Homework 2: Gabor Transforms

Alyssa Jin

02/07/2020

Abstract

This time-frequency analysis assignment including two parts. In first part, I analysis a portion of Handel's *Messiah - Hallelujah*. I use Gaussian window, Mexican hat wavelet and step-function (Shannon) to find the relationships between different Gabor transforms, window width, and spectrograms. In second part, I use Gabor transform to find the difference between recorder and piano version of *Mary has a little lamb*, and reproduce the music score.

I. Introduction and Overview

This time-frequency analysis assignment including two parts.
In part I, I analysis time-frequency of a portion of Handel's *Messiah - Hallelujah* by different window width of the Gabor transforms, and different Gabor windows - Gaussian window, Mexican hat wavelet, and step-function (Shannon) to produce and compare different spectrograms.
In part II, I analysis time-frequency of two waveform audio file, which are played on piano and recorder. I use Gabor filtering in class to reproduce the music score. I compare the spectrograms of two wave form audio file to find the difference of timbre, which is related to center frequency, between piano and recorder.

II. Theoretical Background

II.i Gabor Transform

The Gabor transform is named after Dennis Gabor, who is a Hungarian physicist/mathematician/electrial engineer (Physics Nobel Prize in 1971 for the discovery of holography in 1947). The Gabor transform was first to propose a formal method for localizing both time and frequency. His method involved a simple modification of the Fourier transform kernel. Thus Gabor introduced the kernel

$$g_{t,\omega}(\tau) = e^{i\omega\tau} g(\tau - t)$$

where the new term to the Fourier kernel $g(\tau - t)$ was introduced with the aim of localizing both time and frequency. The Gabor transform, also known as the *short-time Fourier transform (STFT)* is then defined as the following:

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

where the bar denotes the complex conjugate of the function.

1. The energy is bounded by the Schwarz inequality so that

$$|\overline{f}_g(t,\omega)| \leq ||f||||g||$$

2. The energy in the signal plane around the neighborhood of (t,ω) is calculated from

$$|\overline{f}_g(t,\omega)|^2 = |\int_{-\infty}^{\infty} f(\tau)g(\tau-t)e^{-i\omega\tau}d\tau|^2$$

3. The time-frequency spread around a Gabor window is computed from the variance, or second moment, so that

$$\sigma_t^2 = \int_{-\infty}^{\infty} (\tau-t)^2 \left|g_{t,\omega}(\tau)\right|^2 d\tau = \int_{-\infty}^{\infty} \tau^2 \left|g(\tau)\right|^2 d\tau$$
$$\sigma_\omega^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} (\nu-\omega)^2 \left|\overline{g}_{t,\omega}(\nu)\right|^2 d\nu = \frac{1}{2\pi}\int_{-\infty}^{\infty} \nu^2 \left|\overline{g}(\nu)\right|^2 d\nu$$

   where $\sigma_t\sigma_\omega$ is independent of t and ω and is governed by the Heinsenberg uncertainty principle.

4. The Gabor transform is linear so that

$$G[af_1 + bf_2] = aG[f_1] + bG[f_2]$$

5. The Gabor transform can be inverted with the formula

$$f(\tau) = \frac{1}{2\pi}\frac{1}{||g||^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \overline{f}_g(t,\omega)\, g(\tau-t)\, e^{i\omega\tau} d\omega dt$$

   where the integration must occur over all frequency and time-shifting components.

II.iii Gaussian window

The equation of Gaussian window is $g(t) = e^{-a(t-b)^2}$. The Gaussian filtering has a width parameter a and translation parameter b.

II.iv Mexican Hat wavelet

The Mexican Hat wavelet is essentially a second moment of a Gaussian in the frequency domain. The definition of this wavelet and its transform are as follows:

$$\psi(t) = \left(1-t^2\right)e^{-t^2/2} = -d^2/dt^2\left(e^{-t^2/2}\right) = \psi_{1,0}$$
$$\psi(\omega) = \psi_{1,0}(\omega) = \sqrt{2\pi}\omega^2/2$$

The Mexican hat wavelet has excellent localization properties in both time and frequency due to the minimal time-bandwidth product of the Gaussian function.

II.v step-function (Shannon)

In mathematics, a function on the real numbers is called a step function (or staircase function)

if it can be written as a finite linear combination of indicator functions of intervals. Informally speaking, a step function is a piecewise constant function having only finitely many pieces. The function is:

$$g(t) = \begin{cases} 1, x \in A \\ 0, x \notin A \end{cases}$$

III. Algorithm Implementation and Development

III.i Part I Reference to code in book from page 352 to page 357.

(1) **Load file and read basic terms**
I load the data and find the size of the data n is 73113, so I set n to be $73113 - 1 = 73112$, because n need to be an even number. I find t by $linspace.L$ is the length of the interval, which is 8.9247. I rescale the frequency domain by multiplying $\frac{2\pi}{L}$, because I use $fftshift$ in code, so the periodic should be 2π. I find the first and last entries are both zero.

(2) **Explore the window width of the Gabor transform**
Define width and number of steps to find the time-slide vector - $tslide$ in Gabor windows. Use for loop to define functions of Gaussian function, Mexican hat function, and step-function. Plot time signal v and Gabor time filters for Gaussian function, Mexican hat function, and step-function g. The product vg is depicted in the middle panel. I calculate $fftshift$ value and the absolute value of the $fftshift$ for the signal in small window for a specific domain, and plot in the bottom panel.

(3) **Plot the spectrogram**
I use the original domain ks, $pcolor$, $shadinginterp$, and $colormap('hot')$ to plot the spectrogram.

III.ii Part II The analysis steps of piano and recorder are the same, except for elements in data.

(1) **Load file and read basic terms**
I load two wave audio file, and import the data. The size of data n for piano is 701440; the size of data n for recorder is 627712. Because both of the n are even number, so I can keep the n. The length of piano music L is 15.9057; the length of recorder music is 14.2338. I can use n and L to calculate t, k and ks the same as part I.

(2) **Explore the window width of the Gabor transform**
I set $width$ and $numstep$, using $linspace$ to calculate $tslide$ which is from 0 to L. Use for loop to define Gaussian Gabor window. Plot time signal v and Gabor time filters$g$. The product vg is depicted in the middle panel. I calculate $fftshift$ value and the absolute value of the $fftshift$ for the signal in small window for a specific domain, and plot in the bottom panel.

(3) **Plot the spectrogram**
I use the domain $ks/(2 * \pi)$, to have a better look of the spectrograms. I use *pcolor*, *shadinginterp*, and *colormap*('*hot*') to plot the spectrogram.

IV. Computational Results

VI.i Part I

Figure 1 is the original signal and its FFT. Figure 2-5 shows time signal, Gabor filter, and Gabor transform of different filters and width. Figure 2 and figure 3 are Gaussian filter with width 2 and 100. Figure 4 is Mexican hat wavelet with width 2. Figure 5 is step-function (Shannon) with width 2. Comparing figure 2 and 3, I find as the width becomes larger, the wave becomes more narrow. Comparing figure 2 - 4, I find with the same width, the wave of step-function is the narrowest, the wave of Mexican hat wavelet is the widest.

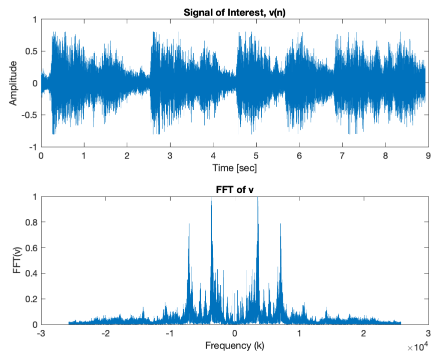
Figures 6-10 show different spectrograms. Figures 6 and 7 show the spectrograms of Gaussian filters of long and short length. In Figure 6 where the frequency are like straight lines continuously without stop. In Figure 7 the frequency arise and suddenly stop, and then arise again. I think this is because as the width becomes larger, the frequency of entering and exiting becomes larger. Figures 8 and 9 show the spectrograms produced by the Mexican Hat wavelet, and step-function (Shannon) of the same size with figure 6. Figure 8, which is the Mexican hat wavelet filter is the clearest among figure 6, 8, and 9. Figure 9 is the worst among figure 6, 8, and 9. The lines in figure 8 looks pretty smooth, but there are lots of irregularity in figure 9. In figure 10, I use Gaussian filter, with the same width as figure 6 and different step numbers 20. The frequency of figure 10 is worse than that in figure 6, so The larger the step numbers, the spectrogram looks clearer and better.

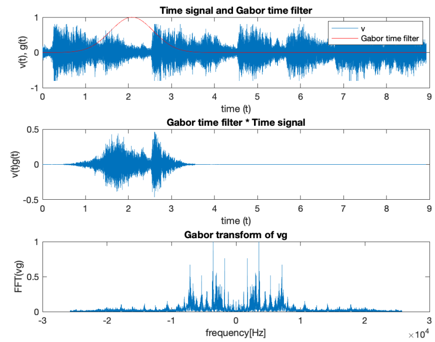
VI.ii Part II

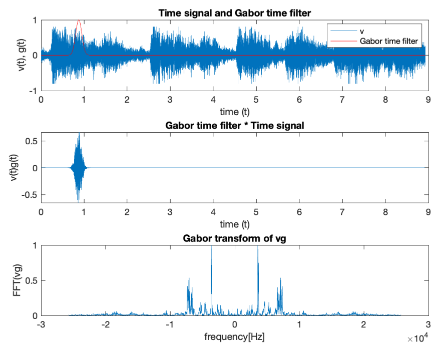
Figures 11 and 12 show the original signal of piano (wav1) and recorder(wav2). Figure 13 and 14 shows the signal of piano (wav1) and recorder(wav2) by Gabor filters, both with width 50 and step numbers 200.
Figures 15 and 16 are spectrograms of signal of piano (wav1) and recorder(wav2). Figure 17 and 18 are zoomed spectrograms of signal of piano (wav1) and recorder(wav2). In figure 15 and 16, it's really difficult to tell the frequency of spectrograms, so to find the music score, I need to look at figure 17 and 18. For piano, the sequences are 2000Hz, 1800Hz, 1600Hz, 18900Hz, 2000Hz, 2000Hz, 2000Hz, 1800Hz, 1800Hz, 1800Hz, 2000Hz, 2000Hz, 2000Hz, 2000Hz, 1800Hz, 1600Hz, 1800Hz, 2000Hz, 2000Hz, 2000Hz, 2000Hz, 1800Hz, 1800Hz, 2000Hz, 1800Hz, 1600Hz.

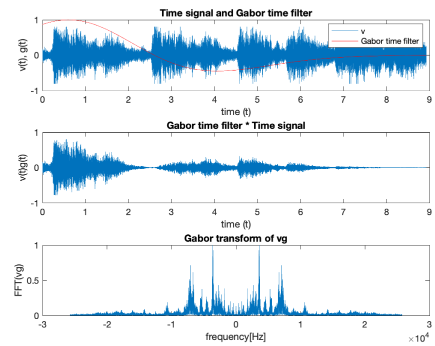
V. Summary and Conclusions

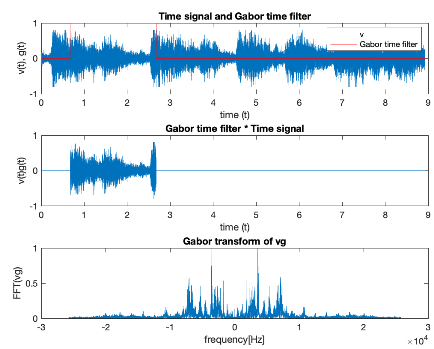
In part I, I find the difference of spectograms affected width, step numbers, and threee different types of Gabor Filter.
In part II, I use Gabor transforms to compare the rewrite the music score, and compare the difference between piano and recorder.

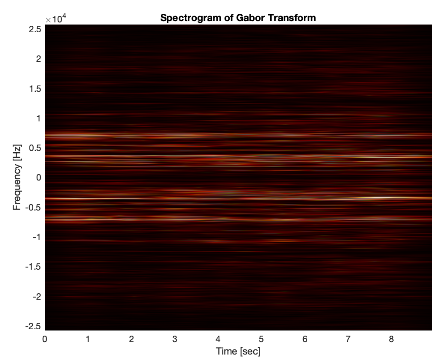

Signal of Interest, v(n)
Amplitude
Time [sec]
FFT of v
FFT(v)
Frequency (k)
×10^4

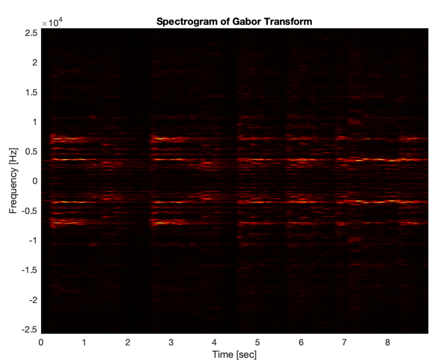

Time signal and Gabor time filter
v
Gabor time filter
v(t), g(t)
time (t)
Gabor time filter * Time signal
v(t)g(t)
time (t)
Gabor transform of vg
FFT(vg)
frequency[Hz]
×10^4

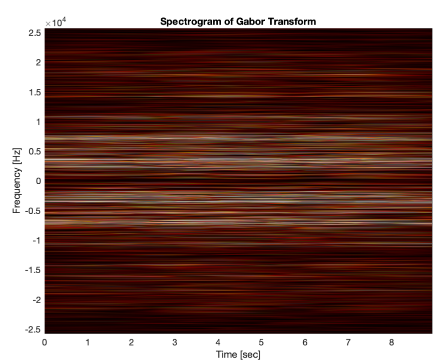

Time signal and Gabor time filter
v
Gabor time filter
v(t), g(t)
time (t)
Gabor time filter * Time signal
v(t)g(t)
time (t)
Gabor transform of vg
FFT(vg)
frequency[Hz]
×10^4

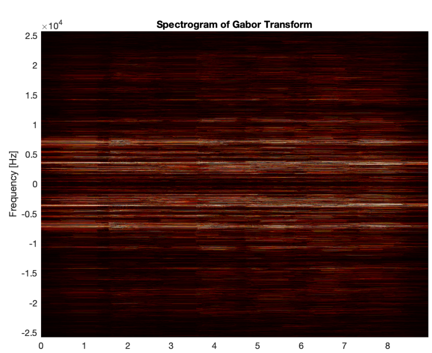

Time signal and Gabor time filter
v
Gabor time filter
v(t), g(t)
time (t)
Gabor time filter * Time signal
v(t)g(t)
time (t)
Gabor transform of vg
FFT(vg)
frequency[Hz]
×10^4

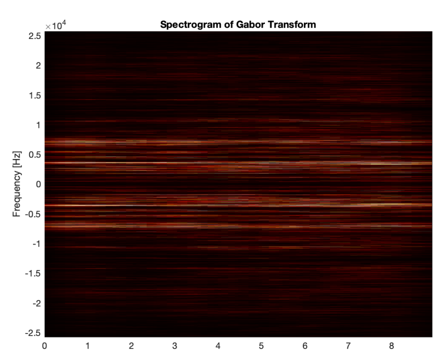

Time signal and Gabor time filter
v
Gabor time filter
v(t), g(t)
time (t)
Gabor time filter * Time signal
v(t)g(t)
time (t)
Gabor transform of vg
FFT(vg)
frequency[Hz]
×10^4

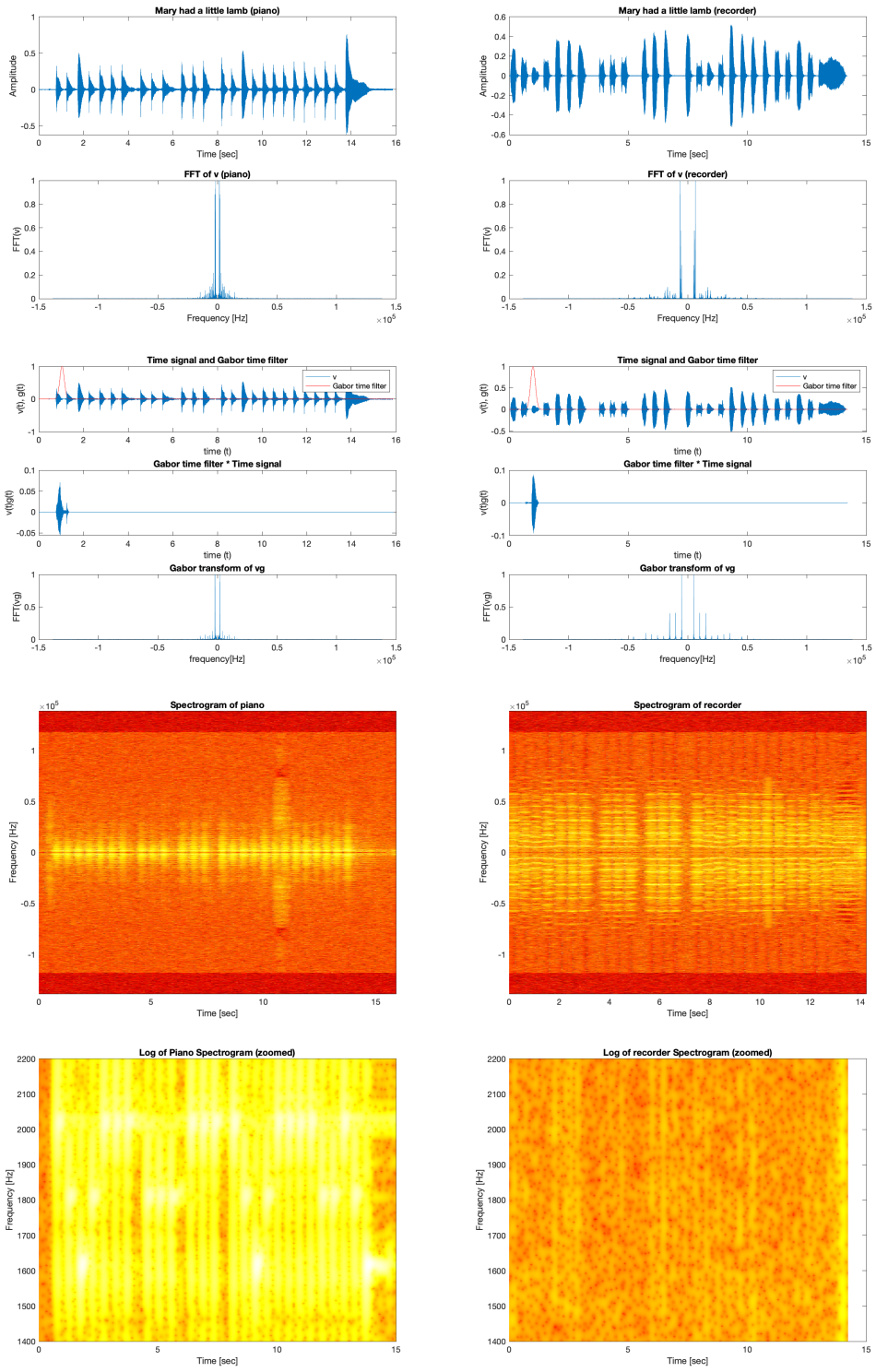

Spectrogram of Gabor Transform
×10^4
Frequency [Hz]
Time [sec]


Spectrogram of Gabor Transform
×10^4
Frequency [Hz]
Time [sec]


Spectrogram of Gabor Transform
×10^4
Frequency [Hz]
Time [sec]


Spectrogram of Gabor Transform
×10^4
Frequency [Hz]
Time [sec]


Spectrogram of Gabor Transform
×10^4
Frequency [Hz]
Time [sec]



Appendix B. MATLAB codes

part I:

```
clc;
clear all;
close all;

%load audio file
load handel;

v = y';
% Play back audio
%p8 = audioplayer(v,Fs);
%playblocking(p8);

% Read the basic terms, calculate time signal, and plot in the upper panel
% of figure 1
v = v(1:end-1);
n = length(v);
L = (n-1) / Fs;
t1=linspace(0,L,n+1); t=t1(1:n);
k=(2*pi/L)*[0:n/2-1 -n/2:-1];
ks=fftshift(k);
figure(1);
subplot(2,1,1)
plot(t,v);
xlabel('Time [sec]');
ylabel('Amplitude');
title('Signal of Interest, v(n)');

% Calculate Fourier transform for the time-frequency, and plot in the lower
% panel of figure 1
vt = fft(v);
subplot(2,1,2);
plot(ks, abs(fftshift(vt)) / max(abs(vt)));
xlabel('Frequency (k)');
ylabel('FFT(v)');
title('FFT of v');
drawnow;
print(gcf,'-dpng','Figure1.png');

% Set width and number of steps, calculate time slide
width = 2;
numstep = 20;
tslide = linspace(0,t(end-1),numstep);
spec = zeros(length(tslide),n);

% Create Gabor time filter for three different Gaussian filters: 1:
% Gaussian, 2: Mexican hat wevelet, and 3: step-function (Shannon)
```

```
filter = {@(x) exp(-width*(x).^2), @(x)(1-(x/width).^2).*exp(-((x/width).^2)/2), ...
    @(x) (x>-width/2 & x< width/2)};
for j=1:length(tslide)
g = filter{1}(t-tslide(j)); % Gabor filter (choose 1-3 for three different filters)
vg = g.*v; % Apply Gabor filter
vgt = fft(vg); % Take fft of filtered data
spec(j,:) = abs(fftshift(vgt)); % Store fft in spectrogram

% plot Gabor transforms
%figure(5);
%subplot(3,1,1), plot(t,v,t,g,'r'), title('Time signal and Gabor time filter'), ...
    %legend('v','Gabor time filter'), xlabel('time (t)'), ylabel('v(t), g(t)');
%subplot(3,1,2), plot(t,vg), title('Gabor time filter * Time signal'), ...
    %xlabel('time (t)'), ylabel('v(t)g(t)');
%subplot(3,1,3), plot(ks, abs(fftshift(vgt))/max(abs(vgt))), ...
    %title('Gabor transform of vg'), xlabel('frequency[Hz]'), ...
    %ylabel('FFT(vg)');
%drawnow;
%print(gcf,'-dpng','Figure5.png');
end


% Plot spectrogram
figure(10);
pcolor(tslide,ks,spec.'), shading interp
colormap('hot')
xlabel('Time [sec]'), ylabel('Frequency [Hz]');
title('Spectrogram of Gabor Transform');
print(gcf,'-dpng','Figure10.png');

Part II:
clc;
clear all;
close all;

% load piano - music 1
[y,Fs] = audioread('music1.wav');
tr_piano=length(y)/Fs; % record time in seconds
v = y.';

% plot figure 1
figure(11);
subplot(2,1,1);
plot((1:length(v))/Fs,v);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Mary had a little lamb (piano)');
```

```
% play back audio
%p8 = audioplayer(y,Fs);
%playblocking(p8);

% start Gabor Transform
L = tr_piano;
n = length(v);
t1 = linspace(0,L,n+1);
t = t1(1:n);
k = (2*pi/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
vt = fft(v);

% Look at frequency components of entire signal
subplot(2,1,2)
plot(ks, abs(fftshift(vt)) / max(abs(vt)));
xlabel('Frequency [Hz]'), ylabel('FFT(v)'), title('FFT of v (piano)');
print(gcf,'-dpng','Figure11.png');

% Create Gabor filter for spectrogram
figure(13);
vgt_spec=[];
width = 50; % Width of filter
numstep = 200; % Number of time steps to take
tslide = linspace(0,L,numstep); % Time discretization

% Create spectrogram using Gabor filter
for j=1:length(tslide)
g = exp(-width*(t - tslide(j)).^2); % Gabor
vg = g .* v;
vgt = fft(vg);
vgt_spec=[vgt_spec; abs(fftshift(vgt))];

% plot Gabor transforms
%figure(13);
%subplot(3,1,1), plot(t,v,t,g,'r'), title('Time signal and Gabor time filter'), ...
    %legend('v','Gabor time filter'), xlabel('time (t)'), ylabel('v(t), g(t)');
%subplot(3,1,2), plot(t,vg), title('Gabor time filter * Time signal'), ...
    %xlabel('time (t)'), ylabel('v(t)g(t)');
%subplot(3,1,3), plot(ks, abs(fftshift(vgt))/max(abs(vgt))), ...
    %title('Gabor transform of vg'), xlabel('frequency[Hz]'), ...
    %ylabel('FFT(vg)');
%drawnow;
%print(gcf,'-dpng','Figure13.png');
end
```

```
% Plot relevant portion of spectrogram
figure(15);
pcolor(tslide,ks,log(vgt_spec.')), shading interp;
colormap('hot');
xlabel('Time [sec]');
ylabel('Frequency [Hz]');
title('Spectrogram of piano');
print(gcf,'-dpng','Figure15.png');

% Plot relevant portion of spectrogram
figure(17);
pcolor(tslide,ks,log(vgt_spec.')), shading interp
axis([0 15 1400 2200])
colormap('hot'), xlabel('Time [sec]'), ylabel('Frequency [Hz]'),
title('Log of Piano Spectrogram (zoomed)')
print(gcf,'-dpng','Figure17.png');

clc;
clear all;
close all;

% load piano - music 1
[y,Fs] = audioread('music2.wav');
tr_recorder=length(y)/Fs; % record time in seconds
v = y.';

% plot figure 1
figure(12);
subplot(2,1,1);
plot((1:length(v))/Fs,v);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Mary had a little lamb (recorder)');

% play back audio
% p8 = audioplayer(y,Fs);
% playblocking(p8);

% start Gabor Transform
L = tr_recorder;
n = length(v);
t1 = linspace(0,L,n+1);
t = t1(1:n);
k = (2*pi/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
vt = fft(v);
```

```
% Look at frequency components of entire signal
subplot(2,1,2)
plot(ks, abs(fftshift(vt)) / max(abs(vt)));
xlabel('Frequency [Hz]'), ylabel('FFT(v)'), title('FFT of v (recorder)');
print(gcf,'-dpng','Figure12.png');

% Create Gabor filter for spectrogram
figure(14);
vgt_spec=[];
width = 50; % Width of filter
numstep = 200; % Number of time steps to take
tslide = linspace(0,L,numstep); % Time discretization

% Create spectrogram using Gabor filter
for j=1:length(tslide)
g = exp(-width*(t - tslide(j)).^2); % Gabor
vg = g .* v;
vgt = fft(vg);
vgt_spec=[vgt_spec; abs(fftshift(vgt))];

% plot Gabor transforms
%figure(14);
%subplot(3,1,1), plot(t,v,t,g,'r'), title('Time signal and Gabor time filter'), ...
    %legend('v','Gabor time filter'), xlabel('time (t)'), ylabel('v(t), g(t)');
%subplot(3,1,2), plot(t,vg), title('Gabor time filter * Time signal'), ...
    %xlabel('time (t)'), ylabel('v(t)g(t)');
%subplot(3,1,3), plot(ks, abs(fftshift(vgt))/max(abs(vgt))), ...
    %title('Gabor transform of vg'), xlabel('frequency[Hz]'), ...
    %ylabel('FFT(vg)');
%drawnow;
%print(gcf,'-dpng','Figure14.png');
end

% Plot relevant portion of spectrogram
figure(16);
pcolor(tslide,ks,log(vgt_spec.')), shading interp;
colormap('hot');
xlabel('Time [sec]');
ylabel('Frequency [Hz]');
title('Spectrogram of recorder');
print(gcf,'-dpng','Figure16.png');

% Plot relevant portion of spectrogram
figure(18);
pcolor(tslide,ks,log(vgt_spec.')), shading interp
```

```
axis([0 15 1400 2200])
colormap('hot'), xlabel('Time [sec]'), ylabel('Frequency [Hz]'),
title('Log of recorder Spectrogram (zoomed)')
print(gcf,'-dpng','Figure18.png');
```