FFT Workshop 2023
September 26th 2023

Due date: Monday October 9th 2023, 23.59h.

Introduction

This assignment can be made using the programming language and libraries of your choice (e.g. Python, C, MATLAB®, etc.; librosa, numpy, etc.). Python and librosa are recommended.

Getting started:
This is just an example using *Python 3.8*, *numpy* and *librosa* that calculates the Fourier transform of an audio file and displays its spectrogram. This and many more examples can be found on https://librosa.org .

1. Make a virtualenv:

```
virtualenv fft --python=python3.8
source ./fft/bin/activate
```

2. Install some packages in this virtual environment:

```
python3.8 -m pip install jupyter
python3.8 -m pip install matplotlib
python3.8 -m pip install librosa
```

3. Start a Jupyter notebook.

```
jupyter notebook
```

4. Enter and run the following code in the notebook to calculate the Fourier transform of an audio file and display a spectrogram:

```
import numpy as np
import matplotlib.pyplot as plt
import librosa
import librosa.display

y, sr = librosa.load(librosa.ex('trumpet'))
D = librosa.stft(y)  # STFT of y
S_db = librosa.amplitude_to_db(np.abs(D), ref=np.max)
plt.figure()
librosa.display.specshow(S_db)
plt.colorbar()
```

Assignment 1 Feature Vectors

a) Under *'Getting started'* we showed how we can compute the Fourier transform of a sound signal. Implement a procedure that takes as input a wav file (8bits resolution, 16kHz sample rate) and gives as output for every part (window) of 512 samples the energy of N=8 frequency bands (for example: [0Hz,1kHz), [1kHz, 2kHz), … , [7kHz, 8kHz)) in the Fourier transformed signal. Calculate these features for the <piano.wav> file which can be found in the <audio_data.zip> file, i.e., a list of 8-dimensional real-valued vectors results.

b) These features can be used to calculate the following code (C_i) for the piano.wav, where:

- C_i = **U (Up)**, if the current max energy band is of a higher frequency than the previous max energy band
- C_i = **D (Down)**, if the current max energy band is of a lower frequency than the previous max energy band
- C_i = **Rx (Repeat-x)**, otherwise (where **x** is the number of consecutive **repeats**)

This codes the signal as a sequence of pitch tendencies, which can be used to recognize melodies. In some respect it resembles the so-called Parsons code.
NB Select N the number of frequency bands and the specific frequency ranges of the N bands in such a way that the features are effective for processing music played by a piano. Justify and explain your choices in the comments of your code.

Submit your coding-routine and a text file with your 'up/down/repeat'-code for the <piano.wav> in a single zip file using Brightspace before Monday October 9th 2023, 23.59h.