ECE 637 Lab 2 - 2-D Random Process

Yifan Fei

Electrical Engineering
Purdue University

Instructor: Prof. Charles A. Bouman

Spring 2018

Section 1: Power Spectral Density of an Image

1:The gray scale image img04g.tif

Figure 1: The gray scale image img04g.tif

2: The power spectral density plots for 3 block sizes

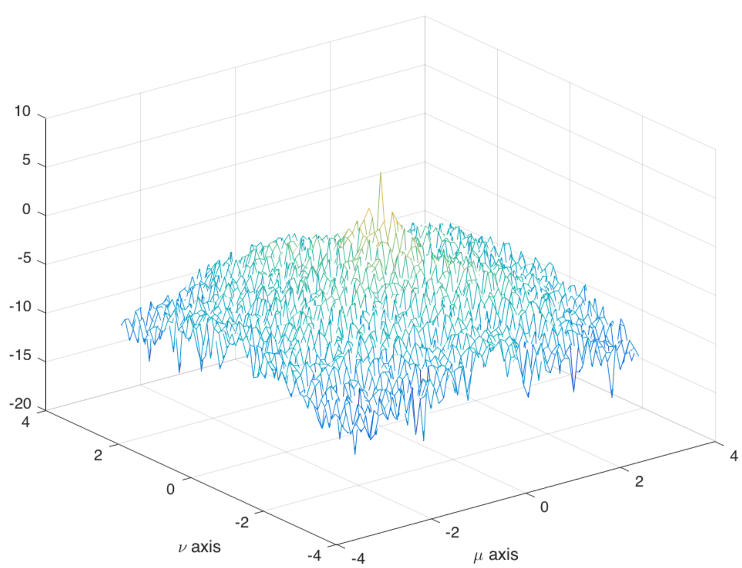

Figure 2: The power spectral density plots for block size of 64*64

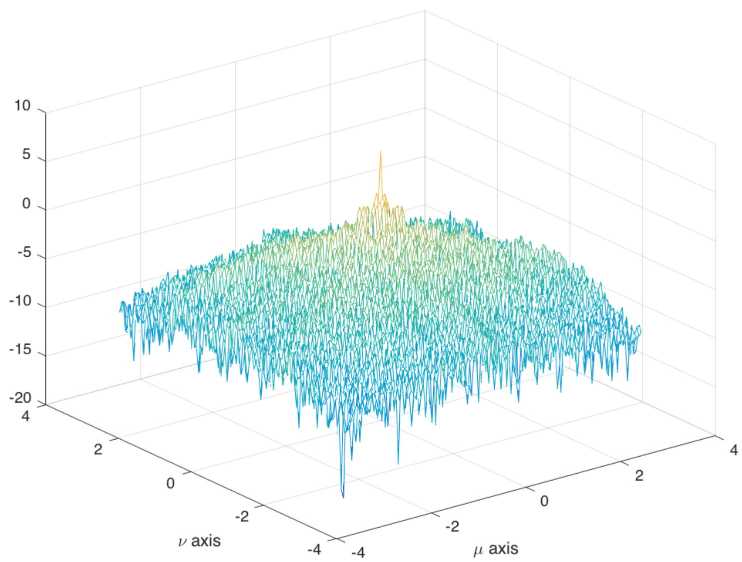

Figure 3: The power spectral density plots for block size of 128*128

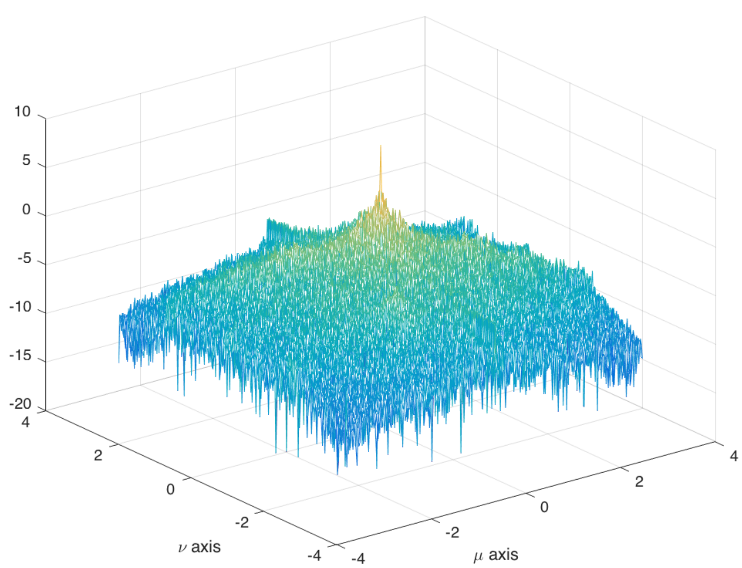

Figure 4: The power spectral density plots for block size of 256*256

Notice the power spectrum estimates remain noisy even when the block size is increased.

3: The improved power spectral density estimate

As is shown, after using 25 non-overlapping image windows and averaging this power spectral density across the 25 windows, the noise is reduced.

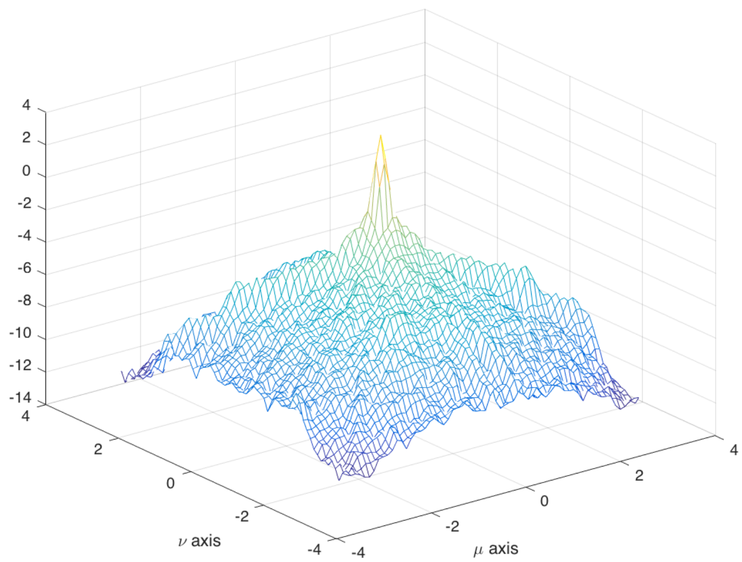

Figure 5: The improved power spectral density estimate

Notice the noise of power spectrum estimates is reduced by windows smoothing.

4: Matlab code for BetterSpecAnal.m

See the attachment

Section2: Power Spectral Density of a 2-D AR Process

0:A derivation of the analytical expression for H(u,v)

Since:

$$H(z_1, z_2) = \frac{Y(z_1, z_2)}{X(z_1, z_2)} = \frac{3}{1 - 0.99z_1^{-1} - 0.99z_2^{-1} + 0.9801z_1^{-1}z_2^{-1}}$$

$$H(e^{ju}, e^{jv}) = \frac{3}{1 - 0.99e^{-ju} - 0.99e^{-jv} + 0.9801e^{-ju}e^{-jv}}$$

$$h(m, n) = \frac{y(m, n)}{x(m, n)} = \frac{3}{0.99\delta(m-1) + 0.99\delta(n-1) - 0.9801\delta(m-1)\delta(n-1)}$$

$$y(m, n) = 3x(m, n) + 0.99y(m, n)[\delta(m-1) + \delta(n-1) - 0.9801\delta(m-1)\delta(n-1)]$$

$$Y(z_1, z_2) = 3X(z_1, z_2) + 0.99z_1^{-1}Y(z_1, z_2) + 0.99z_2^{-1}Y(z_1, z_2) - 0.9801z_1^{-1}z_2^{-1}Y(z_1, z_2)$$

1:The image 255(x + 0.5)

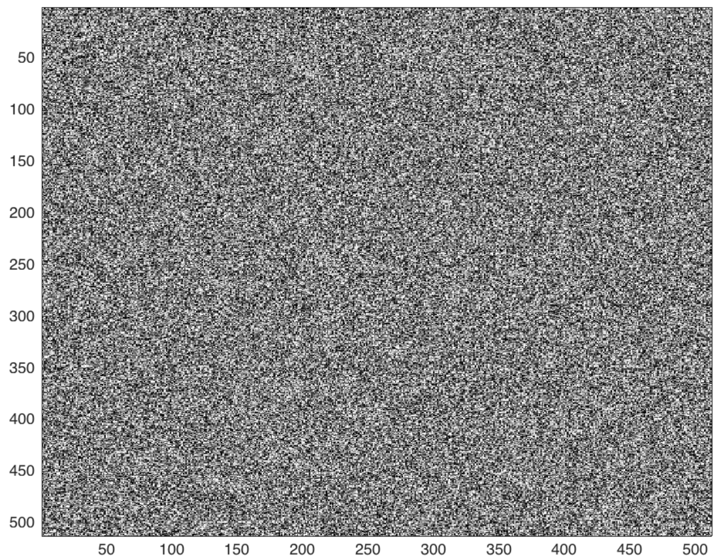

Figure 6: The image 255(x + 0.5)

2: The image(y + 127)

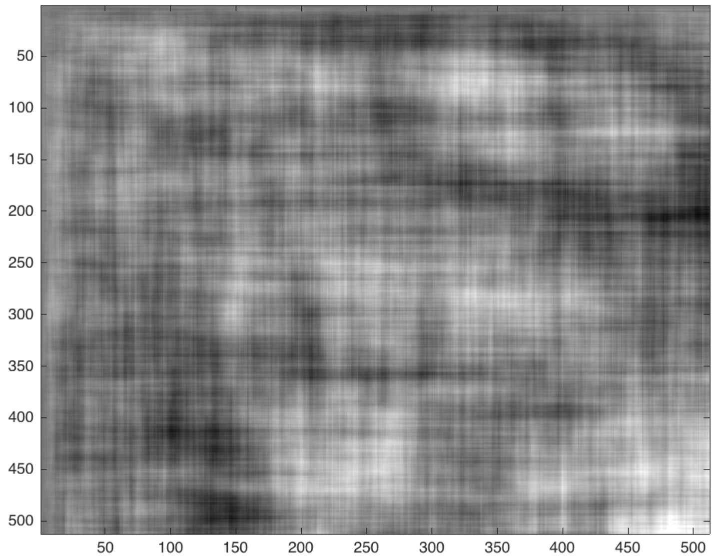

Figure 7: The image y + 127

3: A mesh plot of the function log Sy

Any uniformly distributed random variable in the range (-a/2 to a/2) has an average power(variance) given by $a^2/12$.
So:

$$S_x(u,v) = 1/12$$

$$S_y(u,v) = |H(u,v)|^2 S_x(u,v) = \frac{\frac{9}{(1-0.99e^{-ju}-0.99e^{-jv}+0.9801e^{-ju}e^{-jv})^2}}{12} = \frac{3}{4(1-0.99e^{-ju}-0.99e^{-jv}+0.9801e^{-ju}e^{-jv})^2}$$

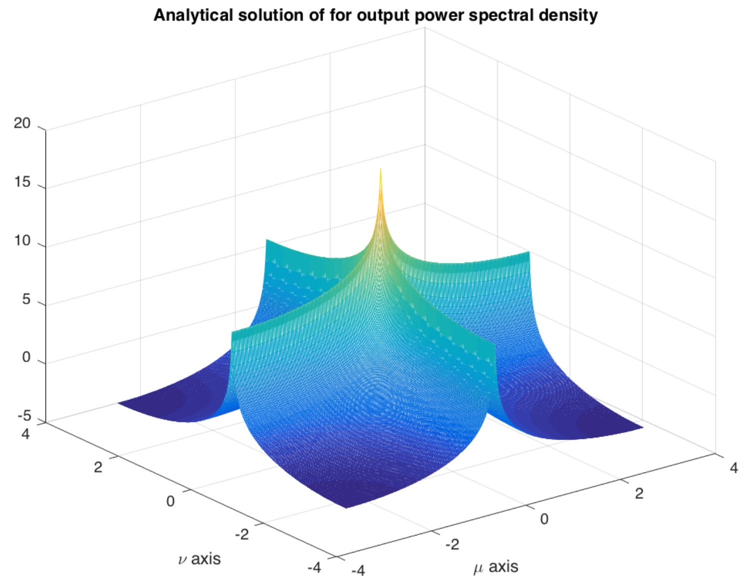

Figure 8: A mesh plot of the function log Sy with analytical solution

Similar to section 1, y array can be generated first. Then the PSD of y can be got via similar method used.

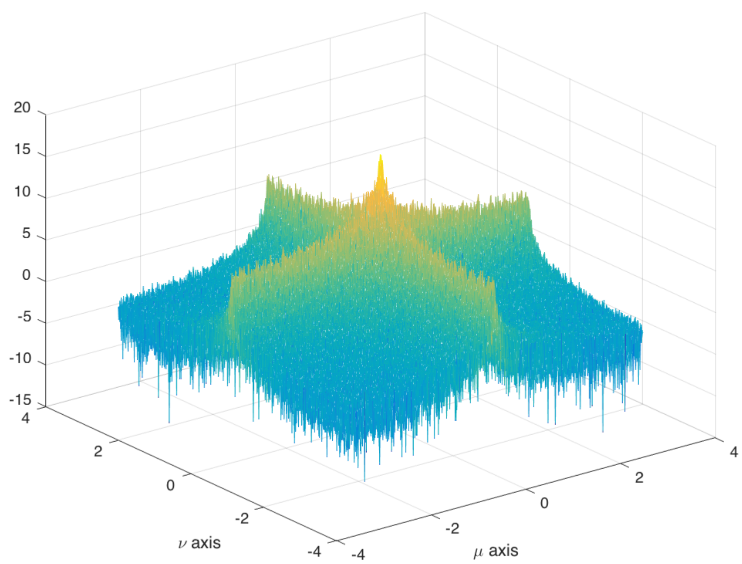

Figure 9: A mesh plot of the function log Sy using real signal

4:A mesh plot of the log of the estimated power spectral density of y using Better- SpecAnal(y)

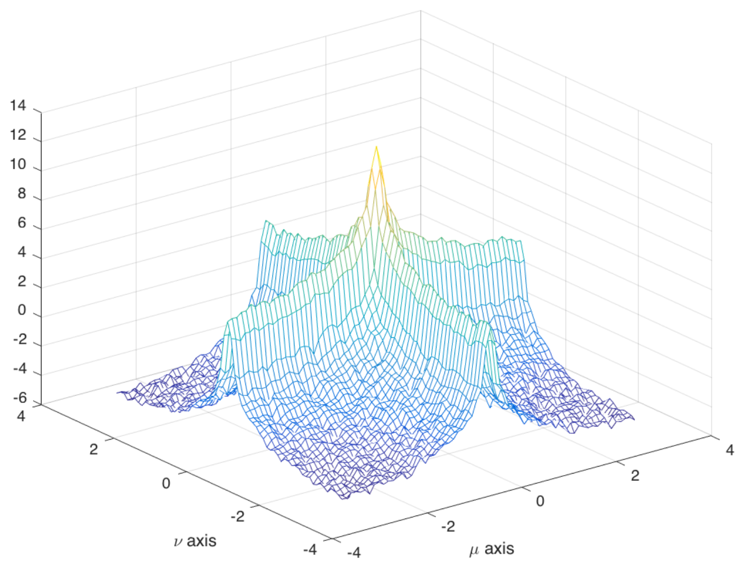

Figure 10: A better mesh plot of the log of the estimated power spectral density of y

5: Matlab code

See attachment.

Attachments: code

MATLAB code for section 3

```
% Clear memory and close existing figures
clear
close

% This line reads in a gray scale TIFF image.
% The matrix img contains a 2-D array of 8-bit gray scale values.

[img] = imread('img04g.tif');

% This line sets the colormap, displays the image, and sets the
% axes so that pixels are square.
% "map" is the corresponding colormap used for the display.
% A gray scale pixel value is treated as an index into the
% current colormap so that the pixel at location (i,j)
% has the color [r,g,b] = map(img(i,j),:) .
```

```
map=gray(256);
colormap(gray(256));
image(img)
axis('image')

X = double(img)/255;
W = hamming(64)*hamming(64)';

% Select an NxN region of the image and store it in the variable "z"

N = 64;
Sum = zeros(64,64);
for i = 1:5
    for j = 1:5
        m = 96 + (i-1)*64;
        n = 224 + (j-1)*64;
        C = X(m:(N+m-1), n:(N+n-1)).*W;
        Z = (1/N^2)*abs(fft2(C)).^2;
        Sum = Sum + Z;
    end
end
Avg = Sum/25;
Avg = log(fftshift(Avg));


% Plot the result using a 3-D mesh plot and label the x and y axises properly.

x = 2*pi*((0:(N-1)) - N/2)/N;
y = 2*pi*((0:(N-1)) - N/2)/N;
figure
mesh(x,y,Avg)
xlabel('\mu axis')
ylabel('\nu axis')
```

MATLAB code for section 4

```
clear all;
clc;

N = 512;
a = -0.5; b = 0.5;
x = (b-a).*rand(N) + a;
x_scaled = 255*(x+0.5);
figure;
colormap(gray(256));
image(uint8(x_scaled));

y = zeros(N);
y(1,1) = 3*x(1,1);
y(2,1) = 3*x(2,1);
y(1,2) = 3*x(1,2);

for i = 2:512
```

```
    for j = 2:512
        y(i,j) = 3*x(i,j) + 0.99*y(i-1,j)+  0.99*y(i,j-1) -  0.99^2*y(i-1,j-1);
    end
end
figure;
colormap(gray(256));
image(uint8(y+127));

Z = (1/N^2)*abs(fft2(y)).^2;

% Use fftshift to move the zero frequencies to the center of the plot
Z = fftshift(Z);

% Compute the logarithm of the Power Spectrum.
Zabs = log( Z );

% Plot the result using a 3-D mesh plot and label the x and y axises properly.

u = 2*pi*((0:(N-1)) - N/2)/N;
v = 2*pi*((0:(N-1)) - N/2)/N;
figure;
mesh(u,v,Zabs)
xlabel('\mu axis')
ylabel('\nu axis')


%% Theoretical plot of Sy
[u,v] = meshgrid(-pi:0.02:pi,-pi:0.02:pi);
Sy = abs(3./(1-0.99.*exp(-1i.*u)-0.99.*exp(-1i.*v)+0.9801.*exp(-1i.*u-1i.*v))).^2./12;
figure;
mesh(u,v,log(Sy))
xlabel('\mu axis')
ylabel('\nu axis')
title('Analytical solution of for output power spectral density  ')

%%
W = hamming(64)*hamming(64)';
N = 64;
% Select an NxN region of the image and store it in the variable "z"
Sum = zeros(N);
for i = 1:5
    for j = 1:5
        m = 96 + (i-1)*64;
        n = 96 + (j-1)*64;
        C = y(m:(N+m-1), n:(N+n-1)).*W;
        Z = (1/N^2)*abs(fft2(C)).^2;
        Sum = Sum + Z;
    end
end
Avg = Sum/25;
Avg = log(fftshift(Avg));


% Plot the result using a 3-D mesh plot and label the x and y axises properly.
```

```
u = 2*pi*((0:(N-1)) - N/2)/N;
v = 2*pi*((0:(N-1)) - N/2)/N;
figure
mesh(u,v,Avg)
xlabel('\mu axis')
ylabel('\nu axis')
```