Application of Gabor Transform in Audio Data

Htet Aung Myin

February 7, 2020

Abstract

The application of the Gabor Transform on three different audio files is described. Differences from different wavelet formulas were observed, along with effects of varying window width, oversampling, and undersampling in spectrograms. The Gabor transform is also applied to observe overtones and derive a musical score for an audio file and identify the timbre of the instrument.

1 Introduction and Overview

The problem given is regarding three audio files. Audio files are presented as a wave with time and amplitude information. By using the Fourier Transform, the audio data can be moved into the frequency domain and Gabor filters can be applied to the data. The first audio file, Handel's Messiah, is analyzed using the Gabor transform to discover different wavelet formulas, window with, and sampling through spectrograms. In the following two audio files, the Gabor transform is then used to write a musical score with their corresponding frequencies and analyze the timbre of two different instruments.

2 Theoretical Background

2.1 Fast Fourier Transform

A Fast Fourier Transform(FFT) is an algorithm that computes the Discrete Fourier Transform (DFT) of a sequence. The Fourier Transforms converts a signal from the spatial domain into a frequency domain, or vice versa but gives no time data. The DFT is defined by the formula:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi ikn/N} \quad k = 0, \ldots, N-1$$

while the inverse Fourier Transform (iFFT) defined by:

$$X_n = \frac{1}{N}\sum_{k=0}^{N-1} x_k e^{2\pi ikn/N} \quad k = 0, \ldots, N-1$$

with $N = 2^n$ In comparison, the FFT runs at $O(NlogN)$ while the DFT runs at $O(N^2)$

2.2 Gabor Transform and Wavelets

The Gabor transform is a windowed Fourier transform that gives both time and frequency properties of a signal. It is also known as the short-time Fourier transform(STFT) and is defined by

$$G[f](t,\omega) = \bar{f}_g(t,\omega) = \int_{-\infty}^{\infty} f(\tau)\bar{g}(\tau - t)e^{-i\omega\tau}d\tau = (f, \bar{g}_{t,\omega})$$

The bar describes the complex conjugate of the function and $g(\tau - t)$ is the filter.

Examples of simple Gabor filters are the Gaussian time-filter, the Mexican-Hat wavelet, and the Shannon filter.

The Gaussian time-filter is described as

$$F(t) = e^{-\alpha(t-t_0)}$$

where α describes the width of the window.

The Mexican Hat Wavelets is described because it is shape and is defined as

$$\psi(t) = (1-(t-t_0)^2) * e^{\alpha(t-t_0)}$$

The Shannon filter is simply a step function that returns 1 in a given width or 0 otherwise.

Note: There are other windowed Fourier transforms other than the Gabor Transform.

2.3 Spectrograms

A spectrogram is a visualization tool that can describe a signal in both the time and frequency domain.

3 Algorithm Implementation and Development

Given data points in audio files, Handel's Messiah and two recordings of Mary Had A Little Lamb, the initial analysis is fairly similar for all three recordings.

1. The data is first read in (Handel's Messiah is part of MATLAB while audio wav files were provided for the two recordings of Mary Had A Little Lamb).

2. External details of the data were extracted such as the length of the file and number of data points. A range of frequency values is also set up. As FFT assumes 2π periodic signal, the frequency values are scaled by $2\pi/L$

3. A suitable time translation and width is determined and for each time translation, the Gabor filter is applied and the FFT of the filtered data is applied across time. Note: Due to computational constraints, the time translation may not be ideal.

4. The output is then displayed in an spectrogram.

3.1 Handel's Messiah

The goal of this section is to discover the consequences of window width, under sampling, over sampling, and different filters. In lines 1-10, the audio file is loaded in and the signal is plotted. Lines 18-19 describes the external details of the audio file while lines 27-44 displays four spectrograms with varying window widths. Line 79-92 tweak around with over sampling and under sampling. Line 94-119 describes a Mexican Hat Wavelet and compares it with a Gaussian filter.

3.2 Mary Had A Little Lamb

The goal of this section is to write a musical score for each respective instruments and analyze the difference in time frequency analysis. Lines 124-138, 195-198 reads in data and extract given data from the audio file. Lines 134-174, 216-244 describes a spectrogram as it translates across time. Lines 182-191 and lines 248-253 describes notes as frequencies and marks it on the histogram.

4 Computational Results

4.1 Results: Handel's Messiah

The MATLAB built in audio file Handel's Messiah is approximately 9 second longs with 73113 points of data. From the spectrograms, we can conclude that the smaller the width(larger α) the clearer the resolution as shown in Figure 1 in contrast to Figure 2. In Figure 3 and 4, it can be concluded that using too large a time translation can result in unclear time resolution while small time translational can be inefficient computationally.

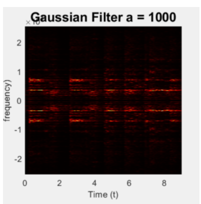

Figure 1: Gaussian Filter with Window Width a = 1000

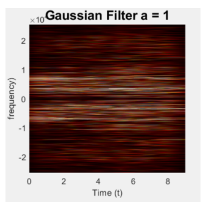

Figure 2: Gaussian Filter with Window Width a = 1

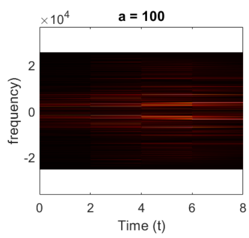

Figure 3: Undersampling in Gaussian Filter (a=1)

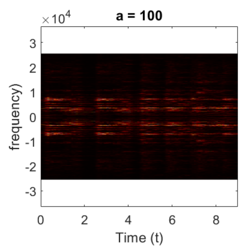

Figure 4: Oversampling in Gaussian Filter (a=1)

From the spectrograms, it can be concluded that different filter functions provide different sets of data as seen in Figure 5 and 6

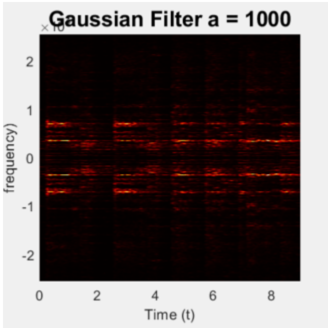

Figure 5: Gaussian Filter with Window Width a = 1000

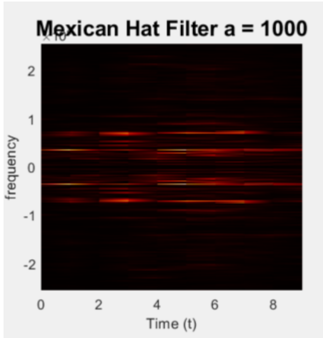

Figure 6: Mexican Hat Filter with Window Width a = 1000

4.2 Results: Mary Had A Little Lamb

With the gabor transform applied to both samples of Mary Had A Little Lamb, a high translation window was picked to give a cursory view of the center frequency. Once the frequency was determined, a smaller translation window is picked. The figures displayed are the best possible window given hardware constraints.

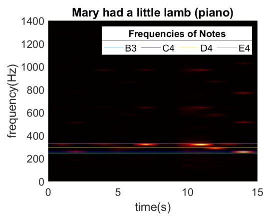

Figure 7: A Large Translation Window Over All Ranges for Piano

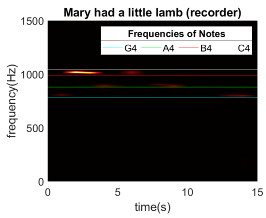

Figure 8: A Large Translation Window Over All Ranges for Piano

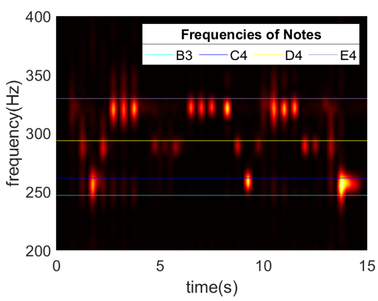

Figure 9: A Large Translation Window Over All Ranges for Piano

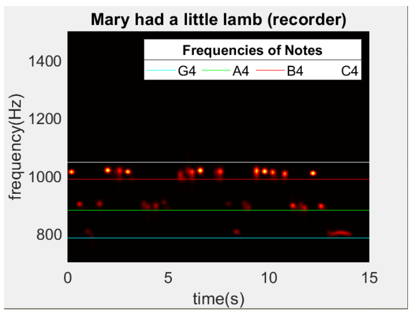

Figure 10: A Large Translation Window Over All Ranges for Piano

As observed from Figure 7 and 8, there are different concentrated frequencies, with 200-400 Hz for the piano and 700 - 1000 Hz for the recorder. Both displayed overtones with the piano being even throughout the frequencies while the overtones in recorder are intense in higher frequencies. These difference in intensities creates the timbre of the instrument.
Frequencies are then converted into a music score. By common musical knowledge, Mary Had A Little Lamb does not contain the sharps so only the base notes were indicated.
The final score derived from the spectogram for the piano is
$E4 - D4 - C4 - D4 - E4 - E4 - E4 - D4 - D4 - D4 - E4 - E4 - E4 - E4 - D4 - C4 - D4 - E4 - E4 - E4 - D4 - D4 - E4 - D4 - C4$

The final score derived from the spectogram for the recorder is
$B4 - A4 - G4 - A4 - B4 - B4 - B4 - A4 - A4 - A4 - B4 - B4 - B4 - B4 - A4 - G4 - A4 - B4 - B4 - B4 - A4 - A4 - B4 - A4 - G4$

5 Summary & Conclusion

The first audio file, Handel's Messiah was analyzed for the behavior of the Gabor filter after changing its window width, time translation, and sample. Oversampling leads to memory problems for the computer and undersampling causes poor time data.

The other two audio files, Mary Had A Little Lamb, were analyzed by finding their central frequencies before converting frequencies into musical scores. The recorder is observed to have higher frequencies than the piano.

6 Appendix A.

- audioread : read in data from wav audio files
- fft: used to fourier transform data into frequency domain
- fftshift: shift frequency to plottable data
- pcolor: plots spectrogram
- colormap: changes color of spectrogram

7 Appendix B.

```

%% loading handel's messiah
clear; close all; clc
load handel
v = y';

plot((1:length(v))/Fs,v);
xlabel('Time [sec]');
ylabel('Amplitude');
title('Signal of Interest, v(n)');

%% play muisc
p8 = audioplayer(v,Fs);
playblocking(p8);


%% Gabor Spectograms
L=9; n=length(v);
t2=linspace(0,L,n+1); t=t2(1:n);

k=(2*pi/L)*[0:(n-1)/2 -(n-1)/2:-1];
ks=fftshift(k);


a_vec = [1000 100 10 1];

for jj = 1:length(a_vec)
    a = a_vec(jj);
    tslide=0:0.09:9;
    Sgt_spec = zeros(length(tslide),n);
    for j=1:length(tslide)
        g=exp(-a*(t-tslide(j)).^2);
        Sg=g.*v;
        Sgt=fft(Sg);
        Sgt_spec(j,:) = fftshift(abs(Sgt));
    end

    subplot(2,2,jj)
```

```
    pcolor(tslide,ks,Sgt_spec.'), shading interp
    title(['Gaussian Filter a = ',num2str(a)],'Fontsize',16)

    xlabel('Time (t)'), ylabel('frequency)')
    colormap(hot)
end


%% Filter plots for oversampling/undersampling


a = 500;
tslide=0:0.1:9;

for j=1:50
    g=exp(-a*(t-tslide(j)).^2);
    Sg=g.*v;
    Sgt=fft(Sg);

    subplot(3,1,1)
    plot(t,v,'k','Linewidth',2)
    hold on
    plot(t,g,'m','Linewidth',2)
    hold off
    set(gca,'Fontsize',16), xlabel('Time (t)'), ylabel('S(t)')

    subplot(3,1,2)
    plot(t,Sg,'k','Linewidth',2)
    set(gca,'Fontsize',16), xlabel('Time (t)'), ylabel('Sg(t)')

    subplot(3,1,3)
    plot(ks,abs(fftshift(Sgt))/max(abs(Sgt)),'r','Linewidth',2);
    set(gca,'Fontsize',16)
    xlabel('frequency (\omega)'), ylabel('FFT(Sg)')
    drawnow
    pause(0.1)
end
```

```

%% Calculate Gabor transform and plot spectrogram
a = 1;
tslide=0:0.1:10;
Sgt_spec = zeros(length(tslide),n);
for j=1:length(tslide)
    g=exp(-a*(t-tslide(j)).^2);
    Sg=g.*S;
    Sgt=fft(Sg);
    Sgt_spec(j,:) = fftshift(abs(Sgt)); % We don't want
        to scale it
end

figure(6)
pcolor(tslide,ks,Sgt_spec.'),
shading interp
set(gca,'Ylim',[-50 50],'Fontsize',16)
colormap(hot)
%% mexican hat wavelet
L=9; n=length(y);
t2=linspace(0,L,n+1); t=t2(1:n);
k=(2*pi/L)*[0:(n-1)/2 -(n-1)/2:-1];
ks=fftshift(k);
St = fft(v);

a_vec = [1000 100 10 1];
for jj = 1:length(a_vec)
    a = a_vec(jj);
    tslide=0:1:9;
    Sgt_spec = zeros(length(tslide),n);
    for j=1:length(tslide)
        g = (1 - (t - tslide(j)).^2).*exp(-a*((t-tslide
            (j)).^2)/2);
        Sg=g.*v;
        Sgt=fft(Sg);
        Sgt_spec(j,:) = fftshift(abs(Sgt));
    end

    subplot(2,2,jj)
    pcolor(tslide,ks,Sgt_spec.'),
    shading interp
```

```
    title(['Mexican Hat Filter a = ',num2str(a)],'
       Fontsize',16)

    xlabel('Time (t)'), ylabel('frequency')
    colormap(hot)
end


%% Part 2, Piano
clear; close all; clc;
[y,Fs] = audioread('music1.wav');
tr_piano=length(y)/Fs; % record time in seconds
plot((1:length(y))/Fs,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Mary had a little lamb (piano)');
%p8 = audioplayer(y,Fs); playblocking(p8);
v = y';

%% spectrogram

n = length(v);
L = length(v)/Fs;
k=(2*pi/(L))*[0:n/2-1 -n/2:-1];
ks=fftshift(k);
t = (1:length(v))/Fs;

a = 500;
tslide = 0:.2:16;
Sgt_spec = zeros(length(tslide),n);

figure(1)
for j = 1:length(tslide)
    g = exp(-a*(t-tslide(j)).^2);


    Sg = g.*v;
    Sgt = fft(Sg);


    Sgt_spec(j,:) = fftshift(abs(Sgt));

```

```
    subplot(3,1,1)
    plot(t,v,'k',t,g,'r');
    set(gca,'Fontsize',16), xlabel('Time (t)'), ylabel('S(t)')
    axis([0 L -0.5 1]);
    subplot(3,1,2)
    plot(t, Sg);
    set(gca,'Fontsize',16), xlabel('Time (t)'), ylabel('Sg(t)')
    axis([0 L -0.5 1]);
    subplot(3,1,3)
    plot(ks,abs(fftshift(Sgt))/max(abs(Sgt)),'r','Linewidth',2);
    xlabel('frequency (\omega)'), ylabel('FFT(Sg)')
    pause(0.1)
end

figure(2)
subplot(1,1,1)
pcolor(tslide,ks/(2*pi),Sgt_spec.'), shading interp
axis([0 15 200 400])
set(gca,'Fontsize',16)
xlabel('time(s)'), ylabel('frequency(Hz)')

colormap(hot)
hold on
title('Mary had a little lamb (piano)');
hold on;


pb = plot([0 16],[246.94 246.94],'c') %B_3
hold on;
pc = plot([0 16],[261.63 261.63],'b'); %C_4
hold on;

pd = plot([0 16],[293.66 293.66],'y'); %D_4
hold on;

pe = plot([0 16],[329.63 329.63],'Color',[.61 .51 .74]) %E_4
hold on;
```

```
leg = legend([pb,pc,pd,pe],{'B3','C4','D4','E4',},'Orientation','horizontal')
leg.Title.String = 'Frequencies of Notes'
%% Part 2 Recorder
clear; close all; clc;
[y,Fs] = audioread('music2.wav');
tr_recorder=length(y)/Fs; % record time in seconds
plot((1:length(y))/Fs,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Mary had a little lamb (piano)');
%p8 = audioplayer(y,Fs); playblocking(p8);
v = y';
%% spectrogram
n = length(v);
L = length(v)/Fs;
k=(2*pi/(L))*[0:n/2-1 -n/2:-1];
ks=fftshift(k);
t = (1:length(v))/Fs;

a = 500;
tslide = 0:1:16;
Sgt_spec = zeros(length(tslide),n);

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


    Sgt_spec(j,:) = fftshift(abs(Sgt));

    subplot(3,1,1)
    plot(t,v,'k',t,g,'r');
    set(gca,'Fontsize',16), xlabel('Time (t)'), ylabel('S(t)')
    axis([0 L -0.5 1]);
    subplot(3,1,2)
    plot(t, Sg);
    set(gca,'Fontsize',16), xlabel('Time (t)'), ylabel(
```

```
        'Sg(t)')
    axis([0 L -0.5 1]);
    subplot(3,1,3)
    plot(ks,abs(fftshift(Sgt))/max(abs(Sgt)),'r','
        Linewidth',2);
    xlabel('frequency (\omega)'), ylabel('FFT(Sg)')
    pause(0.1)
end
figure(2)
subplot(1,1,1)
pcolor(tslide,ks/(2*pi),Sgt_spec.'), shading interp
axis([0 15 0 1500])
set(gca,'Fontsize',16)
xlabel('time(s)'), ylabel('frequency(Hz)')
title('Mary had a little lamb (recorder)');
hold on;
pg = plot([0 16],[783.99 783.99],'c') %G4 freq
hold on;

pa = plot([0 16],[880.00 880.00],'g'); %A4 freq
hold on;

pb=plot([0 16],[987.77 987.77],'r'); %B4 freq
hold on;
pc =plot([0 16],[1046.50 1046.50],'w') %C5 freq
colormap(hot)
hold on
leg = legend([pg,pa,pb,pc],{'G4','A4','B4','C4',},'
    Orientation','horizontal')
leg.Title.String = 'Frequencies of Notes'
```