AMATH 482 Homework 2
Brad Hong
February 10, 2021

Abstract

The contents of this report will contain information on how to take a sound file and convert it into a readable musical score. Through MATLAB we will take clips of some selected songs and filter out specific instruments from them.

Section I. Introduction and Overview

The first main objective is to use the Gabor filtering method to reproduce the music score of the guitar in Guns N' Roses, "Sweet Child O' Mine" and the bass in Pink Floyd's, "Comfortably Numb". Then we will continue working with "Comfortably Numb" to isolate the bass as well as the guitar solo.

Section II. Theoretical Background

Time-frequency analysis is going to be a major part of gathering information about the musical scores. The main thorn of time-frequency analysis is mainly due to the Heisenberg Uncertainty Principle which states that to get information on time we must give up some information about frequency and vice versa. In short, we are not able to get absolute information on both. Thus, we must find a middle ground such that we get some information about both to deduce and analyze the whole picture. One such way to do this is through the Gabor Filtering method.

$$f_g(\tau, k) = \int_{-\infty}^{\infty} f(t)g(t-\tau)e^{-ikt}dt. \quad (1)$$

In this method, there are two variables tau and a. Tau is a variable that represents how far into the function we are going to explore, as in the focal point of time. The variable a is within the g (filtering) function which will specify how wide of an area we want to look at.

To further assist in our analysis, we will be creating spectrograms to better visualize the frequency space over time. Spectrograms are a useful tool that takes the frequency graphs of specific times, at a specified tau and a, and places them side by side to see how they change over time. In other words, we are watching how the frequencies change as tau changes over a specific a value.

Section III. Algorithm Implementation and Development

For our implementation of the Gabor Transform, we will be using the Gaussian as our filtering function, g:

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

Part 1: GNR guitar music score

The first clip to look into was the 15 second guitar riff of the Guns n' Roses song "Sweet Child O' Mine". Since it is an isolated guitar only clip we only had to find the proper window of the Gabor Transform and create a spectrogram out of the changing tau values. To observe the guitar values, the window was limited to -1200 Hz to 1200 Hz. However, the values were mirrored along 0 Hz so the window was shrunk to 200 Hz to 1200 Hz. Finally, I noticed that much of the frequencies are repeating over time except the bottom frequencies so I decided to look at each musical bar individually by shrinking the time window.

Part 2: Floyd Bass music score

This second clip had many more instrumental components compared to the GNR clip. The main challenge was to separate the bass from the rest of the instruments. To do this, I researched that medium bass is usually between 60-150 Hz so I limited the field of view to that. This gave a very clear photo of the bass when the window was set to 0.3 with a tau spacing of 2 seconds.

Part 3: Floyd Guitar music score

The final part of this research was to separate the guitar solo from the Pink Floyd clip. My first instinct was to utilize what I learned from Part 1 and 2 by using smaller segments of the clip as well as setting the frequency window to 250-800 Hz. However, this produced a spectrogram that was extremely hard to read. I presumed that this was due to the strength of the bass frequencies being much stronger relative to the guitar frequencies. To avoid this, I utilized a bandpass filter to only look at frequencies between 250-800. This let me see frequencies of the guitar much clearer but they seemed to be scattered.

Section IV. Computational Results

Part 1: GNR guitar music score

After following the procedure describe in Section IV and running code 1 from Appendix B, the visual seen below was produced:

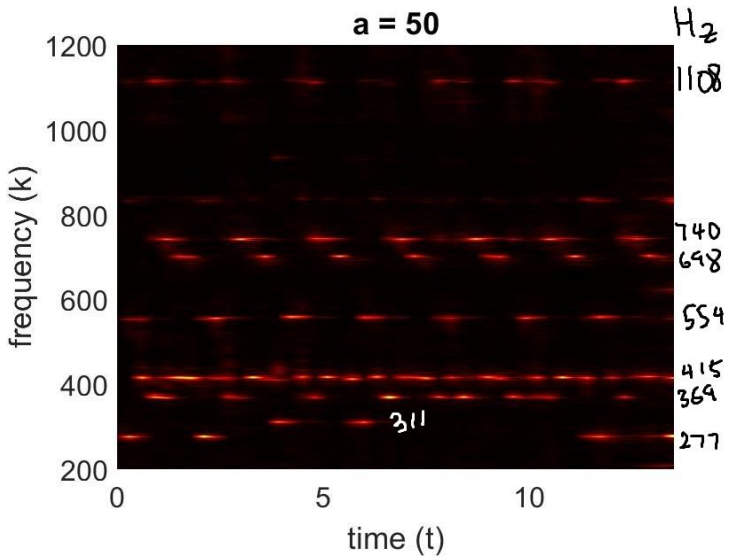

Image 1: The full spectrogram of the GNR.m4a with a width of a = 50

As seen above, a lot of the notes are repeating except the ones on the bottom, taking a closer look at the first 1.5 seconds of the clip, we get this spectrogram:

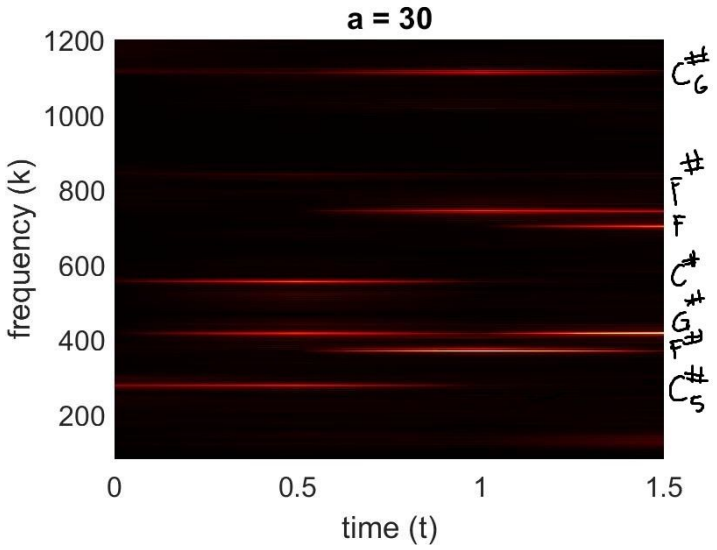

Image 2: A spectrogram over 1.5 seconds

After translating the frequencies on the spectrogram to the notes we got the musical score of C#4, C#5, G#4, F#4, C#6, F#5, F5, G#4.

Part 2: Floyd Bass score

After following the procedure described in Section IV and running code 2 in Appendix B, the spectrogram below was produced:

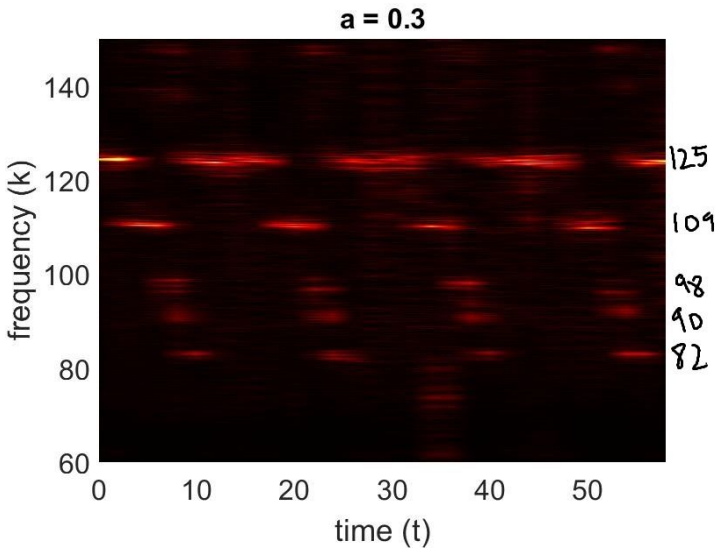

Image 3: Spectrogram of bass riff of Pink Floyd's "Comfortably Numb"

This spectrogram as seen above is a clear descending and repeating riff of 5 notes that goes: B, A, G, F, E all of which are in the 2nd octave.

Part 3: Floyd Guitar score:

Following the procedure from Section IV and running code 3 from Appendix B, we get the resulting spectrogram:

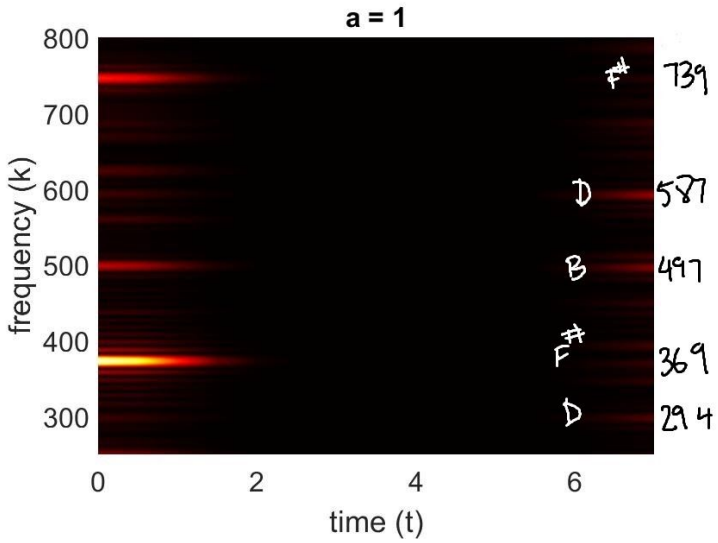

Image 4: Spectrogram of first 6 seconds of guitar solo of "Comfortably Numb"

This spectrogram doesn't really give much in terms of time however increasing a and extending the time to the whole clip would give much less information than the 5 notes listed here. The notes are as follows from top to bottom: F#5, D5, B4, F#4, D4.

Section V. Summary and Conclusions

Utilizing the Gabor Transform to filter specific parts of the music clips, we were able to create spectrograms to visualize the time and frequency on the same plane. I believe this is a powerful tool to find a middle ground for the Heisenberg Uncertainty Principle. Where the Fourier Transform failed in giving us information on time, the adjustability of the Gabor Transform allowed us to gather information on both frequency and time at the same time. Along with other methods of filtering such as the bandpass used in part 3, lots more data can be extracted and analyzed. Although I could not completely figure out the guitar solo, the fact that I could get specific notes out of the noisy clips I was given is amazing to me. If given more time to explore this assignment, I would look towards perfecting the bandpass filter to properly output the guitar solo.

Appendix A. MATLAB functions used and brief implementation explanation

audioread()
This function was used to read audio files into readable data. This allowed us to apply the Gabor transform on the resulting data.

pcolor()
This function is used to create the spectrogram along with other lines of code to visualize the spectrogram closer. I was unable to utilize the spectrogram function as I did not have the Signal Processing Toolbox

bandpass(y, fpass, fs)
This function was used to filter out data. The bandpass function focuses the data on the fpass range. Note: this function comes from the Signal Processing Toolbox but most of the spectrogram programming was already done with pcolor() by the time I decided to download this Toolbox.

Appendix B. MATLAB codes

```
% Code 1
[y, Fs] = audioread('GNR.m4a');
len = length(y);
y1 = y((len/8)*p+1:(len/8)*(p+1));
tr_gnr = length(y)/Fs
t2 = linspace(0,tr_gnr, (len+1));
```

```
t = t2(1:len);
k = (1/tr_gnr)*[0:len/2-1 -len/2:-1];
ks = fftshift(k);
figure(3)
a = 50;
tau = 0:.5:tr_gnr;
Sgt_spec = [];
for j = 1:length(tau)
    g = exp(-a*(t - tau(j)).^2);
    Sg = g.*y';
    Sgt = fft(Sg);
    Sgt_spec(:,j) = fftshift(abs(Sgt));
end
pcolor(tau,ks,Sgt_spec)
shading interp
set(gca,'ylim',[80 1200],'Fontsize',16)
colormap(hot)
%colorbar
xlabel('time (t)'), ylabel('frequency (k)')
title(['a = ',num2str(a)],'Fontsize',16)
```

```
% Code 2: Floyd Bass
[y, Fs] = audioread('Floyd.m4a');
Y = y(1:end-1,1);
len = length(y);
y1 = y((len/8)*p+1:(len/8)*(p+1));
tr_gnr = length(y)/Fs
t2 = linspace(0,tr_gnr, (len+1));
t = t2(1:len);
k = (1/tr_gnr)*[0:len/2-1 -len/2:-1];
ks = fftshift(k);
figure(4)
a = 0.3;
% tau = (tr_gnr1)*p:.5:(tr_gnr1)*(p+1);
tau = 0:2:tr_gnr;
Sgt_spec = [];
for j = 1:length(tau)
    g = exp(-a*(t - tau(j)).^2);
    Sg = g.*y';
    Sgt = fft(Sg);
    Sgt_spec(:,j) = fftshift(abs(Sgt));
end
pcolor(tau,ks,Sgt_spec)
```

```
shading interp
set(gca,'ylim',[60 150],'Fontsize',16)
colormap(hot)
%colorbar
xlabel('time (t)'), ylabel('frequency (k)')
title(['a = ',num2str(a)],'Fontsize',16)

% Code 3: Guitar solo
[y, Fs] = audioread('Floyd.m4a');
y = y(1:end-1,1);
p = 0;
len = length(y);
y1 = y((len/8)*p+1:(len/8)*(p+1));
tr_gnr = length(y)/Fs;
tr_gnr1 = tr_gnr/8;
t2 = linspace((tr_gnr1)*p,(tr_gnr1)*(p+1), (len+1)/8);
t = t2(1:(len+1)/8);
% t2 = linspace(0,tr_gnr, (len+1));
% t = t2(1:len);
k = (1/(tr_gnr1))*[0:(len/8)/2-1 -(len/8)/2:-1];
% k = (1/tr_gnr)*[0:len/2-1 -len/2:-1];
ks = fftshift(k);

figure(5)
a = 0.3;
tau = (tr_gnr1)*p:.5:(tr_gnr1)*(p+1);
% tau = 0:1:tr_gnr;
Sgt_spec = [];
for j = 1:length(tau)
    g = exp(-a*(t - tau(j)).^2);
    Sg = g.*y1';
    Sgt = fft(Sg);
    filtered = bandpass(Sgt, [250 800], Fs);
    Sgt_spec(:,j) = fftshift(abs(filtered));
end
pcolor(tau,ks,Sgt_spec)
shading interp
set(gca,'ylim',[250 800],'Fontsize',16)
colormap(hot)
%colorbar
xlabel('time (t)'), ylabel('frequency (k)')
title(['a = ',num2str(a)],'Fontsize',16)
```