

Math448 Final Project Update – Basic Ideas – Cameron Embree

Basic Ideas needed to understand what you're doing to understand project

GOAL

Implement a system in Mathematica where I can compare multiple samples of audio and classify a piece of audio as being the “same”. I will attempt to create a function that will use examine the Fourier Transform of a audio sample and compare it to the Fourier Transform of previously analyzed audio of the same word. Whichever pre-sampled Fourier transform is closes to the one being tested will be classified as being that word.

BACKGROUND

Audio analysis for sound classification is a growing field and increasingly used (and accurate) tool in mobile phones. I am interested in seeing what tools Mathematica has for audio feature extraction to compare and classify audio. The most likely information I will use for the comparisons is the Fourier Transform of the audio

Import

Use import command of Mathematica to bring in audio that can then be played for testing and gives a quick view of the Fourier Transform of the sample. The user would need knowledge of something like the following:

```
audio = Import["some_audio.wav"];  
audioDat = Import["some_audio.wav", "SampledSoundList"];
```

Fourier Transform - a mathematical **transformation** employed to transform signals between **time** (or spatial) domain and **frequency domain**. It is reversible, being able to transform from either domain to the other. We will use the Fourier transforms by using the following FourierDCT command:

```
FourierDCT[audioDat]
```

The FouierDCT finds the Fourier discrete cosine transform of a list of real numbers. I am using the DCT because it may be easier to compare real values rather than the imaginary from the regular Fourier Transform in Mathematica. To compare the possible sounds, I will create an N by N matrix of the Fourier Transforms of the audio.

```
aDatFTs = Table[Abs[FourierDCT[aDat[[i, j, 1, 1]]]], {i, 1, 4}, {j, 1, 4}]
```

Spectrogram

Used to plots the magnitude of discrete Fourier transforms of partitions of the audio. This is used to examine the differences between the audio amples by hand before breaking them apart and trying ti compare the two. The Command looks something like the following:

```
Spectrogram[audioAt], ListLinePlot[dat, PlotRange -> All]
```