Kevin Leung

CS591 Computational Audio

Wayne Snyder

**Project – Computational Analysis of Standard American Vowel Sounds From IPA**

**1. Introduction**

In starting this project, the first thing I did was learn the vowels of Standard American English from the IPA chart from this page. <https://en.wikipedia.org/wiki/IPA_vowel_chart_with_audio> . These vowels are u, i, ɛ, ɘ, e, ɔ, æ and ɒ. I used the audio provided on the Wikipedia page as and I also recorded my own, imitating the sounds of the .wav files from the Wiki page. You may find the audio from the Wiki page in the folder **Vowels1**. Find the .wav files of the vowels I recorded inside the folders **Vowels2** and **Vowels3**. Only male voices were used. Software used to complete this project are Spyder (Python 3.5) and Anaconda.

**2. Explaining vowelProject.py**

A. *Overview*

I borrowed code from the website <http://stackoverflow.com/questions/23377665/python-scipy-fft-wav-files> in order to find the frequency spectrum of the vowels. (NOTE: This code only works for .wav files in Stereo Mix, not mono).

The main method is where you can choose which folder of audio files and the audio file of which particular vowel you want to choose.

B. *Algorithm*

I found the frequency spectrums of all the vowels from the folders **Vowels1** and **Vowels2** and plotted them. I noticed that some of them look fundamentally different while others shared similarity. For example, the vowel u was the easiest to distinguish because it peaks very high in the beginning while all the minor/local peaks after the peaks in that area become very miniscule – more so than the other vowels. That said, it was easiest to create the algorithm to determine if a wav file is vowel u or not.

\*\*\*I analyzed the Peak1/Peak2 Ratio to determine which vowels share similarities and could be found\*\*\*

**Finding vowel u:**

I found the highest peak which is usually somewhere near the beginning of the plot. Then I calculated where to begin looking for the next highest peak. I calculated this beginning point by doing 0.06 \* len(FS). This is the x value that is 6% of the length of the x-values in the frequency spectrum. After that, I created a limit which is 2% the value of the highest peak. If the 2nd highest peak found after the beginning point is lower than that, we know for sure that the vowel is u.

**Finding vowel i:**

In finding vowel i, I would make sure the 2nd highest peak is in between 35% and 60% the value of the highest point and that the y-value of the 28% of the length of FS (frequency spectrum) is less by 1% of the highest peak (0.01 \* max), because if you notice in the FS, there generally aren’t high values around 28% of the length of the FS.

**Finding vowels ɔ and ɒ:**

* The 2nd highest peak must be in between 2.5% of the highest peak and 11% of the highest peak.
* Vowels **ɘ, e æ or ɛ** do not seem to have any distinguishing features and therefore I was not able to determine if they the .wav file is one of those vowels or not.

**The program would return one of these results:**

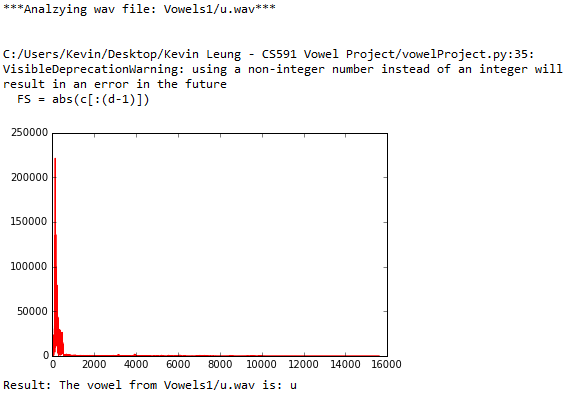
* The vowel from Vowels1/u.wav is: u
* The vowel from Vowels1/i.wav is: i
* The vowel from Vowels1/ɔ.wav is: ɔ or ɒ
* The vowel from Vowels1/ɘ.wav is: either ɘ, e æ or ɛ

**5. Frequency Spectrum Plots:**

Look in the folder to find the file ***Frequency Spectrum Plots.pdf***

**6. Results From Test Cases:**

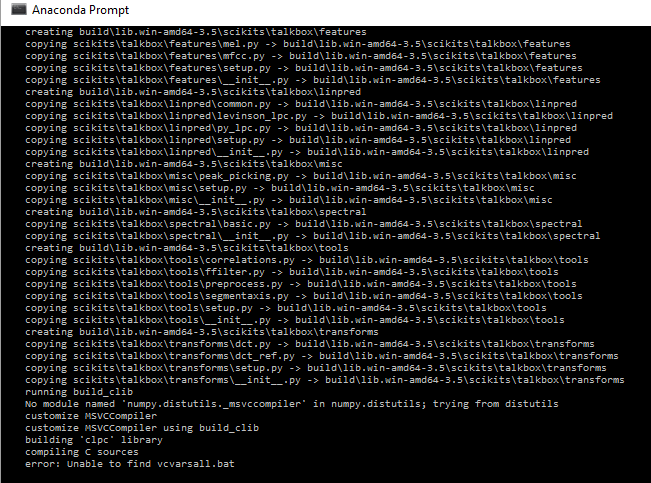
Results generally look something like this:



* **‘u.wav'** correctly identified as “u” 3/3 times.
* **'i.wav'** correctly identified 2/3 times as “i”, i.wav from Vowels3 folder misidentified as “either ɘ, e æ or ɛ”
* **'ɛ.wav'** correctly identified as “either ɘ, e æ or ɛ” 3/3 times
* **'ɘ.wav'** correctly identified 2/3 times as “ɘ”, from ɘ .wav file from Vowels3 folder misidentified as “ɔ or ɒ”
* **'e.wav'** correctly identified 3/3 times as “either ɘ, e æ or ɛ”
* **'ɔ.wav'** correctly identified 3/3 times as “ɔ or ɒ”
* **'æ.wav'** correctly identified 3/3 times as “either ɘ, e æ or ɛ”
* **'ɒ.wav'** correctly identified 1/3 times as “ɔ or ɒ” and 'ɒ.wav' from folders Vowels2 and Vowels3 misidentified as “either ɘ, e æ or ɛ”

**7. Reflection/Conclusions:**

I originally intended to use a program to find estimate formations using LPC in Python, but it was problematic for me as it gave me the error “ImportError: No module named 'scikits.” When I tried installing it on the Anaconda prompt using “pip install --user scikits.talkbox” it gave me this:



Due to this technological barrier, LPC is not covered in this project generally. One thing this project does not address is distinguishing vowels when consonants are present. Through “Acoustic characteristics of American English vowels” by Hillenbrand, Getty, Clark, and Wheeler, I learned that when analyzing the formant frequencies of the vowels, whether the test subject is male, female or a child must be taken into consideration as different results are produced for the average signed and absolute differences. Steps I believe are necessary to further polish this project is to explore programs that deal with Formant Estimation with LPC Coefficients. This is considering that observing the different formants will produce greater accuracy than only analyzing the frequency spectrum. An another note, it was difficult reading the required readings and determining what was important for this problem and ignoring what may be filler. Next time I would focus on only one method and crafting that method rather than exploring different methods all at once, such as ones that are far beyond the scope of this class.

**Sources:**

<http://stackoverflow.com/questions/25107806/estimate-formants-using-lpc-in-python>

<http://stackoverflow.com/questions/23377665/python-scipy-fft-wav-files>

Required readings provided by Prof. Snyder