ColorSynth: An Audiovisual Tool for Understanding Harmonic Ratios

Ethan Rountree
Earlham College
Department of Computer Science
Richmond, Indiana
ehrount15@earlham.edu

ABSTRACT

This paper describes an experiment in visualizing musical harmony. I developed a tool that uses real-time input from a MIDI controller and creates a visualization using graphical matrices. The visualization is designed to demonstrate the harmonic quality of any played intervals through spatialization and color mapping. The color spectrum is mapped around a 12 tone chromatic scale. The frequency values and ratio between played notes are used to scale a sine function modulating color intensity over the x-axis. Horizontal motion is dictated by the complexity of the ratio between two frequencies. Two different color mappings and approaches to scaling the sine wave specialization have been implemented. Resulting visualizations of consonant and dissonant intervals from these variations was compared to evaluated whether consonance and dissonance could be easily distinguished.

KEYWORDS

Music Visualization, Video Synthesis, Harmonic Ratios

1 INTRODUCTION

This work seeks to create an effective music visualization that conveys the harmonic quality of the music being played. Understanding the harmonic qualities of different intervals is important for learning music theory, music composition[13]. Visualizer tools are often used in live performance to add an additional sensory experience. Creating music visualizations that represent the harmonic quality of the notes being played can be useful for translating musical feelings to people with hearing disabilities[9]. I have developed a patch using Max 7 and the Jitter graphics library to create a visualization of live user input from a MIDI keyboard. My program, ColorSynth, can visualize two simultaneous MIDI notes. The objective is to create a visually pleasing output that also conveys information about the quality of the played interval to viewers. ColorSynth uses color to convey tonal qualities and a sine wave based spatialization method to convey the ratio between played frequencies. In order to judge the effectiveness of my work I have developed two variations for both of those visualization parameters and will evaluate their differences in displaying the most consonant and dissonant intervals. Two color variations map notes to colors around the color spectrum; one is a chromatic mapping and the other is based on the circle of fifths. The two spatialization variations scale the size of the sine wave color intensity; one scales it to the relative size of the played frequency, the other is scaled to the interval ratio in just intonation.

The intention of this work was to create a tool that allows a user to explore the harmonic qualities of different intervals and use visual feedback to help them understand the differences between consonant and dissonant intervals. Such a tool could be useful in teaching music theory or express musical qualities to people with hearing impairments. In the final output, it should be easy to visually distinguish consonant and dissonant intervals even for users unfamiliar with music theory. Dissonant intervals should evoke tension and motion and consonant ones should evoke release and rest.

Section 2 of this paper provides relevant background information on the computing tools and music theory useful to understanding this work. Section 3 covers the design of the algorithm I used. Section 4 details the role of each module in my algorithm and its function. Section 5 defines the metrics I will be using to evaluate the results of different variations. Section 6 covers the results of this project. Section 7 is the evaluation of the results. Section 8 contains information on related works. Sections 9 and 10 are the conclusion and acknowledgements.

2 BACKGROUND

2.1 Musical Background

Harmony is the result of two or more musical tones being played together. Harmony is used to build and release tension in music, which makes it an essential component for evoking emotions and driving a piece of music forward. Musical intervals range from consonant, or pleasant, intervals to dissonant, or unpleasant. Consonance occurs when the ratio between two frequencies can be reduced to very low integer values [2]. When the frequency ratio is reducible to low intervals the two waves phase relationship has a low repeat time and results in a stable sound. For more complex ratios the repeat time is longer which makes the interval sound more unstable and less satisfying to the ear. Table 1 compares the ratios of the 12 primary intervals in western music using 7-limit just intonation, a tuning based on reducing the ratio between frequencies to ratios of numbers with factors no higher than 7 for all intervals in a chromatic scale[6][10]. The complexity value adds the number of oscillations for the two waves in one complete phase cycle. The three most consonant intervals (not including unison because unison cannot be played on a single keyboard) are the octave, perfect 5th, perfect 4th, and major 6th. The four most dissonant are the minor 2nd, major 7th, and diminished 5th and major 2nd.

One note about just intonation is that there is no single correct mapping of intervals. Dissonant intervals, by nature are composed of ratios that are not reducible to low integers, so rounding is used in 7-limit intonation. In particular, the Diminished 5th/Augmented 4th is a complex interval ratio and is so dissonant that medieval monks banned it from being used in composition[7]. Other work has

Table 1: Relative Consonant and Dissonant Values of Intervals in Just Intonnantion

Interval	Just Intonation Frequency Ratio	Complexity Value
Minor 2nd	15:14	29
Major 2nd	9:8	17
Minor 3rd	6:5	11
Major 3rd	5:4	9
Perfect 4th	4:3	7
Dim 5th	10:7	17
Perfect 5th	3:2	5
Minor 6th	8:5	13
Major 6th	5:3	8
Minor 7th	7:4	11
Major 7th	15:8	23
Octave/Perfect 8th	2:1	2

approximated the more complex ratio 45:32[13]. The ratio values for just intonation used in Lehnfeld's work were chosen as the values to use in this project because his project was the most closely related to this one, and was also done recently.

2.2 Computing Tools and Data Structures

The development environment used for this project is Max 7. Max is a GUI object-oriented programming language designed for audio processing. A program in Max is called a *patch* and consists of Max objects. Each object consists of a function and inlets and outlets which route the data to and from other objects in the program. An object's function is only called when input is received in its first inlet, and the result is passed through its outlets. Max has sophisticated scheduling capabilities for calling functions in time with a metronome. The majority of computation involved in this project involves working with graphics matrices. The Jitter library provides several objects for initiating and performing operations on graphics matrices within the Max environment [3]. Notably, the jit.matrix object can initiate a multi-plane matrix and provides several methods for setting values for the cells contained. Graphic matrices that display are typically represented using a four plane character matrix. Each plane stores the color values of the alpha, red, green, and blue channels. The jit.op object provides methods for arithmetic functions on all cells in one or more matrices. For example jit.op can be used to add all the values of corresponding cells for two matrices. This project also performs operation on MIDI (Musical Instrument Digital Interface) data. MIDI is a digital transfer protocol for musical data[8]. A MIDI device sends data to an instrument or computer. When a note is played the MIDI controller sends a note on signal as well as information about that note such as pitch value and velocity i.e. how hard the note is played. Note and velocity values range from 0-127. When the note is released a note off signal is sent indicating that the released note is no longer being played.

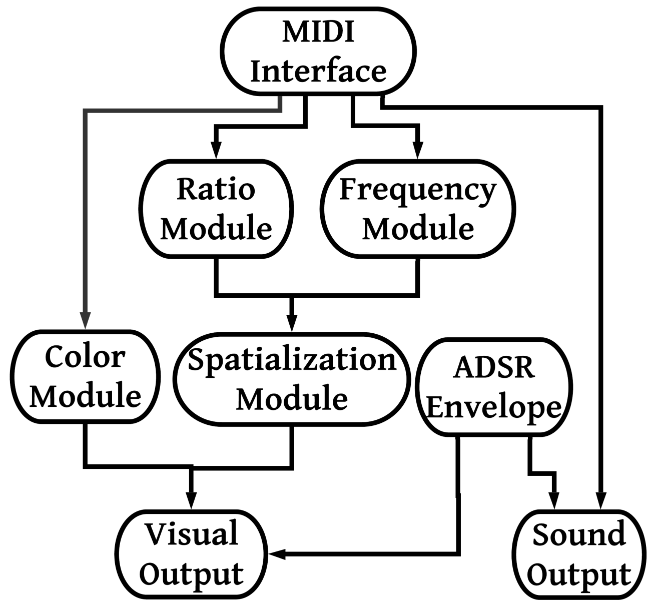

Figure 1: Design Architecture of ColorSynth

3 DESIGN

Figure 1 shows the architecture of ColorSynth and how the individual modules interact with each other. My algorithm takes input of up to 2 simultaneous MIDI notes. The scope of the project was limited to two voices to focus on visualizing the harmonic relationship of a single interval. Harmonies composed of two or more intervals can be significantly more complex than even a very dissonant two tone interval. The color module produces the color value of a given note, as defined in a color table. The spatializer creates sine wave values across the x axis which are used to determine color intensity at that point. The spatializer's sine mapping is mirrored across the y axis to create symmetry. Horizontal motion outwards from this point of origin is used to imply sound emanating from a point of origin in the middle of the visualization. The rate of horizontal motion is directly correlated to the complexity value of the two played notes resulting in a more stable visualization for consonant tones and more motion in dissonant ones. The result of the combination of the color module and the spatializer is a series of colored vertical bars. The amplitude of these bars is controlled by the ADSR envelope and the velocity of the MIDI input to allow for expressiveness both in the attack and decay time and how hard the note was played.

There are two methods for scaling the size of those sine waves, only one of which is used at a time. The first scales the sine waves relative to their frequency, so lower notes are represented with wider bars and higher notes with narrow ones. The second uses the values from a table of just intonation intervals to scale the two notes. The ratio between the sizes of the two notes is approximately the same (the difference being the amount that just intonation varies from the actual tuning system implemented in max). In the frequency-based implementation the size of each note varies based on pitch but for the ratio implementation all intervals of the same size are scaled the same regardless of position on the keyboard. The two mappings used to assign color values are based on two different principals of organization. The first is a chromatic mapping in which the closer two notes are on the keyboard are closer in color value. The second is organized based on the circle of fifths in which notes with a very consonant 3:2 ratio are closest together. The idea was that this mapping might be better at distributing color so that more consonant tones share similar color values.

3.1 Tools

The software I developed to run my experiments was created using Max 7[1]. Max/MSP is a visual programming language designed for music and multimedia projects. I also used many tools from the Jitter library which provides objects and methods for working with multi-plane graphics matrices. My project is designed to be used with a MIDI keyboard as an input device.

4 IMPLEMENTATION

4.1 MIDI Interface

The MIDI interface handles the MIDI input coming from the external keyboard device. My implementation is designed to handle two simultaneous voices. If a note on signal is pressed while there are no other notes held, that note is assigned to the first voice. If there is one other played note, the new note is assigned to voice two. Any notes played while the two voices are already on are discarded. Once assigned to a voice, the MIDI signals note values and velocity are passed to other parts of the program.

4.2 Color Module

The color module takes a MIDI note value as input and returns a 250x250 cell four plane array. First a modulo 12 operation is applied to the incoming note value. This makes it so all notes of the same tonal quality are given the same value (ie all the Cs on the keyboard will have the same output from this module). A table is referenced to return the hue value mapped to that note. The hue value is used to fill three matrices designating RGB values. A fourth matrix contains an alpha channel which is necessary for performing some jitter operations in other parts of the program but all cells in the alpha matrix are always 0.

4.3 Frequency Spatializer

The frequency spatializer contains one variation on how my project maps a note spatially. The midi note is converted into the corresponding frequency. That value is passed into the spatialization module.

4.4 Ratio Spatializer

The ratio spatializer calculates the distance between the two notes played and references a table that contains the ratios of all intervals in just intonation. The values of the numerator and demoninator of this table are then passed to the spatialization module.

Figure 2: A single note's visualization in ColorSynth

4.5 Spatialization Module

This module generates jitter matrices that contain spatial oscillations of each note. A sine function is used to fill cell values in a relative to their horizontal position. The sine function is scaled so that the number of oscillations visible is relative to the note it represents. How the sine function is scaled is determined by which specialization is being used. This sine visualization is mirrored over the y axis so the left and right half of the visualization are always symmetrical. Figure 2 shows the spatial oscillations of a single note.

4.6 ADSR Envelope

The ADSR envelope generates a value that is used to scale the volume and visual intensity of a single note relative the number of milliseconds since a note press. This module outputs a value ranging from 0 to 1 multiplied by the velocity of the note press. The output is routed to the sound output amplitude and amplitude of note visualizations. The four parameters attack, decay, sustain and release allow the user to change how this value changes over time relative to when the note-on signal is received. Attack controls the time in milliseconds after a note press that it takes the envelope to go from 0 to 1. An attack value of 0 means the note will start at full visual intensity and volume, whereas a high value means the note will fade in over time. Decay controls the time it takes to go from to the sustain value. Release controls the ramp from the current value back to 0 when the note off signal is received.

4.7 Visual Output

The Visual Output module takes input matrices from the Color Module, Ratio Spatializer, and Frequency Spatializer and combines them into the final output Matrix. The values in the matrix from the spatializer are subtracted from the values from the color module. All values is the resulting matrix are multiplied by the current output of the ADSR envelope. This process is done for both note voices. The two resulting matrices are then added together for the final output.

4.8 Sound Output

The sound output takes the frequency values of the played MIDI notes. Two sawtooth waves are generated with those frequencies. The amplitude of both voices is multiplied both by the value from the ADSR envelope and a gain control before it is output to the speakers.

Table 2: Color Mapping Variations

Note	Chromatic Color Mapping	Chromatic RGB Value	Fifths Color Mapping	Fifths RGB Value
C		1, 0, 0		1, 0, 0
C#/Db		1, 0.5, 0		0, 0.5, 1
D		1, 1, 0		1, 1, 0
D#/Eb		0.5, 1, 0		0.5, 0, 1
E		0, 1, 0		0, 1, 0
F		0, 1, 0.5		1, 0, 0.5
F#/Gb		0, 1, 1		0, 1, 1
G		0, 0.5, 1		1, 0.5, 0
G#		0, 0, 1		0, 0, 1
A		0.5, 0, 1		0.5, 1, 0
A#/Bb		1, 0, 1		1, 0, 1
B		1, 0, 0.5		0, 1, 0.5

5 EVALUATION METHODS

In order to evaluate the effectiveness of the program, we will examine the visualization in three different ways: Functionality, color mapping, and spatialization. The color mapping and spatialization sections both consist of two variations which will be compared to each other by examining the visualization output resulting from intervals of different dissonant values. In an effective visualization consonant and dissonant tones should be easily distinguishable.

5.1 Functionality

The functionality of my program is evaluated with the following metrics. The MIDI interface handles played notes without problems such as notes getting stuck on or sending the wrong note values to a voice. The user should be able to play the keyboard as they would normally play without causing issues in the visualization. The visualization runs smoothly without noticeable latency or unexpected output.

5.2 Color Variations

There are two ways in which I chose to map colors to the 12 notes of a keyboard. Table 2 shows which colors are assigned to which notes for the Chromatic and Fifths color mappings as well as the RGB float values that correspond with them. The chromatic mapping organizes the keyboard so that notes that are closer to each other are closer in color. Notes that are separated by a half step are a adjacent on the color spectrum. The fifths mapping organizes them so that notes separated by an interval of a perfect fifth are adjacent on the color spectrum; C is represented by red and G which is one fifth higher is orange. One fifth below C is F which is represented by a redish purple color.

5.3 Spatial Variations

Two separate modules contain the two variations in spatialization methods. The frequency method scales the sine wave visualization relative to the actually frequency produced my a played note. The size of a single oscillation is larger for lower notes and smaller for higher notes. An interval of one octave will display twice as many of the higher notes oscillations as the lower notes. This is similar to how how physical sound waves relate to each other. The ratio between the oscillation size of any two notes is equal to the ratio between the frequencies of the sound waves they produce. The ratio method scales it relative to the harmonic interval value from a table of just intonation intervals. The number of oscillations visible is equal to the number of oscillations it takes before the phase cycle of the two tones repeats. The ratio between the size of the two intervals is equal to the ratio of the interval in the just intonation tuning found in table 1. In this method the oscillation size of intervals with a lower complexity is larger than intervals with a higher complexity value.

6 RESULTS

6.1 Functionality Results

As of the time this draft was written most major functionality issues have been worked out however, there are still some things that need to be addressed. One such issue is the way the program operates when there is only a single note being played. Because most of the design choices were made for displaying two notes some features behave strangely without a second input note. The rate of motion is calculated from the complexity of the ratio played. When there is only one note played the motion of that note is determined based on the interval between that note and most recent one played in the second voice. Additionally, when two notes are played using the ratio spatialization method, the table will occasionally output that a ratio that is not unison is unison. More investigation is required to address this bug.

6.2 Color Mapping Results

The visualization of any interval displays a total of four different colors: black between the oscillation visual of both notes, the color of the first note, the color of the second note, and the combination of both notes colors in areas where the oscillation visuals overlap. In the two color mappings tested there were some aspects that were similar. Half of the notes are assigned the same color in both variations. The other half of notes have two different possibilities (see figure 2).

For all notes that have two possible assignments, the color in the chromatic mapping is the inverse of the color in the fifths mapping. For any note X that varies between mappings, there is a note Y 6 steps, or half the octave away, is the color of note X in the other mapping method. For example, in the chromatic mapping method C sharp is orange and G is blue but in the fifths mapping C sharp is blue and G is orange. As a result the color combinations of half of the interval combinations remain the same between mappings and the other half are an inverted version of the alternate mapping.

In both mappings an octave, the most consonant interval, results in a single color because all tones have the same color regardless of octave. A diminished fifth also results in the same color combination regardless of color mapping. This combination is shown in figure 3. For all diminished fifths the color that results from the blend between both notes is always white as the colors are the inverse of each other. For the second most consonant interval, the major fifth, the color combination varies depending on the mapping used. Using the fifths based color mapping the colors of the two notes are adjacent on the color wheel. Figure 4 shows a perfect fifth composed of C and G. C is mapped to red and G is mapped to orange. The result is a combination that is not quite monochromatic. The blend between the two colors is an orange-red that does not stand out distinctly. With the chromatic mapping C is still mapped to red and G is mapped to blue as shown in figure 5. In this mapping the two note's visualizations are much more distinct from each other and the overlap between them creates a bright purple color. A minor second, the most dissonant interval in the just intonation tuning used in this project is the inverse of the major fifth mapping so the color combinations in the fifths based mapping is the same as in figure 5 and in the chromatic mapping it is the same as figure 4. In the fifth's based mapping, a major 7th consisting of C and Bb in figure 6, a relatively dissonant interval, is represented by red and green with the combination of those two colors resulting in yellow. In the chromatic mapping of the same interval, seen in figure 7, the two colors are red and purple and the combined area does not contrast as much.

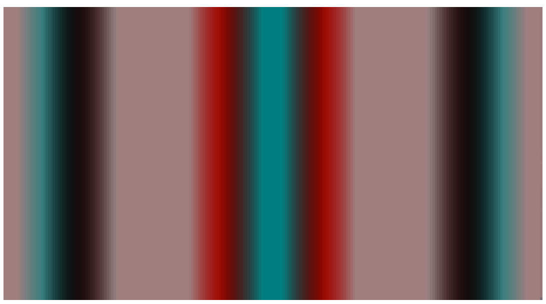

Figure 3: A diminished fifth's color combination is the same in both mappings

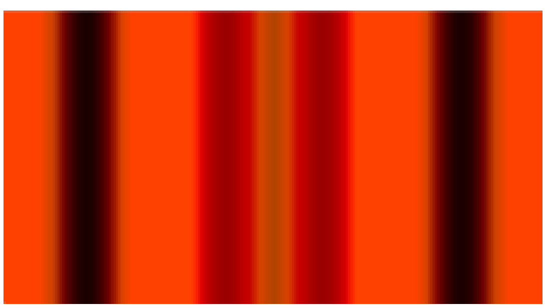

Figure 4: A perfect fifth using the fifths based color mapping

6.3 Spatial Mapping Results

The spatial mapping modules vary the number of oscillations displayed at one time and the size of each single oscillation. When the ratio based spatialization module is used, the number of oscilations shown at a time is dependant on the complexity of the phase cycle of the interval determined by the just intonation table used. All intervals of the same size are spatialized the same size making the

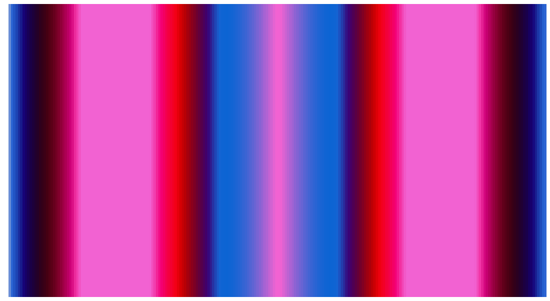

Figure 5: A perfect fifth using the chromatic based color mapping

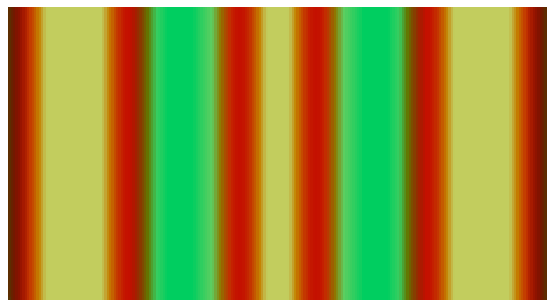

Figure 6: A major seventh using the fifths based mapping

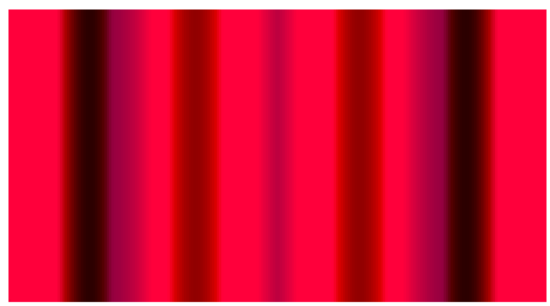

Figure 7: A major seventh using the chromatic based mapping

color the only distinguishing feature between a perfect 5th composed of a C and a G or an A and a E. In addition, the visualization of a C and a G is the same regardless of the octave it is played in. Any single note played is represented using the same spatialization with only the color to distinguish them. In the frequency spatialization module the size of each note's visualization is unique to that note and notes played in different octaves are clearly distinct. Each note's visualization is independent from the value of the other note played or if no other note is being played, whereas the ratio module is entirely dependant on the other note value. One problem with the frequency based module is sets of intervals that differ by one step are so similar to each other that it is difficult to see that they are clearly distinct. This is an issue where highly consonant intervals are only one step away from highly dissonant intervals. The spatialization of a perfect fourth is very similar to a diminished fifth using the frequency based module as can be seen in figures 8 and 10. However using the ratio based module they are clearly distinct as shown in figures 9 and 11.

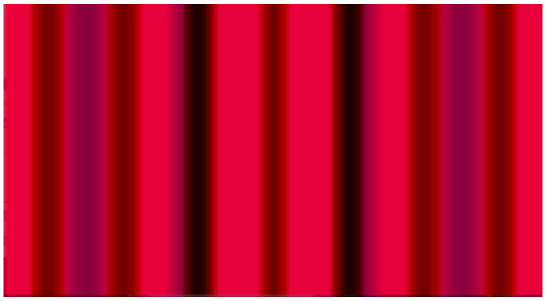

Figure 8: A perfect fourth using the frequency based spatializer

Figure 9: A perfect fourth using the ratio based spatializer

Figure 10: A diminished fifth using the frequency based spatializer

7 EVALUATION

Of the two color variations, the fifths based mapping was more effective at showing differences between consonant and dissonant intervals. In the chromatic mapping, some of the most dissonant intervals are almost the same color whereas the diminished fifth was composed of two colors from opposite ends of the spectrum. Likewise there are consonant intervals made up of very distinct colors, but the octave is the same color. Without any clear pattern

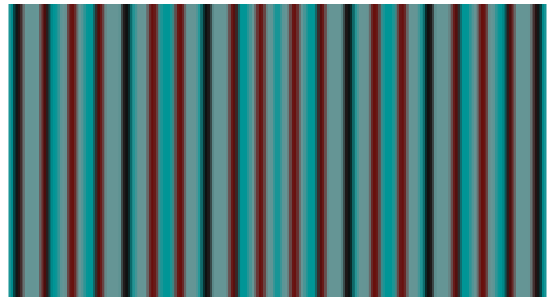

Figure 11: A diminished fifth using the ratio based spatializer

of difference between consonant and dissonant intervals, the color combinations in the chromatic mapping don't convey any meaningful information about the harmonic quality of the interval. In the fifths based mapping, there was a much more clear visual distinctness between consonant and dissonant intervals. The octave, perfect fifth and perfect fourth were all composed of very closely related colors. The minor second, major seventh and diminished fifth were all composed of two contrasting colors. The degree of similarity between the two colors did not perfectly correlate with dissonance value, for example the major third is composed of less similar colors than the major second even though it is less dissonant, however this mapping still was significantly more effective than the chromatic one.

For the spatial variations, the ratio based system performed better at showing distinct visual difference between intervals of different dissonance values. The main issue with the frequency method is for highly consonant and dissonant intervals that are only separated by a half step. The spatialization for these intervals is almost the same even though they don't share the same harmonic quality. This method is much better at showing the difference between very low and high notes. That is a valuable attribute but was not the objective of this project. The ratio method is directly derived from the harmonic value of an interval so it is directly correlated to the dissonance value of that interval. More consonant intervals always show fewer oscillations and more dissonant ones always show more. The principal drawback of this method is that the visualization of a single note looks somewhat awkward this method relies on two notes to work most effectively.

8 RELATED WORK

Related work focused on visualizing harmonic ratios was relatively sparse in so far as we have been able to determine. The most similar project working on visualizing music with attention payed to harmonic ratios was conducted by Mathias Lehnfeld. His approach does not use color and uses a visualization technique derived from oscilliscopes by assigning frequencies to the x and y axes[6]. The visualization is done using gem, a graphics tool provided by Pure Data, an open source language that shares many qualities with Max. One notably difference in his work is the absence of color as his project only visualizes music spatially.

An older music visualization project conducted by Taylor et. al. used a jitter program to modulate visual parameters on a video projection [12]. In this project they also used the mapped of circle of fifths to chromatic notes. In addition to keyboard melodic input, they also used a microphone input and the timbre of the vocalists singing modulated visual parameters. One major difference is their visualization was modulating parameters of a prerecorded video instead of generating them within their program.

Kripper developed a video synthesizer called Vsynth using Jitter tools that was a useful reference in designing my visualization[5]. Kripper's tool has a much broader range of visual possibilities but is designed to accompany music instead of represent it visually. By itself Vsynth is only a visual tool, not an audiovisual one. The STRATIC project created by Simbelis and Lundstro is another notable visual project. Their project uses both digital and analog elements to make abstract projects of colorful horizontal lines[11]. These are generated using a flashing RGB LED and modulating a camera's shutter speed. The frequency of the LED color changes were controlled by audio output from MAX/MSP. Their colorful horizontal lines were influential in the process of designing the visualization used in this project.

A study by Nanayakkara et al. examined how to express musical qualities to people with hearing impairments[9]. In addition to a visualization their work included a specially designed chair to allow the user to physically feel the vibrations. Work conducted by Kim et al. studied music visualization for children[4]. Their focus was on conveying music to children with hearing impairments and helping children learn to play music by providing real time visual feedback. Analysis conducted by Lubar uses her background as both a musician and a painter to analyze color combinations of musical intervals when notes are mapped chromatically across the color spectrum[7]. The color distribution used in her analysis is based on colors as used by painters with red, yellow, and blue being the primary colors. This makes her color mapping slightly different than the one used in this project's chromatic mapping since digital primary colors are red, green and, blue.

9 CONCLUSION

ColorSynth is a capable tool for harmony visualization. The motion speed of the visualization being mapped to ratio complexity makes for distinct differences between consonance and dissonance regardless of which variations are used. In terms of color mappings, the fifths based module performs better at showing distinctness between consonant and dissonant intervals. More consonant intervals share more closely related colors which can indicate resolution to one particular tonal color. Dissonant intervals combine colors further separated on the color wheel which provides more contrast and tension. In the chromatic module there were dissonant intervals comprised of very distinct colors as well as very similar colors. The same is true for the consonant intervals. The chromatic module does have distinct aesthetic qualities that might be preferred if the objective of the project were not to compare harmonic qualities of intervals. In terms of spatialization, the ratio based system was better than the frequency module at showing distinctness between consonant and dissonant intervals. The biggest problem with the frequency based system is the similarity between intervals that are almost the same size but have very different complexity values. However, at the time of this draft the ratio module has some functionality issues that detract from the users experience so the frequency module has the advantage of performing more smoothly overall. In addition the ratio module does not distinguish its spatialization based on lower or higher pitch which allows for more visual control on behalf of the user. If used for performance or visual composition, that might be a feature that users want for another parameter of expressiveness. In future work it would be interesting to develop a system that is not limited to two voices so any song could be played and visualized without the limitation of not being able to play chords.

10 ACKNOWLEDGEMENTS

I would like thank all the people who helped me in the process of executing and researching this project. Xunfei Jiang and David Barbella in the Earlham College Computer Science Department helped guide me in the process of developing my idea and overcoming obstacles. Forrest Tobey from the Earlham College Music Department was an important resource for advice regarding computer music tools and music theory questions.

REFERENCES

[1] Cycling74. 2018. Max8 Max/MSP/Jitter programming languange. https://docs.cycling74.com/max7/
[2] S Errede. 2017. Consonance Dissonance. , 16 pages. https://courses.physics.illinois.edu/phys406/sp2017/Lecture_Notes/P406POM_Lecture_Notes/P406POM_Lect8.pdf
[3] R Jones and B Nevile. 2005. Creating visual music in jitter: Approaches and techniques. *Computer Music Journal* 29, 4 (2005), 55–70.
[4] J Kim, S Ananthanarayan, and T Yeh. 2015. Seen music: ambient music data visualization for children with hearing impairments.. In *In Proceedings of the 14th International Conference on Interaction Design and Children*. ACM, 426–429.
[5] K Kripper. 2018 (active development). Vsynth74 module for Max/MSP/Jitter. https://www.patreon.com/vsynth/overview
[6] M Lehnfeld. 2017. Eden: Visualization Techniques and Sound Synthesis based on Oscilloscope Art. , 53 pages.
[7] K Lubar. 2004. Color intervals: Applying concepts of musical consonance and dissonance to color. *Leonardo* 37, 2 (2004), 127–132.
[8] L Mueth. 1993. MIDI Technology for the Scared to Death: Understanding MIDI opens the door to a powerful resource in music education. Larry Mueth guides music educators through the basics. *Music Educators Journal* 79, 8 (1993), 49–53.
[9] S Nanayakkara, E Taylor, L Wyse, and S Ong. 2009. An enhanced musical experience for the deaf: design and evaluation of a music display and a haptic chair. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 337–346.
[10] D Ryan. 2016. Mathematical Harmony Analysis. *arXiv preprint arXiv:1603.08904* (2016).
[11] V Simbelis and A LundstrÃű. 2016. Synthesis in the Audiovisual.. In *In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 301–304.
[12] R Taylor, P Boulanger, and D Torres. 2006. Real-time music visualization using responsive imagery. In *8th International Conference on Virtual Reality*. 26–30.
[13] F Tobey. 2012. The Joy of Harmony and Composition. http://legacy.earlham.edu/~tobeyfo/musictheory/Book1/home1.html