AMATH482 HW02

Joshua Borgman

University of Washington

Josh Borgman

Tuesday, February 10, 2015

Contents

1. Introduction and Overview

The restrictive relation between obtaining information in the frequency and time domains is a readily apparent problem with unmodified Fourier Transformation of time-variant signals. Throughout this assignment we will aspire to utilize a method of transformation that offers a combination of temporal and frequency information, and observe the benefits and restrictions of this transformation.

2. Theoretical Background

Gabor Transform

The Gabor Transform is an implementation of a windowed Fourier Transform that allows for localization of a signal in both time and frequency domains. The transform is based on a simple modification of the Fourier transform kernel, to yield the Gabor transform kernel:

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

And the associated transform defined as

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

i. There are several properties of the Gabor Transform that are useful including:
   The energy is bounded by the Schwarz inequality
ii. The Gabor transform is linear.
iii. The Gabor transform can be inverted.

In application, the Gabor transforms computed by discretizing the time and frequency domains.

Drawbacks of the Gabor (STFT) arise from the trade off between resolution in the time and frequency domains. In other words, one must trade resolution in one domain to increase resolution in another. The Heisenberg relationship must hold, such that a shorter time-filtering window yields less information of the frequency domain and vice versa.

Spectrograms

Spectrograms allow for visual displays that have both time and frequency resolution. Below is a visual representation of this combination.

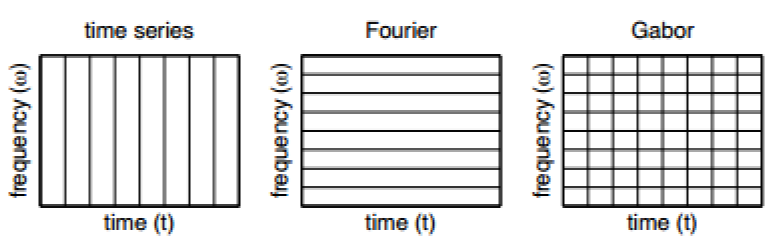


Figure 1: Diagram displaying combination on time and frequency domains

3. Algorithm Implementation and Development

Gabor Transform:

The Gabor Transform is carried out by discretizing the continuous model onto a lattice of time and frequency with sample points given by $\nu = m\omega_0$ and $\tau = nt_0$, respectively. Using these parameters the discrete version of $g_{t,\omega}$ is

$$g_{m,n} = e^{(i2\pi m\omega_0 t)}g(t - nt_0)$$

And transform

$$\tilde{f}(m,n) = \int_{-\infty}^{\infty} f(t)\bar{g}_{m,n}dt$$

If $0 < t_0, \omega_0 < 1$, then the signal is oversampled and the time frames yield excellent localization of the signal in both time and frequency. If $t_0, \omega_0 > 1$, the signal is under sampled and the lattice is incapable of reproducing the signal.

Spectrograms

Spectrograms were generated by concatenating the absolute value of the shifted frequency content matrices of each subsequent windowing during the Gabor Transform. This matrix was then ploted using the pcolor command with the additional inputs of the steps (t_slide) and the vector of shifted wave numbers.

4. Computational Results

The images below contain visualizations of the computational results

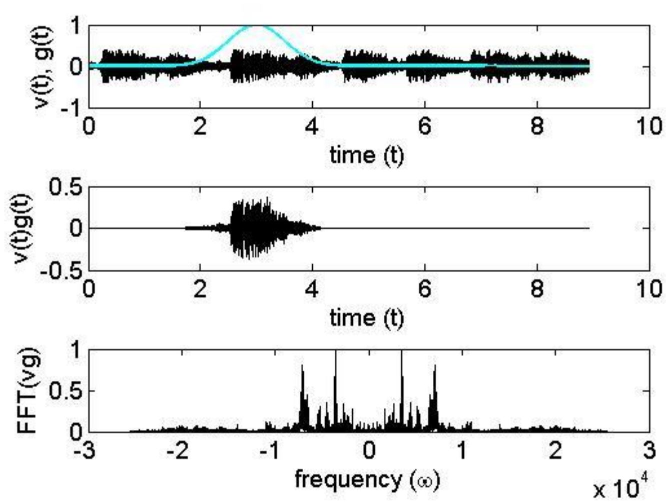


Figure 2: Initial exemplification of Gabor Filter and use seen in both domains.

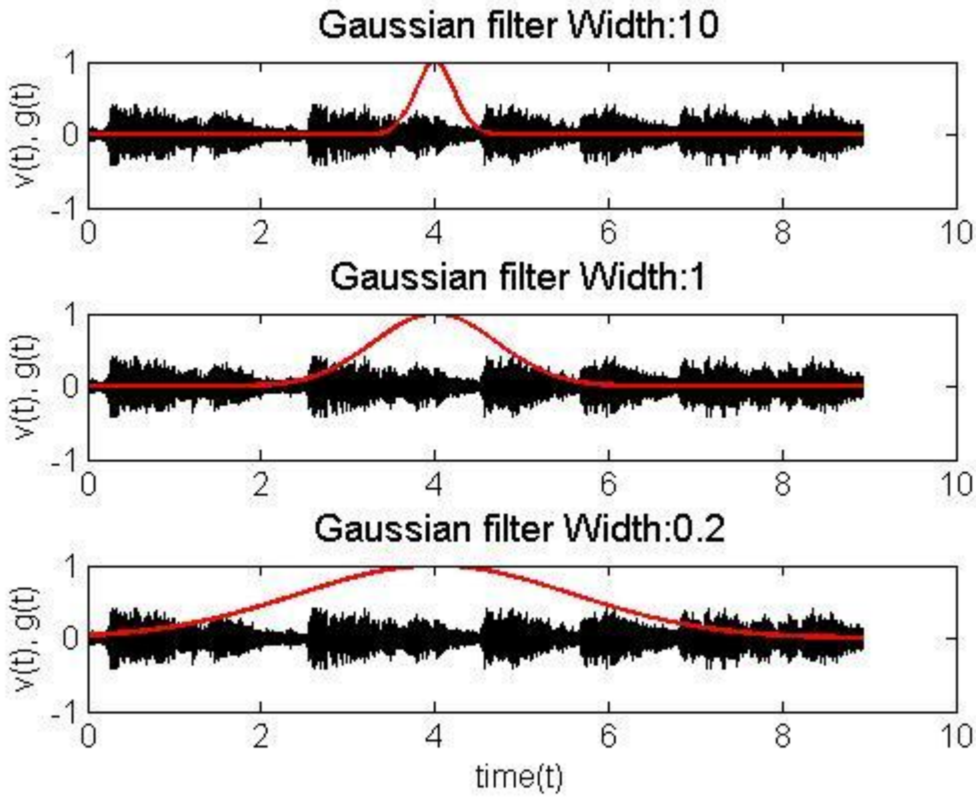


Figure 3: Result of changing width parameter.

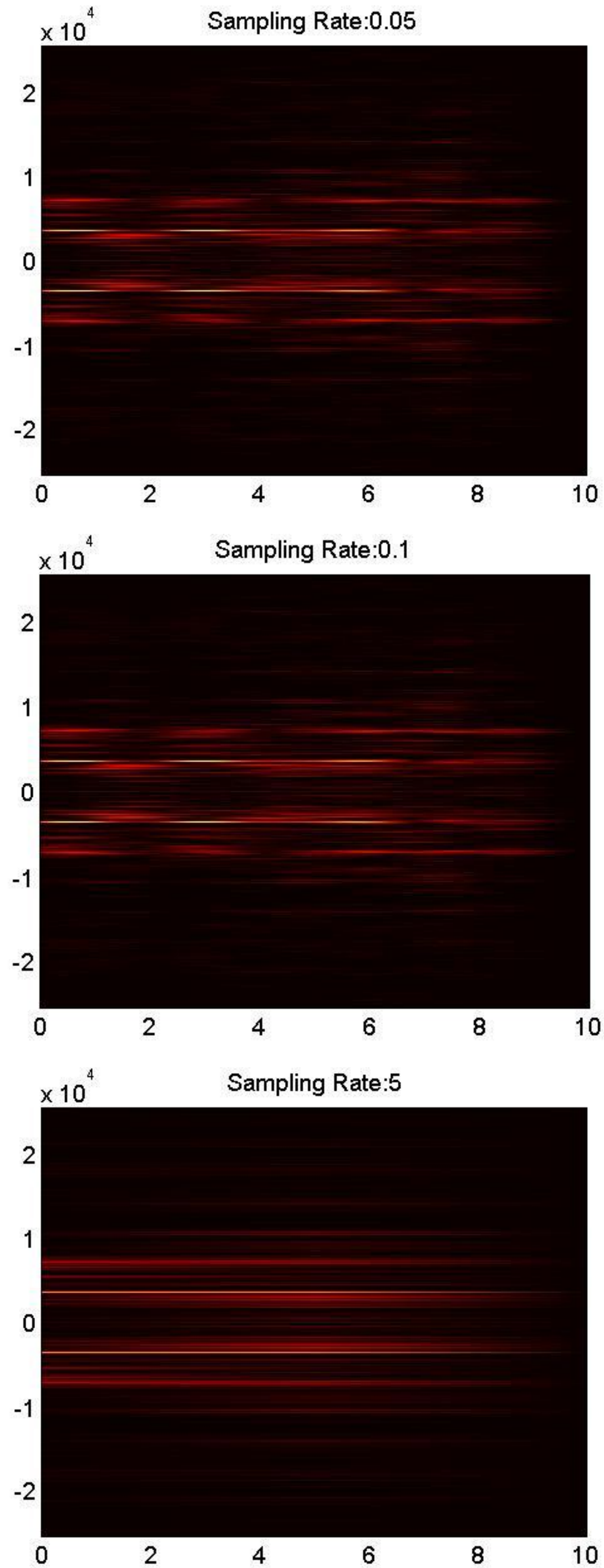


Figure 4: The effect of variation in sampling speeds

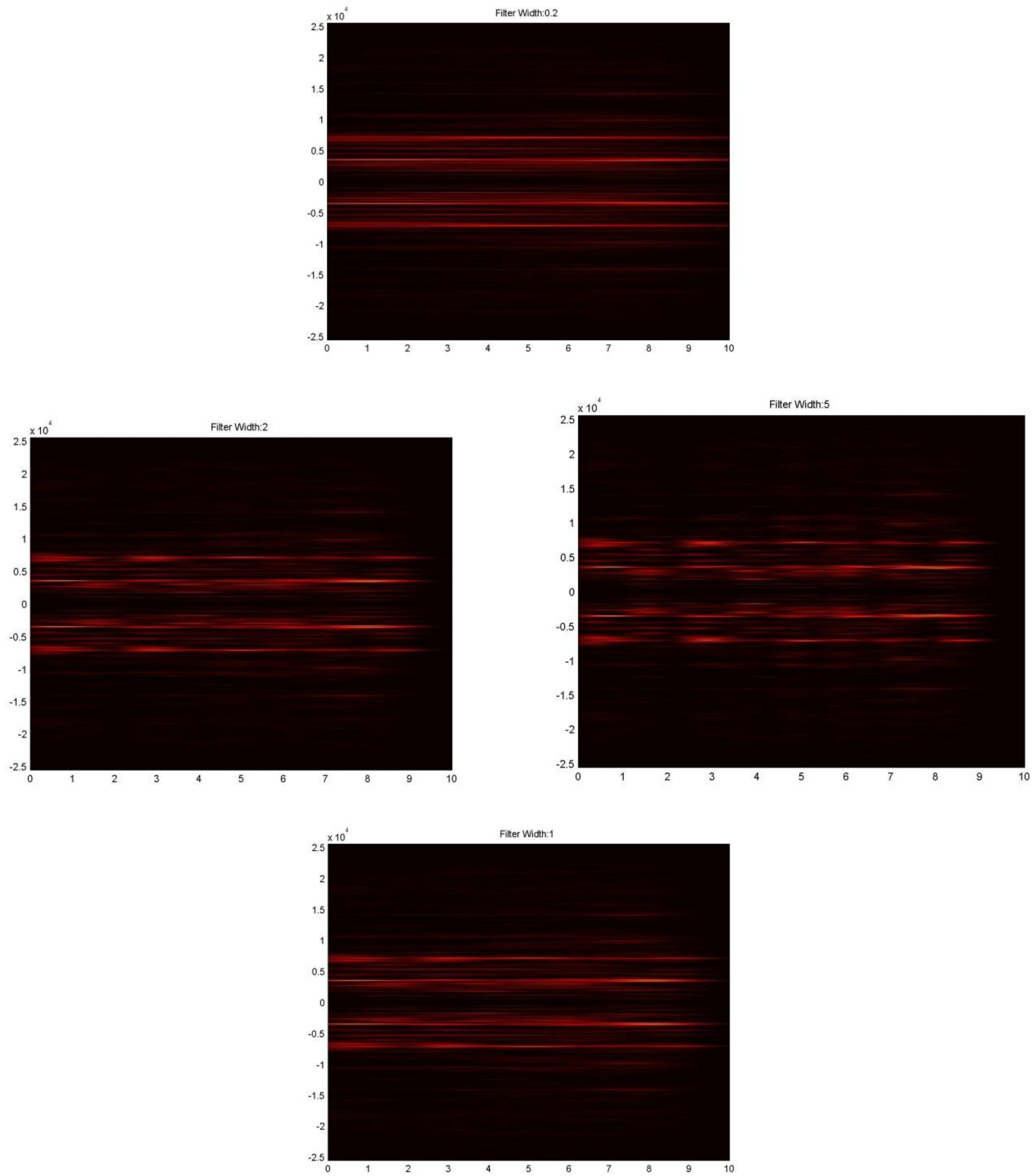
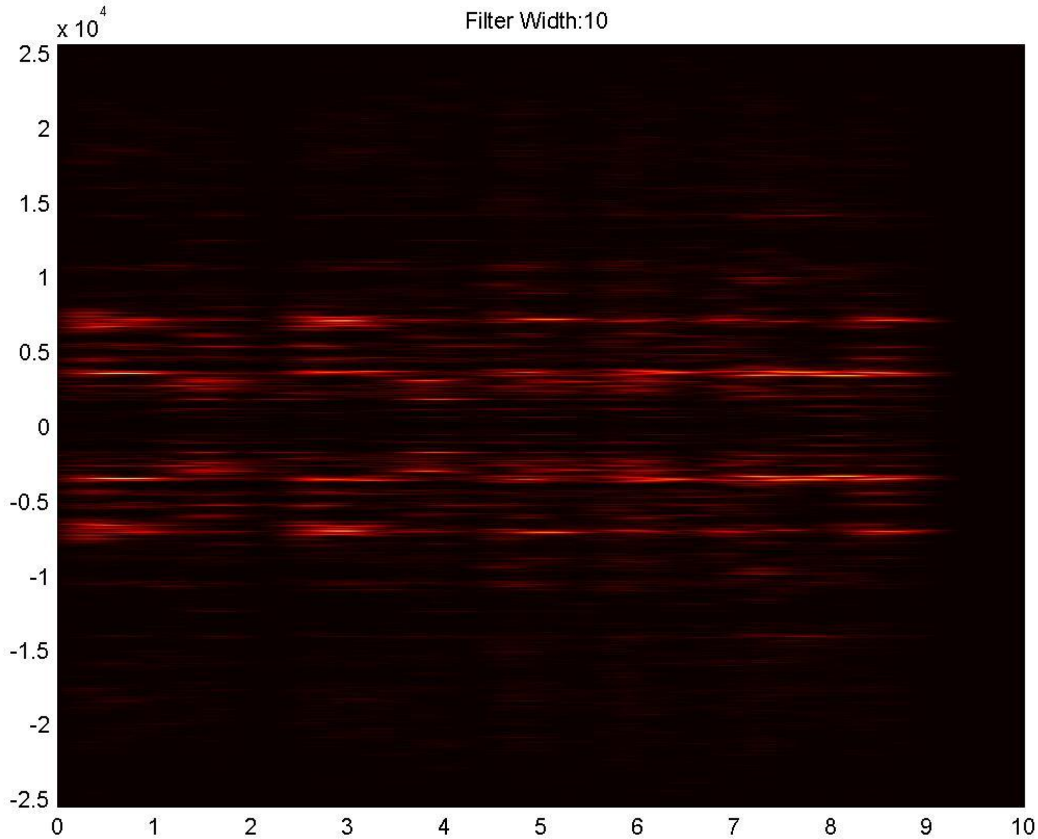


Figure 5: The results of varied filter widths as seen through resultant spectrograms.



5. Summary and Conclusions

The Gabor Transform offers us a unique and widely applicable method by which to transform data to give information in multiple dimensions. This attribute is widely applicable and highly beneficial in areas such as time-frequency analysis and other closely related measurements.

Appendix A:

Appendix B:

```
%Joshua Borgman    AMATH482    Homework2:Gabor Transforms

%Part I

clear all, close all, clc
load handel
v = y'/2;
vt = fft(v);

n = length(v); L = 9;
t =(1:length(v))/Fs;
k = (2*pi/L)*[0:n/2 -n/2:-1]; ks = fftshift(k);

figure(1)
subplot(2,1,1) %Time Domain
plot(t, v)
set(gca, 'Fontsize', [12])
xlabel('Time [sec]'), ylabel('Amplitude');
title('Handels Messiah');

subplot(2,1,2)
plot(ks, abs(fftshift(vt))/max(abs(vt)),'k');
set(gca, 'Fontsize', [12])
xlabel('frequency (\omega)'), ylabel('FFT(v)')

%%
%plays audio file
p8 = audioplayer(v,Fs);
playblocking(p8);

%%
%This secion generates Gaussian filters

figure(2)
width = [10 1 0.2];
for j=1:3
    g = exp(-width(j)*(t-4).^2);
    subplot(3,1,j)
    plot(t,v,'k'), hold on
    plot(t, g, 'r', 'Linewidth', [2])
    set(gca, 'Fontsize', [12])
    ylabel('v(t), g(t)')
    title(strcat('Gaussian filter Width: ', num2str(width(j))), 'Fontsize',
[14]);
end
xlabel('time(t)');
```

```
%%
% Creates Gaussian filter with a=2, b=3, shows overlayed filter and such
close(3)
figure(3)
center = 3;
scale = 2;
g=exp(-scale*(t-center).^2);
vg=g.*v;
vgt=fft(vg);

subplot(3,1,1), plot(t,v,'k'), hold on
plot(t,g,'c','Linewidth',[2])
set(gca,'Fontsize',[14])
ylabel('v(t), g(t)'), xlabel('time (t)')
subplot(3,1,2), plot(t,vg,'k')
set(gca,'Fontsize',[14])
ylabel('v(t)g(t)'), xlabel('time (t)')
subplot(3,1,3), plot(ks,abs(fftshift(vgt))/max(abs(vgt)),'k')
axis([-1e4 1e4 0 1])
set(gca,'Fontsize',[14])
ylabel('FFT(vg)'), xlabel('frequency (\omega)')

%%
%Generates movie that scans audio file

figure(4)
vgt_spec=[];
tslide=0:0.1:10;
for j=1:length(tslide)
    g=exp(-2*(t-tslide(j)).^2); % Gabor
    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')
    subplot(3,1,3), plot(ks,abs(fftshift(vgt))/max(abs(vgt)))
    %axis([-50 50 0 1])
    drawnow
    pause(0.1)
end

%%
%Generates Spectogram of Piece
figure(5)
pcolor(tslide,ks,vgt_spec.'), shading interp
%set(gca,'Ylim',[-50 50],'Fontsize',[14])
colormap(hot)

%%
%Generates Spectograms for various filters

close all
width = [0.2, 1, 2, 5, 10];
for jj=1:length(width)
    figure()
```

```
    vgt_spec=[];
    tslide=0:0.1:10;
    for j=1:length(tslide)
        g=exp(-width(jj)*(t-tslide(j)).^2); % Gabor
        vg=g.*v; vgt=fft(vg);
        vgt_spec=[vgt_spec; abs(fftshift(vgt))];
    end
    pcolor(tslide,ks,vgt_spec.'), shading interp
    colormap(hot)
    title(strcat('Filter Width: ', num2str(width(jj))));
    saveas(jj, strcat('spectogram', num2str(jj), '.jpg'));

end

%%
% See the affect of different sampling rates


samp = [0.05, 0.1, 5];
jj=1;
%for jj=1:length(samp)
    figure()
    vgt_spec=[];
    tslide=0:samp(jj):10;
    for j=1:length(tslide)
        g=exp(-2*(t-tslide(j)).^2); % Gabor
        vg=g.*v; vgt=fft(vg);
        vgt_spec=[vgt_spec; abs(fftshift(vgt))];
    end
    pcolor(tslide,ks,vgt_spec.'), shading interp
    set(gca,'Fontsize',[14])
    colormap(hot)
    title(strcat('Sampling Rate: ', num2str(samp(jj))));
    %saveas(jj, strcat('sampelingRates', num2str(jj), '.jpg'));
%end

%%
%Plays piano recording

tr_piano=16; % record time in seconds
y=wavread('music1'); 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);

%%
% plays recorder recording
figure(2)
tr_rec=14; % record time in seconds
y=wavread('music2'); Fs=length(y)/tr_rec;
plot((1:length(y))/Fs,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Mary had a little lamb (recorder)');
p8 = audioplayer(y,Fs); playblocking(p8);
```

```
%%
% Gabor Filtering of MHALL recorder, noted that there are 24 "notes"

y=y';
ygt_spec=[];
tslide=linspace(0,tr_rec, 25);

n = length(y); L = tr_rec;
t =(1:length(y))/Fs;
k = (2*pi/L)*[0:n/2-1 -n/2:-1]; ks = fftshift(k);
notes = [];
for j=1:length(tslide)
    g=exp(-2*(t-tslide(j)).^2); % Gabor
    yg=g.*y; ygt=fft(yg);
    ygt_spec=[ygt_spec; abs(fftshift(ygt))];
    subplot(3,1,1), plot(t,y,'k',t,g,'r')
    subplot(3,1,2), plot(t,yg,'k')
    subplot(3,1,3), plot(ks,abs(fftshift(ygt))/max(abs(ygt)))
    %axis([-50 50 0 1])
    drawnow
    pause(0.2)
end

%%
% Gabor Filtering of MHALL piano, noted that there are 24 "notes"

y=y';
ygt_spec=[];
tslide=linspace(0,tr_piano, 25);

n = length(y); L = tr_piano;
t =(1:length(y))/Fs;
k = (2*pi/L)*[0:n/2-1 -n/2:-1]; ks = fftshift(k);
notes = [];
for j=1:length(tslide)
    g=exp(-2*(t-tslide(j)).^2); % Gabor
    yg=g.*y; ygt=fft(yg);
    ygt_spec=[ygt_spec; abs(fftshift(ygt))];
    subplot(3,1,1), plot(t,y,'k',t,g,'r')
    subplot(3,1,2), plot(t,yg,'k')
    subplot(3,1,3), plot(ks,abs(fftshift(ygt))/max(abs(ygt)))
    %axis([-50 50 0 1])
    drawnow
    pause(0.2)
end
```