ECES-434 Final Project Overview

Project Description

For this project, you will implement an audio codec. You may choose an algorithm from those discussed in class:

- Sinusoidal Encoding
- Psychoacoustic Masking
- Linear Predictive Coding
- Vector Quantization

Your task is to maximize the performance of your implementation in four areas: **compression ratio**, **audio perception**, **runtime**, and **signal to noise ratio**. During finals week a competition between all students' implementations will take place. Winners will be determined and awarded prizes.

Python Functions

You will generate two Python functions: *encode* and *decode*. The *encode* function takes a vector of samples (samples of the audio clip) and returns a structure containing all information needed to run *decode*. Then *decode* will take your structure and synthesize and output the decoded original signal.

Competition

1. For testing purposes, a number of audio clips will be given to you to evaluate your system. Different clips will be used to evaluate your codec during the competition.
2. The four evaluation categories are weighted as follows:
   (a) **Compression Ratio - 45 %**
   The size of your structure will be compared to the original signal.
   (b) **Perception - 30 %**
   An online survey will be sent out which will allow students to evaluate the sound quality of the audio codecs of their peers. Students will assign scores to the anonymously labeled sound clips resulting from each submitted *decode* function.
   (c) **Runtime - 15 %**
   Both your *encode* and *decode* functions will be timed. The runtime of each of these functions will be summed and used as your complete runtime.

(d) **SNR - 10 %**
The signal to noise ratio will be calculated for each codec using the following equation,

$$\text{SNR} = 10 * log_{10}\left(\frac{A^2_{original}}{(A_{decoded} - A_{original})^2}\right) \tag{1}$$

where $A_{decoded}$ is the amplitude of the output from your *decode* function and $A_{original}$ is the amplitude of the original input signal used in *encode*.

Submissions

You may discuss the project with other students, but each student must create his or her own code implementation and submit an individual notebook. Late submissions will not be accepted. Your submission should be a zip folder consisting of the following:

1. Your Python code (.py file)
   (a) Your functions will be evaluated automatically by a script, so they must have the correct names, *encode* and *decode*.
   (b) Your *encode* function must take only one parameter, a mono 44.1 kHz and 16 bit audio signal. Whatever single structure it returns should be the only input parameter for *decode*.
   (c) The *decode* function should take what is returned from *encode* and return a mono audio signal in the original format.
   (d) **Important: any script that crashes or does not produce a mono audio signal for y = *decode*(*encode*(x)) cannot be evaluated and will not score well.**
2. A Jupyter Notebook file (.ipynb) demonstrating your codec
   (a) Your implementations of the *encode* and *decode* functions should be called within the notebook.
   (b) Audio examples should be provided and labeled.
3. A final project report
   (a) Your report should follow the IEEE Conference template (on Bb Learn).
   (b) It should be a maximum of 4 pages.
   (c) Include an abstract (summary, 150-200 words).
   (d) Focus on your design of codec and results, do not dwell on intro and background.
   (e) Appropriately labeled figures/plots are strongly encouraged.