Reproduce the Guitar and Bass Music Score

Jiajun Bao

Abstract

This project aims to reproduce the music score for the guitar in the 14 seconds clip of the song *Sweet Child O' Mine* by Guns N' Roses and reproduce the music score for bass and find guitar solo in 60 seconds clip of the song *Comfortably Numb* by Pink Floyd, which are two of the greatest rock and roll songs of all time.

1 Introduction and Overview

In this project, the goal is to reproduce the guitar and bass music score, which requires us to find the frequency of the guitar in the GNR clip and frequencies of bass and guitar in the Floyd clip with time information. This means simply analyze the frequency of a signal through Fourier transfer is no longer viable, as Fourier transform can only tell us what frequencies are present, but it can't tell us when certain frequencies occur or how the frequency change over time. Therefore, we use the Gabor Transform to obtain information about a signal in both the time and frequency domain. Additionally, we need to filter out overtones(any frequency greater than the fundamental frequency of the instrument) in order to get a clean score.

2 Theoretical Background

2.1 Fourier Transform and Gabor Transform (Short-Time Fourier Transform)

Fourier Transform is widely used in applications like signal processing. It has advantages when dealing with characterizing stationary or periodic signals, but it loses information about what is happening in the time domain. In this case, as we need to produce music scores in the music clip, we need time information on what notes are being played at different times. As we learned, the Gabor Transform returns some information about time and some information about frequency. The Gabor Transform is representing by:

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

That is, for a fixed τ the function $\hat{f}_g(\tau, k)$ gives us information about the frequency components near time τ, and $g(t)$ is a filter function and we will use a Gaussian filter in this project

2.2 Discrete Gabor Transform

As we are given audio clips, we have a vector representing the signal data in time at discrete points rather than a function, so we need to use the discrete version and consider discrete sets of frequencies and time $k = mw_0$, $\tau = nt_0$, m and n are integers, the Discrete Gabor Transform is representing by:

$$\hat{f}_g(m, n) = \int_{-\infty}^{\infty} f(t)g(t-nt_0)e^{2\pi i m w_0 t}\,dt \tag{2}$$

2.3 Noise Filtering

The filtering method filters around the frequency and helps to remove undesired frequencies. In this case, we use this approach to filter out the overtones and high frequencies to isolate the bass line in *Comfortably Numb*. Mathematically, a filter is a function which we will multiply the signal in the frequency domain. We choose to use Gaussian function in the form of: $F(k) = e^{\tau(k-k_0)^2}$, with τ determining the width of the filter and constant k_0 determining the center of the filter.

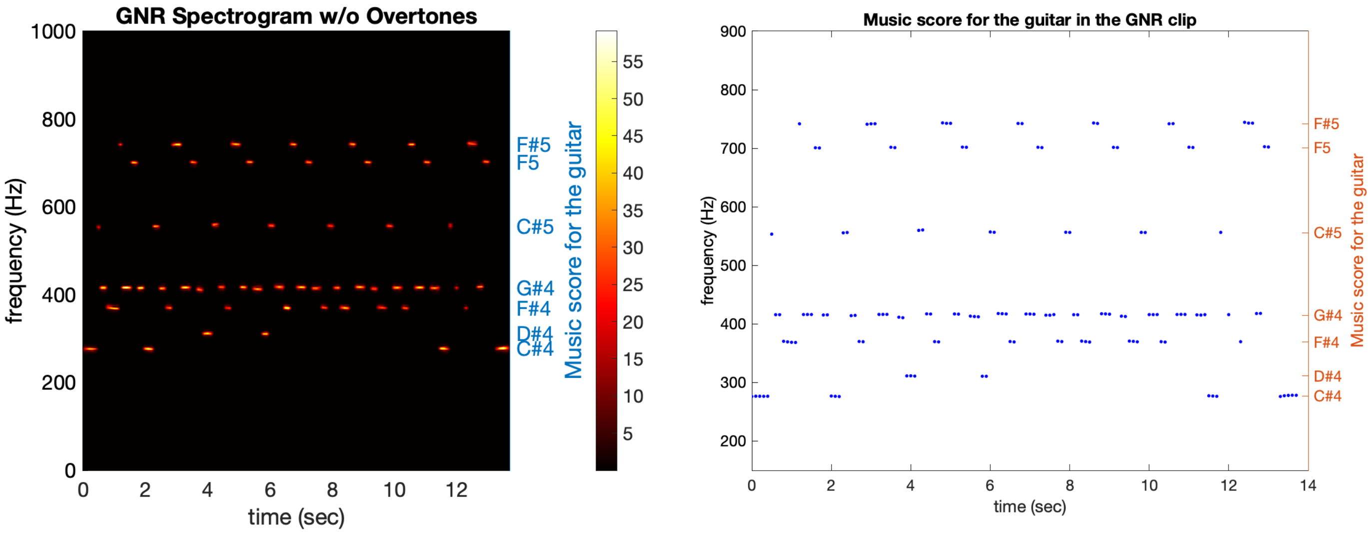
3 Algorithm Implementation and Development

- Read from **GNR.m4a**, get sampled data y and the sample rate Fs for the data, and record time in seconds. Then plot the graph of time and amplitude for the clip of *Sweet Child O' Mine*.
- Define length of time domain, then discretize the time domain. Rescale frequency by multiplying $(1/L)$, as in this case, the frequency is measured in Hertz instead of angular frequency. Then use *fftshift* to shift zero-frequency component to center of spectrum.
- Define the window function specs with size a and center of the window tau, and window function g.
- Create the spectrogram for the guitar in the GNR clip by setting the horizontal direction the value of window center and the vertical direction the frequency according to the frequency range of guitar [0,1000]. Then store the center frequency into a vector from each window.
- In order to get exact frequency for each time step, produce the music score diagram for the guitar in the GNR clip in dot plot format through plotting tau and the center frequency stored in the previous step.
- Label the notes according to the frequency of guitar on both diagram by creating another y axis using *yyaxis* and using *yticks* and *ylabels*.
- In order to get a cleaner spectrogram, we create another spectrogram for the GNR clip without overtone, by find the center frequency using *max* function and define a Gaussian filter around the center frequency, then apply the filter to the windowed signal in frequency space.
- Read from **Floyd.m4a**, get sampled data y and the sample rate Fs for the data, and record time in seconds. Then divide 60 secs Floyd Clip into four 15-secs smaller clips by define four sets of k, y, and tau for the purpose of computing efficiency.
- Repeat the same procedure as finding the guitar note in GNR clips, then create spectrograms according to the range of frequency for bass [0-150] Hz, and match the bass notes in the four divided subparts of Floyd clip.
- Follow the same steps as GNR clip to create another set of spectrogram and dot plots for the Floyd clip without overtone using Gaussian filter.
- To find guitar solo in Floyd clips, eliminate the fundamental frequency of the bass and its overtone and store the new center frequency into a vector from each window, then create four spectrograms for the four 15s subclips of the Floyd clips with frequency range [150-1000] Hz.
- To get the exact frequency at each time step for finding the music score for guitar, the dot plot is needed. By creating a *for* loop to filter out frequencies of bass obtained in previous steps. This is done by setting an error bound around the bass frequency and setting all the frequencies to fall within the bass frequency's error bound to 0. Still, due to the potential overlapping between the guitar and bass frequency and other factors, this method's limitation will be discussed in more detail in Part 5 Summary and Conclusions.

4 Computational Results

4.1 Music Score for the Guitar in the GNR Clip

Based on the above algorithms in section 3, (Please refer to Appendix B for detailed codes in MATLAB), we create the spectrogram and dot plot for the guiatr in the GNR clip according to the guitar fundamental frequency range [0-1000]Hz. The spectrogram with the music notes for the guitar in the GNR clip without overtone is displayed in Figure 1(a), and the dot plot version is plotted in Figure 1(b). Please refer to Appendix A for the Spectrogram of the guitar in the GNR clip before filtering out the overtone. The music scores for the guitar in the GNR clip are: C#4, D#4, F#4, G#4, C#5, F5, and F#5.



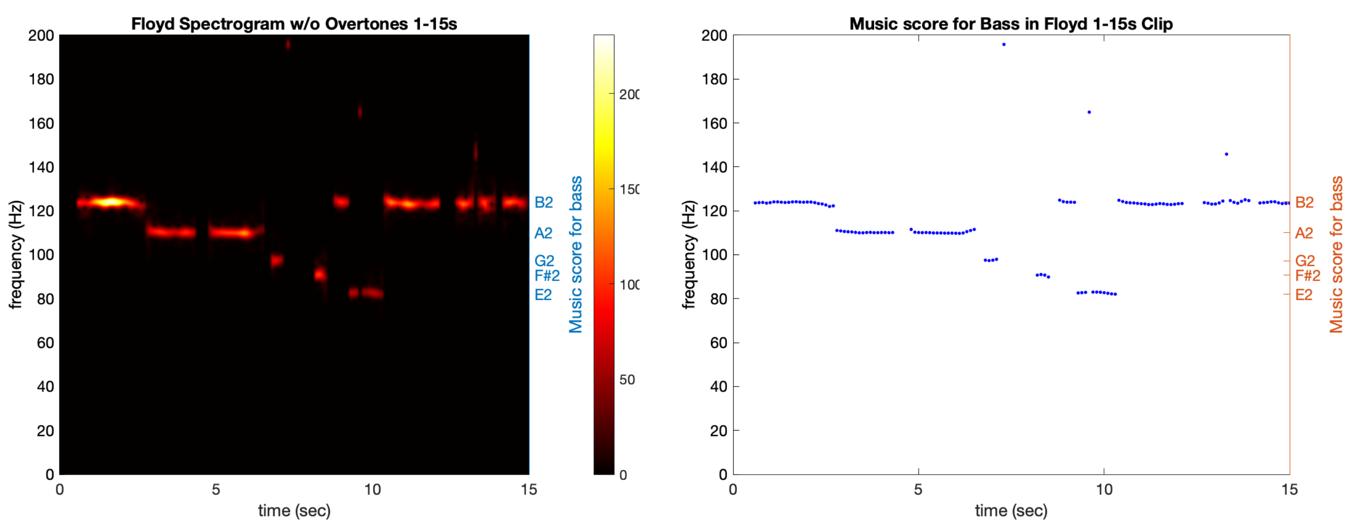
(a) GNR Spectrogram without Overtone

(b) Music score for the guitar in Dot Plot

Figure 1: Music score for the guitar in GNR Clip

4.2 Music Score for the Bass in the Floyd Clip 1-30s

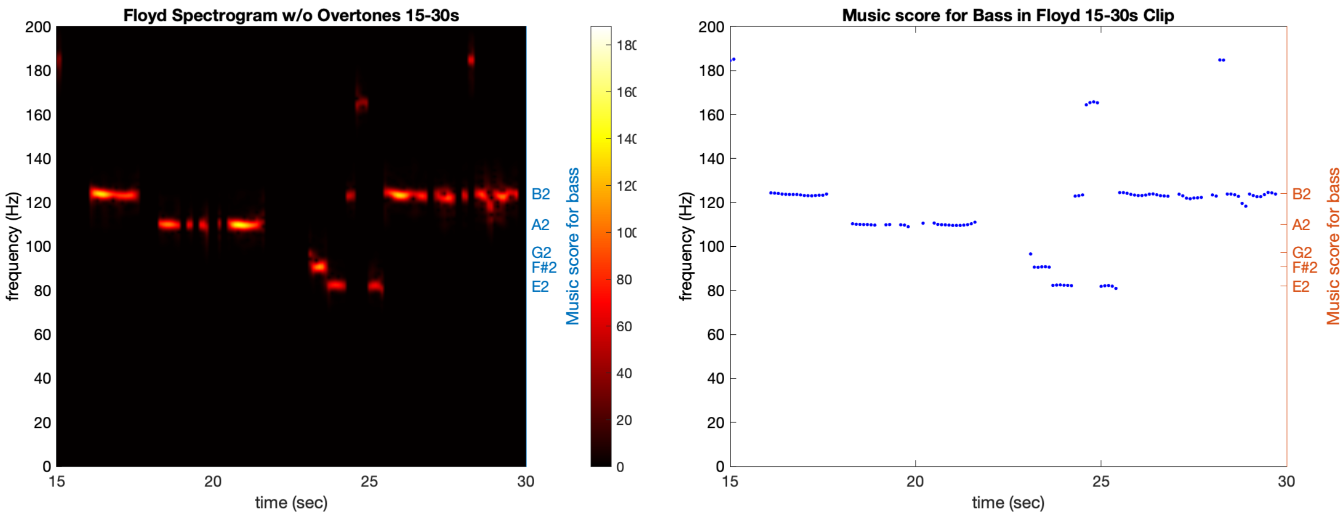
As in the above algorithms in section 3 mentioned, for computing efficiency purpose, the 60 secs Floyd clip was divided into four 15-secs subclip, and we create the spectrogram and dot plot for the bass in the Floyd clip according to the bass fundamental frequency range [50-150]Hz. The first 15 secs spectrogram with the music notes for the bass without overtone is displayed in Figure 2(a), and dot plot version is plotted in Figure 2(b). The second 15 secs spectrogram with the music notes for the bass without overtone is displayed in Figure 3(a), and dot plot version is plotted in Figure 3(b). Please refer to Appendix A for the Spectrogram of the bass in the Floyd clip before filtering out the overtone at first 30s clip. The music scores for the bass in the Floyd clip are: E2, F#2, G2, A2, and B2.



(a) Floyd Spectrogram w/o Overtone 1-15s

(b) Music score for the bass in Dot Plot 1-15s

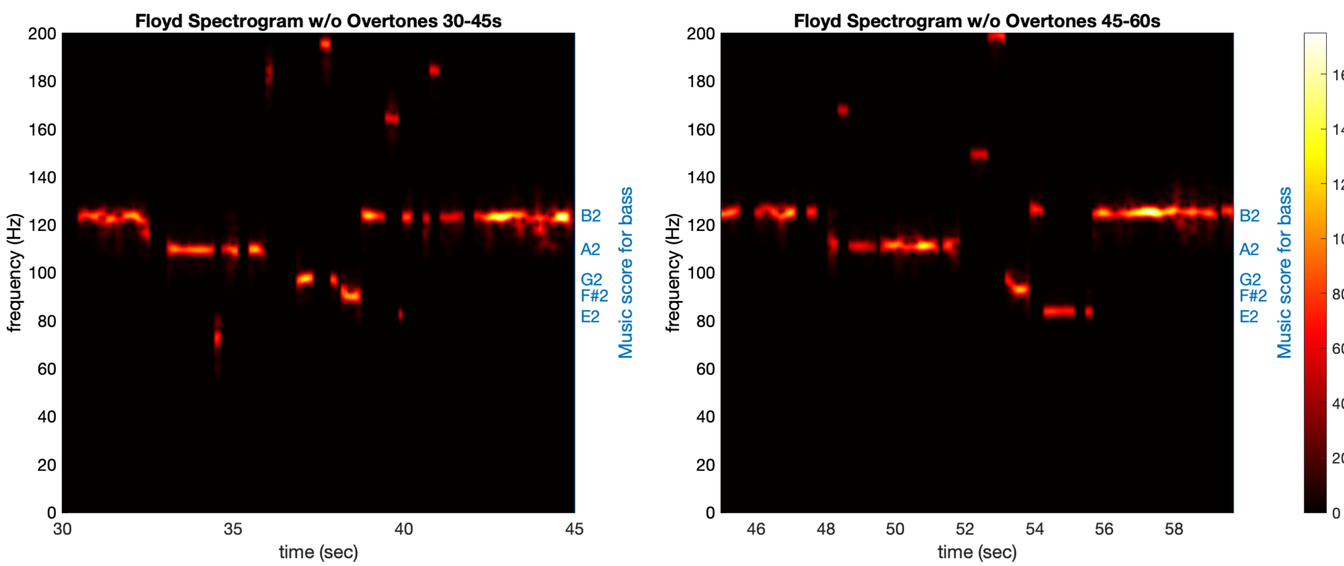
Figure 2: Music Score for the Bass in the Floyd Clip 1-15s



(a) Floyd Spectrogram w/o Overtone 15-30s

(b) Music score for the bass in Dot Plot 15-30s

Figure 3: Music Score for the Bass in the Floyd Clip 15-30s



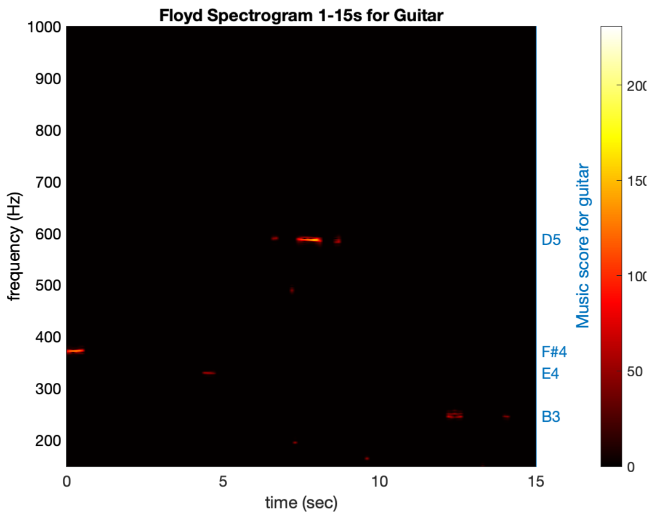
(a) Floyd Spectrogram w/o Overtone 30-45s

(b) Floyd Spectrogram w/o Overtone 45-60s

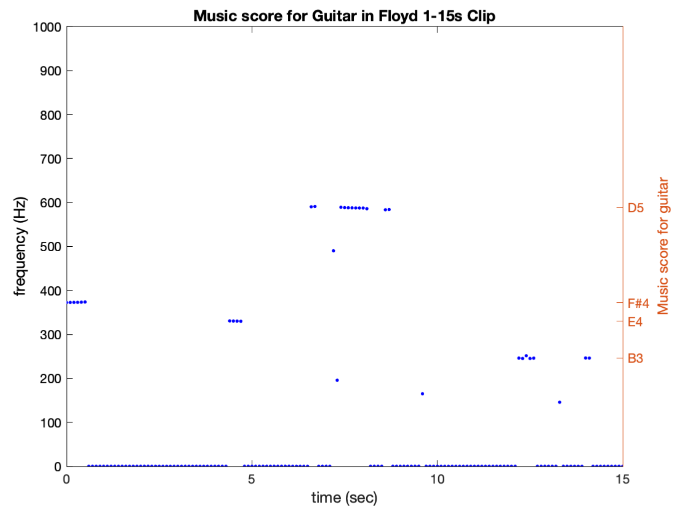
Figure 4: Music Score for the Bass in the Floyd Clip 30-60s

4.3 Music Score for the Bass in the Floyd Clip 30-60s

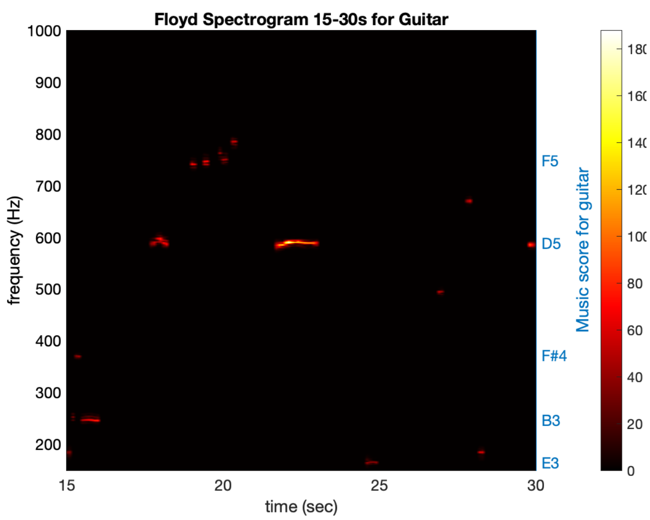
Follow the same procedures creating the 0-15s and 15-30s spectrogram for the bass in the Floyd clip. The 30-45s and 45-60 secs spectrogram are plotted in Figure 4(a) and Figure 4(b). The dot plots for 30-45s and 45-60s are not displayed in this report for simplicity as the bass in Floyd clip has periodic behavior of notes being played and only have very small variation of order in which notes being played compared to first 30 secs dot plot. Please refer to Appendix A for the Spectrogram of the bass in the Floyd clip before filtering out the overtone at 30-45s and 45-60s clip. The music scores for the bass in the Floyd clip are: E2, F#2, G2, A2,and B2.



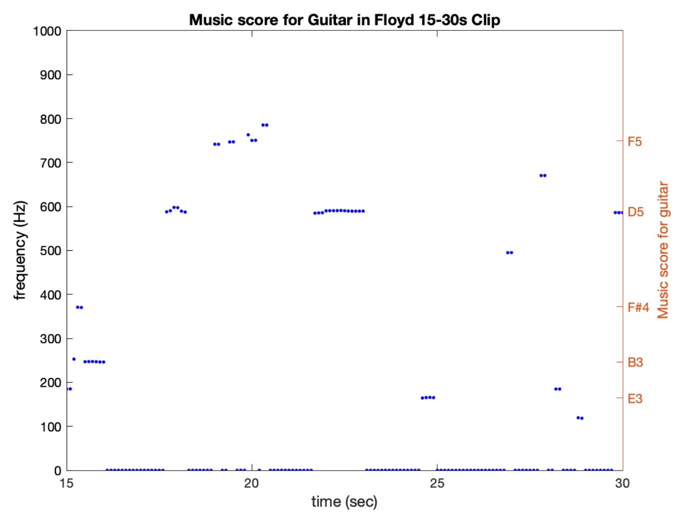
(a) Floyd Spectrogram for Guitar 1-15s



(b) Music Score for the Guitar Dot Plot 1-15s



(c) Floyd Spectrogram for Guitar 15-30s

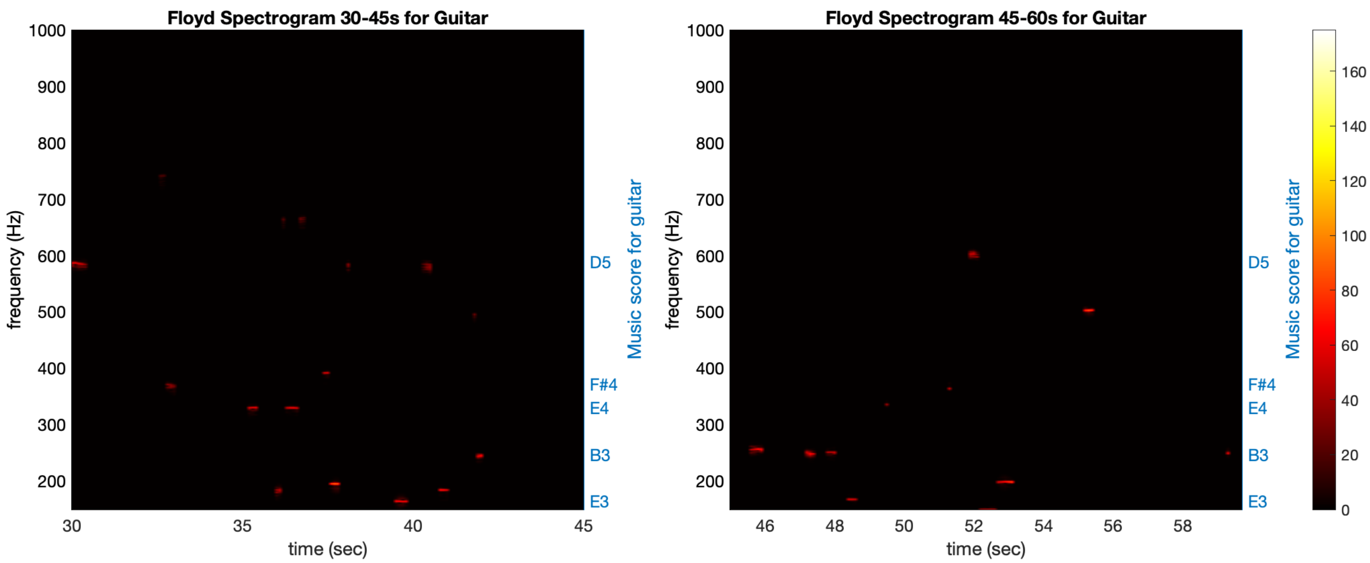


(d) Music Score for the Guitar Dot Plot 15-30s

Figure 5: Music Score for the Guitar in the Floyd Clip 1-30s

4.4 Find Guitar Solo in Floyd Clip

Similar to finding the bass's music score in the Floyd clip, we divide the 60s Floyd clips into four 15s subclips to find the guitar's music score in the Floyd clip. Based on the above algorithms in section 3, (Please refer to Appendix B for detailed codes in MATLAB), we create the spectrogram and dot plot for the guitar in the Floyd clip according to the guitar fundamental frequency range[0-1000Hz] by removing the fundamental frequency and overtone of bass. The first 15 secs spectrogram with the guitar's music notes in the Floyd clip without overtone is displayed in Figure 5(a), and the dot plot version is plotted in Figure 5(b). The second 15 secs spectrogram with the guitar's music notes in the Floyd clip without overtone is displayed in Figure 5(c), and the dot plot version is plotted in Figure 5(d). And the 30-45 secs and 45-60 secs spectrogram are plotted in Figure 6(a) and Figure 6(b). The music scores for the guitar in the Floyd clip are: E3, B3, E4, F#4, D5, F5.



(a) Floyd Spectrogram 30-45s for Guitar

(b) Floyd Spectrogram 45-60s for Guitar

Figure 6: Music Score for the Guitar in the Floyd Clip 30-60s

5 Summary and Conclusions

In this problem, to determine the music score for the guitar in the GNR Clip, and the music score for the bass in the Floyd Clip, and find guitar solo in the Floyd clip, we first construct the spectrogram for each instrument and utilize the dot plot to find the exact frequency in Hz to match the musical notes. The music scores for the guitar in the GNR clip are: C#4, D#4, F#4, G#4, C#5, F5, and F#5. The bass's music scores in the Floyd clip are: E2, F#2, G2, A2, and B2. And The music scores for the guitar in the Floyd clip are: E3, B3, E4, F#4, D5, F5. As mentioned in Part 3 Algorithm Implementation and Development part, the approach that filtering out the fundamental frequency of bass and its overtone to produce the music score of the guitar has limitations:

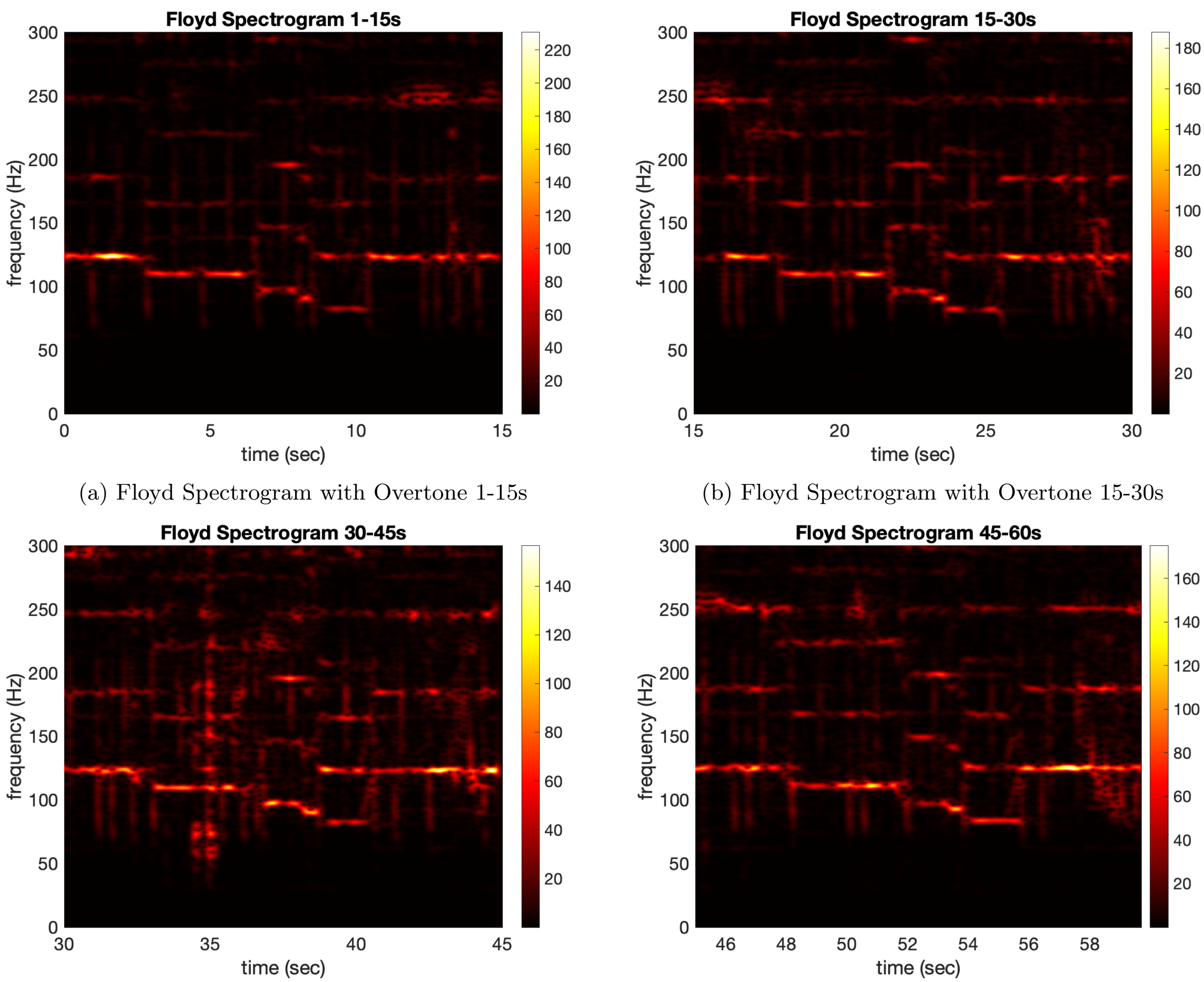
(1) There are potential cases when the guitar in the Floyd clip is played in the same frequency of the bass simultaneously or falls into the error bond we set up around the bass frequency. In this way, removing all the fundamental frequencies of bass and its overtone will also remove those frequencies of the guitar.

(2) There are other instruments involved in the Floyd clip, which are drum and keyboard. Both of their frequency ranges overlap with the frequency range of guitar, so the frequencies we found for guitar may be produced by the other two instruments.

Therefore, further information about the clip, such as whether there are instructions played at the same frequency for a given time, is needed to find the guitar solo in the Floyd clip.

Appendix A MATLAB Functions + Spectrograms with Overtone

- `[y,Fs] = audioread(filename)` reads data from the file named filename, and returns sampled data, y, and a sample rate for that data, Fs.
- `yyaxis right` activates the side of the current axes associated with the right y-axis. Subsequent graphics commands target the right side.
- `yticks(ticks)` sets the y-axis tick values, which are the locations along the y-axis where the tick marks appear.
- `yticklabels(labels))` sets the y-axis tick labels for the current axes.



(a) Floyd Spectrogram with Overtone 1-15s

(b) Floyd Spectrogram with Overtone 15-30s

(c) Floyd Spectrogram with Overtone 30-45s

(d) Floyd Spectrogram with Overtone 45-60s

Figure 7: Music Score for the Bass in the Floyd Clip with overtone

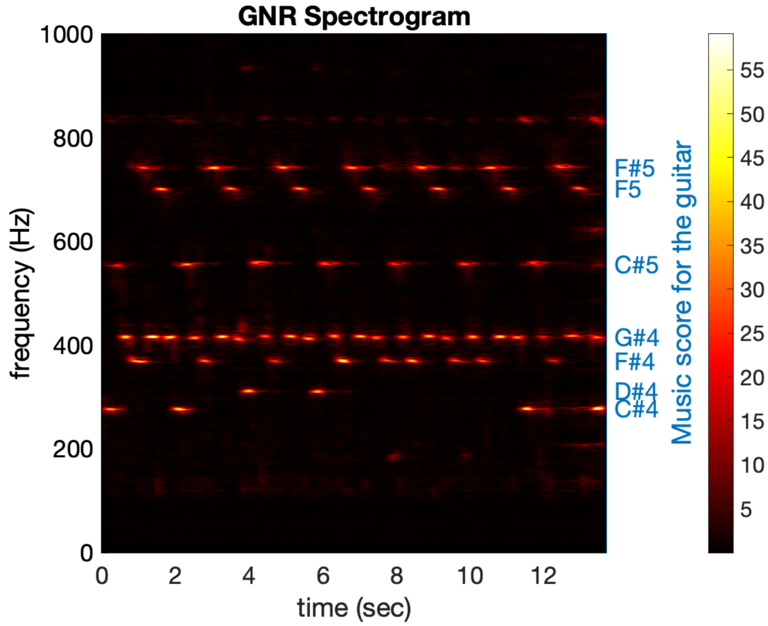


Figure 8: Music Score for the Guitar in the GNR Clip with overtone

Appendix B MATLAB Code

```
%% Part 1.1(a) Music score for the guitar in the GNR clip
[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);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
t2 = linspace(0,L,n+1); t = t2(1:n);

% Produce the GNR Spectrogram
a = 50;
tau = 0:0.1:tr_gnr;
freq_k= [];

for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2); % Window function
   Sg = g.*y';
   Sgt = fft(Sg);
   Sgt_spec(:,j) = fftshift(abs(Sgt));
   [M,I] = max(abs(Sgt));
   freq_k(j) = abs(k(I));
end

figure(2)
pcolor(tau,ks,Sgt_spec)
shading interp
set(gca,'ylim',[0 1000],'Fontsize',16)
```

```
colormap(hot)
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('GNR Spectrogram')
yyaxis right
ylabel('Music score for the guitar')
yticks([277.0, 311.0, 369.0, 415.0, 556.0, 701.0, 742.0])
yticklabels({'C#4','D#4','F#4','G#4','C#5','F5','F#5'})
print(gcf,'-dpng','GNR Spectrogram.png')

% Produce the music score diagram for the guitar in the GNR clip
figure(3)
plot(tau, freq_k, '.','Color','b','LineWidth', 3)
title('Music score for the guitar in the GNR clip','FontSize', 30)
ylim([150 900])
xlabel('time (sec)','FontSize', 12)
ylabel('frequency (Hz)','FontSize', 12)
yyaxis right
ylabel('Music score for the guitar')
yticks([277.0, 311.0, 369.0, 415.0, 556.0, 701.0, 742.0])
yticklabels({'C#4','D#4','F#4','G#4','C#5','F5','F#5'})
set(gca,'FontSize',12)
print(gcf,'-dpng','fig3.png')


%% Part 1.1(b)  Music score for the guitar in the GNR clip w/o overtone
close all; clear all; clc;
figure(1)
[y, Fs] = audioread('GNR.m4a');
tr_gnr = length(y)/Fs; % record time in seconds
L = tr_gnr; n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
t2 = linspace(0,L,n+1); t = t2(1:n);
a = 50;
tau = 0:0.1:tr_gnr;

for j = 1:length(tau)
   g = exp(-a*(t - tau(j)).^2); % Window function
   Sg = g.*y';
   Sgt = fft(Sg);
   Sgt_spec(:,j) = fftshift(abs(Sgt);
   [M1,I1] = max(abs(Sgt));
   filter = exp(-0.01*(k - k(I1)).^2);
   Sgtf = filter.*Sgt;
   Sgt_spec(:,j) = fftshift(abs(Sgtf));
end

figure(1)
pcolor(tau,ks,Sgt_spec)
shading interp
set(gca,'ylim',[0 1000],'Fontsize',16)
colormap(hot)
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('GNR Spectrogram w/o Overtones')
```

```
yyaxis right
ylabel('Music score for the guitar')
yticks([277.0, 311.0, 369.0, 415.0, 556.0, 701.0, 742.0])
yticklabels({'C#4','D#4','F#4','G#4','C#5','F5','F#5'})
print(gcf,'-dpng','GNR Spectrogram no overtone.png')

%% Part 1.2 Music score for the bass in the Floyd clip
close all; clear all; clc;
figure(1)
[y, Fs] = audioread('Floyd.m4a');
tr_gnr = length(y)/Fs; % record time in seconds
plot((1:length(y))/Fs,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Comfortably Numb');
p8 = audioplayer(y,Fs); playblocking(p8);
L = tr_gnr; n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
t2 = linspace(0,L,n+1); t = t2(1:n-1);

% Divide Floyd Spectrogram into 4 time periods with 25s each
k1 = (1/15)*[0:n/8-1 -n/8:-1];
ks1 = fftshift(k1);
k2 = (1/15)*[0:n/8-1 -n/8:-1];
ks2 = fftshift(k2);
k3 = (1/15)*[0:n/8-1 -n/8:-1];
ks3 = fftshift(k3);
k4 = (1/(L-45))*[0:n/8-1 -n/8:-1];
ks4 = fftshift(k4);
y1 = y(1:(length(y)-1)/4);
y2 = y((length(y)-1)/4+1:(length(y)-1)/2);
y3 = y((length(y)-1)/2+1:3*(length(y)-1)/4);
y4 = y(3*(length(y)-1)/4+1:length(y)-1);
t_1st_quarter = t(1:length(t)/4);
t_2nd_quarter = t(length(t)/4+1: length(t)/2);
t_3rd_quarter = t(length(t)/2+1:3*length(t)/4);
t_4th_quarter = t(3*length(t)/4+1:length(t));
tau1 = 0:0.1:15;
tau2 = 15:0.1:30;
tau3 = 30:0.1:45;
tau4 = 45:0.1:tr_gnr;

a = 100;
% Floyd 0 - 15s Spectrogram
for j = 1:length(tau1)
   g1 = exp(-a*(t_1st_quarter - tau1(j)).^2); % Window function
   Sg1 = g1.*y1';
   Sgt1 = fft(Sg1);
   Sgt_spec1(:,j) = fftshift(abs(Sgt1));
end

figure(2)
pcolor(tau1,ks1,Sgt_spec1)
shading interp
set(gca,'ylim',[0 300],'Fontsize',16)
colormap(hot)
```

```
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('Floyd Spectrogram 1-15s')
print(gcf,'-dpng',' Floyd Spectrogram 1-15.png')

% Floyd 15 - 30s Spectrogram
for j = 1:length(tau2)
   g2 = exp(-a*(t_2nd_quarter - tau2(j)).^2); % Window function
   Sg2 = g2.*y2';
   Sgt2 = fft(Sg2);
   Sgt_spec2(:,j) = fftshift(abs(Sgt2));
end

figure(3)
pcolor(tau2,ks2,Sgt_spec2)
shading interp
set(gca,'ylim',[0 300],'Fontsize',16)
colormap(hot)
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('Floyd Spectrogram 15-30s')
print(gcf,'-dpng',' Floyd Spectrogram 15-30s.png')

% Floyd 30 - 45s Spectrogram
for j = 1:length(tau3)
   g3 = exp(-a*(t_3rd_quarter - tau3(j)).^2); % Window function
   Sg3 = g3.*y3';
   Sgt3 = fft(Sg3);
   Sgt_spec3(:,j) = fftshift(abs(Sgt3));
end
figure(4)
pcolor(tau3,ks3,Sgt_spec3)
shading interp
set(gca,'ylim',[0 300],'Fontsize',16)
colormap(hot)
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('Floyd Spectrogram 30-45s')
print(gcf,'-dpng',' Floyd Spectrogram 30-45s.png')

% Floyd 45 - 60s Spectrogram
for j = 1:length(tau4)
   g4 = exp(-a*(t_4th_quarter - tau4(j)).^2); % Window function
   Sg4 = g4.*y4';
   Sgt4 = fft(Sg4);
   Sgt_spec4(:,j) = fftshift(abs(Sgt4));
end
figure(5)
pcolor(tau4,ks4,Sgt_spec4)
shading interp
set(gca,'ylim',[0 300],'Fontsize',16)
colormap(hot)
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('Floyd Spectrogram 45-60s')
print(gcf,'-dpng',' Floyd Spectrogram 45-60s.png')
```

```matlab
%% Part 2 Music score for the bass in the Floyd clip w/o overtone
close all; clear all; clc;
figure(1)
[y, Fs] = audioread('Floyd.m4a');
tr_gnr = length(y)/Fs; % record time in seconds
plot((1:length(y))/Fs,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Comfortably Numb');
p8 = audioplayer(y,Fs); playblocking(p8);
L = tr_gnr; n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
t2 = linspace(0,L,n+1); t = t2(1:n-1);

% Divide Floyd Clip into four 15-secs periods
k1 = (1/15)*[0:n/8-1 -n/8:-1];
ks1 = fftshift(k1);
k2 = (1/15)*[0:n/8-1 -n/8:-1];
ks2 = fftshift(k2);
k3 = (1/15)*[0:n/8-1 -n/8:-1];
ks3 = fftshift(k3);
k4 = (1/(L-45))*[0:n/8-1 -n/8:-1];
ks4 = fftshift(k4);
y1 = y(1:(length(y)-1)/4);
y2 = y((length(y)-1)/4+1:(length(y)-1)/2);
y3 = y((length(y)-1)/2+1:3*(length(y)-1)/4);
y4 = y(3*(length(y)-1)/4+1:length(y)-1);
t_1st_quarter = t(1:length(t)/4);
t_2nd_quarter = t(length(t)/4+1: length(t)/2);
t_3rd_quarter = t(length(t)/2+1:3*length(t)/4);
t_4th_quarter = t(3*length(t)/4+1:length(t));
tau1 = 0:0.1:15;
tau2 = 15:0.1:30;
tau3 = 30:0.1:45;
tau4 = 45:0.1:tr_gnr;
a = 50;
% Floyd 1 - 15s w/o overtone Spectrogram + Score
freq_k1= [];
for j = 1:length(tau1)
   g1 = exp(-a*(t_1st_quarter - tau1(j)).^2); % Window function
   Sg1 = g1.*y1';
   Sgt1 = fft(Sg1);
   [M1,I1] = max(abs(Sgt1));
   filter1 = exp(-0.01*(k1 - k1(I1)).^2);
   Sgtf1 = filter1.*Sgt1;
   [M,I] = max(abs(Sgtf1));
   freq_k1(j) = abs(k1(I));
   Sgt_spec1(:,j) = fftshift(abs(Sgtf1));
end
figure(2)
pcolor(tau1,ks1,Sgt_spec1)
shading interp
set(gca,'ylim',[0 200],'Fontsize',12)
colormap(hot)
colorbar
```

```
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('Floyd Spectrogram w/o Overtones 1-15s')
yyaxis right
ylabel('Music score for bass')
yticks([82.0, 90.7, 97.3, 110.0, 124.0])
yticklabels({'E2','F#2','G2','A2','B2'})
print(gcf,'-dpng','1-15_no_overtone.png')
figure(3)
plot(tau1, freq_k1, '.','Color','b','LineWidth', 3)
title('Music score for Bass in Floyd 1-15s Clip','FontSize',12)
ylim([0 200])
xlabel('time (sec)','FontSize', 12)
ylabel('frequency (Hz)','FontSize', 12)
yyaxis right
ylabel('Music score for bass')
yticks([82.0, 90.7, 97.3, 110.0, 124.0])
yticklabels({'E2','F#2','G2','A2','B2'})
set(gca,'FontSize',12)
print(gcf,'-dpng','1-15_notes.png')

% Floyd 15 - 30s w/o overtone Spectrogram + Score
freq_k2= [];
for j = 1:length(tau2)
   g2 = exp(-a*(t_2nd_quarter - tau2(j)).^2); % Window function
   Sg2 = g2.*y2';
   Sgt2 = fft(Sg2);
   [M2,I2] = max(abs(Sgt2));
   filter2 = exp(-0.01*(k2 - k2(I2)).^2);
   Sgtf2 = filter2.*Sgt2;
   [M,I] = max(abs(Sgtf2));
   freq_k2(j) = abs(k2(I));
   Sgt_spec2(:,j) = fftshift(abs(Sgtf2));
end
figure(4)
pcolor(tau2,ks2,Sgt_spec2)
shading interp
set(gca,'ylim',[0 200],'Fontsize',12)
colormap(hot)
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('Floyd Spectrogram w/o Overtones 15-30s')
yyaxis right
ylabel('Music score for bass')
yticks([82.0, 90.7, 97.3, 110.0, 124.0])
yticklabels({'E2','F#2','G2','A2','B2'})
print(gcf,'-dpng','15-30_no_overtone.png.png')
figure(5)
plot(tau2, freq_k2, '.','Color','b','LineWidth', 3)
title('Music score for Bass in Floyd 15-30s Clip','FontSize', 12)
ylim([0 200])
xlabel('time (sec)','FontSize', 12)
ylabel('frequency (Hz)','FontSize', 12)
yyaxis right
ylabel('Music score for bass')
yticks(  )
yticklabels({'E2','F#2','G2','A2','B2'})
```

```
set(gca,'FontSize',12)
print(gcf,'-dpng','15-30_notes.png')

% Floyd 30 - 45s w/o overtone Spectrogram + Score
for j = 1:length(tau3)
   g3 = exp(-a*(t_3rd_quarter - tau3(j)).^2); % Window function
   Sg3 = g3.*y3';
   Sgt3 = fft(Sg3);
   [M3,I3] = max(abs(Sgt3));
   filter3 = exp(-0.01*(k4 - k3(I3)).^2);
   Sgtf3 = filter3.*Sgt3;
   Sgt_spec3(:,j) = fftshift(abs(Sgtf3));
end
figure(6)
pcolor(tau3,ks3,Sgt_spec3)
shading interp
set(gca,'ylim',[0 200],'Fontsize',12)
colormap(hot)
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('Floyd Spectrogram w/o Overtones 30-45s')
yyaxis right
ylabel('Music score for bass')
yticks([82.0, 90.7, 97.3, 110.0, 124.0])
yticklabels({'E2','F#2','G2','A2','B2'})
print(gcf,'-dpng','30-45_no_overtone.png')

% Floyd 45 - 60s w/o overtone Spectrogram + Score
figure(7)
for j = 1:length(tau4)
   g4 = exp(-a*(t_4th_quarter - tau4(j)).^2); % Window function
   Sg4 = g4.*y4';
   Sgt4 = fft(Sg4);
   [M4,I4] = max(abs(Sgt4));
   filter4 = exp(-0.01*(k4 - k4(I4)).^2);
   Sgtf4 = filter4.*Sgt4;
   Sgt_spec4(:,j) = fftshift(abs(Sgtf4));
end
pcolor(tau4,ks4,Sgt_spec4)
shading interp
set(gca,'ylim',[0 200],'Fontsize',12)
colormap(hot)
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('Floyd Spectrogram w/o Overtones 45-60s')
yyaxis right
ylabel('Music score for bass')
yticks([82.0, 90.7, 97.3, 110.0, 124.0])
yticklabels({'E2','F#2','G2','A2','B2'})
print(gcf,'-dpng','45-60_no_overtone.png')
%% Find Guitar solo
% Floyd 0 - 15s Spectrogram for Guitar
freq_k1= [];
for j = 1:length(tau1)
   g1 = exp(-a*(t_1st_quarter - tau1(j)).^2); % Window function
   Sg1 = g1.*y1';
```

```
   Sgt1 = fft(Sg1);
   [M1,I1] = max(abs(Sgt1));
   filter1 = exp(-0.01*(k1 - k1(I1)).^2);
   Sgtf1 = filter1.*Sgt1;
   [M,I] = max(abs(Sgtf1));
   freq_k1(j) = abs(k1(I));
   Sgt_spec1(:,j) = fftshift(abs(Sgtf1));
end
figure(2)
pcolor(tau1,ks1,Sgt_spec1)
shading interp
set(gca,'ylim',[150 1000],'Fontsize',12)
colormap(hot)
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('Floyd Spectrogram 1-15s for Guitar')
yyaxis right
ylabel('Music score for guitar')
yticks([246.2, 330.0, 372.0, 589.0])
yticklabels({'B3','E4','F#4','D5'})
print(gcf,'-dpng','1-15_guitar.png')
err_bond = 3;
target = [82.0, 90.7, 97.3, 110.0, 124.0];
for i = 1:length(freq_k1)
    for j = 1:length(target)
        if abs(freq_k1(i)-target(j)) < err_bond
            freq_k1(i) = 0;
        end
    end
end
figure(3)
plot(tau1, freq_k1, '.','Color','b','LineWidth', 3)
title('Music score for Guitar in Floyd 1-15s Clip','FontSize', 12)
ylim([0 1000])
xlabel('time (sec)','FontSize', 12)
ylabel('frequency (Hz)','FontSize', 12)
yyaxis right
ylabel('Music score for guitar')
yticks([246.2, 330.0, 372.0, 589.0])
yticklabels({'B3','E4','F#4','D5'})
print(gcf,'-dpng','1-15_guitar_notes.png')

% Floyd 15 - 30s Spectrogram
freq_k2= [];
for j = 1:length(tau2)
   g2 = exp(-a*(t_2nd_quarter - tau2(j)).^2); % Window function
   Sg2 = g2.*y2';
   Sgt2 = fft(Sg2);
   [M2,I2] = max(abs(Sgt2));
   filter2 = exp(-0.01*(k2 - k2(I2)).^2);
   Sgtf2 = filter2.*Sgt2;
   [M,I] = max(abs(Sgtf2));
   freq_k2(j) = abs(k2(I));
   Sgt_spec2(:,j) = fftshift(abs(Sgtf2));
end
figure(4)
```

```
pcolor(tau2,ks2,Sgt_spec2)
shading interp
set(gca,'ylim',[150 1000],'Fontsize',12)
colormap(hot)
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('Floyd Spectrogram 15-30s for Guitar')
yyaxis right
ylabel('Music score for guitar')
yticks([164.5, 246.2, 372.0, 589.0, 749.0])
yticklabels({'E3','B3','F#4','D5','F5'})
print(gcf,'-dpng','15-30_guitar.png')
err_bond = 3;
target = [82.0, 90.7, 97.3, 110.0, 124.0];
for i = 1:length(freq_k2)
    for j = 1:length(target)
        if abs(freq_k2(i)-target(j)) < err_bond
            freq_k2(i) = 0;
        end
    end
end
figure(5)
plot(tau2, freq_k2, '.','Color','b','LineWidth', 3)
title('Music score for Guitar in Floyd 15-30s Clip','FontSize', 12)
ylim([0 1000])
xlabel('time (sec)','FontSize', 12)
ylabel('frequency (Hz)','FontSize', 12)
yyaxis right
ylabel('Music score for guitar')
yticks([164.5, 246.2, 372.0, 589.0, 749.0])
yticklabels({'E3','B3','F#4','D5','F5'})
print(gcf,'-dpng','15-30_guitar_notes.png')

%Floyd 30 - 45s Spectrogram
for j = 1:length(tau3)
   g3 = exp(-a*(t_3rd_quarter - tau3(j)).^2); % Window function
   Sg3 = g3.*y3';
   Sgt3 = fft(Sg3);
   [M3,I3] = max(abs(Sgt3));
   filter3 = exp(-0.01*(k4 - k3(I3)).^2);
   Sgtf3 = filter3.*Sgt3;
   Sgt_spec3(:,j) = fftshift(abs(Sgtf3));
end
figure(6)
pcolor(tau3,ks3,Sgt_spec3)
shading interp
set(gca,'ylim',[150 1000],'Fontsize',12)
colormap(hot)
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('Floyd Spectrogram 30-45s for Guitar')
yyaxis right
ylabel('Music score for guitar')
yticks([164.5, 246.2, 330.0, 372.0, 589.0])
yticklabels({'E3','B3','E4','F#4','D5'})
print(gcf,'-dpng','30-45_guitar.png')
```

```matlab
% Floyd 45 - 60s Spectrogram
for j = 1:length(tau4)
   g4 = exp(-a*(t_4th_quarter - tau4(j)).^2); % Window function
   Sg4 = g4.*y4';
   Sgt4 = fft(Sg4);
   [M4,I4] = max(abs(Sgt4));
   filter4 = exp(-0.01*(k4 - k4(I4)).^2);
   Sgtf4 = filter4.*Sgt4;
   Sgt_spec4(:,j) = fftshift(abs(Sgtf4));
end
figure(7)
pcolor(tau4,ks4,Sgt_spec4)
shading interp
set(gca,'ylim',[150 1000],'Fontsize',12)
colormap(hot)
colorbar
xlabel('time (sec)'), ylabel('frequency (Hz)')
title ('Floyd Spectrogram 45-60s for Guitar')
yyaxis right
ylabel('Music score for guitar')
yticks([164.5, 246.2, 330.0, 372.0, 589.0])
yticklabels({'E3','B3','E4','F#4','D5'})
print(gcf,'-dpng','45-60_guitar.png')
```