AMATH 582 Homework 2

Hyesu Lee

Feb 7, 2020

Abstract

In this report, I will explain the definition of Gabor transformation, difference between Fourier transformation, and factors of Gabor transformation result. Lastly, I will perform Gabor transformation on songs to find music scores.

1 Introduction and Overview

For the first part of the report, I used the Gabor transform to create spectrogram of Handel's Messiah. Then, I recreated the spectrograms with different wavelet, sample size, bandwidth and compare the effects of each factor. For the second part, I used a Gabor filter to create spectrogram for "Mary had a little lamb" played with piano and recorder to compare the music score for each piece. The algorithms are implemented in MATLAB.

2 Theoretical Background

2.1 Time Frequency analysis / Windowed Fourier transform

Fourier transform transform any given time signal in to frequency content of signal. The big drawback of fourier transform is that it only applies to stationary or periodic signals. For example, if a music content is passed onto fourier transform, then it would presents every signal through out the time period. In other words, fourier transform could not capture time resolution. This complication could be easily solved by moving the some wavelet through out the time domain to capture the signal within the time period. Therefore, Gabor introduced the modified Fourier transform kernel equation 1 and Gabor transform, equation 1. The bar notation in equation 1 represents the complex conjugate of the function.[1] In equation 1, the function $g(\tau - t)$ shows that some wavelet window is moving through out the time period to capture frequency at the moment. The wavelet window is centered at τ with width a. Since Gabor transformation captures entire time-frequency content, there are two parameters: t and ω. For the simplicity, I will assume that g is real and symmetric such with property $||g(t)|| = 1$ and $||g(\tau - t)|| = 1$.

$$g_{t_\omega(\tau)} = exp(i\omega\tau)g(\tau - t) \tag{1}$$

The figure 2 illustrate the difference between time series, fourier transform, and gabor transform. Time series captures greate time resolution, but lose all the frequency content. Fourier transform captrues good frequency resolution, but lose all the ttime content. Gabor transform captures both time and freuquency resolution by trading array some accuracy in both measurement.
There is, also, a downside of Gabor transformation. If the wavelet window is short, then the window would have good time content but lose many low frequency content. If the wavelet window is long, then the window

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

Figure 1: Gabor transformation [1]

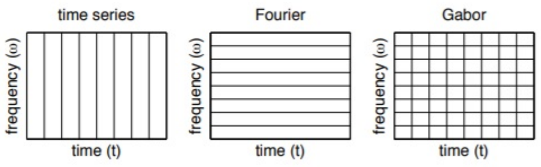


Figure 2: Time series vs. Fourier vs. Gabor [1]

would lose the time content, but be able to capture much more frequency contents. Therefore, the length of the wavelet window should be modified for the matter of the problem.

2.2 Wavelets for Gabor transformation

For Gabor transformation, different wavelets could be used for the purpose of the problem. Gaussian wavelet, equation 2, is the most often used wavelet.
Another wavelet option is Haar wavelet, step function. Haar wavelet is an ideal wavelet for describing localized signals in time or space, but it has poor localization in the frequency domain.
Last common wavelet is the Mexican hat wavelet, equation 4. It has great localization in both time and frequency due to the minimal time-bandwidth product of the Gaussian function.

$$F(t) = exp(-1(t-b)^2) \tag{2}$$

$$\phi(t) = \begin{cases} 1 & \text{if } \frac{t-b}{a} < t < \frac{t-b}{a} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

$$\phi(t) = (1 - (\frac{t-b}{a})^2)e^{-\frac{1}{2}} * (\frac{t-b}{a})^2 \tag{4}$$

3 Algorithm Implementation and Development

3.1 Part1: Handel's Messiah time-frequency analysis

In part 1, I will build the spectrogram of Handel's Messiah and analyze how window width of the Gabor tranform, oversampling/undersampling, and different wavelet window affect the result.
Algorithm to analyze the difference between window width:

1. Load the music and fourier transform the time component.
2. By using the Gaussian window with different widths(short, medium ,long), Gabor transform the music content by multiplying the window to time content.
3. Repeat the procedure 2 for every time unit.
4. Plot the spectrogram to compare the differences.

The width of wavelet window could be done by modifying d in equation 5. short: 200 medium: 1 long: 0.05.

$$g = exp(-d * (\tau - t)^2) \tag{5}$$

Algorithm to analyze the difference between oversampling and undersampling:

1. Load the music and fourier transform the time component.
2. Divide the time period with large units, 2, and small units,0.1.

3. By using the Gaussian window with same width, Gabor transform the music content by multiplying the window to time content.
4. Repeat the procedure 2 for every time unit.
5. Plot the spectrogram to compare the differences.

Algorithm to analyze the difference between wavelets:

1. Load the music and fourier transform the time component.
2. By using the Gaussian window with different widths, Gabor transform the music content by multiplying the window to time content.
3. Repeat the procedure 2 for every time unit.
4. Plot the spectrogram to compare the differences.
5. Plot the spectrogram with Mexican hat wavelet and Haar wavelet.

In part 2, I built two spectrograms of "Mary had a little lamb" played in piano and recorder. Then, used equation 6 to filter the overtone for clear spectrogram. Overtone is generated by the instrument for the center frequency. If ω_0 is center frequency, then $2 * \omega_0, 3 * \omega_0 ....$ are overtones. In order to remove overtone, I found the center frequency of each window by finding the highest point in fourier domain. Lastly, I compared the two spectrogram and define the music score of the song.

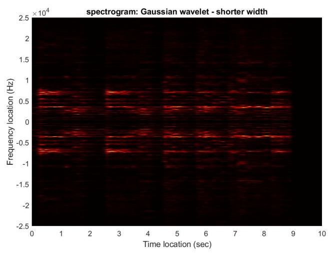
$$F(t) = exp(-\tau(\omega -_0)^2) + exp(-\tau(+_0)^2)) \tag{6}$$

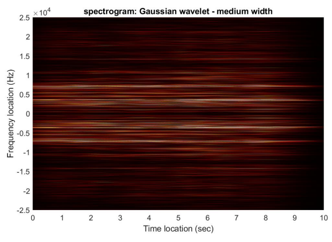
4 Computational Results

4.1 Part 1

4.1.1 Window width

As shown in image 3, 4, 5; the length of the band width affects the accuracy of time content and frequency content. Having smaller wavelet width, image 3 captured great time components, but lost a lot of frequency component. Having wider wavelet width, image 5 captures a lot more frequency content, but it is hard to determine when the frequency occur.





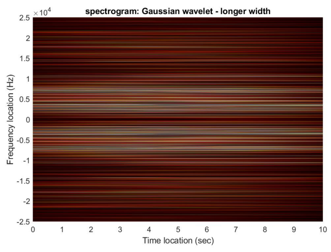


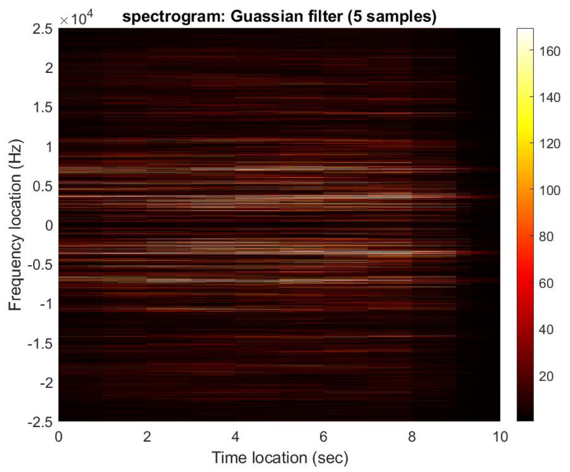
Figure 3: Short width

Figure 4: Medium

Figure 5: Long width

4.1.2 Oversampling vs undersampling

I achieved oversampling and under sampling by changing the movement of Gabor(Gaussian wavelet) window. From the images 7 and image 6, it is obvious that oversampling provides continuous frequency content features. Undersampled image has abrupt discontinuity of frequency content between each time units.



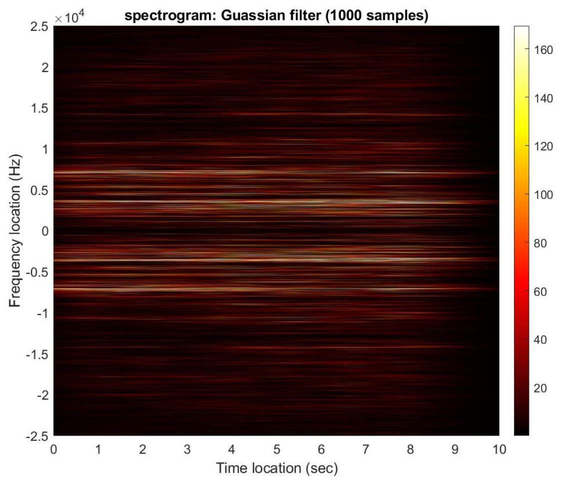
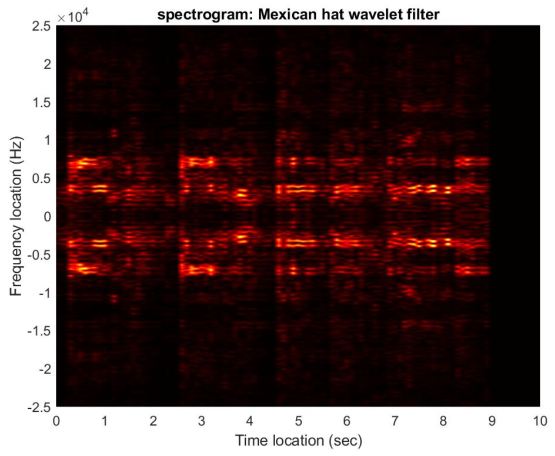


Figure 6: 5 samples: undersampling

Figure 7: 1000 samples: oversampling

4.1.3 Gaussian vs Mexican hat vs Haar

I used all three wavelets to execute Gabor transform



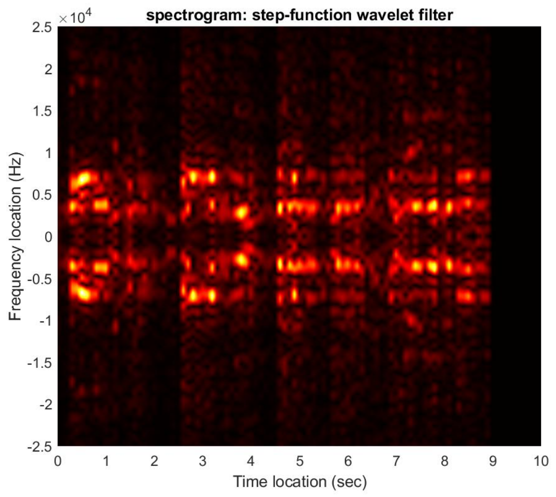


Figure 8: Mexican hat wavelet

Figure 9: Haar wavelet

4.2 Part2

When the spectrogram was built with piano sample and recorder sample, there are some overtones in piano sample. Those overtones occur the multiples of center frequencies. This feature,image 10, indicates that piano has more frequency occurring when one note is played. By filtering out the overtone, it has clear image of the music frequency tones. I used the music scale given in the notes to find the music score for each samples.

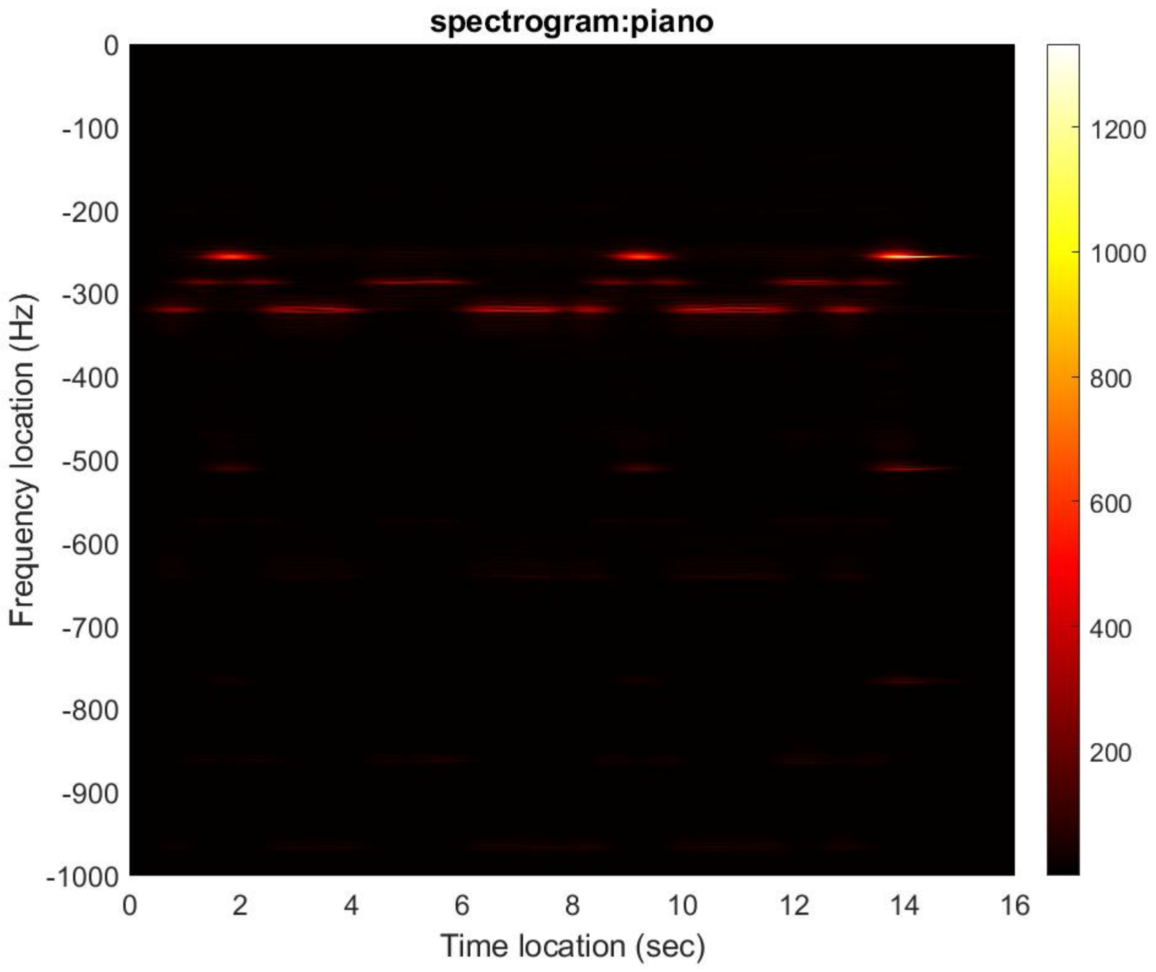


Figure 10: Piano spectrogram with overtones

Lastly, the music score for the song is EDCDEEEDDDEEEEDCDEEEEDDEDC in piano sample and BAGABBBAAABBBBAGABBBBAABAGGG in Recorder sample.

5 Summary and Conclusions

From the Handel's Messiah example, I understood how different factors could affect the time-frequency analysis. From the "Mary had a little lamb", I demonstrated the time frequency analysis on both piano and recorder samples. Also, the music score for the song is EDCDEEEDDDEEEEDCDEEEEDDEDC in piano sample and BAGABBBAAABBBBAGABBBBAABAGGG in Recorder sample.

References

[1] Jose Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.

Appendix A MATLAB Functions

changing x and y using ind2sub Followings are MATLAB Functions used for the assignment:

- `y = fftshift(x)` return the rearranged Fourier transform x which contains zero-frequency component at the center of the array.
- `y = fft(x)` returns Fourier transform of nth dimensional array using a fast Fourier transform algorithm. It needs fftshift() for rearrangement.
- `B = reshape(A, sz)` return reshaped matrix A, using the size vector sz for size of matrix B.
- `[y, x, z] = ind2sub(sz, ind)` returns three arrays containing equivalent multidimensional subscripts corresponding to the linear indices ind for a multidimensional array of size sz. Since the return value format is [row,col], assignment for x-axis and y-axis should be ordered as [y-axis, x-axis].
- `[Y,Fs] = audioread(Filename)` reads an audio file specified by the character vector or string scalar filename. returns the sampled data in Y and the sample rate Fs.
- `pcolor(C)` a pseudo color or ”check board” plot of matrix C.

Appendix B MATLAB Code

```
load handel
v = y'/2;
time = 1: length(v);
plot(time/Fs,v);
xlabel('Time [sec]');
ylabel('Amplitude');
title('Signal of Interest, v(n)');
p8 = audioplayer(v,Fs);
L = time(end)/Fs; % seconds
n = time(end);
k = (2 * pi / L) * [0 : n/2-1 -n/2:-1];
ks = fftshift(k);
t2 = linspace(0,L,n+1);
t = t2(1:n);
vt=fft(v);
plot(abs(fftshift(vt))); % plotting in frequency domain

playblocking(p8);
%% Gabor window

figure(2)
width=[10 1 0.2];
for j=1:3
    g=exp(-width(j)*(t-4).^2);
    subplot(3,1,j)
    plot(time,v,'k'), hold on
    plot(time,g,'r','Linewidth',[2])
    set(gca,'Fontsize',[14])
    ylabel('S(t), g(t)')
end
xlabel('time (t)')
% 2pi frequency == radian frequency
```

```
%vt = fft(v);

%Playing the music
%playblocking(p8);
%%  GABOR TRANSFORM

figure(3)
g=exp(-2*(t-6).^2);
vg=g.*v; vgt=fft(vg);
vgt = vgt(:,1:end-1);

subplot(3,1,1), plot(t,v,'k'), hold on
plot(t,g,'r','Linewidth',[2])
set(gca,'Fontsize',[14])
ylabel('S(t), g(t)'), xlabel('time (t)')

subplot(3,1,2), plot(t,vg,'k')
set(gca,'Fontsize',[14])
ylabel('S(t)g(t)'), xlabel('time (t)')

subplot(3,1,3), plot(ks,abs(fftshift(vgt))/max(abs(vgt)),'k')
axis([-50 50 0 1])
set(gca,'Fontsize',[14])
ylabel('FFT(Sg)'), xlabel('frequency (\omega)')
%% SLIDING GABOR WINDOW Gaussian

figure(4)
vgt_spec=[];
tslide=0:0.1:10
for j=1:length(tslide)
    g=exp(-200*(t-tslide(j)).^2); % Gabor
    vg=g.*v;
    vgt=fft(vg);
    vgt_spec=[vgt_spec;
    abs(fftshift(vgt))];
end
%%
figure(5)
vgt_spec = vgt_spec';
temp = vgt_spec(1:end-1,:);
pcolor(tslide,ks,temp),
shading interp
set(gca,'Ylim',[-2.5*10^4 2.5*10^4],'Fontsize',[14])
colormap(hot)
%title('spectrogram: Guassian filter (5 samples)')
xlabel('Time location (sec)')
ylabel('Frequency location (Hz)')
title('spectrogram: Gaussian wavelet - shorter width')

%%
% Plot the fourier transformed version
val = abs(fftshift(vt));
absval = abs(vt);
```

```
plot(ks,abs(fftshift(vt)) / abs(vt), 'k');
axis([-50 50 0 1])

%% Gabor window------------Step function/Haar wavelemt

figure(2)
width=[10 1 0.2];
for j=1:3
    g=exp(-width(j)*(t-4).^2);
    subplot(3,1,j)
    plot(time,v,'k'), hold on
    plot(time,g,'r','Linewidth',[2])
    set(gca,'Fontsize',[14])
    ylabel('S(t), g(t)')
end
xlabel('time (t)')
% 2pi frequency == radian frequency


%vt = fft(v);

%Playing the music
%playblocking(p8);
%%  GABOR TRANSFORM

figure(3)
g=exp(-2*(t-6).^2);
vg=g.*v; vgt=fft(vg);
vgt = vgt(:,1:end-1);

subplot(3,1,1), plot(t,v,'k'), hold on
plot(t,g,'r','Linewidth',[2])
set(gca,'Fontsize',[14])
ylabel('S(t), g(t)'), xlabel('time (t)')

subplot(3,1,2), plot(t,vg,'k')
set(gca,'Fontsize',[14])
ylabel('S(t)g(t)'), xlabel('time (t)')

subplot(3,1,3), plot(ks,abs(fftshift(vgt))/max(abs(vgt)),'k')
axis([-50 50 0 1])
set(gca,'Fontsize',[14])
ylabel('FFT(Sg)'), xlabel('frequency (\omega)')
%% SLIDING GABOR WINDOW Gaussian

figure(4)
vgt_spec=[];
tslide=0:0.1:10
for j=1:length(tslide)
    g=exp(-200*(t-tslide(j)).^2);
    vg=g.*v;
    vgt=fft(vg);
    vgt_spec=[vgt_spec;
    abs(fftshift(vgt))];
```

```
end
%%
figure(5)
vgt_spec = vgt_spec';
temp = vgt_spec(1:end-1,:);
pcolor(tslide,ks,temp),
shading interp
set(gca,'Ylim',[-2.5*10^4 2.5*10^4],'Fontsize',[14])
colormap(hot)
title('spectrogram: Haar wavelet (width:1)')
xlabel('Time location (sec)')
ylabel('Frequency location (Hz)')
%% Gabor window ----------------Mexican Hat Wavelet

figure(2)
width=[1.4 0.8 0.2];
for j=1:3
    g=(1-((t-4)./width(j)).^2).* exp(-(((t-4)./width(j)).^2)/2);
    subplot(3,1,j)
    plot(time,v,'k'), hold on
    plot(time,g,'r','Linewidth',[2])
    set(gca,'Fontsize',[14])
    ylabel('S(t), g(t)')
    title(['Bandwidth of ', num2str(width(j))])
end
xlabel('time (t)')

% 2pi frequency == radian frequency

%Playing the music
%playblocking(p8);
%%  GABOR TRANSFORM ------------Mexican Hat wavelet

figure(3)
g=(1-((t-4)./0.2).^2).* exp(-(((t-4)./0.2).^2)/2);
vg=g.*v;
vgt=fft(vg);
vgt = vgt(:,1:end-1);

subplot(3,1,1), plot(t,v,'k'), hold on
plot(t,g,'r','Linewidth',[2])
set(gca,'Fontsize',[11])
ylabel('Frequency (Hz)'), xlabel('time (sec)')
legend('Music signal', 'Wavelet')
title('Time-frequency space')

subplot(3,1,2), plot(t,vg,'k')
set(gca,'Fontsize',[11])
ylabel('Frequency (Hz)'), xlabel('time (sec)')
legend('Music signal')
title('Time-frequency space: mutiplied wavelet')

subplot(3,1,3), plot(ks,abs(fftshift(vgt))/max(abs(vgt)),'k')
axis([-50 50 0 1])
```

```
set(gca,'Fontsize',[11])
ylabel('FFT'), xlabel('frequency (\omega)')
legend('Music signal')
title('Fourier space')
%% SLIDING GABOR WINDOW Gaussian

figure(4)
vgt_spec=[];
tslide=0:0.1:10
bandwidth = 200;
for j=1:length(tslide)
    % mexican hat
    g=(1-((t-tslide(j)).*bandwidth).^2).* exp(-(((t-tslide(j)).*bandwidth).^2)/2);
    vg=g.*v;
    vgt=fft(vg);
    vgt_spec=[vgt_spec;
    abs(fftshift(vgt))];
    subplot(3,1,1), plot(t,v,'k',t,g,'r')
    subplot(3,1,2), plot(t,vg,'k')
    temp = abs(fftshift(vgt));
    temp = temp(:,1:end-1);
    subplot(3,1,3), plot(ks,temp/max(abs(vgt)))
    axis([-5 5 0 1])
    drawnow
    pause(0.1)
end
%%
figure(5)
vgt_spec = vgt_spec';
temp = vgt_spec(1:end-1,:);
pcolor(tslide,ks,temp),
shading interp
set(gca,'Ylim',[-2.5*10^4 2.5*10^4],'Fontsize',[14])
colormap(hot)
title(['spectrogram: Mexican hat wavelet filter'])
xlabel('Time location (sec)')
ylabel('Frequency location (Hz)')

%% Gabor window ----------------Step function

figure(2)
width=[1.4 0.8 0.2];
for j=1:3
    g=width(j) * abs(t-4) <= 0.5;
    subplot(3,1,j)
    plot(time,v,'k'), hold on
    plot(time,g,'r','Linewidth',[2])
    set(gca,'Fontsize',[14])
    ylabel('S(t), g(t)')
    title(['Bandwidth of ', num2str(width(j))])
end
xlabel('time (t)')

% 2pi frequency == radian frequency
```

```
%Playing the music
%playblocking(p8);
%%  GABOR TRANSFORM ------------Mexican Hat wavelet

figure(3)
g=1 * abs(t-4) <= 0.5;
vg=g.*v;
vgt=fft(vg);
vgt = vgt(:,1:end-1);

subplot(3,1,1), plot(t,v,'k'), hold on
plot(t,g,'r','Linewidth',[2])
set(gca,'Fontsize',[11])
ylabel('Frequency (Hz)'), xlabel('time (sec)')
legend('Music signal', 'Wavelet')
title('Time-frequency space')

subplot(3,1,2), plot(t,vg,'k')
set(gca,'Fontsize',[11])
ylabel('Frequency (Hz)'), xlabel('time (sec)')
legend('Music signal')
title('Time-frequency space: mutiplied wavelet')

subplot(3,1,3), plot(ks,abs(fftshift(vgt))/max(abs(vgt)),'k')
axis([-50 50 0 0.2])
set(gca,'Fontsize',[11])
ylabel('FFT'), xlabel('frequency (\omega)')
legend('Music signal')
title('Fourier space')
%% SLIDING GABOR WINDOW Gaussian

figure(4)
vgt_spec=[];
tslide=0:0.1:10
bandwidth = 200;
for j=1:length(tslide)
    % mexican hat
    g=bandwidth * abs(t-tslide(j)) <= 0.5;
    vg=g.*v;
    vgt=fft(vg);
    vgt_spec=[vgt_spec;
    abs(fftshift(vgt))];

end
%%
figure(5)
vgt_spec = vgt_spec';
temp = vgt_spec(1:end/2-1,:);
pcolor(tslide,ks,temp),
shading interp
set(gca,'Ylim',[-2.5*10^4 2.5*10^4],'Fontsize',[14])
colormap(hot)
title(['spectrogram: step-function wavelet filter'])
```

```
xlabel('Time location (sec)')
ylabel('Frequency location (Hz)')
```

The following code is for both piano and recorder

```
clc; close all; clear all;
tr_piano=16; % record time in seconds
[y,Fs] = audioread('music1.wav');
Fs=length(y)/tr_piano;
plot((1:length(y))/Fs,y);
xlabel('Time [sec]');
ylabel('Amplitude');
title('Mary had a little lamb (piano)'); drawnow
%p8 = audioplayer(y,Fs); playblocking(p8);

figure(2)

v = y'/2;
time = 1: length(v);
L = time(end)/Fs; % seconds
n = time(end);
k = (2 * pi / L) * [0 : n/2-1 -n/2:-1];
ks = fftshift(k);
ks = ks(:,1:end-1);
t2 = linspace(0,L,n+1);
t = t2(1:n);
vt=fft(v);
plot(abs(fftshift(vt))); % plotting in frequency domain
xlabel('time (seconds)')
ylabel('FFT')
title('Fourier domain: Mary had a little lamb (piano)')

%%
figure(4)
vgt_spec=[];
tslide=0:0.1:L;
for j=1:length(tslide)
    g=200*exp(-5*(t-tslide(j)).^2); % Gabor
    vg=g.*v;
    vgt=fft(vg);
    vgt_spec=[vgt_spec;
    abs(fftshift(vgt))];
end
%%
figure(5)
ks = ks(:,1:end/2-1);
%vgt_spec = vgt_spec';
temp = vgt_spec(1:end,1:end-1);
temp2 = log(temp);

%%
subplot(3, 1, 1);
temp = vgt_spec(:,1:end-1);
%%
pcolor(tslide, ks, temp.');
```

```
caxis([0 800])
shading interp
colormap(hot);
set(gca,'Ylim',[-1000, 0]);
title('With Overtones','fontsize',11, 'fontweight', 'normal');
%%
%
tau = 0.1;
n = size(vgt_spec, 1);
center = zeros(1, n);
for j = 1:n
    vgt_window = vgt_spec(j, :);
    [fmax, idx] = max(abs(vgt_window));
    center(j) = ks(idx);
end
overtones_removed = zeros(size(vgt_spec));
for j = 1:length(center)
    vgt_window = vgt(j, :);
    filter = exp(-tau*((ks-abs(center(j))).^2)) + exp(-tau*((ks-center(j)).^2));
    overtones_removed(j, :) = filter.*vgt_window;
end
save('c_freqs_p');

subplot(3, 1, 2);
pcolor(tau_vec, ks, overtones_removed.');
shading interp
colormap(hot);
set(gca, 'Ylim', [-1000 1000]);
title('No Overtones', 'fontsize', 11, 'fontweight', 'normal');
ylabel('Frequency (Hz)', 'fontsize', 11);

% zooms in on the spectrogram without overtones to more clearly
% see the music score produced by the piano
subplot(3, 1, 3);
pcolor(tau_vec, ks, overtones_removed.');
shading interp
colormap(hot);
set(gca, 'Ylim', [200 350]);
title('No Overtones (Zoomed)', 'fontsize', 11, 'fontweight', 'normal');
xlabel('Time (sec)', 'fontsize', 11);
```

6 Github page

https://github.com/HyesLee99/Data-analysis/tree/master/Gabor%20transformation