AMATH 482 Homework 2

Alex Omusoru

February 10, 2021

Abstract

Portions of two rock and roll songs were extracted and given these, this paper aims to show how various instrumentals can be extracted from the sound. The next step was then to extract musical scores for each instrumental. This goal was accomplished through the use of a Gabor filter and then filtered additionally with a simple gaussian filter as needed. As a result, the music scores of several instrumental portions were able to be extracted from the audios.

1 Introduction and Overview

The goal of this paper has multiple parts. Given audio chunks from "Sweet Child O' Mine by Guns N' Roses and from "Comfortably Numb" by Pink Floyd, this paper intends to extract the guitar line from the Guns N' Roses song (GNR), as well as the bass and guitar lines from the Pink Floyd song (Floyd). To do so, different approaches have to be taken. For the guitar line of GNR, a Gabor Transform and then a spectograph visualization is sufficient. For the bass line of Floyd, a lowpass filter was applied and then followed by a Gabor Transoform and a spectograph visualization. For the guitar line the approach taken is to first filter the audio with the Gabor Transform and then apply a gaussian filter around the secondary central frequency. This ensures that the filtering is done about the higher frequencies the guitar occupies.

2 Theoretical Background

The Gabor Transform is based around the idea that to better filter a time-sensitive signal (such as music) we must use a time-frequency analysis method so as to ensure that time information is not lost as it is with the Fast Fourier Transform. Thus the Gabor Transform is essential in the music decomposition. Generally the Gabor Transform is defined as:

$$\bar{f}_g(\tau, k) = \int_{-\infty}^{\infty} f(t)g(t-\tau)e^{-ikt}dt \tag{1}$$

Where $g(t-\tau)$ represents some filter function centered at τ. In this paper, the gaussian function was used, thus

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

Which introduces a, that represents the width of the filter.

In addition, the implementation of the lowpass filter (See Appendix A:MATLAB Functions for detail on MATLAB implementation) built into MATLAB which was used for the filtration of a music signal composed of several instruments.

Another important model used was the spectrograph. This is a visualization for the aggregation of all the Gabor Transforms over the period of the song. The spectrograph allowed the results of each Gabor transform to be represented in the frequency domain, and as a result allowed for the identification of the central frequencies of each time slot, which when combined gave a visual representation of what notes were played, when they were played, and for how long.

3 Algorithm Implementation and Development

The following algorithm was used for the development of the spectrograph and the application of the Gabor Transform for both GNR and Floyd.

Algorithm 1 Filtration and Coordinate Location

`tau` = 0 to T with step s (where T is the length of the song)
`a` = width parameter
initialize `ygtspec` for storing spectograph data
for $j = length(tau)$ **do**
  Build Gabor Filter using tau(j) and a
  Multiply signal by filter function
  Apply FFT
  Add it to `ygtspec`
end for
Form spectograph using `ygtspec` as the data.

4 Computational Results

Guitar line from GNR The following spectograph results were found:

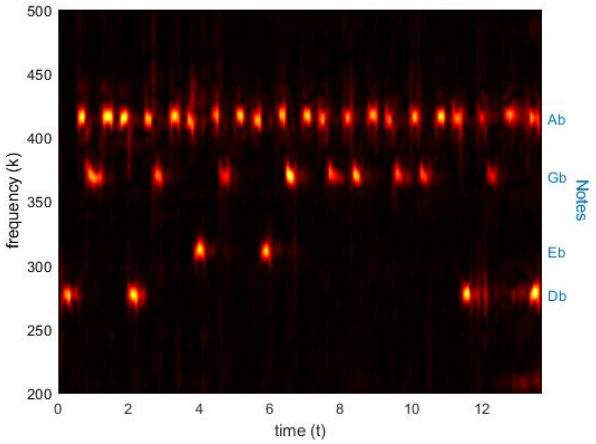

Figure 1: Spectograph of Gabor Transformation for GNR

From this image, the following notes were extracted:

Db Ab Gb Ab Ab
Db Ab Gb Ab Ab
Eb Ab Gb Ab Ab
Eb Ab Gb Ab Ab
Gb Ab Gb Ab Ab
Gb Ab Gb Ab Ab
Db Ab Gb Ab Ab

Bass line from GNR The following spectograph results were found:

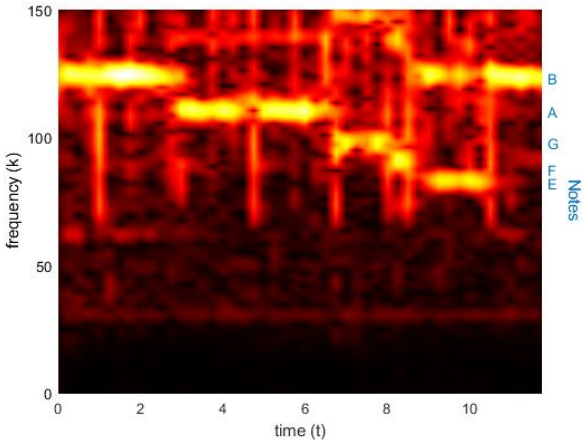

Figure 2: Spectograph of Gabor Transformation for Floyd (Segment 1)

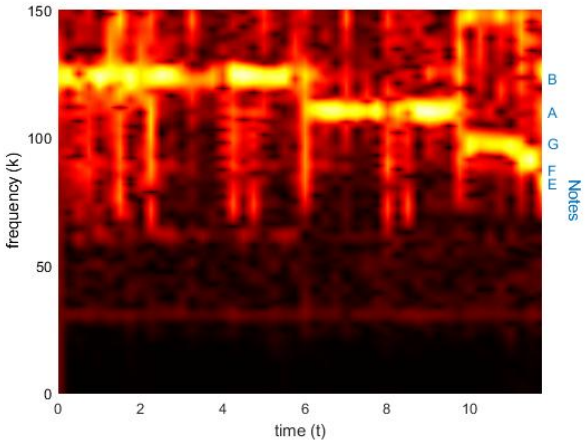

Figure 3: Spectograph of Gabor Transformation for Floyd (Segment 2)

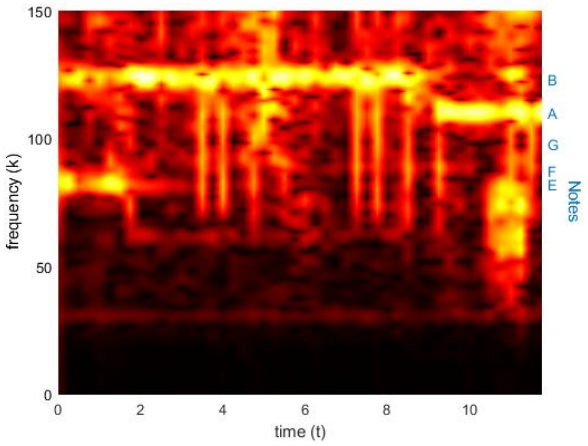

Figure 4: Spectograph of Gabor Transformation for Floyd (Segment 3)

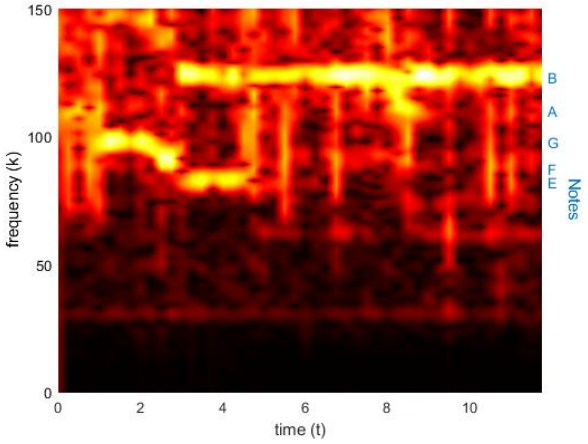

Figure 5: Spectograph of Gabor Transformation for Floyd (Segment 4)

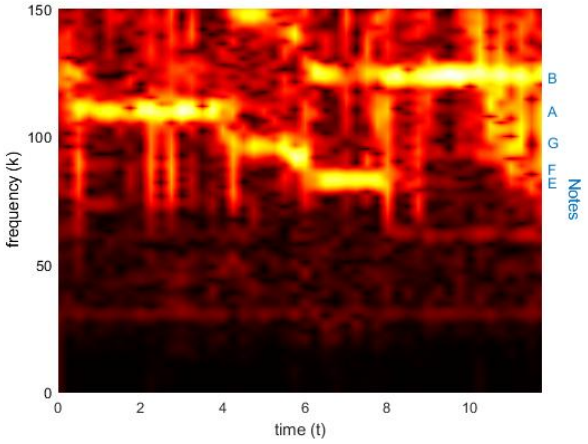

Figure 6: Spectograph of Gabor Transformation for Floyd (Segment 5)

From these images, the following notes were extracted:

B A A G F B E B B B A A G F E B A E A G F B E B A B A G B E B

Guitar line from GNR Despite much effort, the guitar line from Floyd remained too difficult to identify and therefore no data was able to be drawn from the music provided.

5 Summary and Conclusions

Using the Gabor Transform and lowpass filtering, music scores were extracted for the bass line of Floyd and the guitar line of GNR. Unfortunately guitar line of Floyd proved to be too difficult to identify.

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.
- `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). MATLAB® creates this plot as a flat surface in the x-y plane. The surface is defined by a grid of x- and y-coordinates that correspond to the corners (or vertices) of the faces. The grid covers the region X=1:n and Y=1:m, where [m,n] = size(C). Matrix C specifies the colors at the vertices. The color of each face depends on the color at one of its four surrounding vertices. Of the four vertices, the one that comes first in the x-y grid determines the color of the face.
- `pcolor(X,Y,C)` specifies the x- and y-coordinates for the vertices. The size of C must match the size of the x-y coordinate grid. For example, if X and Y define an m-by-n grid, then C must be an m-by-n matrix.
- `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.
- `colormap(map)` sets the colormap for the current figure to the colormap specified by map.

Appendix B MATLAB Code

Listing 1 Code used for GNR

```
%% setup
clear; close all; clc;


% figure(1)
[y, Fs] = audioread('GNR.m4a');
tr_gnr = length(y)/Fs; %record time in seconds
y = y(1:length(y))';

%% fourier transform
L = tr_gnr; 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);

yt = fft(y);
yt_s = fftshift(yt);

[max_num, max_idx] = max(abs(yt_s));
[freq] = abs(ks(max_idx));

%% gabor filtering & spectrogram

tau = 0:0.1:tr_gnr;
a = 400;

ygt_spec = zeros(length(t), length(tau));
for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2);
   yg = g.*y;
   ygt = fft(yg);

   ygt_spec(:,j) = fftshift(abs(ygt));
end

figure(6)
pcolor(tau,ks,abs(ygt_spec))
shading interp
colormap(hot)

xlabel('time (t)')

ylim([200 500])
ylabel('frequency (k)')

yyaxis right
yticks([277.18 311.13 369.99 415.30])
yticklabels({'Db','Eb','Gb','Ab'})
ylim([200 500])
set(get(gca,'YLabel'),'rotation',-90,'VerticalAlignment','bottom')
ylabel('Notes')
```

Listing 2 Code used for Floyd Bass

```
%% setup
clear; close all; clc;

[og_y, Fs] = audioread('Floyd.m4a');
og_y = og_y(1:length(og_y)-1);

n = length(og_y)/5;
splitCount = 1; % break it into 5 pieces, and select which piece to analyze
b = (splitCount-1)*n + 1;
e = splitCount*n;
y = og_y(b:e, 1);

tr_floyd = length(y)/Fs; %record time in seconds

%% fourier transform
L = tr_floyd; 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);

yt = fft(y);
yt_s = fftshift(yt);

plot(ks,abs(yt_s),'r','Linewidth',2);
set(gca,'Fontsize',16)
xlabel('frequency (k)'), ylabel('fft(yt)')

%% gabor filtering & spectrogram
tau = 0:0.25:tr_floyd;
a = 100;
ygt_spec = zeros(length(t), length(tau));
for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2);
   yg = g.*(y');
   ygt = fft(yg);

   ygt_spec(:,j) = fftshift(abs(ygt));
end
pcolor(tau,ks,log(abs(ygt_spec) + 1))
shading interp
colormap(hot)
xlabel('time (t)')
ylabel('frequency (k)')
ylim([0 150])
yyaxis right
yticks([82.41 87.31 98.00 110.00 123.47])
yticklabels({'E','F','G','A','B'})
ylim([0 150])
set(get(gca,'YLabel'),'rotation',-90,'VerticalAlignment','bottom')
ylabel('Notes')
```