Interpreting Physical Data as Musical Melody, Harmony, Rhythm, and Tensions through Color

Benjamin Puca-Mendez[1], and Brian M. Wells[1]
pucamende@hartford.edu
[1]*Department of Physics, University of Hartford, West Hartford, CT 06117*

University of Hartford

In this work we show an alternative way to analyze physical data. Most typically, data is represented using graphs and tables. However, by understanding how to capture the dynamic components that make up musical melody, harmony, rhythm, and tensions as a scalar field or graph we can use music to represent physical phenomena. We have developed a code that can convert sheet music into a mathematical array which is then plotted as three-dimensional surface or line graph. Each musical note is given a frequency on the visible color spectrum which is then displayed against other frequencies to create an image. Here we show a unique approach on how music can be expressed through a visual medium. This interpretation can easily be adapted to convert visual data into an audio format which to the trained ear may allow for a deeper interpretation.

Linear Representation

We start the project by taking a look at one of the simpler aspects of music, scales. A scale is an organized pattern of notes that starts on one note, which is referred to as the tonic, and continues up. For the most part scales go from the tonic up to wither an octave or two above it. Here, we look at several different types of scales.

In order to create line graphs made from scales, the tonic was treated as the origin, or zero. Each space on the $y-$axis of the grid was treated as a semitone apart, and the spaces on the x axis were not given a true unit, since music does not have a definite time function. Instead, the $x-$axis was simply used to separate one note from the previous.

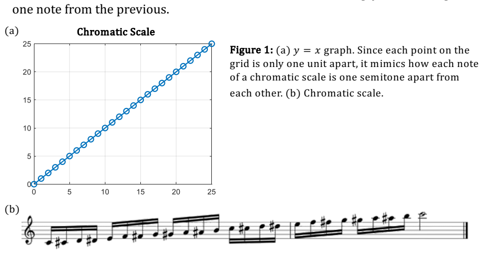


Figure 1: (a) $y = x$ graph. Since each point on the grid is only one unit apart, it mimics how each note of a chromatic scale is one semitone apart from each other. (b) Chromatic scale.

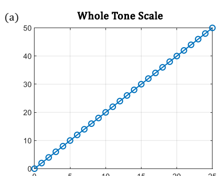


Figure 2: (a) $y = 2x$ graph. (b) A scale whose notes are two semitones, or one whole tone, apart from each other, also known as a whole tone scale.

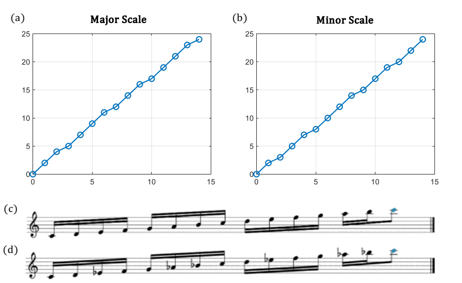


Figure 3: Above are the graphs created. The chart on the right of each shows the data points plotted manually. Each y value is the number of semitones away from tonic each note, or symbol. (a) and (c) major scale, (b) and (d) minor scale.

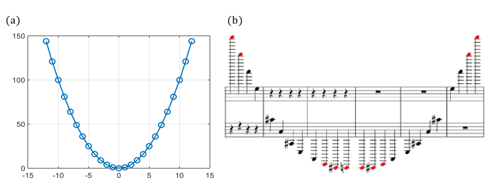
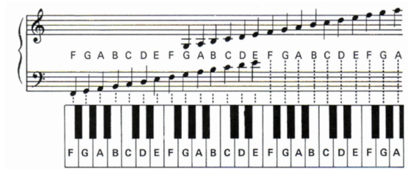


Figure 4: Now, we work backwards, meaning we took a function and tried converting it into notes. For this case, we used (a) a parabola. (b) Here is the sheet music created, where each note is the same number of semitones apart as the points on the parabola. (c) The chart showing each point, or note, used to create the graph. Like the charts in **Figure 1a** and **1b**, each y value is how many semitones each note is from the origin.



Piano with its notes labeled along with sheet music showing where they fall on the staff.

Static Linear Representation

We translate each note to a number in an array which then coincides with a color. Currently these colors are predetermined by the software, MATLAB. We hope to have more control over color choice but will require predefining 88 colors, one for each note on the piano. Each color block represents an eighth note beat. We would like to combine these into rectangles to distinguish between two sequential eight notes.

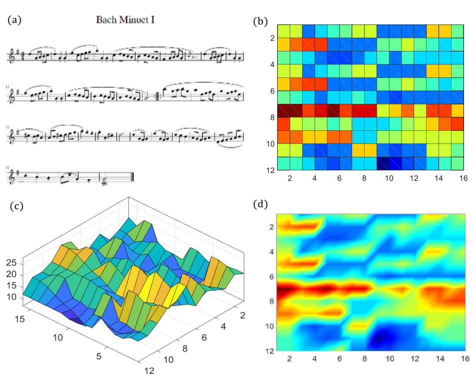
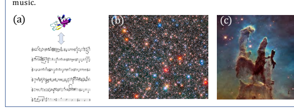


Figure 5: (a) Here is the sheet music of the primary melody from Bach's Minuet I in G. (b) This is the melody as a color plot. We can change the size of the plot as we see fit, currently it is 16 eighth notes across. This plot shows discrete color blocks one for each eighth note beat. (c) This is how (b) looks when rotated, showing the different elevations of each note. (d) This plot shows the interpolation between the nearest neighbor array elements.

Figure 6: (a) Now here is the complete sheet music of the Bach Minuet used. (b) Here is how its plot comes out to be, where every other row after the first is the melody, and every other row after the second is the bass. Because of the larger range of notes, a different note was placed as the origin, so that the colors on both lines would coordinate properly. Unfortunately, a small portion in the lower right section was not able to be plotted for some unknown reason. (c) This is the interpolation of that plot

Some Applications and Uses

As stated before, the goal of our project is to add a new layer of insight to organized and analyzed data. **(1)** This idea has been done before; a similar study was conducted in 2019 by the ACS (American Chemical Society) who used artificial intelligence to convert amino acid sequences into sheet music. We hope to accomplish a similar feat, where instead we take images or data and convert them into sheet music.



(a) Here is the protein lysozyme converted into sheet music by the ACS. Each bar is created by analyzing the vibration patterns of the protein's building blocks, showing an audible representation. Notice how the music doesn't follow any conventional patterns or scales typically used in common music. It is simply used to visualize certain patterns and arrangements created by the protein. (b) Some pictures taken by the Hubble telescope. With our program, we hope to convert these images into sheet music as well. Given the variety of colors displayed in these images, they could be used to analyze the properties of certain celestial bodies, such as what they could be made of or the size and distances of them.

Chord Visualization

This section is our analysis of chords. A chord is a group of two or more notes played at once to create a fuller sound. Typically they are used to amplify a particular note by grouping it with others that will make it sound brighter or darker, depending on which notes are used and in what key signature. Chords with two notes are referred to as double-stops and chords with three notes are triads.

Here, we used our code generated in MATLAB to convert chords into two-dimensional shapes. Each note is a vertex, the lengths of the sides are determined by how far each note is from one another, and, like the previous section with the plotted graph, the notes are given their own colors. Since we are creating shapes here, we only worked with triads and four note chords given the fact that double stops would not give us enough vertices to make a shape.

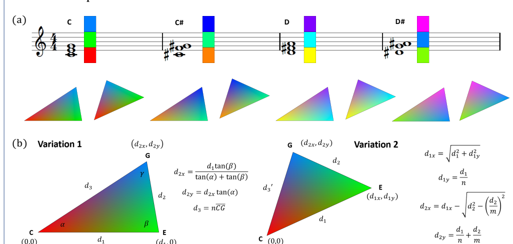


Figure 7: (a) Here are a few examples of triads, all in root position. Each note is assigned a color, but in this case the colors are predetermined. (b) There are two variations for chord representations which are illustrated by third order polygons. As stated before, each vertex is a musical note, assigned a color, with d_1 and d_2 being the intertonal distance. d_3 is a fraction of the intertonal distance to allow for a polygonal representation, this fraction can be controlled. The color blending is done using interpolation allowing the central region color to be a representation of the 'color' assigned to that chord.

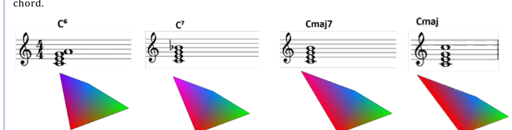


Figure 8: Above are several four-note C major chords with slight variations to them. The little differences shown are reflective in the polygons created from them as well. From left to right, the left-most vertex gets longer and becomes a more prominent shade of red. This is because that vertex represents the highest note of the chord, which is a semitone higher than the chord to the left of it.

Conclusion

Doing these three different procedures bring about different aspects of our research that will eventually come together accordingly. Having the notes portrayed as line functions helps display their intervals from each other, and using colors to differentiate how far apart they are from each other will be able to show tension that is utilized in various pieces of music. Bringing these two together creates the polygons that were formed. The hope is to eventually come to a point where these images can be generated right away and as music is being played, where different shapes and colors are formed and displayed as a song or piece is performed.

References:
(1) Chi-Hua Yu, Zhao Qin, Francisco J. Martin-Martinez, and Markus J. Buehler *ACS Nano* **2019** *13* (7), 7471-7482