Cornell University

CS 4621 Practicum Project Proposal

A♯ – Music Visualizer

Shane MOORE
swm85

Zachary ZIMMERMAN
ztz3

Emre FINDIK
ef343

Joseph VINEGRAD
jav86

October 9, 2014

1 Summary

Our group proposes to create a music visualizer called A♯ (A Sharp) which goes beyond the interpretive scope of current visualization software. We find that the typical music visualization does fine at looking good alongside the music, but fails to go beyond and provide actual interpretation or insight into the song.

As Edward Tufte said in his book *The Visual Display of Quantitative Information*, "At their best, graphics are instruments for reasoning about quantitative information" (Introduction). The job of a music visualizer, therefore, is to provide the user with enough relevant and useful and visual information that they may interpret, on a higher level, the characteristics of the sound which is being visualized. If possible, the visualization could be considered a summary of the song, and even a rudimentary alternative.

We plan to work towards this standard in our music visualizer, A♯.

2 Software description

- Application that takes a song file and outputs a video file to accompany it

- We should think about this, maybe we want to input a song, have it do preprocessing, and then immediately play the video?

- Perhaps also take midi input as a warmup (this is a good idea)

3 Application in Graphics

The graphics portion of our project will focus primarily on rendering and animation.

Given our goal of creating complex visual displays that accurately reflect different types of music, rendering will be a key component of our work. We will likely want to incorporate a variety of shapes, textures, and lighting. In order to accomplish this, we will apply techniques such as texture mapping and shading (under various models). Additionally, our scenes will probably consist of several objects on the screen simultaneously. To account for this, we will need to use ray tracing techniques to determine relative positions of objects. Still, determining what combination of objects and shading models to use for different songs remains a challenge.

Animation is another major area of focus for our project, as we aim to produce a dynamic moving display for our visualizer. To enhance the design and versatility of our visualizer, we will utilize different animation patterns corresponding to different types of music. To implement these, we will apply techniques related to both linear and nonlinear transformations (in 2D and 3D), as well as optimizations for seamlessly animating frame by frame. While we havent yet discussed animation in class, we are scheduled to do so shortly before our first milestone, giving us time to incorporate animation techniques thereafter.

4 Software Architecture

Code in python Modules Sound file analyzation Model for representing different parts of sound Timbre, key/"emotion", amplitude Controller acts as abstraction layer between model and view View (graphical representation) Software representation of the visualizer Renderer Actually render (and play?) the video of the visualization

Also breakdown of work

4.1 Properties of Music for Quantification

- Song genre
- Amplitude at a given frequency
- Sound location (stereo)
- Sound quality (Hilbert scope)
- Tremolo
- Centroid, spread, skewness and kurtosis (of an amplitude envelope)
- Mel-frequency cepstrum
- Rhythm complexity
- Distinguish accompaniment from melody
- Melody: pitch, volume, position in chord

5 Milestone

By the milestone, our group hopes to have the following as part of a proof of concept:

- Warmup: midi support, later on, wav support
- Working interfaces for each of the sound libraries we choose to use
- Integration of data from all of the libraries into one format
- A rudimentary visual representing every meaningful piece of insight we can glean from the music