AMATH 482 Homework 2

Clayton Heath

February 10, 2021

Abstract

We use Gabor filtering in order to reproduce the music score for the guitar in a clip from the song *Sweet Child O' Mine* and the bass in *Comfortably Numb*. We will then explore how to use these methods in order to piece together in the music score of the guitar and bass from *Comfortably Numb* as well as the guitar from *Sweet Child O' Mine*.

1 Introduction and Overview

We will first convert our songs into vectors. Then, for *Sweet Child O' Mine*, we will apply the *Gabor transform*, and plot its spectrogram, and use that to find what notes are being played by the Guitar. However, *Comfortably Numb* has a bass, guitar, piano, and drums, so it takes a little more work to get the music score. First by applying the *Gabor transform* and plotting its spectrogram, we will be able to clearly see the bass line, and can filter around it to get the notes that are being played. We then can completely cut out the low frequencies, removing the bass, and filter again to get notes of the guitar.

2 Theoretical Background

We use the *Gabor Transform* to be able to information on the frequency over time of a signal:

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

This gives us information about the frequency components near some fixed τ. Also, the function $g(t-\tau)$ is some filter that we use to ensure our transform is only looking at the frequency information near τ. g must be real and symmetric and $||g||_2 = 1$. We will be using the Gaussian as g, but really any filter can be used as long as it meets those requirements.

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

In our Gaussian, the constant a determines the window size of the filter. This is important because if we make the window too big, we will just recover the *Fourier Transform*, where we have all of the frequency information but no information about time. If the the window is too small, then we would be losing all of the frequency information.

Since we are wanting to transform data, we are going to use the *discrete Gabor transform* where $k = m\omega_0$ and $\tau = nt_0$ for m, n are integers and ω_0, t_0 are positive constants representing the frequency resolutions:

$$\tilde{f}_g(m,n) = \int_{-\infty}^{\infty} f(t)g(t-nt_0)e^{2\pi i m\omega_0 t} \tag{3}$$

3 Algorithm Implementation and Development

First we will look at *Sweet Child O' Mine*. Under the `Loading Songs` section of the MATLAB code we import and convert the clip of the song into a vector and calculate the length of the song in seconds.

Next, under the `Guns and Roses` section, we prepare for plotting the *discrete Gabor transformed* to the song. We see a lot of similar steps to that of preparing to plot the `Fourier transform` of data. We don't scale our frequency space `k` by 2π since we are working with Hertz. We make the `tslide` vector, which starts at 0 and sums up the length of our song with step-size of 0.1. The smaller the step size, the clear of an image we will get when we plot our spectrogram, however, it also increases the computational cost. 0.1 was chosen as it produces a pretty clear image while not freezing my computer. We also make a 2-dimensional matrix `spec` which will contain our *Gabor transformed* song. It is 2-dimensional since the *Gabor transform* retains both time and frequency, thus turning our 1-dimensional array into 2-dimensions. We also make an empty `maxFreq` vector to hold the frequency that highest value at each time step. We also have `a=100` for our Gaussian used in the *Gabor transform* as seen in Equation 2. We then apply the *descrete Gobor transform* and plot our `spec` matrix using `pcolor()`. We can see this spectrogram in Figure 1 and the analysis in the Computational Results section.

Next, we will start working on *Comfortably Numb*. Under the `Loading Songs` section, we follow the same steps as before to import and convert the clip, however, this clip is almost a minute long, and working with the entire clip causes my computer to freeze, so we take a trimmed vector of the clip containing only the first quarter of it, or about 15 seconds.

Then, under the `Guns and Roses` section, we look to apply the *Pink Floyd* and plot the sectrogram. We follow the same setup as we did for *Sweet Child O' Mine* and plot the spectrogram as seen in Figure 2. There is a lot more going on in this image, but we can see a bright bass line between 50-150 Hertz. So, now we want to filter around those bass frequencies in order to get a clearer picture of what notes are being played. We do this under the `PF Filtered` section. We follow the same steps as before fora *Gabor transform*, however, for each time step we apply a Gaussian filter to the transformed data. We have $\tau = 1$ and the filter is centered at the `maxFreq()` at the time step as collected from the the first *Gabor transform* we did. We then plot the spectrogram as seen in Figure 3.

Now, we want to find the music score the guitar in *Comfortably Numb*. But, as seen, the bass line contains most of the max values, so we don't have anything to filter around. So what we are going to do is remove all of the frequencies less than 150 since that is where the bass lies. So under the `PF Guitar` section, we *Fourier transform* our clip and build our `ks` vector as before containing all of the frequencies. Then, we run a for loop over all of the values of frequencies, and if a frequency is less than 150, we set the value of our *Fourier transformed* clip to 0 and that same index. We then use `ifft()` to bring our song back to time and out of frequency space. Then, under the `PF Bass Removed` section, we trim our new clip to the first 15 seconds, and then apply the *Gabor transform* and track the maximum values and plot the spectrogram as before (Figure 4). We can see now that our bass line down at the bottom is completely gone, so we use these new maximum values under section `PF Guitar Filtered` to filter around as done before using a Gaussian, then plotting the spectrogram as seen in Figure 5.

4 Computational Results

When figuring out the notes that are played from the guitar from the *Sweet Child O' Mine* clip, we look at the spectrogram from Figure 1. Since the guitar is the only instrument in the clip, we can assume that the bright spots between 200 and 800 Hertz to be the notes that the Guitar played since the bright spots above 800 are overtones from lower frequencies. Thus by looking at the spectrogram as well as the maximum values from the MATLAB code, we see that the frequencies of these notes are about 276, 310, 369, 415, 553, 702, and 742. Thus translating these frequencies to notes, we get $C^{\#}$, $D^{\#}$, $F^{\#}$, $G^{\#}$, $C^{\#}$, F, and $F^{\#}$ where the first $C^{\#}$ is just above middle C, and the second occurrence of a note is an octave above the first. One might think the the notes an octave above are overtones, but we know they aren't since they are not occurring at the same times as seen in the spectrogram.

We can see the spectrogram of the *Gabor transformed* first 15 seconds of *Comfortably Numb* clip in Figure 2. Here we can see the obvious hot spots of the bass between 50 and 150 Hertz. So, when filtering around those hot spots, we get Figure 3, which has a much clearer picture of the notes being played by the bass. But we notice some holes, and spots that seem wrong, these holes and spots show where the overtones of the base had a higher value than the notes the bass was playing. But by using both of the Figures as well as the vector of the maximum values, we see that the notes being played have frequencies 83, 90, 98, 110,

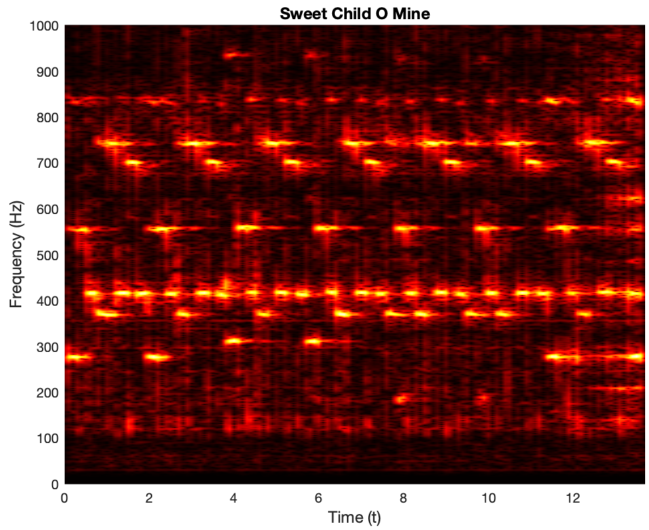

Figure 1: Here is the spectrogram of the Gabor transformed guitar solo from *Sweet Child O' Mine*.

124. Which translates to E, F$^{\#}$, G, A, and B.

Cutting out all frequencies below 150 removes the bass and gives the spectrogram in Figure 4. Then, by filtering around the maximum values, we get the spectrogram in Figure 5. These frequencies we see are going to be the guitar from the song mixed in with some of the overtones from the bass. We did not remove the overtones because if we did, we would lose some of the guitar as well since the guitar might play notes that match the overtone.

5 Summary and Conclusions

We were able to use the *Gabor transform* to find the notes played by the guitar in the *Sweet Child O' Mine* clip and the bass in the *Comfortably Numb* clip. We were also able to get a less accurate, but still reasonable spectorgram filtered to show the guitar notes from *Comfortably Numb*. It is not completely accurate since it still contains overtones from the bass line and possibly some notes from the piano and drums that are in the background. If we had some sort of way of figuring out when the guitar was playing the same note as an overtone, then we could get rid of all the overtones except the ones that match the guitar, and get a really accurate breakdown of the notes being played. However, the spectrogram produced for the guitar is pretty decent for what we are given.

Appendix A MATLAB Functions

- `fftn(A)` returns the fast Fourier transform of `A` for 1 dimension.
- `fftshift(X)` returns the values in the order: $\left\{\hat{x}_{-\frac{N}{2}}, \hat{x}_{-\frac{N}{2}+1}, ..., \hat{x}_{-1}, \hat{x}_0, \hat{x}_1, ..., \hat{x}_{\frac{N}{2}-1}\right\}$ given that $X = \left\{\hat{x}_0, \hat{x}_1, ..., \hat{x}_{\frac{N}{2}-1}, \hat{x}_{\frac{N}{2}}, \hat{x}_{-\frac{N}{2}+1}, ..., \hat{x}_{-1}\right\}$. X is a matrix where each row is a copy of `x`, and `Y` is a matrix where each column is a copy of `y`. The grid represented by the coordinates `X` and `Y` has `length(y)` rows and `length(x)` columns.
- `pcolor(X,Y,C)` creates a pseueocolor plot where the x and y coordinates are specified by `X` and `Y` and the color is determined by `C`.

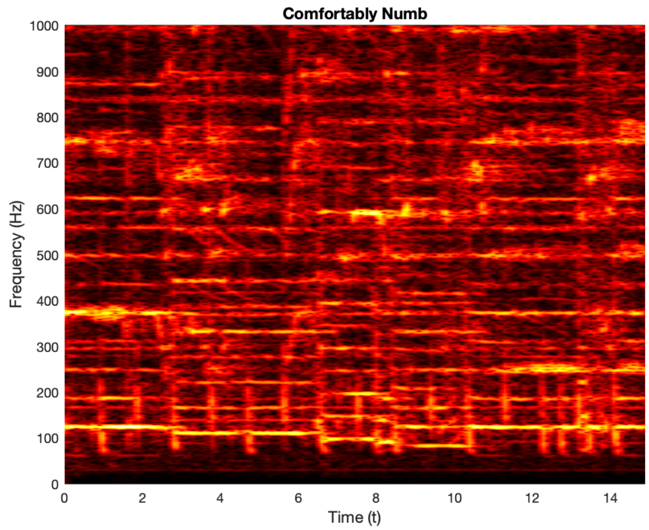

Figure 2: Here is the spectrogram of the Gabor transformed first 15 seconds of *Comfortably Numb*.

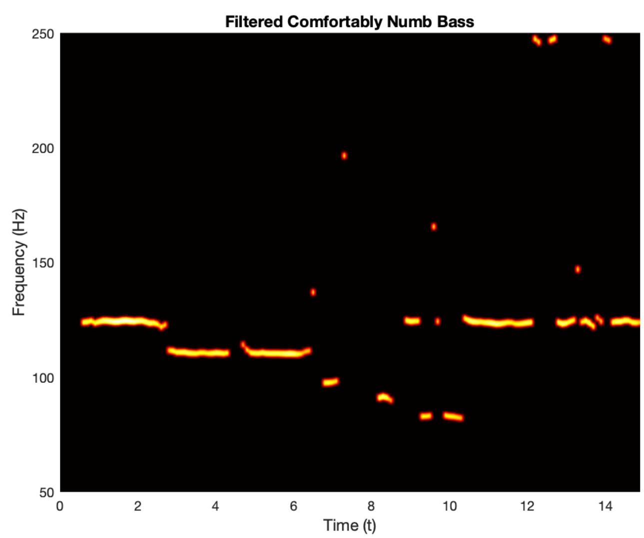

Figure 3: Here is the spectrogram of the Gabor transformed first 15 seconds of *Comfortably Numb* filtered around the maximum values.

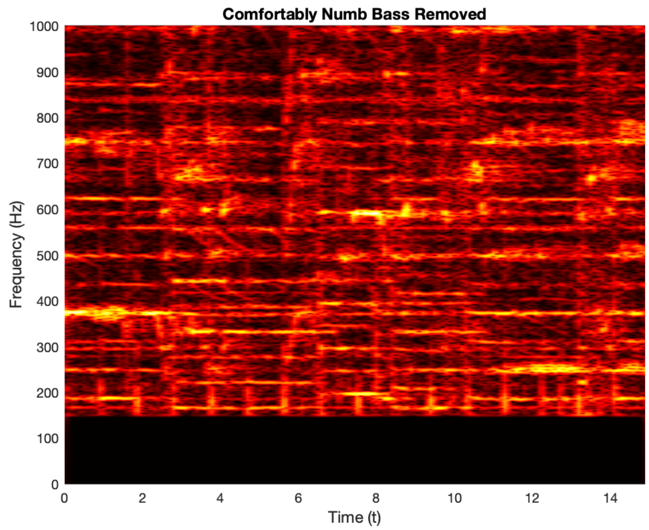

Figure 4: Here is the spectrogram of the Gabor transformed first 15 seconds of *Comfortably Numb* with all frequencies below 150 cut off.

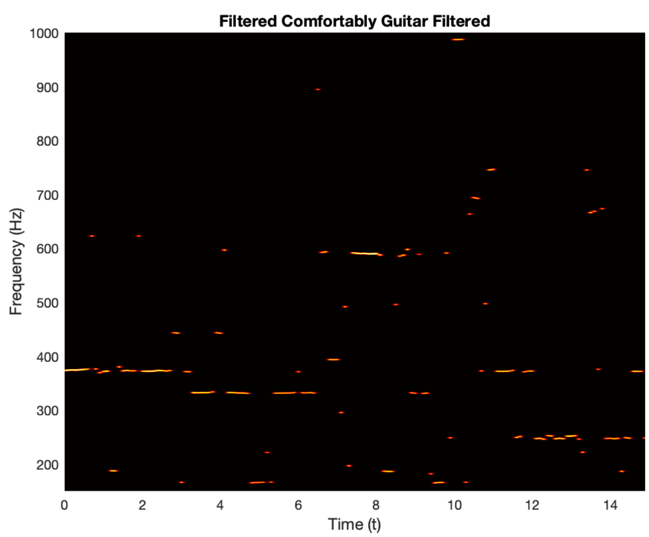

Figure 5: Here is the spectrogram of the Gabor transformed first 15 seconds of *Comfortably Numb* with all frequencies below 150 cut off filtered around the maximum values.

Appendix B MATLAB Code

Add your MATLAB code here. This section will not be included in your page limit of six pages.

```matlab
%% Loading Songs
clear all; close all; clc

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

[PF, PFfs] = audioread('Floyd.m4a');
PF = PF(1:end-1); % odd length
tr_PF = length(PF)/PFfs; % record time in seconds
tPF = PF(1:length(PF)/4);
ttr_PF = length(tPF)/PFfs;


%% Guns and Roses

a = 100;
L = tr_gnr; n = length(GNR);
t2 = linspace(0,L,n+1); t = t2(1:n);
k = (1/L)*[0:n/2-1  -n/2:-1];
ks = fftshift(k);
tslide = 0:0.1:tr_gnr;
spec = zeros(length(tslide),length(GNR));
maxFreq = zeros(length(tslide),1);
for j = 1:length(tslide)
    gaus = exp(-a*(t-tslide(j)).^2);
    gab = gaus.*transpose(GNR);
    gabt = fft(gab);
    spec(j,:) = abs(fftshift(gabt));
    AbsSgt = fftshift(abs(gabt));
    [maxv, idx] = max(AbsSgt);
    maxFreq(j) = ks(idx);
end

figure(1)
pcolor(tslide, ks, log((abs(spec.')+1))), shading interp, colormap hot
set(gca, 'Ylim', [0 1000])
title('Sweet Child O Mine', 'Fontsize', 12)
xlabel('Time (t)', 'Fontsize', 12)
ylabel('Frequency (Hz)', 'Fontsize', 12)
notes = [276 553 415 369 310 702 742];

%% Pink Floyd

a = 100;
L = ttr_PF;
n = length(tPF);
t2 = linspace(0,L,n+1); t = t2(1:n);
k = (1/L)*[0:n/2-1  -n/2:-1];
ks = fftshift(k);
tslide = 0:0.1:ttr_PF;
spec = zeros(length(tslide),n);
maxFreq = zeros(length(tslide),1);
for j = 1:length(tslide)
    gaus = exp(-a*(t-tslide(j)).^2);  %Gabor
    gab=gaus.*transpose(tPF);
    gabt = fft(gab);
    spec(j,:) = abs(fftshift(gabt));
    AbsSgt = fftshift(abs(gabt));
    [M, ind] = max(AbsSgt);
    maxFreq(j) = ks(ind);
end
figure(2)
pcolor(tslide, ks, log((abs(spec.')+1))), shading interp, colormap hot
set(gca, 'Ylim', [0 1000])
title('Comfortably Numb', 'Fontsize', 12)
xlabel('Time (t)', 'Fontsize', 12)
ylabel('Frequency (Hz)', 'Fontsize', 12)
notes = [124 110 98 90 83];
```

```matlab
%% PF Bass Filtered

tau = 1;
spec = zeros(length(tslide),n);
for j = 1:length(tslide)
    gaus = exp(-a*(t-tslide(j)).^2);  %Gabor
    gab=gaus.*transpose(tPF);
    gabt = fft(gab).*fftshift(exp(-tau*(ks-abs(maxFreq(j))).^2));
    spec(j,:) = abs(fftshift(gabt));
end

figure(3)
pcolor(tslide, ks, log((abs(spec.')+1))), shading interp, colormap hot
set(gca, 'Ylim', [50 250])
title('Filtered Comfortably Numb Bass', 'Fontsize', 12)
xlabel('Time (t)', 'Fontsize', 12)
ylabel('Frequency (Hz)', 'Fontsize', 12)

%% PF Guitar

L = tr_PF;
n = length(PF);
k = (1/L)*[0:n/2-1  -n/2:-1];
ks = fftshift(k);
FPF = fftshift(fft(PF));
for j = 1:length(ks)
    if abs(ks(j)) < 150
        FPF(j) = 0;
    end
end
nPF = ifft(fftshift(FPF));
%% PF Bass Removed

tPF = nPF(1:length(nPF)/4);
ttr_PF = length(tPF)/PFfs;
a = 100;
L = ttr_PF; n=length(tPF);
t2 = linspace(0,L,n+1); t = t2(1:n);
k = (1/L)*[0:n/2-1  -n/2:-1];
ks = fftshift(k);
tslide = 0:0.1:ttr_PF;
spec = zeros(length(tslide),n);
maxFreq = zeros(length(tslide),1);
for j = 1:length(tslide)
    gaus = exp(-a*(t-tslide(j)).^2);  %Gabor
    gab=gaus.*transpose(tPF);
    gabt = fft(gab);
    spec(j,:) = abs(fftshift(gabt));
    AbsSgt = fftshift(abs(gabt));
    [M, ind] = max(AbsSgt);
    maxFreq(j) = ks(ind);
end
figure(4)
pcolor(tslide, ks, log((abs(spec.')+1))), shading interp, colormap hot
set(gca, 'Ylim', [0 1000])
title('Comfortably Numb Bass Removed', 'Fontsize', 12)
xlabel('Time (t)', 'Fontsize', 12)
ylabel('Frequency (Hz)', 'Fontsize', 12)
```

```
%% PF Guitar Filtered

tau = 1;
spec = zeros(length(tslide),n);
for j = 1:length(tslide)
    gaus = exp(-a*(t-tslide(j)).^2);  %Gabor
    gab=gaus.*transpose(tPF);
    gabt = fft(gab).*fftshift(exp(-tau*(ks-abs(maxFreq(j))).^2));
    spec(j,:) = abs(fftshift(gabt));
end

figure(5)
pcolor(tslide, ks, log((abs(spec.')+1))), shading interp, colormap hot
set(gca, 'Ylim', [150 1000])
title('Filtered Comfortably Guitar Filtered', 'Fontsize', 12)
xlabel('Time (t)', 'Fontsize', 12)
ylabel('Frequency (Hz)', 'Fontsize', 12)
```