Audio Spectral Centroid Visualization

Eric Anderson
Student at Ontario Tech University
Markham, Canada
eric.anderson1@ontariotechu.net

Abstract – Throughout this paper there the development process of music visualization software is discussed. As well as coving who this project was intended for.

Index Terms – Frequency, Librosa, Music, Python, Software, Spectral Centroid, Tkinter, Visualization

I. INTRODUCTION

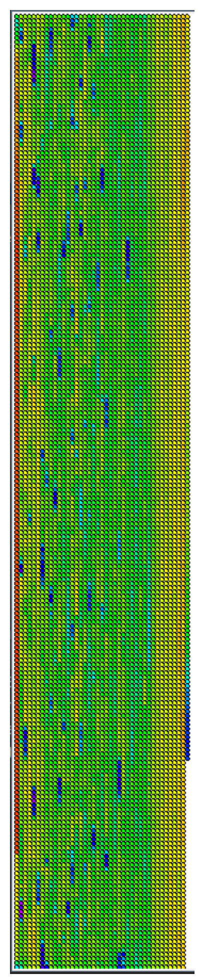
Music plays a large role in many peoples lives. In the fourth quarter of 2021 Spotify had approximately 406 million active users worldwide, and that is just one way to consume music [1]. So, it clearly plays an important role in society. Despite this though not many people ever think about what music looks like, and the ones that do look, may not find what they are looking for. A quick look at the images Google shows when searching “music visualization” reveals a lot of snap shots of active visualization. This is not really all that useful if you want to understand more than just an instance in time of any musical piece. So, the goal for this project was to better represent music across a time period rather than just an instance.

II. THE TECHNOLOGY

When going through the possible ways to visualize an album of music python seemed like the best solution. With its large variety of packages and script style of coding more time could be put into visualization development. For example, instead of coding an algorithm to get the spectral centroids needed from an audio file, librosa was opted for as it could acquire the necessary data with less lines of code. It also saved on development time. The default python package tkinter was also used as it seemed to be the best way to achieve the visualization required for the project. Although other libraries were used in the python program librosa and tkinter are the most critical to its functionality. With these technologies it was then possible to create the first iteration of the program.

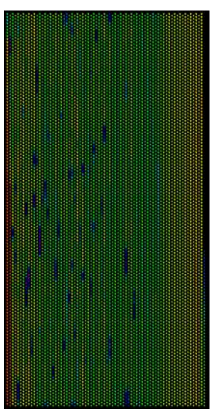
III. ITERATION ONE

When the program is run at this stage it will prompt the user to enter in a location of a folder. In that folder it expects to find at least one WAV file for it to load into the program. From here, the spectral centroid data is stored as a list and scaled to be a percentage of the max hertz value in the list. The scaled audio samples are multiplied by a max hue value which determines the colour of a point when plotted. As far as following the end result you need to alternate the direction you are reading. First starting from the top right and looking down. Then looking to the next column over and reading up that column. When the program is completed, you will have an image of an entire album showing the mean frequency of the audio at multiple points in time.



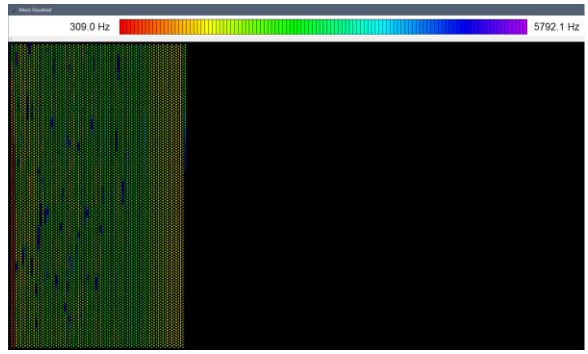
IV. ITERATION TWO

The first iteration of the program was a great start but it had multiple problems. It was more of a test of concept which means it was not very user friendly and had some restrictions. In the first iteration the result was just shown at the end of the program. As a result of this the program suffered from a directionality problem largely to the use of circles as data points. So, in this iteration the circles were abandoned in favor of triangles as they could be used as arrows to provide the necessary directionality. The improvement of directionality was taken one step further though. Instead of loading every sample a one moment they are loaded over time which helps with providing direction. A progress bar was also added as a result of this visualization change as it is informative for the user to know when the program will finish its visualization. Another restriction from the first iteration was the audio length. If an album greater than about 40 minutes was loaded into the program the end of the data would be cut off and unseeable. To solve this a scrollbar was introduced. Black triangles are also added at the end of a track to distinguish between songs. The last major change in this iteration was switching the background colour from white to black. By making this change it focused the eyes on the bright coloured data points instead of the white background.



V. ITERATION THREE

The last version of the program did not make too many drastic changes but the ones that were made improved useability. In the second version of the program the scrollbar's location was not updated to a new location when the window was full. As such, it was more difficult to follow the data as it was loaded. This version solves this by updating the scrollbar's location when the window is full of data. Version 3 also adds a legend so that the output of one run of the program can be compared to another.



VI. Audience

This visualization software can be used by anyone who is interested in what music looks like. Though it is more directed to audio engineers, music writers and researchers. The nature of the program allows a comparison between different albums or songs. This allows these groups to analyze why some songs are so popular and others are not, for example. Then they can apply their findings to their own work the way they would like.

VII. CONCLUSION

With the use of python, it was possible to visualize an album of music. Throughout the three iterations of the program, we went from proof of concept, to a user-friendly program and then to a version that is actually useful for audio enthusiasts. Although iteration three of the program is great and has come a long way from its initial conceptualization there is still more that can be done. Due to time restraints or code complexity some features did not make the cut to the final program. Which leave the potential for an iteration four in the future that allows the mean volume to be seen and the ability to listen to the track audio while the visualization is rendering. Software is a constant fluctuating development so there is always room to improve and visualize new musical information.

REFERENCES

[1] M. C. Götting, "Spotify Maus Worldwide 2021," Statista, 10-Feb-2022. [Online]. Available: https://www.statista.com/statistics/367739/spotify-global-mau/#:~:text=In%20the%20fourth%20quarter%20of,million%20in%20just%20one%20year. [Accessed: 04-Mar-2022].