated using the following formula and a reference pitch of A_4 = 440.0 Hz (French, 2009). This table can be seen in the appendix table A.1.

$$f_n = f_0 * r^n, \qquad where\ r = 2^{\frac{1}{12}}\ and\ f_0 = 440.0\ Hz \tag{3.1}$$

3.7 Test Methodology

There were two stages of testing for this study:

1. MATLAB was used to generate sine waves accurate to 0.01 Hz. One sine wave was generated for each of the 47 notes in the guitar range. These sine waves were saved as .wav files and then fed into the visualization. The .wav format was chosen because it is an uncompressed audio file format. The output values from the visualization were recorded and analyzed to determine accuracy.

2. A guitar was tuned with the PolyTune® 2 tuning pedal. The guitar was then connected to the visualizer using the *Rocksmith®* cable. Each open string was plucked individually using a pick, and the output values from the visualization were recorded and analyzed to determine accuracy.

During the two stages of testing, both the note name and note transcription were observed to determine correctness. The data for both the note name and note transcription was binary; values were either yes (correct) or no (incorrect).

3.8 Summary

Chapter three provided the framework and methodology that were used in the research study. It also detailed the design of the visualization, the strategy and method of pitch detection chosen, and the testing methodology.

CHAPTER 4. RESULTS

Chapter four provides the results of the research study.

4.1 Pitch Detection

Autocorrelation was a respectable choice for the pitch detection method, but it had several limiting factors. The *Rocksmith®* 1/4-inch to USB cable was limited to a sampling frequency of 48kHz (Ubisoft, 2012), so the sampling rate of the whole system was limited to 48kHz. Given that the visualization needed to run in real time, the buffer length (or window size) of the audio samples was kept to $N = 16384$. This resulted in frequency resolution of:

$$\Delta f = \frac{48000\ Hz}{16384} = 2.93\ Hz \tag{4.1}$$

The fundamental frequency (pitch) values were not always stable in the visualization window; in these cases an average of the repeating values was taken. The fundamental frequency (pitch) values did not always meet the desired ±2¢ accuracy. Table 4.1 contains the results from open string live guitar testing. While the desired ±2¢ accuracy was not always met, the data showed that the detected pitch values were all accurate to at least ±8¢ over the range. Interestingly, the testing done with the MATLAB generated sine waves produced data that showed that the pitch values were not quite so accurate. The sine wave detected pitch values were accurate to at least ±18¢ over the full range of the guitar. Figure in the appendix shows how sporadic the accuracy values were for sine waves across the guitar range.

Table 4.1. Live Guitar Open String Testing Data - Pitch (f_0)

		Live Guitar Test 1			Live Guitar Test 2			Live Guitar Test 3		
Note Name	**Calculated f_0 (Hz)**	**Measured f_0 (Hz)**	**% Error**	**Accuracy Range (¢)**	**Measured f_0 (Hz)**	**% Error**	**Accuracy Range (¢)**	**Measured f_0 (Hz)**	**% Error**	**Accuracy Range (¢)**
E_2	82.407	82.333	0.0897	+2	82.616	0.2538	+6	82.474	0.0814	+2
A_2	110.000	110.092	0.0836	+2	110.092	0.0836	+2	110.092	0.0836	+2
D_3	146.832	146.789	0.0295	−2	146.789	0.0295	−2	147.239	0.2769	+6
G_3	195.998	196.721	0.3690	+8	195.918	0.0407	+2	196.721	0.3690	+8
B_3	246.942	247.423	0.1949	+4	246.517	0.1720	−4	247.423	0.1949	+4
E_4	329.628	331.034	0.4267	+8	331.034	0.4267	+8	331.034	0.4267	+8

4.2 Note Name and Transcription

All notes were correctly named and transcribed in both tests. Table 4.2 shows the open string test results. Given that the note names and associated locations were based upon the fundamental frequency value, as long as the frequency was within the calculated ±50¢ range for the note, the note was named and transcribed correctly. Figure 4.1 shows the visualization with a live guitar input. The note G_3 was correctly named and transcribed, and the pitch was displayed with an accuracy of ±2¢. In addition to the note information, the visualization displayed an FFT of the audio spectrum and the raw audio input with the Hamming window applied.

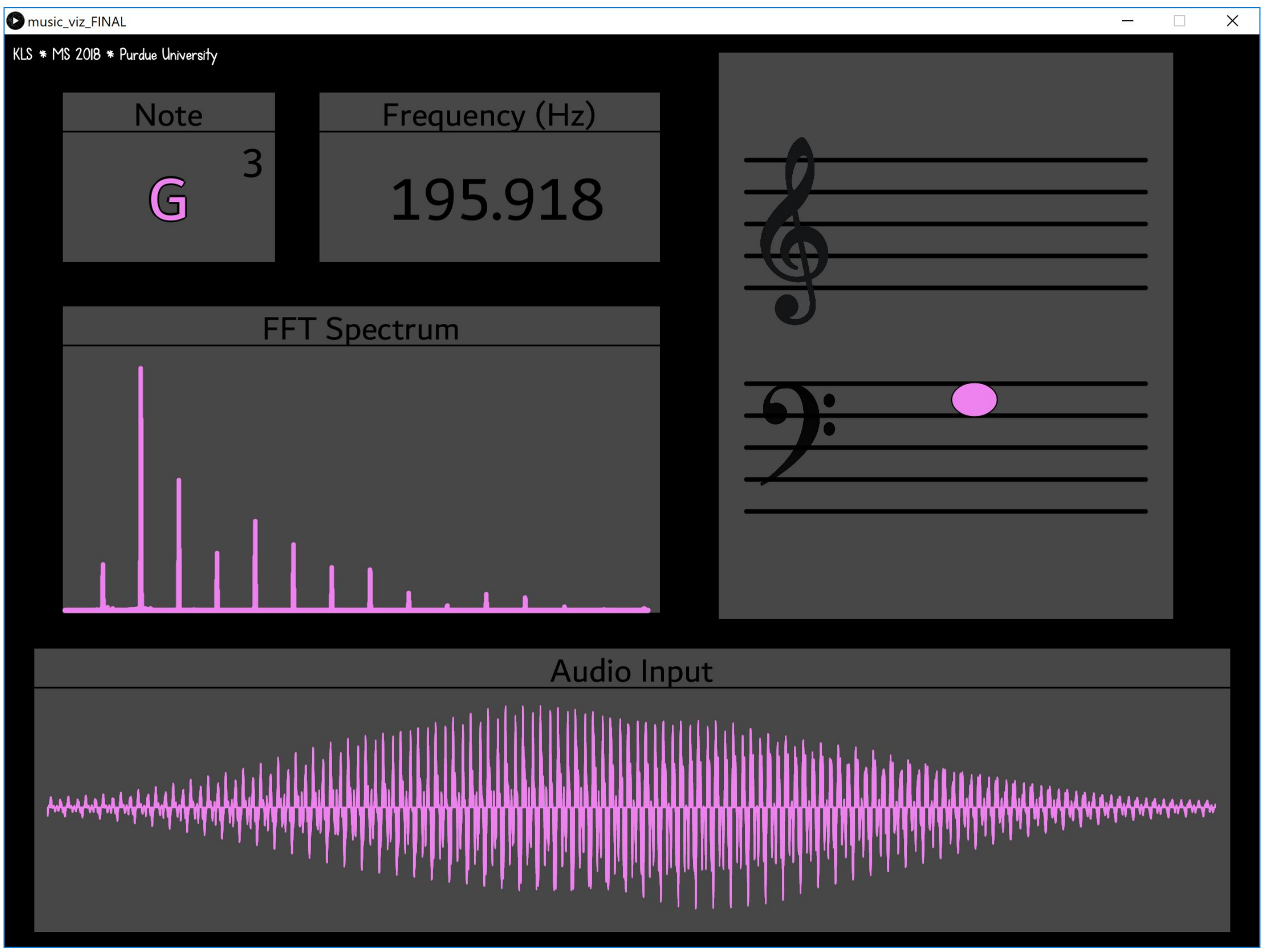


Figure 4.1. The created visualization with a live guitar input. Audio input shown with the Hamming window applied.

Table 4.2. Live Guitar Open String Testing Data - Notes

		Live Guitar Test 1			**Live Guitar Test 2**			**Live Guitar Test 3**		
Note Name	**Calculated f_0 (Hz)**	**Correct Name**	**Correct Position**	**Correct Color**	**Correct Name**	**Correct Position**	**Correct Color**	**Correct Name**	**Correct Position**	**Correct Color**
E_2	82.407	yes	yes	yes	yes	yes	yes	yes	yes	yes
A_2	110.000	yes	yes	yes	yes	yes	yes	yes	yes	yes
D_3	146.832	yes	yes	yes	yes	yes	yes	yes	yes	yes
G_3	195.998	yes	yes	yes	yes	yes	yes	yes	yes	yes
B_3	246.942	yes	yes	yes	yes	yes	yes	yes	yes	yes
E_4	329.628	yes	yes	yes	yes	yes	yes	yes	yes	yes

4.3 Overall Outcome

This study confirmed one hypothesis and rejected the other. The accepted hypotheses for this study were the following:

$H1_0$: The individual note's pitch was *not* correctly identified and displayed with an accuracy of ±2¢.

$H2_\alpha$: The individual note name was correctly displayed and transcribed into a standard music notation.

4.4 Summary

Chapter four reviewed the results of the research study.

CHAPTER 5. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Chapter five provides a summary of the research, conclusions from the research, and recommendations for further research.

5.1 Summary

A visualization was created with *Processing*©*3* that could take input from a live guitar and correctly display the note name, correctly transcribe the note onto the Western music staff, display the audio input, display the FFT of the audio input, and display the pitch value, all in real time. Autocorrelation was chosen as the method of pitch detection. The pitch was not always accurate to ±2¢, but the pitch was accurate to at least ±18¢ over the full range of the guitar.

5.2 Conclusions

Processing©*3* was a suitable choice for the visualization development software. *Processing*©*3* boasts many powerful features and functions available in various libraries, as well as a large network of users with a plethora of examples and helpful forums. Autocorrelation was not the ideal choice for pitch detection, as it is unable to achieve the desired accuracy without some further processing, modifications, or increasing the processing time.

5.3 Recommendations

For future iterations of this project, it would be prudent to look into the possibility of a zoom FFT to achieve the desired ±2¢ accuracy. However, given the limitations of the FFT, it may also be possible to achieve more accurate results if the FFT is not included in the visualization. Some other improvements to the visualization could include the ability to choose a reference tuning pitch, various selectable “modes” that could gear the visualizations towards a certain aspect of music, the ability to cover ranges of other instruments, and a recording feature. The creation of selectable “modes” for the visualization could be useful because it is difficult to create a visualization which can clearly and cleanly visualize all technical aspects of the music while still being visually entertaining and fun to use.

5.4 Summary

Chapter five provided a summary of the research, conclusions from the research, and recommendations for further research.

LIST OF REFERENCES

Bain, M. N. (2008). Real time music visualization: A study in the visual extension of music (Master's thesis, The Ohio State University). *ProQuest Theses and Dissertations*. Retrieved from https://etd.ohiolink.edu/rws_etd/document/get/osu1213207395/inline

Bent, I., Hughes, D., Provine, R., Rastall, R., Kilmer, A., Hiley, D., ... Chew, G. (n.d.). Notation. *Grove Music Online*. Retrieved from http://www.oxfordmusiconline.com/subscriber/article/grove/music/20114

Blinddog. (2012, July 17). *Atari helps you see the music.* Retrieved from http://www.retroist.com/2012/07/17/atari-helps-you-see-the-music/

Brougher, K., & Mattis, O. (2005). *Visual music: Synaesthesia in art and music since 1900.* Thames & Hudson.

Brown, R. (1976, March). *Audio activated video display.* Google Patents. Retrieved from https://www.google.com/patents/US4081829 (US Patent 4,081,829)

Cantareira, G. D., Nonato, L. G., & Paulovich, F. V. (2016, Nov). MoshViz: A detail + overview approach to visualize music elements. *IEEE Transactions on Multimedia*, *18*(11), 2238–2246. doi: 10.1109/TMM.2016.2614226

Chasin, M. (2003, July). Music and hearing aids. *Hearing Journal*, *56*(7), 36–41. doi: 10.1097/01.HJ.0000292553.60032.c2

Collopy, F. (2004, October 19). *Color scales?* Retrieved from http://rhythmiclight.com/archives/ideas/colorscales.html

Cousins, C. (2012, June 11). *Color and cultural design considerations.* Retrieved from https://www.webdesignerdepot.com/2012/06/color-and-cultural-design-considerations

deCheveigné, A., & Kawahara, H. (2002, April). YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, *111*(4), 1917–1930. Retrieved from http://audition.ens.fr/adc/pdf/2002_JASA_YIN.pdf doi: 10.1121/1.1458024

DePrisco, R., Malandrino, D., Pirozzi, D., Zaccagnino, G., & Zaccagnino, R. (2017). Understanding the structure of musical compositions: Is visualization an effective approach? *Information Visualization*, *16*(2), 139–152. Retrieved from https://doi.org/10.1177/1473871616655468 doi: 10.1177/1473871616655468

Fantasia (1940). (2015, September 29). Retrieved from https://disneymoviesfanatic.wordpress.com/2015/09/29/fantasia-1940/

Forsberg, G. (2009). An audio-to-midi application in java (Master's thesis, Luleå University of Technology). *DiVA portal*. Retrieved from http://ltu.diva-portal.org/smash/get/diva2:1023441/FULLTEXT01.pdf (ISRN: LTU-EX–09/073–SE)

Fourney, D. W., & Fels, D. I. (2009, Sept). Creating access to music through visualization. In *Proceedings of the 2009 IEEE international conference on science and technology for humanity (TIC-STH)* (pp. 939–944).

Frantz, A. (n.d.). *Difference between pitches and notes.* Retrieved from https://www.key-notes.com/blog/difference-between-pitches-and-notes

French, R. M. (2009). *Engineering the guitar: Theory and practice*. Springer Science+Business Media, LLC. doi: 10.1007/978-0-387-74369-1

French, R. M. (2012). *Technology of the guitar*. Springer Science+Business Media, LLC. doi: 10.1007/978-1-4614-1921-1

Gerson-Kiwi, E., & Hiley, D. (n.d.). Cheironomy. *Grove Music Online*. Retrieved from http://www.oxfordmusiconline.com/subscriber/article/grove/music/05510

Griscom, W. (2014). Visualizing sound: Cross-modal mapping between music and color (Doctoral dissertation, University of California, Berkeley). *ProQuest Dissertations and Theses*. Retrieved from https://search-proquest-com.ezproxy.lib.purdue.edu/docview/1667095472?accountid=13360 (UMI No. 3686303)

Hiraga, R., Watanabe, F., & Fujishiro, I. (2002). Music learning through visualization. In *Proceedings of the second international conference on WEB delivering of music (WEDELMUSIC'02)* (pp. 101–108). doi: 10.1109/WDM.2002.1176199

Jones, T. D. (1972). *The art of light and color*. Van Nostrand Reinhold Inc.

Klemenc, B., Ciuha, P., & Solina, F. (2011, November). Educational possibilities of the project colour visualization of music. *Organizacija*, *44*(3), 67–75. Retrieved from https://search-proquest-com.ezproxy.lib.purdue.edu/docview/1323852688?accountid=13360 doi: 10.2478/v10051-011-0006-9

Lee, Y., & Fathia, R. N. (2016, October). Interactive music visualization for music player using processing. In *22nd international conference on virtual system multimedia (VSMM), Malaysia* (pp. 1–4). doi: 10.1109/VSMM.2016.7863205

Malinowski, S. (n.d.). Retrieved from http://www.musanim.com

Maltin, L. (2015, November 12). *When disney got trippy.* Retrieved from http://www.bbc.com/culture/story/20151112-when-disney-got-adult-and-trippy

McLeod, P. (2008). Fast, accurate pitch detection tools for music analysis (Doctoral dissertation, University of Otago, Dunedin). *CiteSeerX*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.231.7315 doi: 10.1.1.231.7315

Morris, P., Dalley, K., Johnston, M., Keislar, D., & Keller, J. (n.d.). Retrieved from http://www.musicnotation.org/systems

Muhammad, G. (2011, April). Extended average magnitude difference function based pitch detection. *The International Arab Journal of Information Technology*, *8*(2), 197–203. Retrieved from http://www.ccis2k.org/iajit/PDF/vol.8,no.2/12-1093.pdf

Musical symbol 2228728. (2016). Retrieved from http://clipart-library.com/clipart/8i65B9pLT.htm

Nave, R. (n.d.). *Cents and calculating cents.* Retrieved from http://hyperphysics.phy-astr.gsu.edu/hbase/Music/cents.html

Nettl, B. (2005). *The study of ethnomusicology: Thirty-one issues and concepts* (New ed.). University of Illinois Press.

O'Meara, A., Compton, K., Gluck, G., & Shevchenko, A. (n.d.). Retrieved from http://media.soundspectrum.com/images/flashplaceholder_wc.jpg

Outram, B. I. (2016, March). Synesthesia audio-visual interactive-sound and music visualization in virtual reality with orbital observation and navigation. In *Proceedings of the 2016 IEEE international workshop on mixed reality art (MRA)* (pp. 7–8).

Pon, A. (2012). Vuzik: Exploring a medium for painting music (Master's thesis, University of Calgary). *ProQuest Dissertations and Theses*. Retrieved from https://search-proquest-com.ezproxy.lib.purdue.edu/docview/1322082496?accountid=13360

Pool, B. (1993, Apr 03). *Symphony of silence interpreter uses sign language to describe music to deaf concert-goers.* Retrieved from https://search-proquest-com.ezproxy.lib.purdue.edu/docview/282028221?accountid=13360

Popova, M. (n.d.). The music animation machine. Retrieved from https://www.brainpickings.org/2010/11/09/stephen-malinowski-music-animation-machine/

The Schøyen collection series. (n.d.). Retrieved from https://www.schoyencollection.com/music-notation

Singh, C. P., & Kumar, T. K. (2014). Pitched musical instrument sounds: A comparative performance evaluation. In *Proceedings of the international conference on advances in computing, communications, and informatics (ICACCI)* (pp. 1876–1880).

Smith, S. M., & Williams, G. N. (1997, October). A visualization of music. In *Proceedings of the 8th IEEE visualization '97 conference* (pp. 499–503).

Suttoni, C. (1989). *An artist's journey: Franz liszt*. The University of Chicago Press.

tc electronic. (n.d.). *Polytune 2 blacklight.* Retrieved from https://www.tcelectronic.com/Categories/Tcelectronic/Guitar/Tuners/POLYTUNE-2-BLACKLIGHT/p/HE134#googtrans(en|en)

tc electronic. (2014, September 1). *Polytune 2 english manual.* Retrieved from http://cdn-downloads.tcelectronic.com/media/2830605/tc_electronic_polytune_2_manual_english.pdf

Thibeault, M. D. (2011). Learning from looking at sound: Using multimedia spectrograms to explore world music. *General Music Today*, *25*(1), 50–55. Retrieved from https://doi.org/10.1177/1048371311414050 doi: 10.1177/1048371311414050

Ubisoft. (2012, September 26). *Rocksmith game manual.* Retrieved from http://static9.cdn.ubi.com/customersupportfaqfiles/archived/547372_Rocksmith%20PC%20manual%20with%20Configuration_ENG.PDF

VonDemKnesebeck, A., & Zölzer, U. (2010, September). Comparison of pitch trackers for real-time guitar effects. In *Proceedings of the 13th international conference on digital audio effects (DAFx-10).*

Wolfman, U. R. (2016, February 6). Scriabins color symbolism in music. Retrieved from http://www.interlude.hk/front/scriabins-color-symbolism-music

Yang, H., Lay, Y., Liou, Y., Tsao, W., & Lin, C. (2007). Development and evaluation of computer-aided music-learning system for the hearing impaired. *Journal of Computer Assisted Learning*, *23*(6), 466–476. Retrieved from `http://onlinelibrary.wiley.com.ezproxy.lib.purdue.edu/doi/10.1111/j.1365-2729.2007.00229.x/full` doi: 10.1111/j.1365-2729.2007.00229.x

APPENDIX A. DATA TABLES

Table A.1 shows the calculated values that were used to determine whether or not the pitch was within ±2¢ of the fundamental frequency (f_0).

Table A.1.: Calculated Pitch Values

Note Name	Reference A_4 (Hz)	$r = 2^{(1/12)}$	n	f_0 (Hz)	f_0–2¢ (Hz)	f_0+2¢ (Hz)
E_2	440.0	1.05946	–29	82.407	82.312	82.502
F_2	440.0	1.05946	–28	87.307	87.206	87.408
$F\sharp_2$ / $G\flat_2$	440.0	1.05946	–27	92.499	92.392	92.606
G_2	440.0	1.05946	–26	97.999	97.886	98.112
$G\sharp_2$ / $A\flat_2$	440.0	1.05946	–25	103.826	103.706	103.946
A_2	440.0	1.05946	–24	110.000	109.873	110.127
$A\sharp_2$ / $B\flat_2$	440.0	1.05946	–23	116.541	116.406	116.676
B_2	440.0	1.05946	–22	123.471	123.328	123.614
C_3	440.0	1.05946	–21	130.813	130.662	130.964
$C\sharp_3$ / $D\flat_3$	440.0	1.05946	–20	138.591	138.431	138.752
D_3	440.0	1.05946	–19	146.832	146.663	147.002
$D\sharp_3$ / $E\flat_3$	440.0	1.05946	–18	155.563	155.384	155.743
E_3	440.0	1.05946	–17	164.814	164.623	165.004
F_3	440.0	1.05946	–16	174.614	174.413	174.816
$F\sharp_3$ / $G\flat_3$	440.0	1.05946	–15	184.997	184.784	185.211
G_3	440.0	1.05946	–14	195.998	195.771	196.224
$G\sharp_3$ / $A\flat_3$	440.0	1.05946	–13	207.652	207.413	207.892
A_3	440.0	1.05946	–12	220.000	219.746	220.254
$A\sharp_3$ / $B\flat_3$	440.0	1.05946	–11	233.082	232.813	233.351

continued on next page

Table A.1.: *continued*

Note Name	**Reference A_4 (Hz)**	$r = 2^{(1/12)}$	**n**	f_0 **(Hz)**	f_0**–2¢ (Hz)**	f_0**+2¢ (Hz)**
B_3	440.0	1.05946	–10	246.942	246.657	247.227
C_4	440.0	1.05946	–9	261.626	261.323	261.928
$C\sharp_4$ / $D\flat_4$	440.0	1.05946	–8	277.183	276.863	277.503
D_4	440.0	1.05946	–7	293.665	293.326	294.004
$D\sharp_4$ / $E\flat_4$	440.0	1.05946	–6	311.127	310.768	311.487
E_4	440.0	1.05946	–5	329.628	329.247	330.009
F_4	440.0	1.05946	–4	349.228	348.825	349.632
$F\sharp_4$ / $G\flat_4$	440.0	1.05946	–3	369.994	369.567	370.422
G_4	440.0	1.05946	–2	391.995	391.543	392.449
$G\sharp_4$ / $A\flat_4$	440.0	1.05946	–1	415.305	414.825	415.785
A_4	440.0	1.05946	0	440.000	439.492	440.509
$A\sharp_4$ / $B\flat_4$	440.0	1.05946	1	466.164	465.626	466.703
B_4	440.0	1.05946	2	493.883	493.313	494.454
C_5	440.0	1.05946	3	523.251	522.647	523.856
$C\sharp_5$ / $D\flat_5$	440.0	1.05946	4	554.365	553.725	555.006
D_5	440.0	1.05946	5	587.330	586.651	588.008
$D\sharp_5$ / $E\flat_5$	440.0	1.05946	6	622.254	621.536	622.973
E_5	440.0	1.05946	7	659.255	658.494	660.017
F_5	440.0	1.05946	8	698.456	697.650	699.264
$F\sharp_5$ / $G\flat_5$	440.0	1.05946	9	739.989	739.134	740.844
G_5	440.0	1.05946	10	783.991	783.086	784.897
$G\sharp_5$ / $A\flat_5$	440.0	1.05946	11	830.609	829.650	831.570
A_5	440.0	1.05946	12	880.000	878.984	881.017
$A\sharp_5$ / $B\flat_5$	440.0	1.05946	13	932.328	931.251	933.405
B_5	440.0	1.05946	14	987.767	986.626	988.908
C_6	440.0	1.05946	15	1046.502	1045.294	1047.712

continued on next page

Table A.1.: *continued*

Note Name	**Reference A_4 (Hz)**	$r = 2^{(1/12)}$	**n**	**f_0 (Hz)**	**f_0–2¢ (Hz)**	**f_0+2¢ (Hz)**
$C\sharp_6$ / $D\flat_6$	440.0	1.05946	16	1108.731	1107.450	1110.012
D_6	440.0	1.05946	17	1174.659	1173.303	1176.017

Note	Min Hz (-50c)	Freq -18c	Freq -16c	Freq -14c	Freq -12c	Freq -10c	Freq -8c	Freq -6c	Freq -4c	Freq -2c	Measured Frequency	Freq +2c	Freq +4c	Freq +6c	Freq +8c	Freq +10c	Freq +12c	Freq +14c	Freq +16c	Freq +18c	Freq +20c	Max Hz (+49c)
E_2	80.061	81.555	81.649	81.743	81.838	81.932	82.027	82.122	82.217	82.312	**82.474**	82.502	82.598	82.693	82.789	82.884	82.980	83.076	83.172	83.268	83.364	84.773
F_2	84.822	86.404	86.504	86.604	86.704	86.804	86.905	87.005	87.106	87.206	**88.2**	87.408	87.509	87.610	87.711	87.813	87.914	88.016	88.118	88.220	88.322	89.813
$F^{\#}_2/G^{b}_2$	89.865	91.542	91.648	91.754	91.860	91.966	92.072	92.179	92.285	92.392	**93.2**	92.606	92.713	92.820	92.927	93.034	93.142	93.250	93.357	93.465	93.573	95.154
G_2	95.209	96.985	97.097	97.210	97.322	97.434	97.547	97.660	97.773	97.886	**98.1**	98.112	98.226	98.339	98.453	98.567	98.680	98.795	98.909	99.023	99.138	100.812
$G^{\#}_2/A^{b}_2$	100.870	102.752	102.871	102.990	103.109	103.228	103.348	103.467	103.587	103.706	**104.5**	103.946	104.066	104.187	104.307	104.428	104.548	104.669	104.790	104.911	105.033	106.807
A_2	106.869	108.862	108.988	109.114	109.240	109.366	109.493	109.619	109.746	109.873	**110.09**	110.127	110.254	110.382	110.509	110.637	110.765	110.893	111.021	111.150	111.278	113.158
$A^{\#}_2/B^{b}_2$	113.223	115.336	115.469	115.602	115.736	115.870	116.004	116.138	116.272	116.406	**116.7**	116.676	116.811	116.946	117.081	117.216	117.352	117.487	117.623	117.759	117.895	119.887
B_2	119.956	122.194	122.335	122.476	122.618	122.760	122.902	123.044	123.186	123.328	**123.7**	123.614	123.756	123.899	124.043	124.186	124.330	124.473	124.617	124.761	124.905	127.015
C_3	127.089	129.460	129.609	129.759	129.909	130.059	130.210	130.360	130.511	130.662	**130.7**	130.964	131.115	131.267	131.419	131.571	131.723	131.875	132.027	132.180	132.333	134.568
$C^{\#}_3/D^{b}_3$	134.646	137.158	137.316	137.475	137.634	137.793	137.952	138.112	138.271	138.431	**139.5**	138.752	138.912	139.072	139.233	139.394	139.555	139.717	139.878	140.040	140.202	142.570
D_3	142.652	145.314	145.482	145.650	145.818	145.987	146.155	146.324	146.494	146.663	**147.239**	147.002	147.172	147.342	147.512	147.683	147.854	148.025	148.196	148.367	148.538	151.048
$D^{\#}_3/E^{b}_3$	151.135	153.954	154.132	154.311	154.489	154.668	154.846	155.025	155.204	155.384	**155.844**	155.743	155.923	156.104	156.284	156.465	156.646	156.827	157.008	157.189	157.371	160.029
E_3	160.122	163.109	163.298	163.486	163.675	163.865	164.054	164.244	164.433	164.623	**165.517**	165.004	165.195	165.386	165.577	165.769	165.960	166.152	166.344	166.536	166.729	169.545
F_3	169.643	172.808	173.008	173.208	173.408	173.608	173.809	174.010	174.211	174.413	**174.545**	174.816	175.018	175.220	175.423	175.626	175.829	176.032	176.235	176.439	176.643	179.627
$F^{\#}_3/G^{b}_3$	179.731	183.084	183.295	183.507	183.719	183.932	184.144	184.357	184.570	184.784	**186.047**	185.211	185.425	185.639	185.854	186.069	186.284	186.499	186.715	186.931	187.147	190.308
G_3	190.418	193.970	194.195	194.419	194.644	194.869	195.094	195.320	195.545	195.771	**196.721**	196.224	196.451	196.678	196.906	197.133	197.361	197.589	197.818	198.046	198.275	201.624
$G^{\#}_3/A^{b}_3$	201.741	205.505	205.742	205.980	206.218	206.456	206.695	206.934	207.173	207.413	**207.792**	207.892	208.133	208.373	208.614	208.855	209.097	209.338	209.580	209.823	210.065	213.614
A_3	213.737	217.724	217.976	218.228	218.480	218.733	218.986	219.239	219.492	219.746	**220.183**	220.254	220.509	220.764	221.019	221.274	221.530	221.786	222.043	222.299	222.556	226.316
$A^{\#}_3/B^{b}_3$	226.446	230.671	230.938	231.205	231.472	231.739	232.007	232.275	232.544	232.813	**234.146**	233.351	233.621	233.891	234.161	234.432	234.703	234.974	235.246	235.518	235.790	239.773
B_3	239.912	244.387	244.670	244.953	245.236	245.519	245.803	246.087	246.372	246.657	**247.423**	247.227	247.513	247.799	248.085	248.372	248.659	248.947	249.234	249.523	249.811	254.031
C_4	254.178	258.919	259.219	259.518	259.818	260.119	260.419	260.720	261.022	261.323	**262.295**	261.928	262.231	262.534	262.837	263.141	263.445	263.750	264.055	264.360	264.666	269.136
$C^{\#}_4/D^{b}_4$	269.292	274.316	274.633	274.950	275.268	275.586	275.905	276.224	276.543	276.863	**277.457**	277.503	277.824	278.145	278.466	278.788	279.111	279.433	279.756	280.080	280.403	285.140
D_4	285.305	290.627	290.963	291.300	291.636	291.973	292.311	292.649	292.987	293.326	**294.479**	294.004	294.344	294.684	295.025	295.366	295.707	296.049	296.391	296.734	297.077	302.095
$D^{\#}_4/E^{b}_4$	302.270	307.909	308.265	308.621	308.978	309.335	309.693	310.051	310.409	310.768	**311.688**	311.487	311.847	312.207	312.568	312.929	313.291	313.653	314.016	314.379	314.742	320.059
E_4	320.244	326.218	326.595	326.973	327.351	327.729	328.108	328.487	328.867	329.247	**331.034**	330.009	330.390	330.772	331.154	331.537	331.920	332.304	332.688	333.073	333.458	339.090
F_4	339.286	345.616	346.016	346.416	346.816	347.217	347.618	348.020	348.422	348.825	**350.365**	349.632	350.036	350.441	350.846	351.251	351.657	352.064	352.471	352.878	353.286	359.254
$F^{\#}_4/G^{b}_4$	359.461	366.167	366.591	367.014	367.439	367.863	368.289	368.714	369.141	369.567	**369.231**	370.422	370.850	371.279	371.708	372.138	372.568	372.999	373.430	373.861	374.294	380.616
G_4	380.836	387.941	388.389	388.838	389.288	389.738	390.188	390.639	391.091	391.543	**393.443**	392.449	392.902	393.356	393.811	394.266	394.722	395.178	395.635	396.092	396.550	403.249
$G^{\#}_4/A^{b}_4$	403.482	411.009	411.484	411.960	412.436	412.913	413.390	413.868	414.346	414.825	**417.391**	415.785	416.265	416.747	417.228	417.711	418.193	418.677	419.161	419.645	420.130	427.227
A_4	427.474	435.449	435.952	436.456	436.961	437.466	437.971	438.478	438.985	439.492	**440.367**	440.509	441.018	441.528	442.038	442.549	443.060	443.573	444.085	444.599	445.113	452.631
$A^{\#}_4/B^{b}_4$	452.893	461.342	461.875	462.409	462.944	463.479	464.015	464.551	465.088	465.626	**466.019**	466.703	467.242	467.782	468.323	468.864	469.406	469.949	470.492	471.036	471.580	479.546
B_4	479.823	488.775	489.340	489.906	490.472	491.039	491.606	492.175	492.744	493.313	**494.845**	494.454	495.026	495.598	496.171	496.744	497.319	497.893	498.469	499.045	499.622	508.062
C_5	508.355	517.839	518.438	519.037	519.637	520.237	520.839	521.441	522.044	522.647	**521.739**	523.856	524.461	525.068	525.675	526.282	526.891	527.500	528.109	528.720	529.331	538.273
$C^{\#}_5/D^{b}_5$	538.584	548.631	549.265	549.900	550.536	551.172	551.809	552.447	553.086	553.725	**558.140**	555.006	555.648	556.290	556.933	557.577	558.221	558.866	559.512	560.159	560.807	570.280
D_5	570.609	581.255	581.926	582.599	583.273	583.947	584.622	585.298	585.974	586.651	**585.366**	588.008	588.688	589.369	590.050	590.732	591.415	592.098	592.783	593.468	594.154	604.191
$D^{\#}_5/E^{b}_5$	604.540	615.818	616.530	617.242	617.956	618.670	619.385	620.101	620.818	621.536	**623.377**	622.973	623.693	624.414	625.136	625.859	626.582	627.306	628.031	628.757	629.484	640.118
E_5	640.487	652.436	653.190	653.945	654.701	655.458	656.216	656.974	657.734	658.494	**657.534**	660.017	660.780	661.544	662.309	663.074	663.841	664.608	665.376	666.145	666.915	678.181
F_5	678.573	691.232	692.031	692.831	693.632	694.434	695.236	696.040	696.845	697.650	**695.652**	699.264	700.072	700.881	701.691	702.503	703.315	704.128	704.941	705.756	706.572	718.508
$F^{\#}_5/G^{b}_5$	718.923	732.335	733.181	734.029	734.877	735.727	736.577	737.429	738.281	739.134	**738.462**	740.844	741.701	742.558	743.416	744.276	745.136	745.997	746.859	747.723	748.587	761.232
G_5	761.672	775.882	776.779	777.677	778.575	779.475	780.376	781.278	782.182	783.086	**786.885**	784.897	785.804	786.713	787.622	788.532	789.444	790.356	791.270	792.185	793.100	806.498
$G^{\#}_5/A^{b}_5$	806.964	822.018	822.968	823.920	824.872	825.825	826.780	827.736	828.692	829.650	**827.586**	831.570	832.531	833.493	834.457	835.421	836.387	837.354	838.321	839.290	840.261	854.454
A_5	854.948	870.898	871.905	872.912	873.921	874.932	875.943	876.955	877.969	878.984	**888.889**	881.017	882.036	883.055	884.076	885.098	886.121	887.145	888.171	889.197	890.225	905.263
$A^{\#}_5/B^{b}_5$	905.786	922.684	923.751	924.818	925.887	926.958	928.029	929.102	930.176	931.251	**941.176**	933.405	934.484	935.564	936.646	937.728	938.812	939.898	940.984	942.072	943.161	959.093
B_5	959.647	977.550	978.680	979.811	980.944	982.077	983.213	984.349	985.487	986.626	**979.592**	988.908	990.051	991.196	992.342	993.489	994.637	995.787	996.938	998.090	999.244	1016.123
C_6	1016.710	1035.678	1036.875	1038.074	1039.274	1040.475	1041.678	1042.882	1044.087	1045.294	**1043.478**	1047.712	1048.923	1050.135	1051.349	1052.565	1053.781	1054.999	1056.219	1057.440	1058.662	1076.545
$C^{\#}_6/D^{b}_6$	1077.167	1097.263	1098.531	1099.801	1101.072	1102.345	1103.619	1104.895	1106.172	1107.450	**1116.279**	1110.012	1111.295	1112.580	1113.866	1115.153	1116.442	1117.733	1119.025	1120.318	1121.613	1140.560
D_6	1141.219	1162.509	1163.853	1165.198	1166.545	1167.894	1169.244	1170.595	1171.948	1173.303	**1170.732**	1176.017	1177.376	1178.737	1180.100	1181.464	1182.829	1184.197	1185.566	1186.936	1188.308	1208.381

Figure A.1. Data collected from testing with a MATLAB generated sine wave. The green cells show the distribution of cent accuracy.

APPENDIX B. TC ELECTRONIC® POLYTUNE®2

The following pages are excerpts from the *tc electronic*® PolyTune® 2 user's manual (tc electronic, 2014), stating the accuracy of the tuner. Details about the bypass switch are also provided. The tuner can be seen in figure B.1.

Figure B.1. The *tc electronic*® PolyTune® 2 Blacklight (tc electronic, n.d.).

Introduction

PolyTune 2: Polyphonic tuning revisited

As the world's first polyphonic tuner, the original PolyTune took the hearts of guitarists by storm. Features such as the MonoPoly technology (which automatically detects whether you want to tune a single string or all strings) made tuning a bass or guitar faster and easier than ever before.

But at TC Electronic, we are all about moving forward, and so the obvious question was: How can we improve on perfection?

Enter PolyTune 2.

Brighter than a thousand suns

When you need to tune, you need to tune. And the last thing you want to worry about at that moment is a display that is either too bright for the club stage or unusable in the bright sun of a late afternoon gig. The display on PolyTune 2 is primed with some of the brightest LEDs you have ever laid your eyes on. And the ambient light sensor makes sure you get just the right amount of brightness. It's the best of/for both worlds!

Strobe tuning

TC has received many requests from the guitar community for a strobe tuner, so we added a strobe mode, which is both lightning fast and ultra precise. And with a pitch detection accuracy of **±0.1 Cent,** (that's 1/1000 of a semitone!) this is the right tool for fine-tuning your precious instrument – wherever you may be.

Total recall

PolyTune 2 stores your preferences. From pitch reference to selected tuning mode, it hangs on to this information even after it is powered down – making sure you only have to set these parameters once. And just to be safe, it will display the current settings when you plug in your instrument. One less thing to worry about!

Of course, PolyTune 2 still has all the features from the original PolyTune that users know and love.

- PolyTune®: Tune all strings simultaneously
- For guitars and basses
- Supports Drop-D and capo tuning modes
- True bypass with silent tuning
- DC output for powering other pedals

PolyTune 2 will allow you to tune your instrument faster and easier than ever before, so you can go back to doing the one thing we know you care about: playing your music.

Enjoy!

True bypass

Here at TC, we have a simply philosophy: When you are using one of our products, you should hear something great – and if you don't, you shouldn't hear it at all. This is why this pedal sports True Bypass. When it is bypassed, it is really off and has zero influence on your tone, resulting in optimum clarity and zero loss of high-end. Also please note that the pedal lets your dry, unprocessed sound pass without ever converting it to digital – keeping your original tone pure and without any latency.

Changing the battery

If you need to change the battery of your pedal, proceed as follows:

- Unscrew the thumb-screw on the back of the pedal and detach the back-plate.
- Unmount the old battery and attach the new battery to the battery clip making sure the polarity is correct.
- Remount the back-plate.

Notes regarding batteries

- Batteries must never be heated, taken apart or thrown into fire or water.
- Only rechargeable batteries can be recharged.
- Remove the battery when the pedal is not being used for a longer period of time to save battery life.
- Dispose batteries according to local laws and regulations.

Technical specifications

- Input Connector Type:
  1 Standard ¼" jack – mono/TS
- Output Connector Type:
  1 Standard ¼" jack – mono/TS
- Tuning Range: A0 (27.5 Hz) to C8 (4186 Hz)
- Tuning Accuracy: ±0.1 cent
- Reference Pitch: A4 = 435 to 445 Hz (1 Hz steps)
- Input Impedance: 500 kΩ (pedal on)
- Power Input: Standard 9V DC, centre negative >100 mA (power supply not included)
- Battery option: Standard 9V (battery not included)
- Current Draw: 45 to 50 mA (typical use)
- Dimensions (W x D x H):
  72 x 122 x 50 mm / 2.8 x 4.8 x 2.0"

- Weight: 300 g / 10.6 oz (incl. battery)

  Due to continuous development, these specifications are subject to change without notice.

Getting support

If you still have questions about the product after reading this manual, please get in touch with **TC Support:**

http://tcelectronic.com/support/

APPENDIX C. UBISOFT ROCKSMITH®

The following pages are excerpts from the Ubisoft *Rocksmith*® PC user's manual (Ubisoft, 2012). The manual states the system requirements and the sampling frequency for the 1/4-inch to USB interface cable.

Rocksmith PC
Configuration and FAQ

September 26, 2012

Contents:

Rocksmith Minimum Specs

Rocksmith is built to run well on a majority of PCs released in the last 2-3 years. Our minimum PC specification is:

- Intel Core2Duo E4400 @ 2.00 Ghz or AMD Athlon64 3800+ @ 2.0 Ghz
- 2 GB
- 256MB NVIDIA GeForce 8600GT or AMD Radeon HD 2600XT video card
- USB 2.0
- Windows Vista or Windows 7

Our recommended PC specification is:

- Intel Core2Duo E6750 @ 2.6GHz or AMD Athlon 64 X2 6000+ @ 3.0GHz
- 4 GB Memory
- 512MB NVIDIA GeForce 200-series or AMD Radeon HD 3000-series video card
- USB 2.0
- Windows Vista or Windows 7

Our PC specifications are intended as rough guideline, and are based on mainstream desktop PCs using these specifications. Very often laptop or budget PC manufacturers may carry the same specifications but won't have nearly as good performance, so your experience running Rocksmith on a laptop or budget PC may be less than optimal.

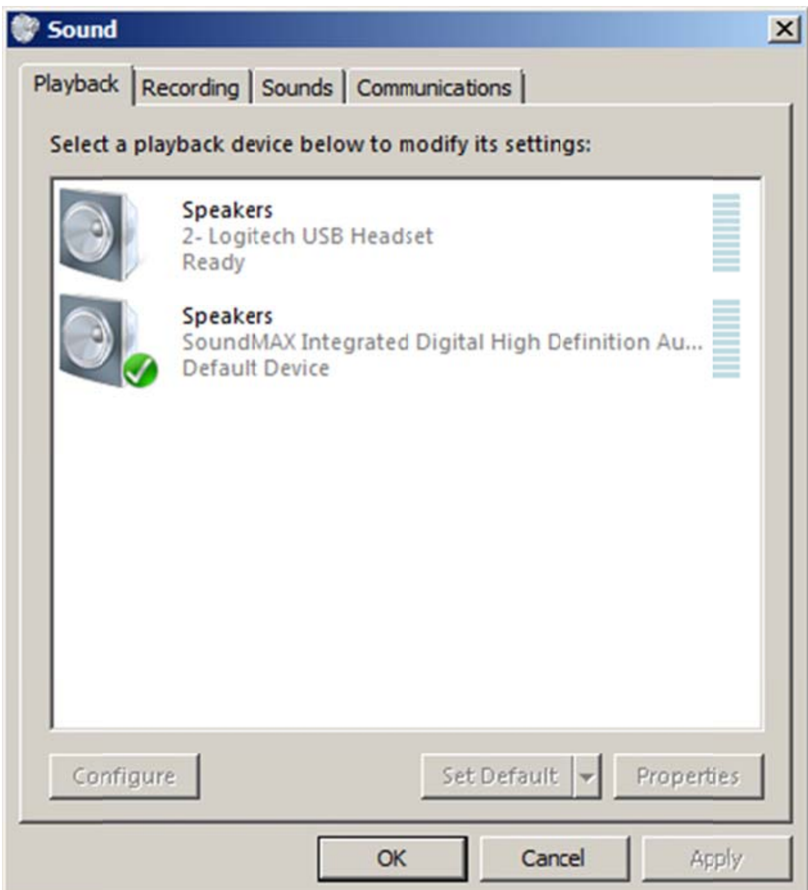
Audio Device Configuration

Proper configuration of your audio hardware is important for achieving the best performance and gameplay experience with Rocksmith on your PC. If the audio system isn't configured correctly, you may experience a significant amount of lag or latency between strumming your guitar and hearing the sound.

For best results, you'll need to configure both your output device and the Rocksmith Real Tone Cable. Both devices should be set to operate using a sample rate of 48 kHz using a 16 bit sample size. Audio output should also be set for 2 channel playback. Finally, configure both devices to allow applications to take exclusive control of the device.

Configuring Audio Output

Open your Sound control panel, and select the “Playback” tab. Though different audio drivers provide a variety of configuration software, here's an example of what it should look like:



Your desired output device should be set as the “Default Device”. Select that device and click the “Configure” button.