AMATH 482 Homework 2

Fiona Fu

February 2020

Abstract

Analyzing a portion of rock & roll songs by using Gabor Transform to get the spectrogram representing the music scores for each clip.

1 Introduction and Overview

By using the Gabor Transform to modify the vectors representing the music, we had the spectrogram of the main melody performed by certain instruments shown in a graph, and our goal is to:

1.1 Reproduce the music score.

1.1.1 Find Guitar in the GNR clip

In order to get the main music score, we use the Gabor filtering to create a spectrogram with clear corresponding notes.

1.1.2 Find bass in the Floyd clip

Same as the process of reproducing the music score for guitar, we also use Gabor filtering to filter for the bass in this clip.

1.2 Isolate bass in Floyd clip

By filtering certain frequency based on spectrogram, we can get information for the notes on frequency domain and time domain.

1.3 Isolate guitar solo in Floyd clip

A challenging task that requires to reconstruct the music score and identify the guitar in the portion of *Comfortably Numb*.

2 Theoretical Background

As we learned from our lecture [4], ***Gabor transform***, or the short-time Fourier transform (STFT) is given by:

$$\tilde{f}_g(\tau,k) = \int_{-\infty}^{\infty} f(t)g(t-\tau)e^{-ikt}\,dt$$

This can be explained that for a fixed filter center τ, the function gives us information about frequency components near time τ, and the filtered function represented by $f(t)g(t-\tau)$ means shifting the filter function g(t) by τ and multiply the original function.

The introduction of Gabor transform is to solve the question "how can we get both time and frequency information from a signal?" when Fourier transform is limited to get information in both time and frequency domain. By combining the filtering method in Lecture 2 and applying to time domain, we form the basis of STFT, also known as Gabor transform.

When writing a report to visualize things in the time-frequency domain, we could stack all the Fourier transforms next to each other to make a plot with the horizontal direction being the value τ, which is the window center, and the vertical direction--the frequency. This is called a ***spectrogram***, giving a still picture that represents the changing of the Fourier transform as the window slides over the domain.

As we learned from lecture [7], we should keep the choice for the ***window function*** to be simple by using a Gaussian:

$$g(t-\tau) = e^{-a(t-\tau)^2}$$

where τ determines the center of the window and a > 0 determines the width of the window.

3 Algorithm Implementation and Development

Read the data and get the signals of GNR clip, which is the same for Floyd clip.

```
[y, Fs] = audioread('GNR.m4a');
tr_gnr = length(y)/Fs; % record time in seconds
```

To get wavenumber in Hz by scaling the wavenumbers by 1/(length of record time in seconds). It's the same for Floyd clip, the only difference is the L and we split the long clip into three pieces, 20 seconds each. So L = 20 for Floyd Bass.

```
% GNR Guitar
L = 14;  n = length(y);
t2 = linspace(0,L,n+1); t = t2(1:n);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
```

We specify the window size a to be 20 and τ to be 0.1 here.

```
a = 20;
tau = 0:0.1:14;
```

Plotting the spectrogram follow the formula of Gabor transform by multiplying g(t), shifted by τ, to the original function. Same process and formula for Floyd clip except the range of y-axis.

```
for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2); % Window function
   Yg = g.*transpose(y); % Signal filter
   Ygt = fft(Yg);
   Ygt_spec(:,j) = fftshift(abs(Ygt));
end
pcolor(tau,ks,log(abs(Ygt_spec)+ 1))
shading interp
set(gca,'ylim',[0 1000],'Fontsize',12)
colormap(hot)
colorbar
```

By changing the range of y-axis scaled 0 to 300, we can identify the bass notes clearly as shown in figure 3, 4, 5 below.

In order to isolate for the frequency of guitar, we exclude the frequency under 150 Hz to prevent from the effects of Bass frequency. Then the range will be from 200 to 1000 Hz.

```
for j = 1:length(tau)
    g = exp(-a*(t - tau(j)).^2); % Window function
    Yf = g.*transpose(y1); %Signal filter
    Yft = fft(Yf);
    [f_max, ind] = max(Yft);
    freq_max = [freq_max;k(ind)];
    Yft_spec(:,j) = abs(fftshift(Yft));
end
```

W filter the signals by keeping the maximum frequency in the Fourier domain and plug back the location of maximum point to the wavenumber, and finally we had the isolated guitar solo as shown in Figure 6.

4 Computational Results

GNR-Guitar:

In Figure 1, the main notes played by guitar is clearly identifiable, and the closer look at frequency around 200 to 500 Hz in Figure 2.

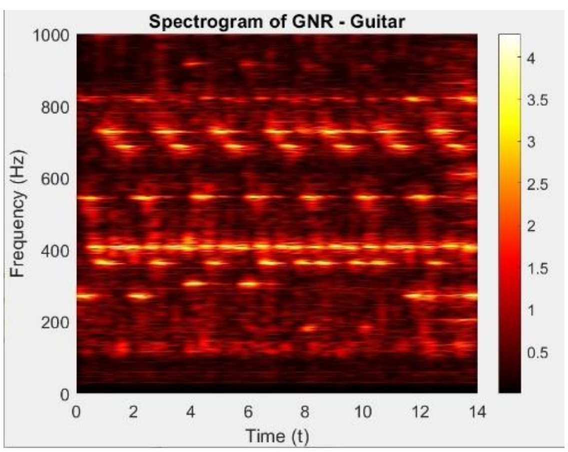

Figure 1

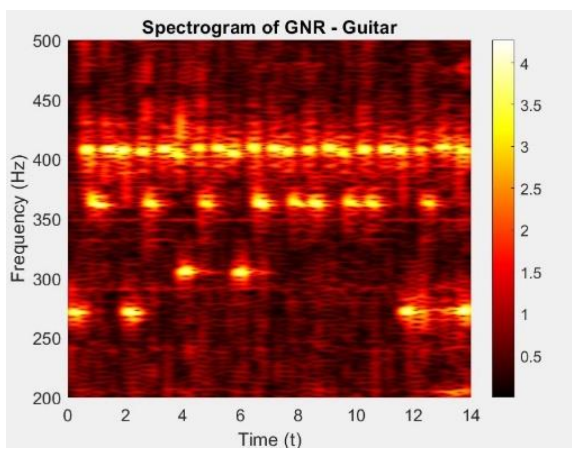

Figure 2

The music score of the guitar in GNR has three different chords, each repeated for two times, which played in the following sequence:

Start with $C^{\#}$	Start with $D^{\#}$	Start with $F^{\#}$	Start with $C^{\#}$
$C^{\#}C^{1\,\#}G^{\#}\ F^{\#}\ F^{1\,\#}G^{\#}\ D^{1\,\#}G^{\#}$	$D^{\#}C^{1\,\#}G^{\#}\ F^{\#}\ F^{1\,\#}G^{\#}\ D^{1\,\#}G^{\#}$	$F^{\#}C^{1\,\#}G^{\#}\ F^{\#}\ F^{1\,\#}G^{\#}\ D^{1\,\#}G^{\#}$	$C^{\#}C^{1\,\#}G^{\#}\ F^{\#}\ F^{1\,\#}G^{\#}\ D^{1\,\#}G^{\#}$

There exist four chords in this clip and only the start score is different, so we get the final music score printed as:

$C^{\#}C^{1\,\#}G^{\#}\ F^{\#}\ F^{1\,\#}G^{\#}\ D^{1\,\#}G^{\#}$, $C^{\#}C^{1\,\#}G^{\#}\ F^{\#}\ F^{1\,\#}G^{\#}\ D^{1\,\#}G^{\#}$, $D^{\#}C^{1\,\#}G^{\#}\ F^{\#}\ F^{1\,\#}G^{\#}\ D^{1\,\#}G^{\#}$, $D^{\#}C^{1\,\#}G^{\#}\ F^{\#}\ F^{1\,\#}G^{\#}\ D^{1\,\#}G^{\#}$, $F^{\#}C^{1\,\#}G^{\#}\ F^{\#}\ F^{1\,\#}G^{\#}\ D^{1\,\#}G^{\#}$, $F^{\#}C^{1\,\#}G^{\#}\ F^{\#}\ F^{1\,\#}G^{\#}\ D^{1\,\#}G^{\#}$, $C^{\#}C^{1\,\#}G^{\#}\ F^{\#}\ F^{1\,\#}G^{\#}\ D^{1\,\#}G^{\#}$, $C^{\#}$.

Floyd-Bass:

Below in Figure 3, 4, 5, we got the spectrogram of Floyd clip and tried to identify the bass notes.

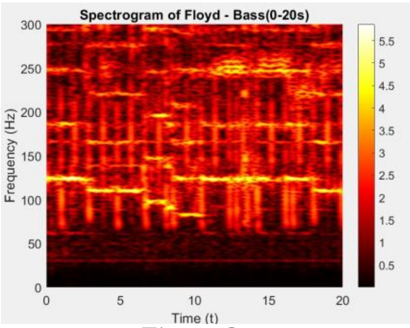

Figure 3

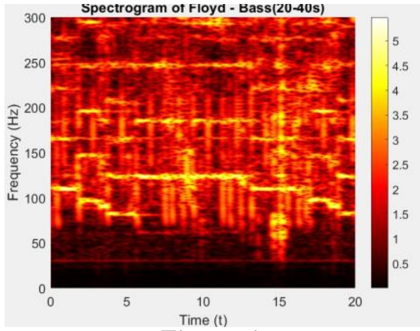

Figure 4

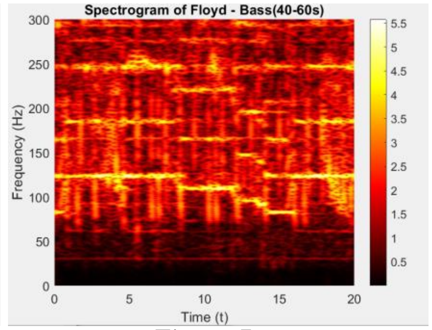

Figure 5

The music score of the bass in Floyd clip can be identified by the relatively strong notes, so we can reproduce it as about:

Repeated: B(for about 12 seconds), A(for 5 seconds), G, $F^{\#}$, E each for about 1 second.

The frequency of the bass is around 80 to 150 Hz with a deep, strong sound in the background.

Isolated Guitar Solo:

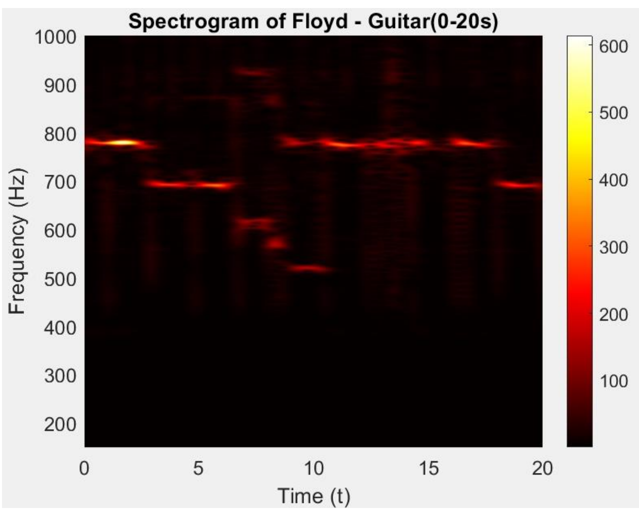

Figure 6

We can find the identifiable notes of the guitar solo in the first 20 seconds in Figure 6.

3 Summary and Conclusions

- The clip played by guitar in *Sweet Child O' Mine* can be reproduced in the music score of the repeated three chords, each start with a different note while other notes are all the same.
- The bass in Comfortably Numb are mainly ranged from 50 to 150 Hz, each lasting

for 4 series of one long chord, represented by C^{-1}: B--, A-, G, $F^{\#}$, E within the 60 seconds clip.

- It's hard to isolate the guitar solo in Comfortably Numb although it's clearly identifiable in the music clip. It still consists of a few similar chords with the popular melody.

References

[1] Jose Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.

[2] MathWork. *https://ww2.mathworks.cn/help/matlab/ref/fft.html*

[3] Jason J. Bramburger, Lecture Notes for AMATH482, Winter, 2021

[4] Katherine, Slack Q&A, https://app.slack.com/client/T01GMTJJGN5/C01KP1MD4LT/thread/C01KP1MD4LT-1612301094.018000>

Appendix A MATLAB Functions

- y = linspace(x1,x2,n) returns a row vector of n evenly spaced points between x1 and x2.
- fftn(x) returns multidimensional Fourier transform of the array using fast Fourier transform algorithm.
- pcolor(C) creates a pseudocolor plot using values in matrix C.
- Y = fftshift(x) rearranges the fourier transform by moving the zero frequency component to the center of the array x

Appendix B MATLAB Code

```
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);

% GNR Guitar
L = 14;  n = length(y);
t2 = linspace(0,L,n+1); t = t2(1:n);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);

% Spectrogram
a = 20;
tau = 0:0.1:14;

for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2); % Window function
   Yg = g.*transpose(y); %Signal filter
   Ygt = fft(Yg);
   Ygt_spec(:,j) = fftshift(abs(Ygt));
end
figure(2)
pcolor(tau,ks,log(abs(Ygt_spec)+ 1))
shading interp
set(gca,'ylim',[0 1000],'Fontsize',12)
colormap(hot)
colorbar
xlabel('Time (t)'), ylabel('Frequency (Hz)')
title('Spectrogram of GNR - Guitar')

% Floyd Bass

figure(1)
[y1, Fs] = audioread('Floyd.m4a');
% y1 = y1((1:2635920),1);
y1 = y1((1:878640),1);
% y1 = y1((1757281:2635920),1);
tr_gnr = length(y1)/Fs; % record time in seconds
plot((1:length(y1))/Fs,y1);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Comfortably Numb');
% p8 = audioplayer(y1,Fs); playblocking(p8);

% FT of Floyd (divided into 3 clips, 20s each)
L = 20;  n = length(y1);
```

```
t2 = linspace(0,L,n+1); t = t2(1:n);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);

% Spectrogram
a = 20;
tau = 0:0.1:20;

for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2); % Window function
   Yf = g.*transpose(y1); %Signal filter
   Yft = fft(Yf);
   Yft_spec(:,j) = fftshift(abs(Yft));
end

figure(2)
pcolor(tau,ks,log(abs(Yft_spec)+ 1))
shading interp
set(gca,'ylim',[150 1000],'Fontsize',12)
colormap(hot)
colorbar
xlabel('Time (t)'), ylabel('Frequency (Hz)')
title('Spectrogram of Floyd - Guitar(0-20s)')
% Floyd Guitar

figure(1)
[y1, Fs] = audioread('Floyd.m4a');
% y1 = y1((1:2635920),1);
y1 = y1((1:878640),:);
% y1 = y1((1757281:2635920),1);
tr_gnr = length(y1)/Fs; % record time in seconds
plot((1:length(y1))/Fs,y1);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Comfortably Numb');
% p8 = audioplayer(y1,Fs); playblocking(p8);

% FT of Floyd (divided into 3 clips, 20s each)
L = 20;  n = length(y1);
t2 = linspace(0,L,n+1); t = t2(1:n);
k = (2*pi/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);

% Spectrogram
a = 5;
tau = 0:0.1:20;
freq_max = [];
for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2); % Window function
   Yf = g.*transpose(y1); %Signal filter
   Yft = fft(Yf);
   [max, index] = max(Yft);
```

```
    freq_max = [freq_max;k(index)];
    Yft_spec(:,j) = abs(fftshift(Yft));
end
figure (2)
pcolor(tau, ks, Yft_spec)
shading interp
set(gca,'ylim',[150 1000],'Fontsize',12)
colormap(hot)
colorbar
xlabel('Time (t)'), ylabel('Frequency (Hz)')
title('Spectrogram of Floyd - Guitar(0-20s)')
```

Listing 1: Code from MATLAB