# AMATH 482 Homework 2

Jiatian Xie

Feb 11, 2020

**Abstract**

In this assignment, I analyzed a portion of two of the greatest rock and roll songs, Sweet Child O' Mine by Gun N' Roses and Comfortably Numb by Pink Floyd. I called the first song as GNR and the second as Floyd in this report. The main goal of this assignment was to reproduce the music score for the guitar in the GNR flip and filter the clean bass in the Floyd clip. I used Gabor filtering and Gaussian filter in order to get a good clean score.

## 1 Introduction and Overview

Since we have two famous RockRoll songs which contained several electronic instruments. The clips contained lots of overtones. In order to filter out overtones and isolate the sound of a typical sound, I will create spectrum and apply it on these clips and find out the music scores. This assignment includes three parts:

### 1.1 Create Spectrum

At first, I tried to create spectrum of each of the cilps and found that I was able to clean up the music score for guitar in the GNR clip and the bass in the Floyd clip. After searching, I found the frequency of guitar is above 300Hz and the frequency of bass is about 80Hz to 250Hz.

### 1.2 Filter

Then, I would like to isolate the bass in the Floyd clip by using a filter. In this assignment, I simply used a Gaussian filter.

### 1.3 Guitar solo in Comfortably Numb

At last, I wanted to mix the previous two tasks. I was going to isolate the sound of guitar solo in Floyd clip.

## 2 Theoretical Background

As I learned from the lectures and lecture notes, **Gabor Transform** is a kind of Fourier Transform and it contains two parameters:

1. $\tau$: where the filter is centered
2. $\alpha$: the width of the window(sub-domain)

The Gabor Transform is

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

The function g is real and symmetric. And the $L_2$-norm(energy) of g is set to unity. g(t) can be anything here, and I am gonna use a Gaussian here.

Consider the width of window:

- Big window: just using the FT over the entire signal, so we can get all frequency information.
- Small window: very little frequency info

Gabor transform with middle window size is Heisenberg Uncertainty Principle in action.

Since now, we have known how to visualize things in time-frequency domain, just multiplying by a window function or filter. We are able to see the visualization with different windows. However, in order to get a complete result, we want to stack all Fourier transform. This is called a **spectrum**. The main idea of spectrum is stacking all the information in frequency domain. Therefore, we can tell with the most precision where spots are bright.
In problem 2, we are supposed to use a filter. From Google search, I found the frequency of bass is about 80-250 Hz. Then, I can use the 1D Gaussian filter to isolate the bass.

$$F(k) = e^{-\tau*(k-k_0)^2}$$

and $\tau > 0$ is the width of the filter, $k_0$ is the center.

# 3 Algorithm Implementation and Development

Before solving the problem, firstly, I wanted to convert frequency to music notes. From slack, I saw we can multiplying the frequency with $2^{1/12}$ to get the note. See Algorithm 1

**Algorithm 1:** Music Notes Converter

Read audio from `GNR.mp4` and `Floyd.mp4`
Create the array of all note names
`freq = 440 * (2`$^($`1/12)).`$^($$-57:42$);
**for** $ii = 0:8$ **do**
`allNotes = [allNotes regexprep(noteNames, 'N', num2str(ii))];`
**end for**

Then, in order to reproduce the music score for each of clip, I made use of the Gabor filering and create a spectrum of the result. See algorithm 2, which illustrated my progress with GNR clip.

**Algorithm 2:** Gabor Filtering  Spectrum

Set reasonable value of a
`tau = 0:0.3:trgnr`
Rescale the Fs since this problem assumes 1/L1 periodic signals
Get the `fftshift` of rescaled Fs
**for** j = 1:length(tau) **do**
Create the Gabor filter g
mutiplying gyg `= g.*y`
`ygt = fft(yg);`
`spec(:,j) = fftshift(abs(ygt));`
**end for**

At last, to draw a spectrum of the slacking signals, we can use `pcolor` and `colormap(hot)`. Also, in order to include both frequency and music notes, I used `yyaxis`.

For the second problem, which asked me to isolate the bass, I was trying to create a Gaussian filter for each of the time realization. For example, we can start with $\tau = 0$ and find the max value and the corresponding index at that time. The max value we found was the center frequency. Therefore, I was able to set up a Gaussian filter for that specific time and filter the data. See algorithm 3.

**Algorithm 3:** isolate the bass in Comfortably Numb

**for** j = 1:length(tau) **do**
`[mxv,idx] = max(spec(:,j));`Get the fundamental frequency by indexingCreate a Gaussian filter in frequency domain based on the frequency we found previouslyMultiplying the filter with the spec matrix
**end for**

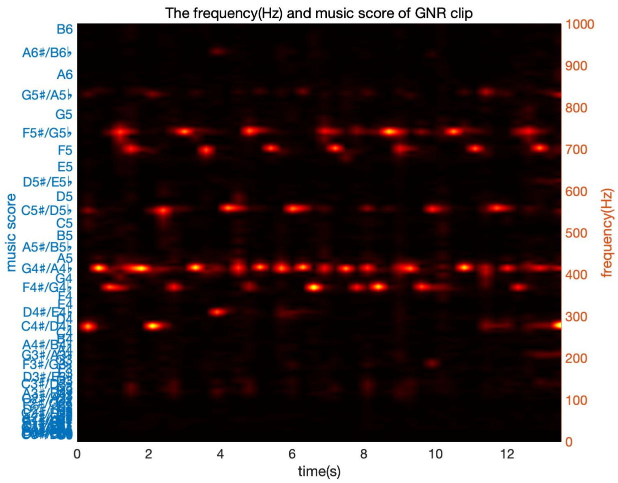


Figure 1: Here is a spectrum of the GNR clip through Gabor transform

The last question - isolate the guitar solo in the Floyd clip. My thought was combining previous two approaches.

1. According to the spectrum of problem 1, we can only consider the frequency higher than 400Hz.
2. Create Gaussian filter based on the center frequency like I did in problem 2.

There is some problem with Matlab - I could not use high pass filter and low pass filter. And therefore, I could not produce a spectrum of music notes for guitar solo. However, I tried to directly used the same Gaussian filter in problem 2 and limited y to [400 1000].

# 4 Computational Results

Based on the spectrum that gotten from my algorithm, I was able to determine the music notes. For the GNR clip, see figure 1, I can see the spots easily and I add notes in y axis as well, which is convenient for us to see the corresponding notes for each clear spot.

But for the Floyd clip, it was kind of difficult, see figure 2, it was hard to see the spots and determine the notes. However, I was able to use another Gaussian filter based on the fundamental frequency in each time realization. See figure 2, obviously, we can see a clearer spectrum with music notes in frequency domain.

Now, the third problem confused me a lot since firstly, I wanted to use `highpass()` to filter the frequency that was lower than 250Hz, but it appeared an error. So I just used the original data and only draw the frequency that was higher than 400Hz. Luckily, See figure 2,the spectrum looks not bad.

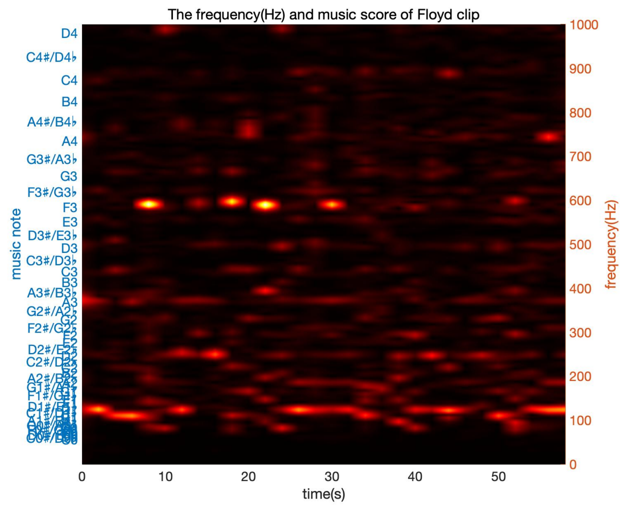


Figure 2: Here is a spectrum of the Floyd clip through Gabor transform

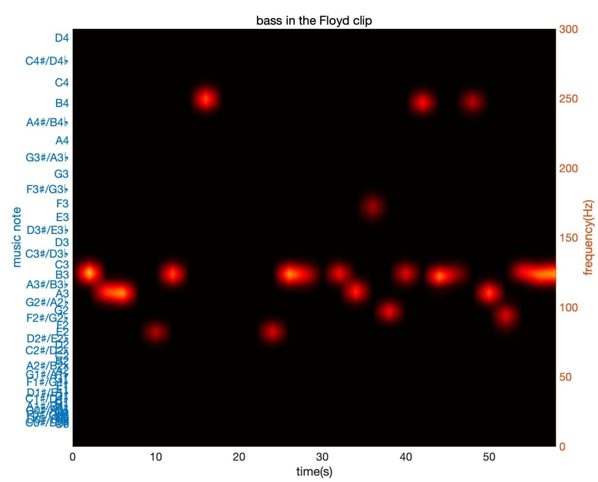


Figure 3: Here is a spectrum of the bass in the Floyd clip, the spot means a music note and the its duration.

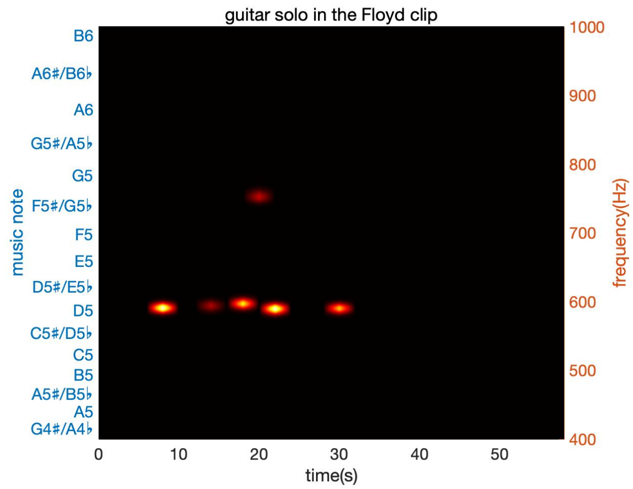


Figure 4: Here is a spectrum of the guitar solo in the Floyd clip, the spot means a music note and the its duration.

# 5 Summary and Conclusions

This assignment aims to reproduce the music score for a specific instrument. For the GNR clip, which guitar dominates the most part, it was easy for me to directly use Gabor transform and draw a spectrum since I can see the light spot that represented the music score. However, for the Floyd clip, it was complex, it was hard to determine the music score in the first step, so I just created a new Gaussian filter to filter the data in each time and the spectrum became better. Also, finding the guitar solo is super hard for me since I did not know how to filter the frequency below 300 Hz. But I still got a clear spectrum illustrating it.

I really enjoy doing this project although I am not familiar with both music score and rockroll. Isolating the sound of one instrument was really interesting and amazing.

# Appendix A MATLAB Functions

- `[y,Fs] = audioread(filename)` reads data from the file named filename, and returns sampled data, y, and a sample rate for that data, Fs.
- `newStr = regexprep(str,expression,replace)` replaces the text in str that matches expression with the text described by replace. The regexprep function returns the updated text in newStr.
- `y = linspace(x1,x2,n)` returns a row vector of `n` evenly spaced points between `x1` and `x2`.
- `Y = fft(X)` computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.
- `X = ifft(Y)` computes the inverse discrete Fourier transform of Y using a fast Fourier transform algorithm. X is the same size as Y.
- `pcolor(C)` creates a pseudocolor plot using the values in matrix C. A pseudocolor plot displays matrix data as an array of colored cells (known as faces).
- `set(H,Name,Value)` specifies a value for the property Name on the object identified by H.
- `yyaxis left` activates the side of the current axes associated with the left y-axis. Subsequent graphics commands target the left side.

## Appendix B MATLAB Code(1)

## Appendix C MATLAB Code(2)

## Appendix D MATLAB Code(3)

```
clear all; close all; clc
%%
[y1,Fs1] = audioread('GNR.m4a');
tr_gnr1 = length(y1)/Fs1;

[y2,Fs2] = audioread('Floyd.m4a');
tr_floyd = length(y2)/Fs2;

% create the list of notes
noteNames = {...
    'AN' 'AN/BN' 'BN' 'CN' 'CN/DN' 'DN' 'DN/EN' 'EN' 'FN' 'FN/GN' 'GN' 'GN/AN'};
note_freq = 440 * (2^(1/12)).^(-57:42);
allNotes = [];
for ii = 0:8
    allNotes = [allNotes regexprep(noteNames, 'N', num2str(ii))]; end %#ok<AGROW>
allNotes = allNotes(4:end-5);

% GNR
a = 800;
tau = 0:0.3:tr_gnr1;
y1 = y1.';
n1 = length(y1);
L1 = length(y1)/Fs1;
tp = linspace(0,L1,n1+1); t1 = tp(1:n1);
k = (1/L1)*[0:n1/2-1 -n1/2:-1]; % Notice the 1/L instead of 2*pi/L
ks = fftshift(k);
for j = 1:length(tau)
    g = exp(-a*(t1-tau(j)).^2);
    yg1 = g.*y1;
    ygt1 = fft(yg1);
    ygt1_spec(:,j) = fftshift(abs(ygt1));
end
figure(3)
yyaxis left
pcolor(tau,ks,ygt1_spec)
shading interp
set(gca,'ylim',[0,500], 'Fontsize', 16)
set(gca,'yTick',note_freq)
set(gca, 'YTicklabel',allNotes)
colormap(hot)
ylabel('music note')
yyaxis right
pcolor(tau,ks,ygt1_spec)
shading interp
set(gca,'ylim',[0,500], 'Fontsize', 16)
ylabel('frequency(Hz)'), xlabel('time(s)')
title('The frequency(Hz) and music score of GNR clip')
```

Listing 1: My code of solving this problem(1)

```
% Floyd
a = 1000;
y2 = y2.';
y2 = y2(1:end-1);
n2 = length(y2);
L2 = length(y2)/Fs2;
tau = 0:2:tr_floyd;
tp = linspace(0,L2,n2+1); t2 = tp(1:n2);
k = (1/L2)*[0:n2/2-1 -n2/2:-1]; % Notice the 1/L instead of 2*pi/L
ks2 = fftshift(k);
for j = 1:length(tau)
    g = exp(-a*(t2-tau(j)).^2);
    yg2 = g.*y2;
    ygt2 = fft(yg2);
    ygt2_spec(:,j) = fftshift(abs(ygt2));
end
figure(4)
yyaxis right
pcolor(tau, ks2,ygt2_spec)
shading interp
set(gca,'ylim',[0,300], 'Fontsize', 16)
colormap(hot)
xlabel('time(s)'), ylabel('frequency(Hz)')
yyaxis left
pcolor(tau, ks2,ygt2_spec)
shading interp
set(gca,'ylim',[0,300], 'Fontsize', 16)
set(gca,'yTick',note_freq)
set(gca, 'YTicklabel',allNotes)
colormap(hot)
ylabel('music note')
title('The frequency(Hz) and music score of Floyd clip')

bass = zeros(n2,length(tau));
for j = 1:length(tau)
    [mxv,idx] = max(ygt2_spec(:,j));
    k0 = abs(ks2(idx));
    g = exp(-0.01*(ks2-k0).^2);
    bass(:,j) = g.'.* ygt2_spec(:,j);
end
figure(5)
yyaxis right
pcolor(tau,ks2,bass)
shading interp
set(gca,'ylim',[0,300], 'Fontsize', 16)
colormap(hot)
xlabel('time(s)'), ylabel('frequency(Hz)')
yyaxis left
pcolor(tau, ks2,bass)
shading interp
set(gca,'ylim',[0,300], 'Fontsize', 16)
set(gca,'yTick',note_freq)
set(gca, 'YTicklabel',allNotes)
colormap(hot)
ylabel('music note')
title('bass in the Floyd clip')
```



Listing 2: My code of solving this problem(2)

```
guitar = zeros(n2,length(tau));
for j = 1:length(tau)
    [mxv,idx] = max(ygt2_spec(:,j));
    k0 = abs(ks2(idx));
    g = exp(-0.01*(ks2-k0).^2);
    guitar(:,j) = g.'.* ygt2_spec(:,j);
end
figure(6)
yyaxis right
pcolor(tau,ks2,guitar)
shading interp
set(gca,'ylim',[400,1000], 'Fontsize', 16)
colormap(hot)
xlabel('time(s)'), ylabel('frequency(Hz)')
yyaxis left
pcolor(tau, ks2,guitar)
shading interp
set(gca,'ylim',[400,1000], 'Fontsize', 16)
set(gca,'yTick',note_freq)
set(gca, 'YTicklabel',allNotes)
colormap(hot)
ylabel('music note')
title('guitar solo in the Floyd clip')
```

Listing 3: My code of solving this problem(3)