AMATH 482 Homework 2

Biyao Li

February 6, 2020

Abstract

In this assignment, I will be investigating two music clips, Sweet Child O' Mine by Guns N' Roses and Comfortably Numb by Pink Floyd, by transforming the data in time domain to frequency domain using Gabor transform. Using the data in frequency domain, I will create a spectrogram that will help us to analyze the music score of bass and guitar of two clips.

1 Introduction and Overview

When analyzing music data in spatial domain, we have a lot of information on when each note is being played but we don't know any information about the frequency of each note. From the previous assignment, I used the Fast Fourier Transform to put the data into frequency domain but we will lose the time information of each frequency. To find the right balance between time and frequency, I will be using the Gabor transform which allows us to isolate each chunk of data in the time domain by applying a filter and then apply the Fourier transform to each chunk. Then by finding the central frequency of each chunk of transformed data, we will have the frequency/music score of the note that was played in each clip. In the latter part of this assignment, I will isolate the bass notes and find the guitar solo from the Comfortably Numb music clip by applying Gaussian and Shannon filters around the frequency of bass notes and its overtones in the frequency domain.

2 Theoretical Background

2.1 Fourier Transform

Recall from the previous assignment, the Fourier Transform of a function $f(x)$ is defined as:

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

and the Inverse Fourier Transform of $\hat{f}(x)$ as:

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

As I mentioned in the introduction, the Fourier Transform will lose the time information of the data and to know when each note was played in frequency domain, we need to use a Gabor Transform instead.

2.2 Gabor Transform

Dennis Gabor defines Gabor Transform as:

$$\hat{f}(x) = \int_{-\infty}^{\infty} f(t) g(t-\tau) e^{-ikt} dt \tag{3}$$

where $g(t-\tau)$ is the filter function that is shifted by τ so that we will have the frequency component near the time τ. In this assignment, I will be using a Gaussian filter such that $g(t-\tau)$ is equal to

$$e^{-a(t-\tau)^2} \tag{4}$$

where a represent the window size of the filter. If a is a large value, we will have a small window size which means that there would be no frequency components. On the other hand, if a is a small value, we will have a wide window and there would be no time information. In this assignment, I will be using different values of a to investigate the difference between the corresponding spectrograms.

2.3 Spectrogram

Spectrogram is way to visualize the frequency spectrum vary with time. In this assignment, all spectrogram are constructed by stacking the Gabor transformed data next to each other such that the x-axis will be all values of τ and y-axis will be value of frequency in Hz.

2.4 Overtone

Overtones are integer multiple of a fundamental frequency. For example, if the fundamental frequency of a bass note is 200 Hz then its first overtone is at 400 Hz and second overtone is at 600 Hz.

2.5 Shannon Filter

The Shannon filter is a step function that takes values of 0 and 1 and it will remove all frequencies under or above a certain threshold. This filter will be used for finding the guitar solo of Comfortably Numb.

3 Algorithm Implementation and Development

3.1 Loading music clip

The provided code of this assignment will convert the $m4a$ formatted music file into vector that represent the music by using the MATLAB function **audioread**. Line 2 to 11 in appendix B is used to convert Sweet Child O' Mine and 48 to 56 is used to convert Comfortably Numb. Note that the **audioread** function returns two values where y represent the total amount of sampled data and Fs represents the sampling rate per second. Hence, $\frac{y}{Fs}$ is the time of each clip.

3.2 Defining domain

First, I define L as the time slot of each clip and the frequency domain ks that range from $(-\frac{n}{2}, \frac{n}{2}-1)$. Even though Fast Fourier Transform assumes a 2π periodic signal, in order to have the frequency with unit of Hertz so that we can find the corresponding note, we only need to scale the frequency domain by $\frac{1}{L}$. Furthermore, I used **fftshift** on frequency domain so that the domain have negative frequencies first and then the positive frequencies. Lines 13 to 19 of the code in appendix B define the domains for Sweet Child O' Mine and Lines 58 to 64 define the domains for Comfortably Numb.

3.3 Gabor Transform

To perform Gabor Transform on the given music clip data, I defined **tau** as a vector of time which ranges from 0 to L with increment of 0.1 seconds for Sweet Child O' Mine and 0.5 seconds for Comfortably Numb. This vector represents the time that each Gaussian filter is centered on. When looping through each element of τ, we need to perform the following procedure:

- Apply the Gaussian Filter (4) shifted by the current element of **tau** to the given data of the music clip.
- Apply **fft** and **fftshift** function to the filtered data so that the data will be in frequency domain.

- Find the maximum value of each transformed data which will give us the most dominating frequency in each time step. By calling the **max** function on the transformed data which will return the index of the maximum value and we can find the maximum frequency by plugging in the index into the frequency domain **ks**.

At the end of each iteration of the for loop, I will store the maximum frequency into a row vector **notes** and store the transformed data into a matrix **Sgt_spec** which will be used to generate the spectrogram. Note that for Comfortably Numb, I split the clip into 4 portions and performed the above procedure separately and this is due to memory shortage issue with MATLAB. This part of the code in appendix B are lines 21 to 30 for Sweet Child O' Mine and lines 68 to 118 for Comfortably Numb.

3.4 Spectrogram

To produce the spectrogram, I used the **pcolor** function that takes in three parameters: vector **tau**, frequency domain **ks**, and **Sgt_spec** which is a matrix of Fourier transformed data. This part of the code in appendix B are lines 40 to 46.

3.5 Constructing the music score

To have a good visualization of the music score for both clips, I created a plot for the maximum frequencies stored in the vector **notes** where the x-axis is the time span **tau** and y-axis is frequency in Hz. In order to know the corresponding note of each maximum frequency, I used the **yticks** and **yticklabels** function to create labels of note on the y-axis. With the labels, we can easily tell the note of each point in the figure 3. This part of the code in appendix B are lines 32 to 38 for Sweet Child O' Mine and lines 120 to 127 for Comfortably Numb.

3.6 Isolating the bass in Comfortably Numb

For the second part of this assignment, the goal is to isolate the bass notes that is at the bottom of the spectrogram for Comfortably Numb. This should results in a spectrogram without the overtones of the bass notes and notes created by other instruments. This can be achieved by applying a Gaussian filter to the Gabor transformed data around the centre frequency that is in-between 100 to 250 Hz which is the range of frequency for bass. This part of the code in appendix B are lines 129 to 137.

3.7 Recreating the Guitar solo in Comfortably Numb

For the last part of this assignment, I will try to find the guitar solo in the clip of Comfortably Numb. From the spectrogram for isolated bass in figure 4, it is obvious that the frequency of bass range from 50 to 250. Hence, we can apply a Shannon filter that set the value with the frequency less than 255 in the Gabor transformed data to 0 so that all bass notes will be filtered. To take care of the overtones of each bass note, we can apply a special Gaussain filter to each of the overtone where we filter the frequency that is close to the center frequency(overtone). This can be done by subtracting a normal Gaussian filter from 1. This part of the code in appendix B are lines 139 to 159.

4 Computational Results

4.1 Spectrogram

For the music clip of Sweet Child O' Mine, I plotted two spectrograms in Figure 1 that correspond to Gabor transforms with the α value of 1 and 100. As I mentioned previously, as α get larger, the window size of Gabor filter get smaller and we will lose a lot of information about frequency. This can be supported by the right part of figure 1 since we can see that each note is really localized in time when the window size is small. Comparing to a large window size on the left, we don't have a lot of time information of each note since it is more localized in frequency.

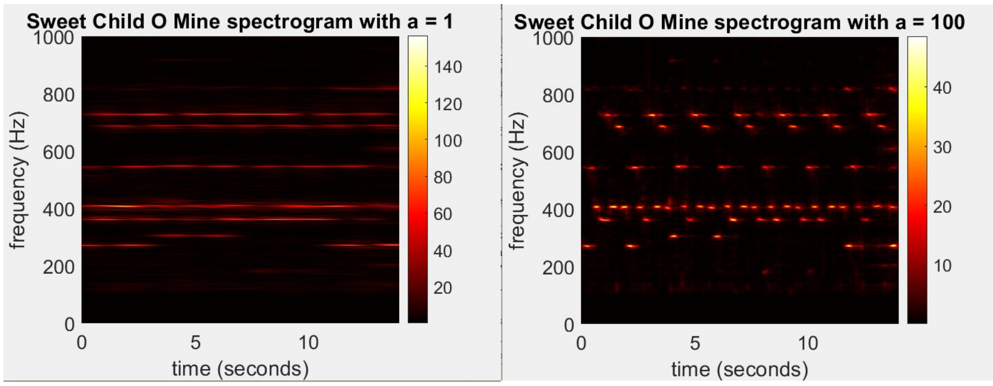


Figure 1: Spectrograms for Sweet Child O' Mine with α of 1 and 100

For the spectrogram of Comfortably Numb in Figure 2, I plotted it with taking the Gabor transform of α equals 1000. From the spectrograms, we can see that there are many yellow spots that is between 50 Hz to 200 Hz and these yellow spots represents the strongest frequency of that time. If we listen to the given music clip, we can tell that the most dominate instrument is the bass which implies that these yellow spots are the notes for bass. Similarly, since the most dominant instrument in the clip of Sweet Child O' Mine is guitar, the yellow spots in the spectrogram on the right of figure 1 represent the note of the guitar played at each time. Moreover, the spectrogram in Figure 2 is for the first 15 seconds of Comfortably Numb since the bass notes have a repeated pattern which is labelled inside the white box of Figure 2. This pattern is more visible in plot on the right of Figure 3.

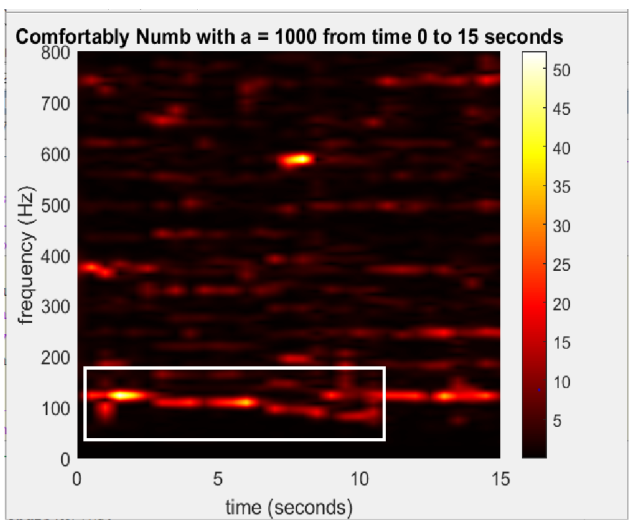
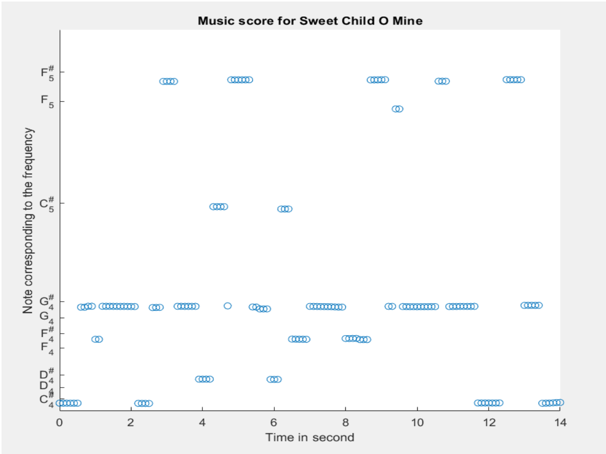


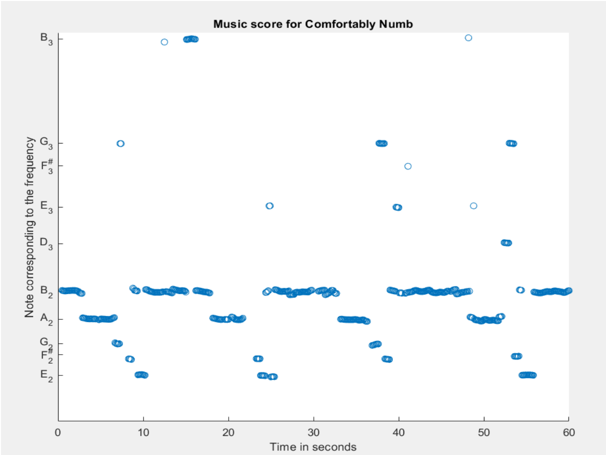
Figure 2: Spectrograms for Comfortably Numb with α of 100

4.2 Music score

Based on the previous observations of the spectrograms, it appears that the maximum/strongest frequencies in each of the Gabor transformed data of Sweet Child O' Mine represents the note played by the guitar while the maximum frequenies in range of 100 to 250 in the Gabor transformed data of Comfortably Numb represent the note played by the bass. By following the procedure in **3.5**, we will get figure 3 that includes the music score for Sweet Child O' Mine on the left and the music score for Comfortably Numb on the right. Note that the frequency for figure 3 range from 0 to 800 Hz and frequency for figure 4 range from 0 to 250 Hz.



(a) Music score for Sweet Child O' Mine with



(b) Music score for Comfortably Numb

Figure 3: Music scores

4.3 Isolated bass

By following the procedure of **3.6**, we should get the spectrogram in Figure 4 where the only frequencies that is visible are the frequencies of the bass notes which is exactly what we want for the part 2 of this assignment. Moreover, the pattern of the bass notes from time 0 to 10 seconds become more visible without the overtones and unrelated notes.

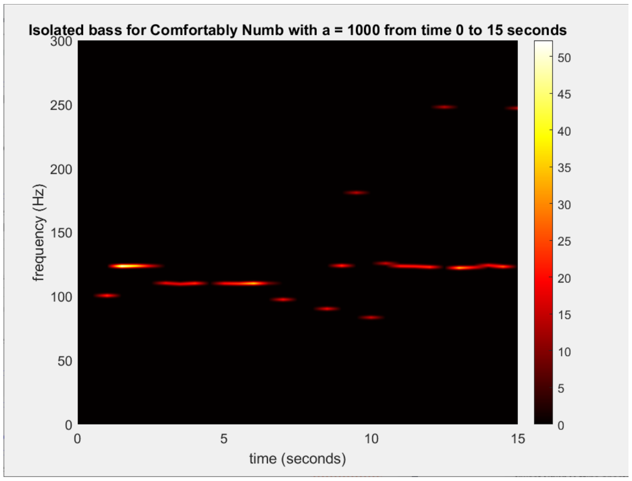


Figure 4: Isolated bass notes for Comfortably Numb from 0 to 15 seconds

4.4 Guitar Solo

Running the last section of the code in appendix B should output the spectrogram in Figure 5. As we hoped, the bass notes and the overtones are mostly filtered and there are still a couple of visible notes left. Based on my research, the frequencies for guitar typically range from 100 to 1000 Hz which is why I think the two circled notes in the spectrogram are the notes of guitar. Moreover, we can see that there are some unfiltered overtones under the circled note at 8 seconds which could belong to other instruments in the clip such as drums and keyboard.

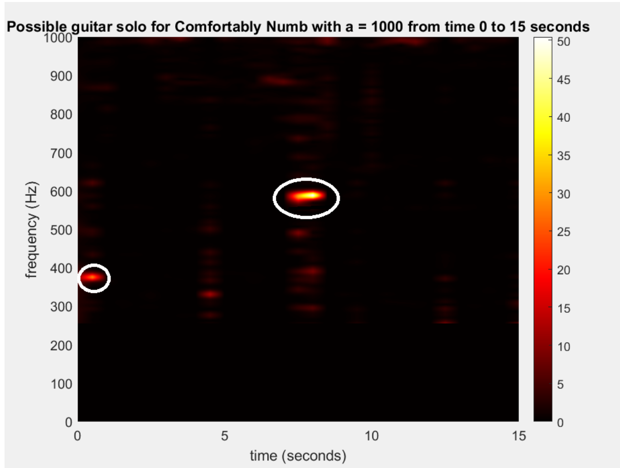


Figure 5: Guitar solo in Comfortably Numb spectrogram from time 0 to 15 seconds

5 Summary and Conclusions

In this assignment, we are given two music clip data in the time domain and we applied the Gabor transform to put the data into the frequency domain. The main advantage of the Gabor transform is that it allows us make the data more localized in time or frequency by adjusting the window width. With the Gabor transformed data, we plotted spectrograms that allows us to visualize the data in frequency domain and it is clear that the central frequencies represent the notes played by the guitar and the bass. Then, we created the music score for both clip using the central frequency at each time step. For the latter part of this assignment, we applied filters in the frequency domain in attempt to isolated the bass notes and the guitar solo in the spectrogram of Comfortably Numb. This assignment is really rough for me, mainly because I had a lot of issues with MATLAB not having enough memory and making my computer not able to function properly. But I think all of the frustrations is worth it after seeing these nice results.

Appendix A MATLAB Functions

Add your important MATLAB functions here with a brief implementation explanation. This is how to make an **unordered** list:

- `[y, Fs] = audioread(filename)` reads data from the file named filename, and returns sampled data, y, and a sample rate for that data, Fs.
- `Y = fftshift(X)` rearranges a Fourier transform X by shifting the zero-frequency component to the center of the array.
- `[M,I] = max(A)` return a maximum value M in the matrix A and the corresponding index of M.
- `scatter(x,y)` creates a scatter plot with circles at the locations specified by the vectors x and y. This type of graph is also known as a bubble plot.
- `pcolor(C)` creates a pseudocolor plot using the values in matrix C.
- `yticks(ticks)` sets the y-axis tick values, which are the locations along the y-axis where the tick marks appear.
- `yticklabels(labels)` sets the y-axis tick labels for the current axes.

Appendix B MATLAB Code

Add your MATLAB code here. This section will not be included in your page limit of six pages.

```
clear all; close all; clc
%% Load GNR song
figure(1)
[y, Fs] = audioread('GNR.m4a');
tr_gnr = length(y)/Fs;
% record time in seconds
plot((1:length(y))/Fs,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Sweet Child O Mine');
p8 = audioplayer(y,Fs);
playblocking(p8);
%% Filtering for GNR
yT = y';
L = 14;
n = length(yT);
t2 = linspace(0,L,n+1); t = t2(1:n);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
tau = 0:0.1:14;
a = 100;
notes = zeros(1,length(tau));
for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2);
   Sg = g.*yT;
   Sgt = fft(Sg);
   Sgts = fftshift(abs(Sgt));
   [mv,idx] = max(Sgts);
   notes(j) = abs(ks(idx));
   Sgt_spec(:,j) = Sgts;
end
%% Recreating the music score
scatter(tau,notes);
yticks([277.18,293.66,311.13,349.23,369.99,392,415.3,554.37,698.46,739.99]);
yticklabels({'C^{#}_{4}','D_{4}','D^{#}_{4}','F_{4}','F^{#}_{4}','G_{4}','G^{#}_{4}','C^{#}_{5}','F_{5}
ylim([260 800]);
title('Music score for Sweet Child O Mine')
xlabel('Time in second');
ylabel('Note corresponding to the frequency');
%%
pcolor(tau,ks,guitar)
shading interp
set(gca,'ylim',[0 1000],'Fontsize',12)
colormap(hot)
colorbar
xlabel('time (seconds)'), ylabel('frequency (Hz)')
title('Possible guitar solo for Comfortably Numb with a = 1000 from time 0 to 15 seconds')
%% Load Floyd song
figure(1)
[y, Fs] = audioread('Floyd.m4a');
tr_gnr = length(y)/Fs;
% record time in seconds
```

```
plot((1:length(y))/Fs,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Comfortably Numb');
% p8 = audioplayer(y,Fs);
% playblocking(p8);
%% Initiate the frequency of the each note
yT = y';
yT = yT(1:2635920);
n = length(yT);
L = ceil(tr_gnr);
t2 = linspace(0,L,n+1); t = t2(1:n);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
big_time = 0:0.5:L;
notes = zeros(1,length(big_time));
%% Floyd's clip 0 to 15 sec
tau = 0:0.5:15;
a = 1000;
for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2);
   Sg = g.*yT;
   Sgt = fft(Sg);
   Sgts = fftshift(abs(Sgt));
   [mv,idx] = max(Sgts);
   notes(j) = abs(ks(idx));
   Sgt_spec1(:,j) = Sgts;
end
%% Floyd's clip 15 to 30 sec
tau = 15.5:0.5:30;
nidx = j+1;
a = 1000;
for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2);
   Sg = g.*yT;
   Sgt = fft(Sg);
   Sgts = fftshift(abs(Sgt));
   [mv,idx] = max(Sgts);
   notes(nidx) = abs(ks(idx));
   nidx = nidx + 1;
   Sgt_spec2(:,j) = Sgts;
end
%% Floyd's clip 30 to 45 sec
tau = 30.5:0.5:45;
a = 1000;
for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2);
   Sg = g.*yT;
   Sgt = fft(Sg);
   Sgts = fftshift(abs(Sgt));
   [mv,idx] = max(Sgts);
   notes(nidx) = abs(ks(idx));
   nidx = nidx + 1;
   Sgt_spec3(:,j) = Sgts;
end
```

```
%% Floyd's clip 45 to 60 sec
tau = 45.5:0.5:60;
a = 1000;
for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2);
   Sg = g.*yT;
   Sgt = fft(Sg);
   Sgts = fftshift(abs(Sgt));
   [mv,idx] = max(Sgts);
   notes(nidx) = abs(ks(idx));
   nidx = nidx + 1;
   Sgt_spec4(:,j) = Sgts;
end
%% Recreating music score for Floyd
tau = 0:0.5:60;
scatter(tau,notes);
yticks([82.41,92.5,98,110,123.47,146.83,164.81,185,196,246.94]);
yticklabels({'E_{2}','F^{#}_{2}','G_{2}','A_{2}','B_{2}','D_{3}','E_{3}','F^{#}_{3}','G_{3}','B_{3}'});
ylim([60,250]);
title('Music score for Comfortably Numb');
xlabel('Time in seconds');
ylabel('Note corresponding to the frequency');
%%
cent_freq = 1:1:121;
tau = 0:0.5:15;
for j = 1:length(tau)
   Sgts = Sgt_spec1(:,j)';
   currF = notes(cent_freq(j));
   filter = exp(-0.5*(ks - currF).^2);
   Sgtsf = Sgts .* filter;
   bass(:,j) = Sgtsf;
end
%%
cent_freq = 1:1:121;
tau = 0:0.5:15;
for j = 1:length(tau)
    Sgts = Sgt_spec1(:,j)';
    currF = notes(cent_freq(j));
    filter = ks > 255;
    Sgtsf = Sgts .* filter;
    overtone1 = 1 - exp(-0.0001*(ks - currF*2).^2);
    Sgtsf = Sgtsf .* overtone1;
    overtone2 = 1- exp(-0.0001*(ks - currF*3).^2);
    Sgtsf = Sgtsf .* overtone2;
    overtone3 = 1 - exp(-0.0001*(ks - currF*4).^2);
    Sgtsf = Sgtsf .* overtone3;
    overtone4 = 1 - exp(-0.0001*(ks - currF*5).^2);
    Sgtsf = Sgtsf .* overtone4;
    overtone5 = 1 - exp(-0.0001*(ks - currF*6).^2);
    Sgtsf = Sgtsf .* overtone5;
    overtone6 = 1 - exp(-0.0001*(ks - currF*7).^2);
    Sgtsf = Sgtsf .* overtone6;
    guitar(:,j) = Sgtsf;
end
```