CS3570 Introduction to Multimedia

Homework #2

Due: 11:59pm, 4/14/2017

Q1. (50%) Create your own FIR filters to filter audio signal:

"**HW2_Mix.wav**" is a mix of 3 songs. Based on the ideal impulse responses given in slide #76 and the windowing functions in slide #79, you need to design and apply different FIR filters in time domain to separate the **three audio signals** from the given audio file. You can refer to the algorithm on slide #80.

1. **(20%)** Implement **3 different FIR filters** to separate the three audio signals with:
   - **Blackmann** window function (You have to pick the appropriate window size and cut-off frequency).
2. **(5%)** Implement **1-D convolution on the input signal** of the given audio with your filters.
3. **(8%) Plot** the result.
   - The spectrum of the input signal.
   - The spectrums of the output signals.
   - The spectrums of the filters.
   - The shapes of the filters (time domain).
4. **(3%) Store** the filtered audio files and name **"FilterName_[para1]_ [para2].wav"**.
5. In the report
   - **(4%)** Discuss how you determine the filters.
   - **(5%)** How you implement the filter and convolutions to separate the mixed song.
   - **(5%)** Compare spectrum and shape of the filters**.**
   - Anything worth mentioning.

Q2. (50%) Music classification through spectrograms and human eyes:

In this unit, we have learned that we can transform an audio signal from time domain to frequency domain to obtain its spectrogram. The spectrogram of an audio signal contains temporal and spectral information and we can visually observe its important characteristics. Therefore, we can consider using spectrograms to recommend or classify music based on the visual contents of them.

Here we have a music collection, 12 audio files performed by 4 different instrument classes (guitar, piano, violin, drum), each one with 3 files. Given 4 test spectrograms, you need to classify each of them through your inspection on the spectrogram image to determine which instrument class it should be.

You can read the slide **"Supplement"** for implementation details. You can also read the paper **"AUDIO CLASSIFICATION FROM TIME-FREQUENCY TEXTURE"** to see how the author use spectrograms for music classification.

1. **(20%)** Implement **"Short-time Fourier transform"** to **plot** and **store** spectrograms of all the audio files in the "audio" folder (it is your music collection). Name your stored spectrogram image "InstrumentName_para.png", i.e. guitar_01.png for guitar_01.wav.
2. **(8%)** For each test spectrogram, please classify which class it should be through your visual inspection.
3. In the report
   - **(11%)** Discuss what you observe on spectrograms of different classes of instruments.
   - **(11%)** Discuss how you implement **"Short-time Fourier transform"**.
   - Anything worth mentioning.

Reminder

- **Follow the instructions (hints) and spec in the sample codes given by TA.**
- You cannot use Matlab built-in function "**conv**", "**spectrogram**" in this homework.
- Plot the spectrum and shape in appropriate range for better visualization.
- Please save the report as **"[YourID]_report.pdf"**.
- Please compress all the .m files, output audio/image files and report into a zip file and name it **"HW2_[YourID].zip"**.