AMATH 482 Homework 2

Erik Huang

February 10, 2021

Abstract

Experiment how to generate time frequency spectrogram from a music file using Gabor transform, and reproduce music scores using filter functions.

1 Introduction and Overview

Our goal is to determine the music score of specific instruments from a given music file in m4a format. The 2 music files we will be using are Sweet Child O' Mine by Guns N' Roses and Comfortably Numb by Pink Floyd. We will need to generate a frequency-time spectrogram using a technique called Gabor Transform. The resulting spectrogram can demonstrate notes playing at specific frequency at specific time, for the purpose of time-frequency-analysis. In order to obtain a good clean diagram, we might need to apply some filters.

The first part of this assignment is to analyze the music score for guitar in Sweet Child O' Mine and bass in Comfortably Numb, as these two instruments are dominating in the sound file.
The next part of the assignment will focus on isolating the bass from the Comfortably Numb and obtain a spectrogram that only consists of guitar notes. To achieve this, we will tune our filter function until we are able to distinguish the frequency for guitar and bass.

2 Theoretical Background

2.1 Gabor Transform (Short-Time Fourier Transform)

Fourier transform is a common tool used to analyze the frequency of a signal. However, it has a drawback that it doesn't obtain the information of when certain frequencies occur or how frequencies change over time. In other words, Fourier transform can determine whether the frequency is present, but it cannot tell when the frequency is present. Since we are analyzing a music file, obtaining a time-frequency relationship is necessary. For this purpose, we will use a variation of Fourier transform: The Gabor Transform, or the Short-Time Fourier Transform.

The input to be transformed is first multiplied by a Gaussian function, and the resulting function is then transformed with a Fourier transform to derive the time-frequency analysis. For an input function $f(t)$, the transform can be given as:

$$\tilde{f}_g(\tau, k) = \int_{-\infty}^{\infty} f(t)g(t-\tau)\mathrm{e}^{-\mathrm{i}kt}\ \mathrm{d}t$$

In this function, $g(t)$ is the filter function, shifting by τ and multiplying with the input function $f(t)$. The value τ describes where the filter is centred. So that by shifting τ, we can apply Fourier Transform over a time interval. Thus, for a fixed τ, the function $\tilde{f}_g(\tau, k)$ gives the information about frequency components near time τ.

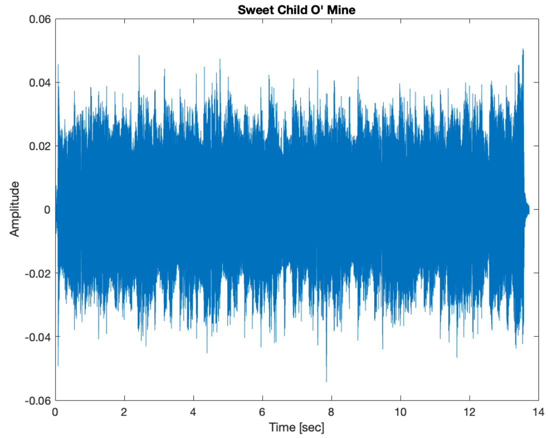

Figure 1: Here is the sample graph for GNR.m4a

2.2 Filter function in Gabor Transform

There are many choices for the filter function, and the result would be dependent on them. In this assignment, we will use the Gaussian function as our filter function. This is the default approach since this is also the choice for Gabor when he designs the Gabor Transform. The function is defined as:

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

In this function, value a describes the width of the window that we are considering. The width of the window has a trade off effect: If the window is huge(a is too large), then we are just doing Fourier Transform over the entire signal, which has all the frequency information but no time information; If the window is extremely small(a is small), then we will lose track of frequency and obtain more information in individual times. We will need to tune the parameter for value a and strike a balance in order to obtain the best result for time-frequency-analysis.

3 Algorithm Implementation and Development

3.1 Applying Gabor Transform to both music files

The provided music files, "GNR.m4a" and "Floyd.m4a", are in m4a form. These files can be imported and converted into vectors by the `audioread` command. Next, we need to further convert this vector in order to apply Gabor Transform on it. The sample rate is at 48 kHz, so that to obtain the time interval, we can simply divide the total number of sample points by the sample rate. If we plot the data now, the result would be figure 1. To apply Gabor transform on the signal, we can do the multiplication of the signal function $f(t)$ and the filter function $g(t-\tau)$ within an iteration. In this algorithm, we need to manually tune the parameter τ and a, which corresponds to the time step and window width.

Algorithm 1: Gabor Transform on music files

for $i = 1 : length(\tau)$ **do**
  Define the shifted filter function $g(t-\tau)$
  Run FFT on the product of $f(t)$ and $g(t-\tau)$
  Extract the maximum frequency
  Save the result in a matrix
end for

For Sweet Child O' Mine, I set $\tau = 0.1$ to get enough time steps and $a = 5000$ to highlight more on frequency. This iteration steps through each time step and performs the Gabor Transform. During this process, The Fast Fourier Transforms is used to transform the result from spatial domain into frequency domain. In the case of Sweet Child O' Mine, we assume that the extraction of the maximum frequency is the frequency of guitar because the music file only includes the guitar sound.

In the case of Floyd.m4a, which is Comfortably Numb by Pink Floyd, we can use the same algorithm described above. However, we need to tune the parameters differently. Since the length of the audio is 60 seconds, I set $\tau = 1$ to avoid the inflation of time steps. I keep $a = 5000$ to highlight more on frequency.

The time-frequency-analysis for Comfortably Numb gets a bit more complicated. The music file consists of two musical instruments, guitar and bass. Thus we must extract frequencies for specific instruments. In this part, we will consider the regular frequency range of bass, which is between 20 and 300 Hz, and trim our result.

3.2 Subtracting the frequency information of bass

By completely isolating the frequency information of bass, we will be able to obtain some information of the guitar solo. This process can be done with some modification to our algorithm during the iteration:

Algorithm 2: Gabor Transform with Bass sound eliminated

for $i = 1 : length(\tau)$ **do**
  Define the shifted filter function $g(t - \tau)$
  Run FFT on the product of $f(t)$ and $g(t - \tau)$
  Extract the maximum frequency
  Set any frequency above 300 to be 0
  Set any frequency below 20 to be 0
  Save the result in a matrix
end for

By simply deleting all the frequencies in the frequency range of bass, we will get information that purely consist of guitar sounds. However, guitar's frequency range overlaps with bass's range. Hence, it's very likely that we also deleted some information of guitar notes during this process.

4 Computational Results

Figure 2,3,4 are spectrograms demonstrating the note played by instruments.

5 Summary and Conclusions

From the result, we can see the effect of Gabor Transform. Using the transform, we can generate spectrograms that clearly display the notes being played during the time interval. It's very possible for us to determine the music scores from simple audio files like these. To some extent, we can also distinguish different instruments based on their frequency ranges.

Appendix A MATLAB Functions

Major Functions list:

- `audioread` Reads audio data from the local directory.
- `fftn` N-D Fast Fourier Transform
- `fftshift` Shift zero-frequency component to center of spectrum.

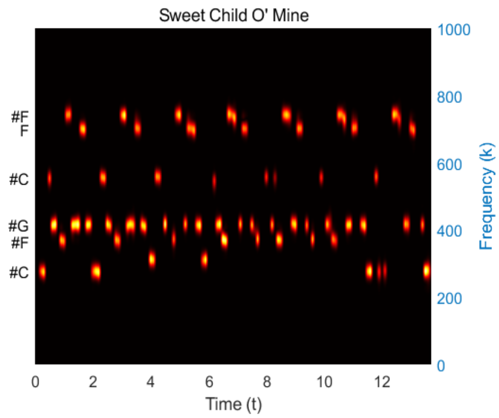

Figure 2: Spectrogram of Sweet Child O'Mine

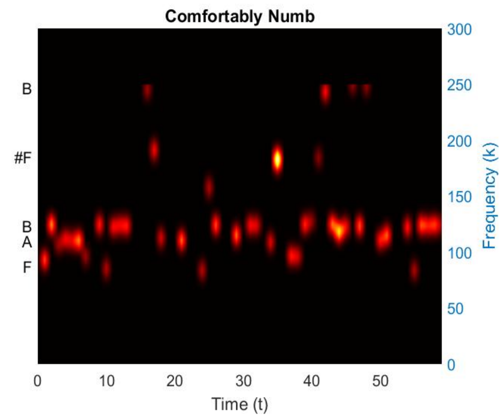

Figure 3: Spectrogram of Comfortably Numb(Bass only)

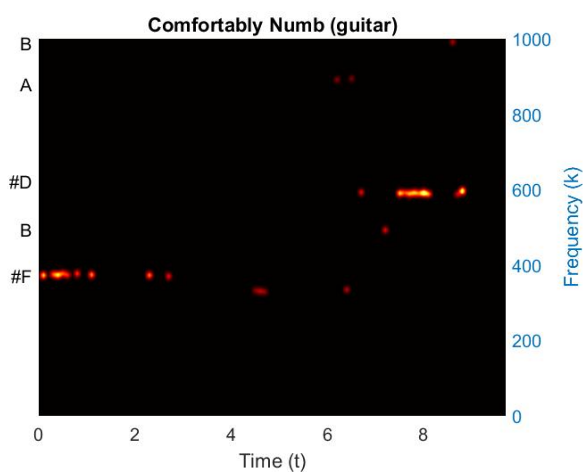

Figure 4: Spectrogram of Comfortably Numb(Guitar only)

- `colormap` Sets the color configuration for the current graph.
- `yticks` Sets the y-axis tick values
- `yticksslabel` Sets the y-axis tick labels

Appendix B MATLAB Code

```
% Clean workspace
clear all; close all; clc
%% Figure of sampling versus time for GNR
figure(1)
[y, Fs] = audioread('GNR.m4a');
tr_gnr = length(y)/Fs; % record time in seconds

t = (1:length(y))/Fs;
plot(t,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Sweet Child O'' Mine');
% p8 = audioplayer(y,Fs); playblocking(p8); % Sound player

%% Gabor Transform for GNR
n = length(y);
L = tr_gnr;
k = (1/L)*[0:(n/2 - 1) -n/2:-1];
ks = fftshift(k);
% create signal
S = y';
tau = 0:0.1:L;
a = 5000;
Sgt = zeros(length(y), length(tau));

for i = 1:length(tau)
    g = exp(-a*(t-tau(i)).^2); % Gaussian
    Sft = fft(g.*S);
    [M,I] = max(abs(Sft(:)));
    Sgt(:,i) = fftshift(abs(Sft));
end


figure(2)
pcolor(tau,ks,Sgt)
shading interp
set(gca,'ylim',[0 1000],'Fontsize',12)
colormap(hot)
yticks([277, 367, 415, 554, 698, 740])
yticklabels({'#C', '#F', '#G', '#C', 'F', '#F'})
yyaxis right
set(gca,'ylim',[0 1000],'Fontsize',12)
xlabel('Time (t)'), ylabel('Frequency (k)');
title('Sweet Child O'' Mine');

%% Clear space
clear all; close all; clc
```

```
%% Gabor Transform for Floyd
[y, Fs] = audioread('Floyd.m4a');
n = length(y);
trgnr = n / Fs;

k = (1 / trgnr) * [0:(n/2 - 1) (-n/2):-1];

tau = 0:1:trgnr;
a = 6000;
S = y';
Sgt_spec = zeros(n - 1, length(tau));

for j = 1:length(tau)
    g = exp(-a * (t - tau(j)).^2);
    Sgt = fft(g.*S);

    Sgt = Sgt(1:n-1);
    [maximum, index] = max(abs(Sgt));
    filter = exp(-0.01 * (k - k(index)).^2);
    Sgtf = Sgt .* filter;
    Sgtf(k > 250) = 0;
    Sgtf(k < 60) = 0;
    Sgt_spec(:, j) = fftshift(abs(Sgtf));
end

%%
figure(2)
pcolor(tau,ks,Sgt_spec)
shading interp
set(gca,'ylim',[0 300],'Fontsize',12)
colormap(hot)
yticks([87.307, 110.00, 123.47, 185.00, 246.94])
yticklabels({'F','A', 'B', '#F', 'B'})
yyaxis right
set(gca,'ylim',[0 300],'Fontsize',12)
xlabel('Time (t)'), ylabel('Frequency (k)');
title('Comfortably Numb');

%% Gabor Transform for Floyd
[y, Fs] = audioread('Floyd.m4a');
n = length(y);
trgnr = n / Fs;

k = (1 / trgnr) * [0:(n/2 - 1) (-n/2):-1];

tau = 0:1:trgnr;
a = 6000;
S = y';
Sgt_spec = zeros(n - 1, length(tau));

for j = 1:length(tau)
    g = exp(-a * (t - tau(j)).^2);
    Sgt = fft(g.*S);
```

```
    Sgt = Sgt(1:n-1);
    [maximum, index] = max(abs(Sgt));
    filter = exp(-0.01 * (k - k(index)).^2);
    Sgtf = Sgt .* filter;
    bass = Sgtf;
    bass(k > 250) = 0;
    bass(k < 60) = 0;
    % subtract bass freq
    guitar = Sgtf - bass;
    Sgt_spec(:, j) = fftshift(abs(guitar));
end

%%
figure(2)
pcolor(tau,ks,Sgt_spec)
shading interp
set(gca,'ylim',[0 1000],'Fontsize',12)
colormap(hot)
yticks([369.99, 493.88, 622.25, 880.00, 987.77])
yticklabels({'#F','B', '#D', 'A', 'B'})
yyaxis right
set(gca,'ylim',[0 1000],'Fontsize',12)
xlabel('Time (t)'), ylabel('Frequency (k)');
title('Comfortably Numb (guitar)');
```