Keefer Rourke
Mr. Fryer-Davis
6 May 2015
ICS4UI

Fast Fourier Transforms and Frequency Visualisation of Audio Files – Design

There are three major parts to this project:

- Opening, parsing, and playing audio files
- Performing the fast Fourier transform
- Visualising the waveform using OpenGL

Dealing with Audio Files

A fast Fourier transform needs to be performed on uncompressed audio, so an uncompressed audio container format should be used for simplicity. The two most common uncompressed audio formats are AIFF and WAV, however this project will likely only be concerned with the latter. The decision to work with WAV audio over AIFF audio was an arbitrary one, but AIFF support might be added later.

WAV files are (quite simply and nicely) composed of three different chunks or sections: the "`RIFF` chunk", the "`fmt ` chunk", and the "data chunk".

- The RIFF chunk contains the four byte chunk ID, `RIFF`, followed by a four byte integer which describes the file size in bytes, and other four byte form type identifier, `WAVE`.
- The fmt chunk contains the four byte chunk ID, `fmt ` (note the space), followed by a four byte integer which decribes the wave format size in bytes, followed by a data block of the size described which contains the wave format information.
- The data chunk contains the four byte chunk ID, `data`, followed by a four byte integer which describes the size in bytes of the PCM data, and the actual PCM data.

This can be read into the program by a single function, which verifies that the file is valid (has all three chunks with valid IDs), and loads it into a buffer to be played through the devices speakers.

Performing the FFT

While there are several FFT algorithms that can be used, by far the most common is the Cooley-Tukey algorithm which recursively breaks down discrete Fourier transforms into smaller transforms in order to reduce computing time from $O(n^2)$ to $O(n \log n)$.