AMATH 482 Homework 2

Zhiyang Cheng

February 10, 2021

1 Introduction and Overview

This assignment we are analyzing a portion of two of the greatest rock and roll songs of all time using two files. The first one is GNR.m4a (14 second clip) and the second one is Floyd.m4a (60 second clip). These files play clips of the songs Sweet Child O' Mine by Guns N' Roses and Comfortably Numb by Pink Floyd, respectively. We converted those pieces of music into a vector in order to filter and analyze them.
The code and report is trying to reproduce the music score for the guitar in the GNR clip, and the bass in the Floyd clip using Gabor filtering and Fourier Transform.

2 Theoretical Background

2.1 Fourier Transform

The Fourier Transform (Eqn.1) can take a discrete function in time space and convert it into frequency space by breaking up the function into sines and cosines.

$$f(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ikx} dx \tag{1}$$

The Fourier is used to analyze which frequencies are presented in the songs but it cannot locate those frequencies in the time domain.

2.2 Gabor filtering

Gabor filtering is another important method that is been used in this report. A gabor filter set with a given direction gives a strong response for locations of the target images that have structures in this given direction. Gabor filtering (Eqn.3) in this assignment is to locate the frequencies in the time domain. Multiplying by the filter g(t$-\tau$) gave:

$$f(t, \omega) = \int_{-\infty}^{\infty} f(t) g(t - \tau) e^{-ikt} dt \tag{2}$$

Based on this method, we can determine the music notes from the given files by getting the information of frequencies at different time (t).

3 Algorithm Implementation and Development

3.1 Setting up

The first several lines of Matlab code in Appendix B is to import the song data from GNR.m4a and Floyd.m4a. We firstly set up time and frequency domain based on the audio files and transpose the vector data. Aftering getting the frequency domain using FFT, for our purpose, we then transfer those data in to Hertz.

3.2 Analyze

Nextly we created a spectrogram matrix to store the frequency information. We then use a for loop to multiply the audio data at each time with the Gabor filter. We tried several times with different constant to see the best value for Garbor tranform. We lastly use the pcolor fucntion in matlab to plot our spectrogram. It appears as a hotmat with the matlab command colormap. The x axis is the time(s) and the y axis is labled as the music notes with corresponding result (notes) on it.
We repeated those steps for three parts of code to get the 'GNR Guitar Music Score', 'Floyd Bass Music Score' and 'Guitar Floyd Music Score.'

4 Computational Results

My computational results is below including three graphs including the 'GNR Guitar Music Score', 'Floyd Bass Music Score' and 'Guitar Floyd Music Score.'
All the graphs have specified music notes on the y-axis and time(s) as the x-axis.

5 Summary and Conclusions

By running the Matlab code below in the Appendix B, we got the spectrograms for the audio files GNR.m4a and Floyd.m4a which are 'GNR Guitar Music Score', 'Floyd Bass Music Score' and 'Guitar Floyd Music Score.'
Figure 1 is the music score for the guitar in the GNR clip. Firgure 2 is the bass in the Floyd clip. Those two are more clearly identifiable than figure 3 which is the Guitar Floyd Music Score.

6 References

https://dsp.stackexchange.com/questions/1603/what-is-the-gabor-filter-and-what-are-its-main-uses
https://www.thefouriertransform.com/

Appendix A MATLAB Functions

- `audioread('file.m4a')` returns a row vector with the audio data and the sampling rate for the data
- `abs(x)` computes and returns the absolute value of (each element in) x
- `fftn(x)` Computes the Fast Fourier Transform of the object x
- `fftshift(x)` shifts the transformed function x
- `pcolor(X,Y,C)` setting up the x- and y-coordinates for the vertices. The size of C would be he size of the x-y coordinate grid.For example, if X and Y define an m-by-n grid, then C must be an m-by-n matrix.
- `colormap(map)` sets the colormap for the current figure to the specified one

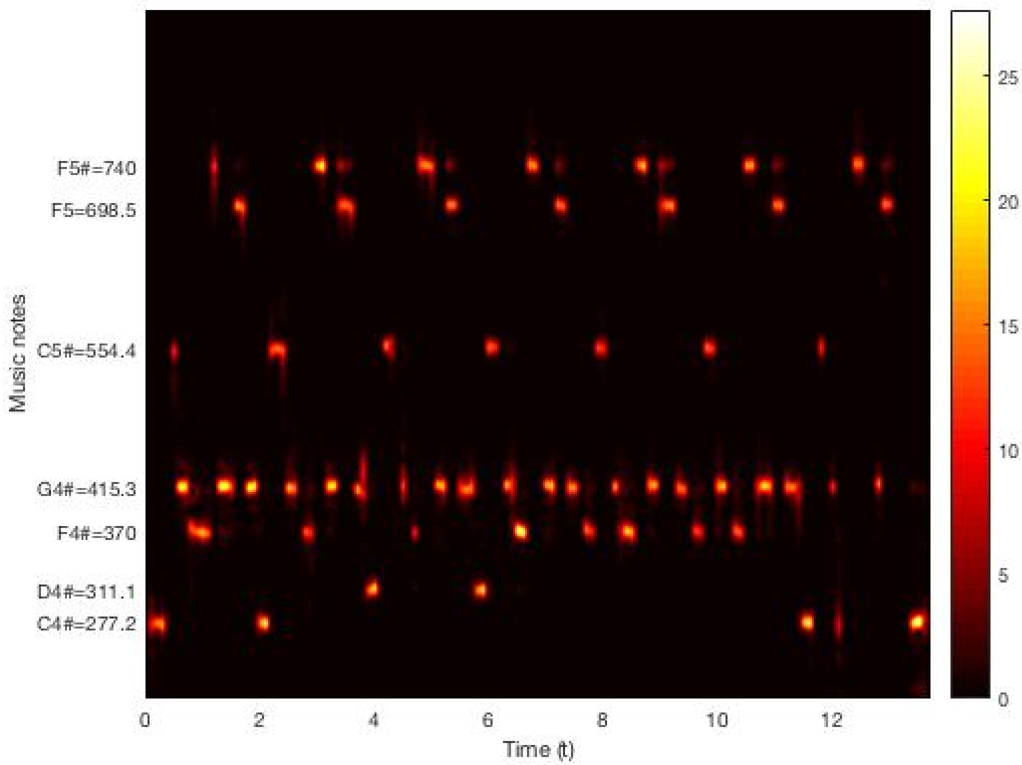

Figure 1: Here is a graph for the GNR Guitar Music Score.

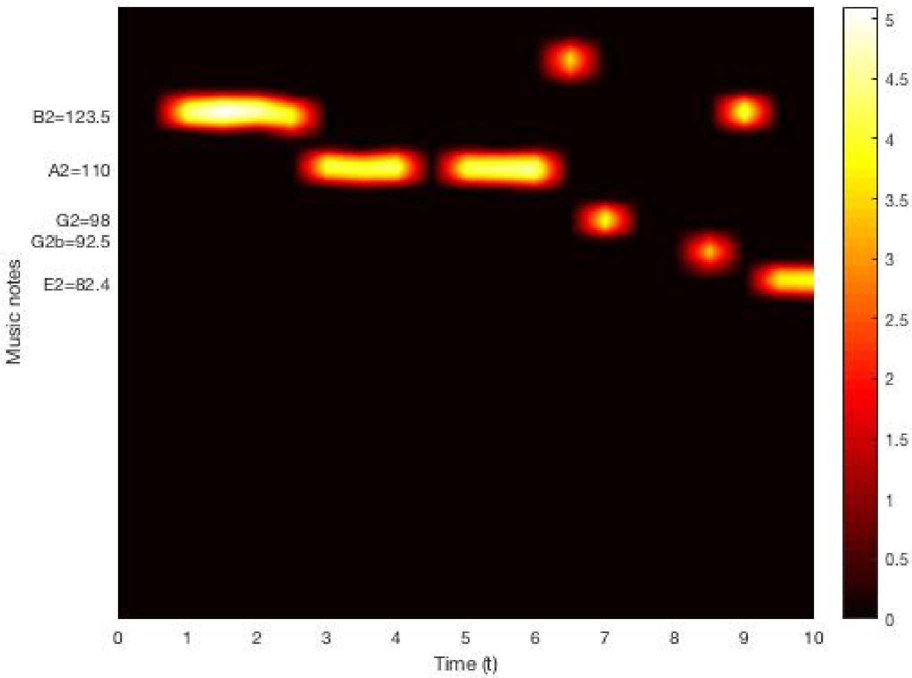

Figure 2: Here is a graph for the Floyd Bass Music Score.

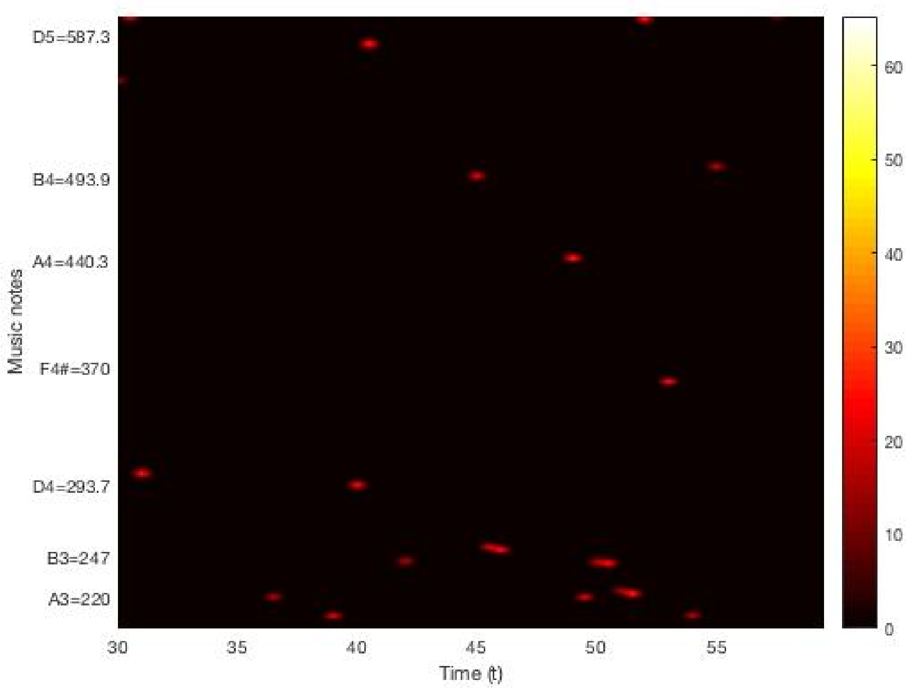

Figure 3: Here is a graph for Guitar Floyd Music Score.

Appendix B MATLAB Code

```
%% Part 1 GNR Guitar Frequencies
clear; close all; clc;
[y1, Fs1] = audioread('GNR.m4a');
tr_gnr = length(y1)/Fs1; % record time in seconds

Ltime = tr_gnr;
le = length((1:length(y1))/Fs1);
ti = linspace(0,Ltime,le+1);
t = ti(1:le);
k = (1/Ltime)*[0:le/2-1 -le/2:-1];
kshift = fftshift(k);
y = y1';

val = 500;

tau = 0:0.1:Ltime;
filt_Yft_spec = zeros(length(y), length(tau));

for j = 1:length(tau)
    filter = exp(-val*(t - tau(j)).^2);
    Yf = filter .* y;
    Yft = fft(Yf);

    [Max, Ind] = max(abs(Yft));
    [Max_Ind] = ind2sub(size(Yft), Ind);
    Max_Val = abs(k(Max_Ind));

    fft_tau = 0.0001;
    fft_filt = exp(-fft_tau*(k - Max_Val).^2);
    filt_Yft = fft_filt .* Yft;

    filt_Yft_spec(:,j) = (fftshift(abs(filt_Yft)));
end

figure(1);
pcolor(tau, kshift, abs(filt_Yft_spec));
colormap(hot)
shading interp
ylim([200 900])
colorbar
xlabel('Time (t)'), ylabel('Music notes')
yticks([277.2, 311.1, 370, 415.3, 554.4, 698.5, 740])
yticklabels({'C4#=277.2', 'D4#=311.1', 'F4#=370', 'G4#=415.3', 'C5#=554.4', 'F5=698.5', 'F5#=740'})

%% Part 2 - Floyd Bass Fequencies
clear; close all; clc;
[y2, Fs2] = audioread('Floyd.m4a');
tr_floyd = length(y2)/Fs2;

Ltime = tr_floyd;
le = length((1:length(y2)-1)/Fs2);
ti = linspace(0,Ltime,le+1); t = ti(1:le);
```

```
k = (1/Ltime)*[0:le/2-1 -le/2:-1];
kshift = fftshift(k);

y = y2(1:length(y2)-1)';

val = 100;
tau = 0:0.5:10;
filt_Yft_spec = zeros(length(y), length(tau));

for j = 1:length(tau)
    gabor = exp(-val*(t - tau(j)).^2);
    Yf = gabor .* y;
    Yft = fft(Yf);

    [Max, Ind] = max(abs(Yft));
    [Max_Ind] = ind2sub(size(Yft), Ind);
    Max_Val = abs(k(Max_Ind));

    fft_tau = 0.1;
    fft_filt = exp(-fft_tau*(k - Max_Val).^2);
    filt_Yft = fft_filt .* Yft;

    filt_Yft_spec(:,j) = (fftshift(abs(filt_Yft)));

end

figure(2);
pcolor(tau,kshift,log(abs(filt_Yft_spec)+1));
colormap(hot)
shading interp
ylim([0 150])
colorbar
xlabel('Time (t)'), ylabel('Music notes')
yticks([82.4, 92.5, 98, 110, 123.5])
yticklabels({'E2=82.4', 'G2b=92.5', 'G2=98', 'A2=110', 'B2=123.5'})

%% Part 3 - Floyd Guitar Frequencies
% clear; close all; clc;
[y2, Fs2] = audioread('Floyd.m4a');
tr_floyd = length(y2)/Fs2;

Ltime = tr_floyd;
le = length((1:length(y2)-1)/Fs2);
ti = linspace(0,Ltime,le+1); t = ti(1:le);
k = (1/Ltime)*[0:le/2-1 -le/2:-1];
kshift = fftshift(k);

y = y2(1:length(y2)-1)';

prefilt_Yft = fftshift(fft(y));
prefilt_Yft(abs(prefilt_Yft) >= 300) = 0;
prefilter_Y = ifft(ifftshift(prefilt_Yft));

val = 100;
```

```
tau = 30:0.5:Ltime;
filt_Yft_spec = zeros(length(y), length(tau));

for j = 1:length(tau)
    gabor = exp(-val*(t - tau(j)).^2);
    Yf = gabor .* prefilter_Y;
    Yft = fft(Yf);

    [Max, Ind] = max(abs(Yft));
    [Max_Ind] = ind2sub(size(Yft), Ind);
    Max_Val = abs(k(Max_Ind));

    fft_tau = 0.1;
    fft_filt = exp(-fft_tau*(k - Max_Val).^2);
    filt_Yft = fft_filt .* Yft;

    filt_Yft_spec(:,j) = (fftshift(abs(filt_Yft)));

end

figure(3);
pcolor(tau,kshift,filt_Yft_spec);
colormap(hot)
shading interp
ylim([200 600])
colorbar
xlabel('Time (t)'), ylabel('Music notes')
yticks([220, 247, 293.7, 370, 440.3, 493.9, 587.3])
yticklabels({'A3=220', 'B3=247', 'D4=293.7', 'F4#=370', 'A4=440.3','B4=493.9','D5=587.3'})
```