Amath 482 Homework 2

Oscar Zhang

February, 6th, 2021

Abstract

In this project, we are given two different music file to use different filter and frequency to find music score for the file. The file we are given are two music file, we first need to convert them into vector form so that we can proceed further filter operation to get a clean data. After we get the data in vector form, we could then apply gabor filter to find out the fundamental frequency so that we could filter the overtone data.

1. Introduction

1.1 Main methodology

In order to filter out the guitar and bass music score, we should first convert the music file into vector form so that we are able to do operation on our data. After that, we should apply gabor filter to our data to find out the frequency domain. By this way, we can see that different instruments have different music score and frequency, according to the frequency, we are then able to filter the overtones and keep the part we need.

1.2 Data and programming language

For this project, we are going to use the matlab to solve the problem and the data we have are two music files which needed to be transformed into vector form data.

2. Theoretical Background

Gabor Transform

The main method we are using in this assignment is the gabor filter and gabor transform, which is:

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

With the inverse form:

$$f(t) = \frac{1}{2\pi\|g\|_2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_g(\tau,k)g(t-\tau)e^{ikt}dkd\tau$$

The idea is that when given a function g(t), we then are able to shift it by τ unit and multiplied by a function f(t) so that we could get filtered function$f(t)g(t-\tau)$ with filter center at τ.

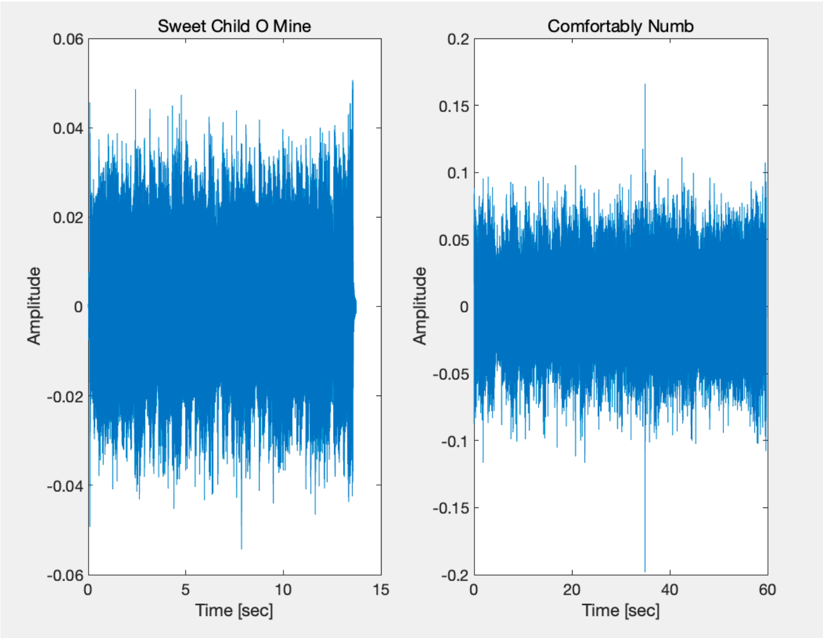
We also need to understand the concept of fundamental frequency – which is the lowest frequency for certain type of instrument and by this frequency, we are then able to filter out the overtone, which is the frequency that is higher than the fundamental frequency.

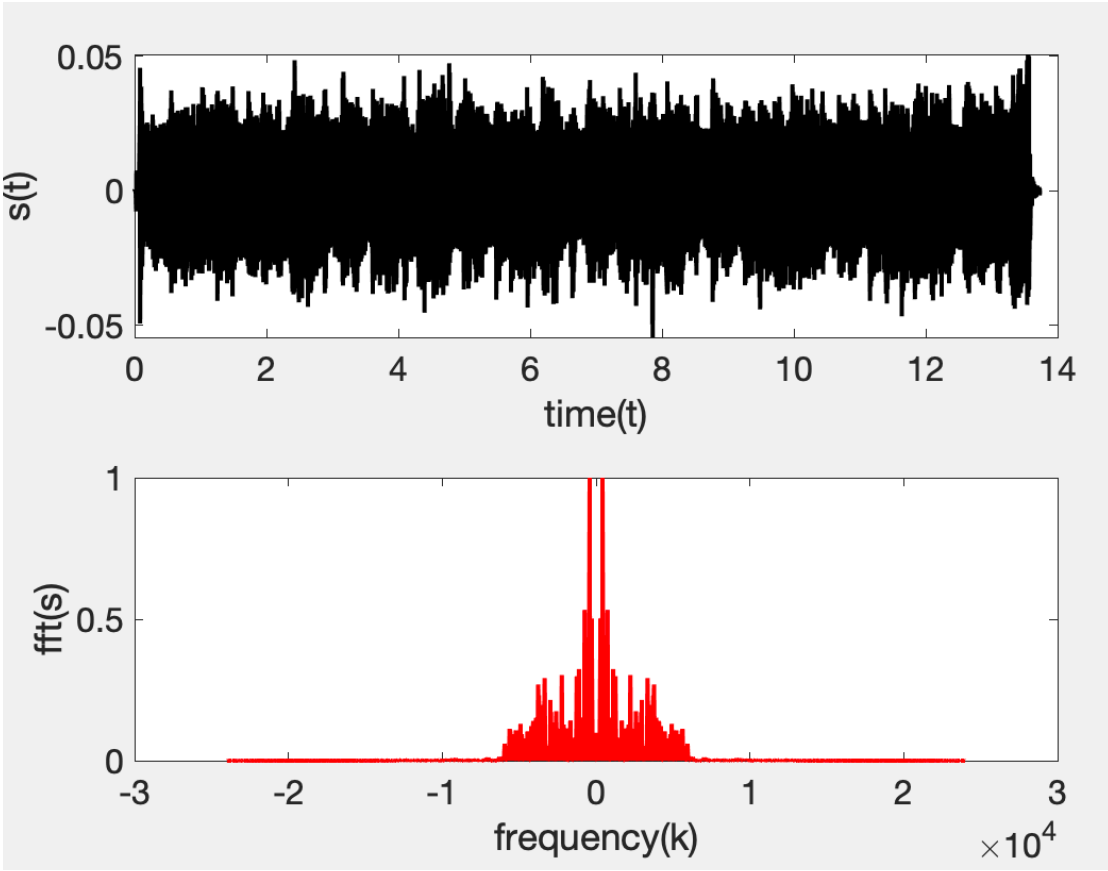
3 Algorithm Implementation and Development

For our assignment, given that two music file, we should first use the command [y, Fs] = audioread(file name) to convert the music file into vector form. Then we need to compute the length of the music in time so that we could proceed our gabor transform.

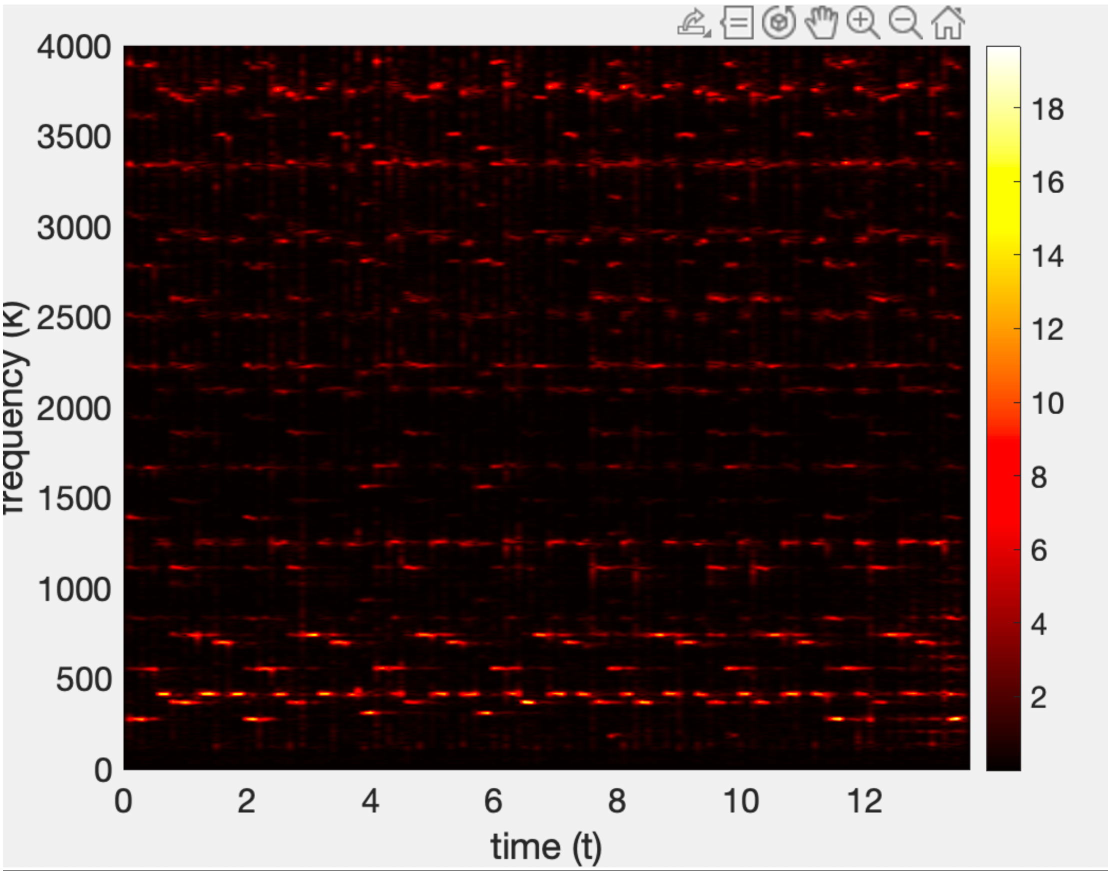
We apply command “fft” to y and we set k = (1/L) * [0:n/2-1 -n/2:-1] where L is the length of the music in second and n is the length of y, a vector that recording every points location in at every timestep. Then we set our a and tau, the width and the center of the gabor transform. Typically, for GNR.m4a, we set a = 1000 and tau = 0.1 and for Floyd.m4a file, we set a = 1000 and tau = 1. After this, we run a for loop for both music file and set a function g for the g(t). Multiplying g and y we get a new function sg and apply command “fft” to it. After this, we are then going to apply spectrogram to our new function and get the plot we need.

4 Computational Results





s(t)
time(t)
fft(s)
frequency(k)
$\times 10^4$



frequency (k)
time (t)

5 Conclusion

For this assignment, our goal is that when we are given certain kinds of data that mixed with different frequency, we are then able to apply the gabor transform so that we could categorize the frequency by its magnitude and based on this method, we can filter out the data we don't like and keep the data we need to run the spectrogram which is a better way to see the frequency distribution by the time.

Appendix A. Matlab Functions.

1. fftshift(x) = rearrange data x by putting all zero frequency data to the center of the data.
2. [,] = audioread(file name) = convert the audio file into vector form.
3. for: i = x:y = run a for loop by (y-x) times, repeat the code in the for loop each time, after each time, the i implements by 1.

Appendix B. Matlab codes

```
%% plot original signal graphs
figure(1)
subplot(1, 2, 1)
[y, Fs] = audioread('GNR.m4a');
tr_gnr = length(y) / Fs;
plot((1:length(y))/ Fs, y)
xlabel('Time [sec]')
ylabel('Amplitude')
title('Sweet Child O Mine')

subplot(1, 2, 2)
[y, Fs] = audioread('Floyd.m4a');
tr_fld = length(y) / Fs;
plot((1:length(y))/ Fs, y)
xlabel('Time [sec]')
ylabel('Amplitude')
title('Comfortably Numb')
%% GNR
[y, Fs] = audioread('GNR.m4a');
tr_gnr = length(y)/Fs;
y = y';

st = fft(y);
n = length(y);
L = tr_gnr;
t2 = linspace(0, L, n+1);
t = t2(1:n);
k = (1/L) * [0:n/2-1 -n/2:-1];
ks = fftshift(k);

figure(2)
subplot(2, 1, 1)
plot(t, y, 'k', 'Linewidth', 2)
set(gca, 'Fontsize', 16); xlabel('time(t)'); ylabel('s(t)')

subplot(2, 1, 2)
plot(ks, abs(fftshift(st))/max(abs(st)), 'r', 'Linewidth', 2);
set(gca, 'Fontsize', 16); xlabel('frequency(k)'); ylabel('fft(s)')
```

```
subplot(2, 1, 2)
plot(ks, abs(fftshift(st))/max(abs(st)), 'r', 'Linewidth', 2);
set(gca, 'Fontsize', 16); xlabel('frequency(k)'); ylabel('fft(s)')

a = 1000;
tau = 0:0.1:L;

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

   figure(3)
   subplot(3,1,1) % Time domain
    plot(t,y,'k','Linewidth',2)
    hold on
    plot(t,g,'m','Linewidth',2)
    set(gca,'Fontsize',16), xlabel('time (t)'), ylabel('S(t)')

    subplot(3,1,2) % Time domain
    plot(t,Sg,'k','Linewidth',2)
    set(gca,'Fontsize',16,'ylim',[-1 1]), xlabel('time (t)'), ylabel('S(t)*g(t-\tau)')

    subplot(3,1,3) % Fourier domain
    plot(ks,abs(fftshift(Sgt))/max(abs(Sgt)),'r','Linewidth',2);
    set(gca,'Fontsize',16), xlabel('frequency (k)'), ylabel('fft(S(t)*g(t-\tau))')
    drawnow
    pause(0.1)
    clf
end

a = 1000;
tau = 0:0.1:L;

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

figure(4)
pcolor(tau,ks,Sgt_spec)
shading interp
set(gca,'ylim',[0 4000],'Fontsize',16)
colormap(hot)
colorbar
xlabel('time (t)'), ylabel('frequency (k)')


%% Floyd
[y, Fs] = audioread('Floyd.m4a');
tr_fld = length(y)/Fs;
y = y(1:length(y) - 1);
y = y';

st = fft(y);
n = length(y);
L = tr_fld;
t2 = linspace(0, L, n+1);
t = t2(1:n);
k = (1/L) * [0:n/2-1 -n/2:-1];
ks = fftshift(k);

figure(3)
subplot(4, 1, 1)
plot(t, y, 'k', 'Linewidth', 2)
set(gca, 'Fontsize', 16); xlabel('time(t)'); ylabel('s(t)')

subplot(4, 1, 2)
plot(ks, abs(fftshift(st))/max(abs(st)), 'r', 'Linewidth', 2);
set(gca, 'Fontsize', 16); xlabel('frequency(k)'); ylabel('fft(s)')

a = 1000;
tau = 0:1:L;

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

```
   g = exp(-a*(t - tau(j)).^2);
   Sg = g.*y;
   Sgt = fft(Sg);

   figure(4)
   subplot(5,1,1) % Time domain
    plot(t,y,'k','Linewidth',2)
    hold on
    plot(t,g,'m','Linewidth',2)
    set(gca,'Fontsize',16), xlabel('time (t)'), ylabel('S(t)')

    subplot(5,1,2) % Time domain
    plot(t,Sg,'k','Linewidth',2)
    set(gca,'Fontsize',16,'ylim',[-1 1]), xlabel('time (t)'), ylabel('S(t)*g(t-\tau)')

    subplot(5,1,3) % Fourier domain
    plot(ks,abs(fftshift(Sgt))/max(abs(Sgt)),'r','Linewidth',2);
    set(gca,'Fontsize',16), xlabel('frequency (k)'), ylabel('fft(S(t)*g(t-\tau))')
    drawnow
    pause(0.1)
    clf
end

a = 1000;
tau = 0:1:L;

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

figure(5)
pcolor(tau,ks,Sgt_spec)
shading interp
set(gca,'ylim',[0 4000],'Fontsize',16)
colormap(hot)
colorbar
xlabel('time (t)'), ylabel('frequency (k)')
```