AMATH 482 Homework 2

Yiqing He

February 10th, 2021

Abstract

This report discusses the application of Gabor filtering on two pieces of Rock & Roll music, *Sweet Child O' Mine* and *Comfortably Numb* to reproduce and analyze the music score. Through this process, I will create spectrograms to visualize frequencies at specific points in time.

Introduction and Overview

At the beginning of the assignment, I import the two digital music files to MATLAB. Then, I used the Gabor filtering to reproduce the music score for the guitar in the GNR clip and the bass in the Floyd clip. I also plotted the spectrogram to visually give the information. Then, I used afilter in frequency space to try to isolate the bass in Comfortably Numb, through which I filtered out overtones first. Last but not least, I looked at smaller portions of the clip to guide the reconstruction of the music score.

Theoretical Background

This assignment mainly used the concept "Gaber filtering", which is given precisely by

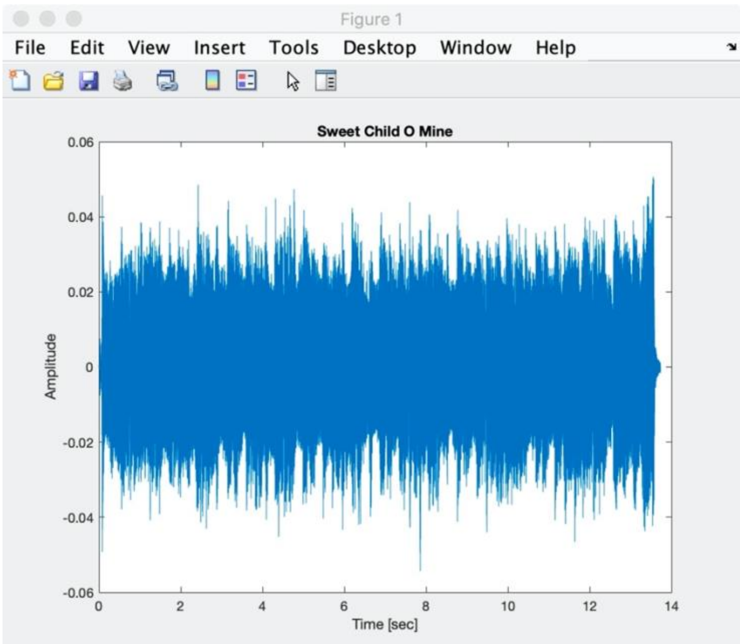
$$\tilde{f}_g(\tau, k) = \int_{-\infty}^{\infty} f(t)g(t-\tau)\mathrm{e}^{-\mathrm{i}kt}\mathrm{d}t.$$

This function is the Gabor transform, or STFT. In this function, the results are dependent on the choice of the filter g(t). Besides, there are many choices for g and some commonly used assumptions are 1) the function g is symmatric; 2) L_2-norm of g is set to unity.

The inverse of the Gabor transform is given by

$$f(t) = \frac{1}{2\pi\|g\|_2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{f}_g(\tau, k)g(t-\tau)\mathrm{e}^{\mathrm{i}kt}\mathrm{d}k\mathrm{d}\tau.$$

Algorithm Implementation and Development



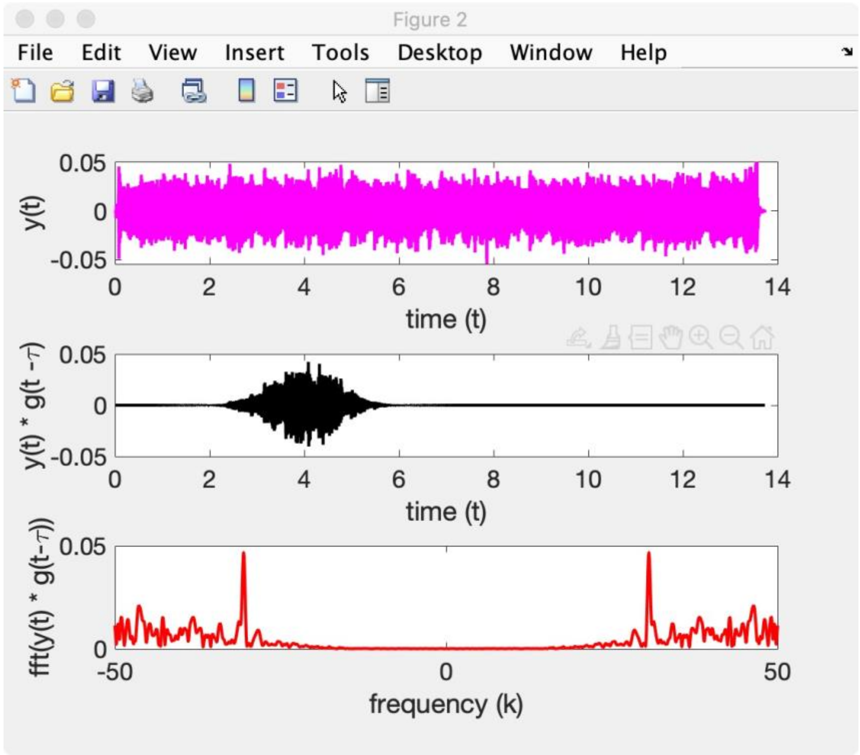
Both of the music pieces are converted to a vector representing the music, so I could easily edit the music by modifying the vector. The command “audioread” is used to read the two m4a files. Then, the two pieces could be played by the command “audioplayer” and “playblocking”. I got the graph on the left hand side for the song of *Sweet Child O’ Mine*.

I plotted three diagrams to illustrate the relationship between y(t) & time (t), y(t) * g(t-τ) & time (t) and fft(y(t) * g(t-τ)) & frequency (k). For these graphs, I used a center of window of 4 and the window size of 1.

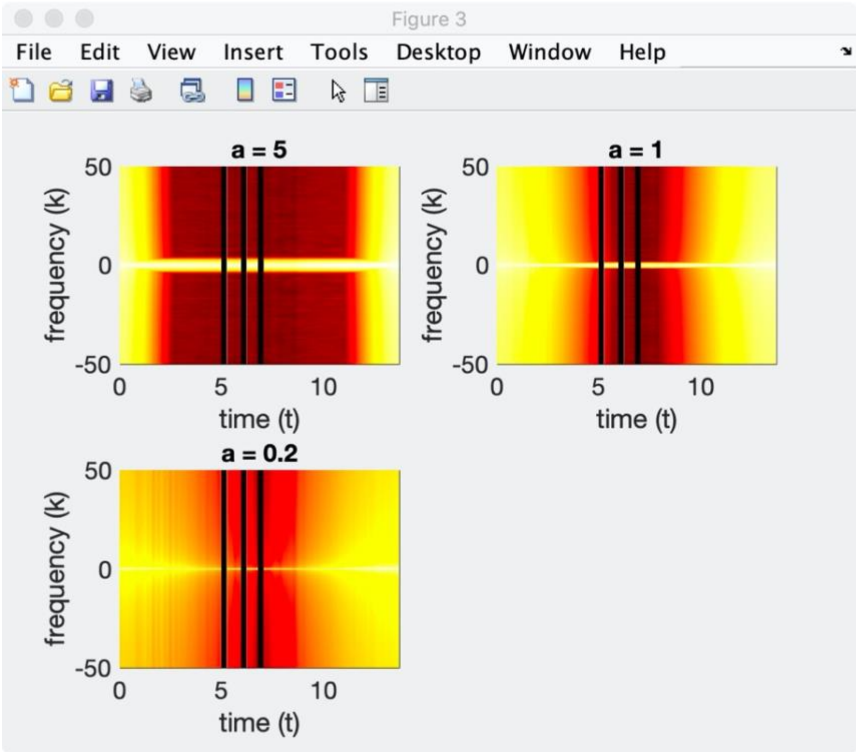
Then comes the most important part, the spectrograms. As in the lecture, I chose Gaussian for g(t). Then, the center of the window, tau, and the window size, a, should be determined. For the Gabor window, I used 5, 1 and 0.2 as the window size and I used 100 numbers from 0 to L as the center of the window. L is the length(y) divided by Fs.

Computational Results

The graph for y(t) * g(t-τ) & time (t) shows that somethings happens around time 4 in between 2 and 6. And the graph for fft(y(t) * g(t-τ)) & frequency (k) elucidates that there is information around -30 and 30, and the peaks are 0.05.



Below are the spectrograms. When a = 5, that the window size is comparatively small, there is more information of time and less information frequency. As the window size becomes larger, frequency is more localized.



Summary and Conclusions

from the graphs I could see that a = 5 gibes the best time solution, which means the most precision where those bright, low frequency spots are located in time. The high frequency content from the beginning of the signal decays in time. Though things are blurrier in time for a = 1, there is a slightly better resolution in the frequency domain.

Appendix A

audioread: reads data from the file named filename, and returns sampled data, y, and a sample rate for that data, Fs
audioplayer: creates an audioplayer object for signal Y, using sample rate Fs
playblocking: plays audio from audioplayer object, holds control until playback completesset: sets graphics object properties
pcolor: creates a pseudocolor plot
shading interp: varies the color in each line segment and face by interpolating the colormap index or true color value across the line or face
colormap: sets the colormap for the current figure

Appendix B

```
clc;clear all
figure(1)
[y, Fs] = audioread('GNR.m4a');
tr_gnr = length(y)/Fs; % record time in seconds
plot((1:length(y))/Fs,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Sweet Child O Mine');
p8 = audioplayer(y,Fs); playblocking(p8);

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);
a = 1;
tau = 4;
g = exp(- a * (t - tau).^2);
y = y.';
yg = y .* g;
```

```matlab
ygt = fft(yg);
ygt_spec = fftshift(abs(ygt));
figure(2)
subplot(3, 1, 1)
plot(t, y, 'k', 'Linewidth', 2)
hold on
plot(t, y, 'm', 'Linewidth', 2)
set(gca, 'Fontsize', 16), xlabel('time (t)'), ylabel('y(t)')
subplot(3, 1, 2)
plot(t, yg, 'k', 'Linewidth', 2)
set(gca, 'Fontsize', 16), xlabel('time (t)'), ylabel('y(t) * g(t -\tau)')
subplot(3, 1, 3)
plot(ks, abs(fftshift(ygt))/max(abs(ygt)), 'r', 'Linewidth', 2); axis([-50 50
0 0.05])
set(gca, 'Fontsize', 16), xlabel('frequency (k)'), ylabel('fft(y(t) * g(t-
\tau))')

figure(3)
a = [5 1 0.2];
tau = linspace(0, L, 100)
for jj = 1:length(a)
    ygt_spec = [];
    for j =  1:length(tau)
        g =  exp(-a(jj)*(t - tau(j)).^2);
        y = y.';
        yg = y(j) .* g;
        ygt = fft(yg);
        ygt_spec(:, j) = fftshift(abs(ygt));
    end
    subplot(2,2,jj)
    pcolor(tau, ks, log(ygt_spec) + 1)
    shading interp
    set(gca, 'ylim', [-50 50], 'Fontsize', 16)
    colormap(hot)
    xlabel('time (t)'), ylabel('frequency (k)')
    title(['a = ', num2str(a(jj))], 'Fontsize', 16)
end
```