**CSC 545/645 Computer Speech, Music and Images**

**Exercise No. 11c, Week 14, due April 25, 2021**

**Visualizing the Structure of Music 2: Adding Sound and Time**

**Goal**

Display a self-similarity matrix of music with audio play and passage of time

**Background**

In a previous exercise, you created and displayed a matrix showing the self-similarity of music. It’s hard to know, however, how that matrix relates to song structure. In this exercise, we will add the audio and illustrate how the audio relates to the similarity matrix.

**Procedure**

Start with your music visualization code from the previous exercise.*.*

The goal of this exercise is to play the audio of the selected song and to show, on the similarity matrix, the passage of time. Playing the song is simple, and you can see how to do it in the songPlayer program you wrote in a previous exercise, or in the showSpectra example program on trace. At any rate, all you have to do is create an AudioPlayer by calling the Minim object and start it playing:

songPlayer = minim.loadFile(fname, fftSize);

…

songPlayer.play();

where *fname* is the name of the file to play and *minim* is the Minim object. Starting play should probably be the last thing in *setup()*. The AudioPlayer is global because you’ll have to use it, in *draw()*, to synchronize audio play time with the similarity matrix display.

One way to show the passage of time is to draw a colored (red?) line down the diagonal of the similarity matrix in the display window. The trick, though, is to synchronize elapsed time with position on the matrix.

If you assume the similarity matrix represents the entire song, then the horizontal (and vertical) endpoint of the progress line should cover an equal percentage of the image as the percentage of song played. An AudioPlayer has two relevant methods: AudioPlayer.length() gives you the length of the audio file, in milliseconds. AudioPlayer.position() gives you the current play position in the file, also in milliseconds. The line should cover the same percentage of the width of the file as the percentage of the file played.

You should probably set the line color in *setup();*  *stroke(255, 0, 0)* will set the line color to red. You may also want to set *strokeWeight()* to 3 or 5 so the line will show up better. However, it’s not clear that a diagonal line is the best way to show the passage of time. I suggest you start there, with a red line drawn from (0, 0) to (frame number, frame number), where frame number is the index of the pixel that represents the percentage of the song covered so far. But there may well be a better way. It might be nice to draw a rectangle encompassing the part of the matrix covered so far. You could also draw the part of the matrix covered by the play so far by drawing that part in a different color—using, in effect, “greenscale” or “redscale” rather than grayscale. Or perhaps you can think of a more effective representation.