Music To Color Barcode

Zoë James, Molly Robins, Elizabeth Shaher
University of Rochester Electrical & Computer Engineering Department

Introduction

When consuming music we rely solely on our ears to help us understand different characteristics of a piece. These characteristics, such as loudness, pitch, and timbre, are important distinguishing features that can define the music we listen to. We believe that we could use other senses to understand these characteristics and how they relate to a piece as a whole.

We came across a TED talk by Neil Harbisson who is color blind and sees the world through frequencies. He created a piece of hardware that would see colors and translate them to different frequencies. We sought out to do the opposite, by creating a program that could take in music and sounds and output color. We decided that printing a color for every frame over time as a single static barcode would best represent our data.

Background & Research

Relationships between Sound and Color

From our research we found that there is no direct relationship between color and sound. They are both frequency based, and color is created through additive wave synthesis, but there is no corresponding color that can be associated with a specific sound. With no direct mathematical, or psychological relationship between color and sound we had to design our own.

Key Component Analysis

The components that were chosen for musical analysis were inspired by components used for genre classification. Genre classification uses timbre, rhythmic content, and pitch content to characterize a piece and its content. We were able to decide on three components that could be used to describe a piece, varied noticeably over time, and were relevant to calculate. We chose loudness, to mirror how energy was included in genre classification; the spectral centroid, to represent timbre and brightness; and the loudest frequency as this would be largely affected by any prominent instruments present in the piece.

Results & Discussion

The red color was associated with Loudness. The value was calculated using the formula below.

$$Red = 255 - (3 \cdot |dB|)$$

Our reason for choosing to multiply our decibel level by 3 is that the most quiet sections of music are about -80dB. When there is no scaling on the decibel value the barcodes become overwhelmingly red. Since green and blue are both based on frequency, they can be calculated the same way. To associate a value between 0 and 255 with a frequency that can range anywhere from 0 to 20,000 we took the common frequency bands used for mixing into account. We multiplied the band number of the input frequency by 20 to account for the specific band. We then modulated the frequency over 135 and added it to the multiple of 20. This allowed us to account for differences between frequencies within the same band.

The Beatles, "Hey Jude"

The Proclaimers, "I'm Gonna Be (500 Miles)"

Electric Light Orchestra, "Mr. Blue Sky"

One of the best applications of our visualization software is the ability to use it as a way to represent and recall data. Since it is not possible to perfectly recreate the song, or any piece of sound from the color barcode, we can use them as representations of the audio.

To show how the barcodes are individual to the song, we created barcodes for two different versions of the same song, "Bohemian Rhapsody". The first barcode in the figure below was made using original song, and the second barcode in the figure was created using a cover of the song that emulated the style, but was performed by a different artist and produced by a different team.

Queen, "Bohemian Rhapsody"

Panic! At The Disco, "Bohemian Rhapsody"

The barcodes for these two versions of the same song are visually different, and can be properly matched within our database. However, it is also clear that these songs are very similar. The frequencies present and the volume of each section of the song are relatively similar. They look like the same song, but can be differentiated, the same way that they sound like the same song but can easily be differentiated.

References

1. Libretexts, "Electromagnetic Radiation," Chemistry LibreTexts, 23-Apr-2019. [Online]. [Accessed: 30-Apr-2019].
2. Musical genre classification of audio signals - IEEE Journals Magazine. [Online].[Accessed: 30-Apr-2019].
3. N. Harbisson, "I listen to color," TED. [Online]. [Accessed: 01-May-2019].
4. "Understanding Hexadecimal Colors is Simple," Pluralsight, 19-Jan-2015. [Online]. [Accessed: 30-Apr-2019].

Methods

This project was written in MATLAB, as the program provides many tools for audio analysis and visualization. The code is comprised of a single main script that calls two functions: one that calculates the key components, and another that translates those key components into an RGB color code. Each of these functions operate on a single frame. The main script itself is where the sound file is input, and the barcode is generated.

Before analysis, the input is buffered into frames with a frame length of 512 samples, using a rectangular window and a hop size of zero samples. After the buffering process, the input is processed frame-by-frame through the spectral analysis function followed by the sound-to-color converting function. The output of the sound-to-color conversion is then plotted as a stem plot with a wide line thickness, without the stem heads.

To create the RGB color code, the red value was associated with loudness, the green value was associated with brightness (spectral centroid), and the blue value was associated with pitch (loudest frequency).

ColorBar Database

We created a small database of songs with their associated barcode. Our code can read and accurately match the barcode to the correct song, including matching specific versions of the same piece of music.

Our database currently works by matching an input matrix with one of the matrices stored from previous musical outputs, meaning a new song can not be identified, the barcode must exist in the database beforehand. When a song does exist in the database, the identification program will output the title and artist of the song, then play a 30 second clip of the piece of music.