AMATH 482 Homework 2

Tommy Dong

February 9, 2021

Abstract

In this homework, one is supposed to utilize Gabor Transformation to extract frequencies with the assistance of other techniques. In particular, two music clips were offered and the goal is to isolate different instruments out of the clips. This paper contains the process of isolating different frequencies along with theoretical backgrounds and conclusion.

1 Introduction and Overview

There are two music clips, one is from "Sweet Child O' Mine" by Guns N' Roses and the other is "Comfortably Numb" by Pink Floyd. These are two of the greatest rock and roll songs of all time. The goal is to isolate the music score for guitar and bass for both music clips, and there are several steps required.

1.1 Data Description

There are two music clips. One is about 14 seconds long and the other takes about 60 seconds. Using MatLab function we could transform them into two vectors containing critical information.

1.2 Goal

1.2.1 Identify the music score

The first thing required is to identify the music score from both music clips. In particular, one need to use Gabor filtering to reproduce the music score of guitar for "Sweet Child O' Mine" and the music score of bass for "Comfortably Numb".

1.2.2 Isolating bass from "Comfortably Numb"

For "Sweet Child O' Mine", it is a guitar solo which doesn't require any filtering for different instruments. For "Comfortably Numb", it consists varies instruments, and the goal is to use certain techniques to isolate the bass from the entire "Comfortably Numb" clip.

1.2.3 Guitar Solo for "Comfortably Numb"

Following the isolation of the bass from " Comfortably Numb", now the new object is to extract guitar solo from it as contact as possible.

2 Theoretical Background

2.1 Gabor Transform

Gabor Transformation is mainly focus on shifting the center of the frequency while filtering the signal, where $g(t)$ is the filtering function and multiply by $f(t)$ yields the filtered the function $f(t)g(t-\tau)$, where the function centered at τ.

$$\tilde{f}_g(\tau, k) = \int_{-\infty}^{\infty} f(t)g(t-\tau)e^{-ikx}\,dk \tag{1}$$

2.2 Fourier Transform

This is the Fourier Transform equation. It can change a function of time ,x, into a function of frequency, k.

$$\hat{f}(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-\mathrm{i}kx}\, dx \tag{2}$$

2.3 Gaussian Equation

Gaussian equation is the filter in this project for its bell-shape property can assist to amplify the signal desired and remove the influence of other signals if filter around the center signal.

$$f(x) = e^{-\tau(x-x_0)^2} \tag{3}$$

3 Algorithm Implementation and Development

1. Gabor Filtering
   Gabor filtering is based on Fourier Transform, but Gabor Transform is more advanced for it can in some level manage to certify not only the frequency signals but also locate the time where signals appear. To achieve that, one needs to set up a window around the center frequency, and then shift the center frequency along with time in order to gather the entire information. One problem Gabor Transformation facing is that it cannot be accurate on both getting frequency and locate the time the frequency appear. The challenge is to find a balance between these two information so that we can have a whole image of the data as accurate as possible.

   In this project, we pick a relatively small window size, $a = 100$, in order to balance between getting clean signal and locate the time period it showed up. As we implemented the algorithm, we get a neat spectrogram heat map like Figure 1, on which the red and yellow lines represent notes played in the music clips.

2. Filtering frequency
   Different from last project which the real signals are hidden in noises, this time we want to isolate the signal for bass in "Comfortably Numb". The theoretical background is similar though, using the property of filtering function to amplify the frequency signatures of target frequencies in order to acquire a clean music score.

   For this project, what we did is first find out the loudest sound played in each time window. Then we applied filter around that frequency in order to clean the spectorgram heat map. It worked like a charm and we managed to recoginze notes played by bass in "Comfortably Numb" from Figure 3 and recorded them in Table 2.

4 Computational Results

4.1 Guitar from "Sweet Child O' Mine"

We obtained the notes played by guitar in "Sweet Child O' Mine" from its spectrogram and listed it in Table 1. Since "Sweet Child O' Mine" is a guitar solo, all the notes are played by guitar and it didn't require any filtering for exclude other instruments.

Notes	Ocatave	Frequency
C#/Db	3	277.18
D#/Eb	3	311.13
F#/Gb	3	369.99
G#/Ab	3	415.30
C#/Db	4	554.37
F	4	698.46
F#/Gb	4	739.99
C#/Db	5	1108.7
D	5	1174.7
D#/Eb	5	1244.5

Table 1: Guitar notes played in "Sweet Child O' Mine"

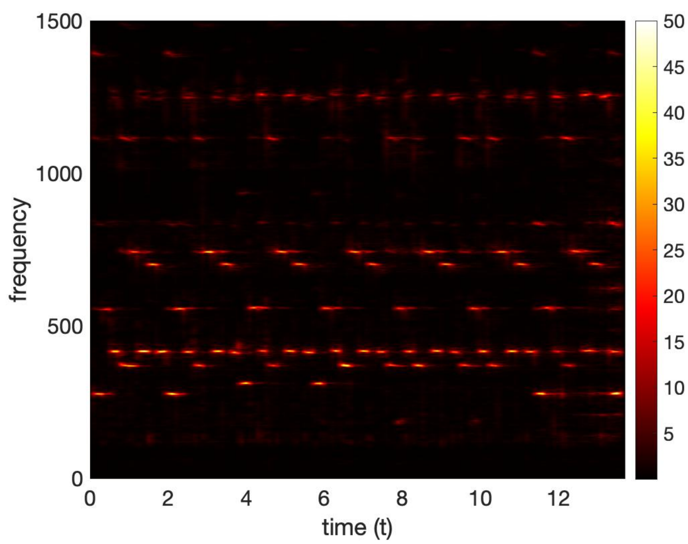

Figure 1: Spectrogram of the notes in "Sweet Child O' Mine"

4.2 Bass from "Comfortably Numb"

For "Comfortably Numb", it contains several instruments. The goal is to identify the notes played by bass and isolate them out. Bass has a property that the frequencies it produces are all between 60 Hz and 250 Hz according to my Google search. So when producing spectrogram, I limited the frequency axis between 0 and 300 in order to concentrate on bass notes' search. Figure 2 is the spectrogram for all the notes from "Comfortably Numb" within the 0 and 300 frequency interval. Figure 3 is the result of applying filter function on certain notes in order to acquire clean music score. From Figure 3, 5 notes were identified and listed in Table 2.

Notes	Ocatave	Frequency
E	1	82.407
F#/Gb	1	92.499
G	1	97.999
A	2	110.00
B	2	123.47

Table 2: Guitar notes played in "Sweet Child O' Mine"

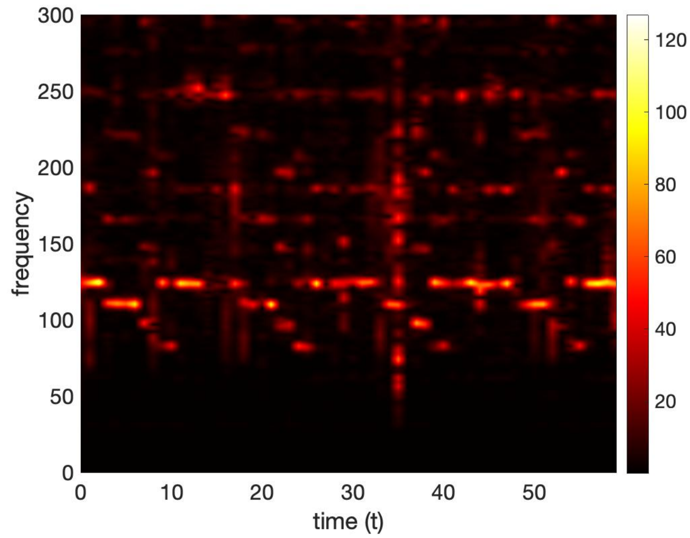

Figure 2: Spectrogram of the all notes in "Comfortably Numb"

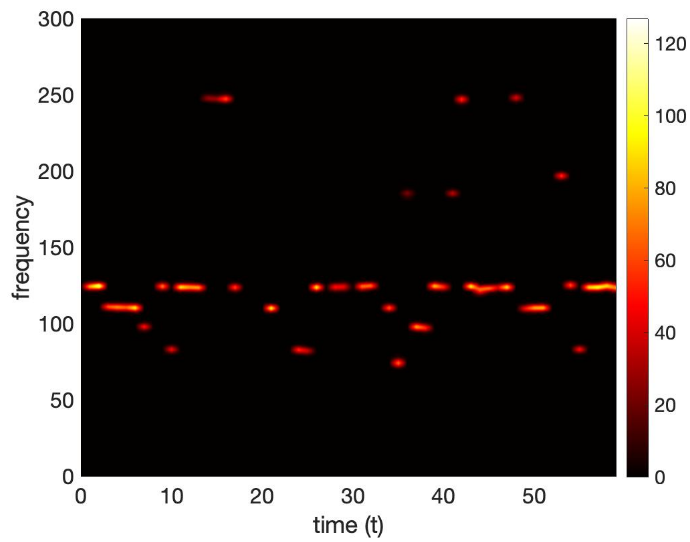

Figure 3: Spectrogram of "Comfortably Numb" with only notes played by bass

Notes	Ocatave	Frequency
E	2	164.81
F#/Gb	2	185.00
G	2	196.00
B	3	246.94
E	3	329.63
F#/Gb	3	369.99
G	3	392.63
A	4	440.00
B	4	493.88
D	4	587.33
F#/Gb	4	739.99
G	4	783.99

Table 3: Guitar solo notes played in "Comfortably Numb"

4.3 Guitar Solo for "Comfortably Numb"

For isolating guitar solo from "Comfortably Numb", we initially ruled out the notes played by bass from last part of the question. Then we followed the similar procedure as the last part, but instead we centered and filtered around notes possibly played by guitar now. There existed a problem of overtone of the bass which made some notes' frequencies. The solution I implemented is to check the times of each potential guitar notes being played, for the guitar in the song repeats all along, and it may be possible to rule out notes only played once or twice in the song. In the end, we identified several notes played by the guitar and listed them in Table 3. The spectrogram after filtering is Figure 4.

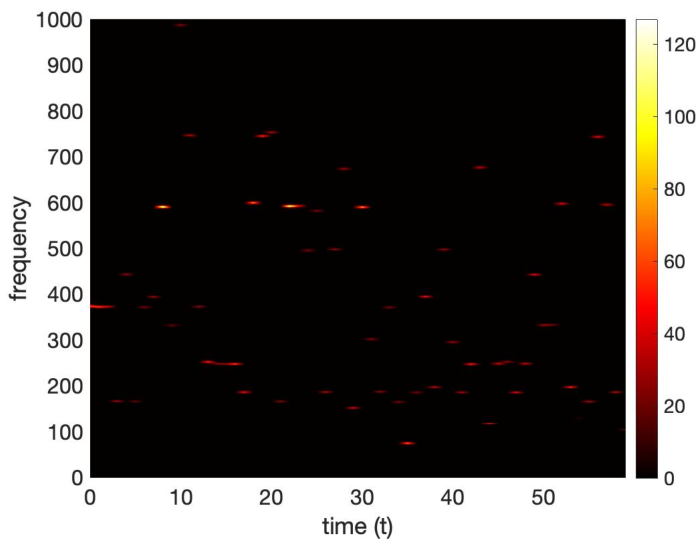

Figure 4: Spectrogram of "Comfortably Numb" with only notes played by bass

5 Summary and Conclusions

The primary focus for this project is to utilize Gabor Transformation to get the result. There exists several problems, like the balancing between accurate frequency signature and accurate time period which required to pick window sizes cautiously. Also, in real life implementation, the time of iteration is also crucial for a gigantic data may crush the computer if iterate too many times.

For a load of frequencies across time, it would be better to look at them with a limited time period instead of the whole range. Also, with the help of spectrogram combining the information acquired from Gabor Transformation, we can get a general view of frequencies and the time period they exist. In this case, we utilize these techniques to identify notes by its frequency and then by using Gabor filtering to locate the time window this note was played. For identifying different instruments, we again applied Gaussian function to amplify the desired frequencies in order to get a clean music score.

Appendix A MATLAB Functions

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

- `y = linspace(x1,x2,n)` returns a row vector of `n` evenly spaced points between `x1` and `x2`.
- `[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 = fft(X)` computes the discrete Fourier transform (DFT) of `X` using a fast Fourier transform (FFT) algorithm.
- `Y = fftshift(X)` rearranges a Fourier transform `X` by shifting the zero-frequency component to the center of the array.
- `[M,I] = max(A)` returns the linear index into `A` that corresponds to the overall maximum value in `A`.

Appendix B MATLAB Code

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

```
%% Clean workspace
clear all; close all; clc
format shortG
%% import files
[y1, Fs1] = audioread('GNR.m4a');
tr1 = length(y1)/Fs1;
[y2, Fs2] = audioread('Floyd.m4a');
tr2 = length(y2)/Fs2;
y2=y2(1:length(y2)-1);
%% Part 1
% GNR
n1 = length(y1);
t1 = linspace(0,tr1,n1+1); t1 = t1(1:n1);
k1 = (1/tr1)*[0:n1/2-1 -n1/2:-1];
ks1 = fftshift(k1);
a = 100;
tau1 = 0:0.1:tr1;
gnr_notes=zeros(length(tau1),1);
for j = 1:length(tau1)
   g1 = exp(-a*(t1 - tau1(j)).^2);
   yg1 = g1.'.*y1;
   yg1t = fft(yg1);
   [M,i]=max(abs(yg1t));
   gnr_notes(j)=k1(i);
   yg1t_spec(:,j) = fftshift(abs(yg1t));
end
round(gnr_notes,2)
figure(1)
pcolor(tau1,ks1,yg1t_spec)
shading interp
set(gca,'ylim',[0 1500],'Fontsize',16)
colormap(hot)
colorbar
xlabel('time (t)'), ylabel('frequency')
```

Listing 1: Part 1

```
%%
% Floyd
n2 = length(y2);
t2 = linspace(0,tr2,n2+1); t2 = t2(1:n2);
k2 = (1/tr2)*[0:n2/2-1 -n2/2:-1];
ks2 = fftshift(k2);
a = 100;
tau2 = 0:1:tr2;
gau_t2au=0.1;
floyd_notes=zeros(length(tau2),1);
for j = 1:length(tau2)
   g2 = exp(-a*(t2 - tau2(j)).^2); % Window function
   yg2 = g2.'.*y2;
   yg2t = fft(yg2);
   yg2t_spec(:,j) = fftshift(abs(yg2t)); % We don't want to scale it

   [M,idx2]=max(abs(yg2t));
   floyd_notes(j)=k2(idx2);

   gau_filter=exp(-gau_tau2*(k2-abs(k2(idx2))).^2);
   yg2t_filtered=gau_filter'.*yg2t;
   yg2t_spec_filtered(:,j)=fftshift(abs(yg2t_filtered));
end
% k2(i)
figure(2)
pcolor(tau2,ks2,yg2t_spec)
shading interp
set(gca,'ylim',[0 300],'Fontsize',16)
colormap(hot)
colorbar
xlabel('time (t)'), ylabel('frequency')
figure(3)
pcolor(tau2,ks2,yg2t_spec_filtered)
shading interp
set(gca,'ylim',[0 300],'Fontsize',16)
colormap(hot)
colorbar
xlabel('time (t)'), ylabel('frequency')
floyd_notes
```

Listing 2: Part 2

```matlab
%% Part 3
% Floyd
n3 = length(y2);
t3 = linspace(0,tr2,n3+1); t3 = t3(1:n3);
k3 = (1/tr2)*[0:n3/2-1 -n3/2:-1];
ks3 = fftshift(k3);
a = 100;
tau3 = 0:1:tr2;
freq_bass=[82.407;92.499;97.999;110.0;123.47];
floyd_notes_guitar=zeros(length(tau3),1);
gau_tau3=0.1
for j = 1:length(tau3)
   g3 = exp(-a*(t3 - tau3(j)).^2);
   yg3 = g3.'.*y2;
   yg3t = fft(yg3);

   for i=1:length(yg3t)
       for k=1:5
           if abs(abs(k3(i))-freq_bass(k))<=5
               yg3t(i)=0;
               break;
           end
       end
   end
%    yg3t_spec(:,j) = fftshift(abs(yg3t));
   [M,idx3]=max(abs(yg3t));
   floyd_notes_guitar(j)=abs(k3(idx3));

   gau_filter=exp(-gau_tau3*(k3-abs(k3(idx3))).^2);
   yg3t_filtered=gau_filter'.*yg3t;
   yg3t_spec_filtered(:,j)=fftshift(abs(yg3t_filtered));
end
floyd_notes_guitar
figure(3)
pcolor(tau3,ks3,yg3t_spec_filtered)
shading interp
set(gca,'ylim',[0 1000],'Fontsize',16)
colormap(hot)
colorbar
xlabel('time (t)'), ylabel('frequency')
```

Listing 3: Part 3