Purdue University

ECE 637 Laboratory Exercise 1
2-D Random Processes

Tong Shen

January 28, 2017

1 Power Spectral Density of an Image

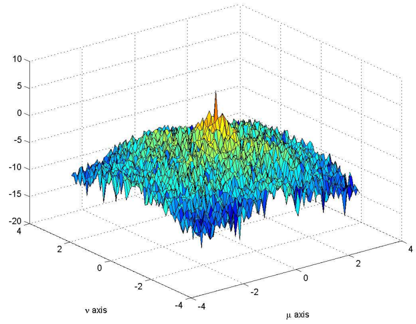
In this section, we will use MATLAB to read, analyze the power spectral density of gray img04.tif. Different windows are used to get the normalized energy spectrum. Also, a better PSD calculation based on hamming window and averaging method is implemented to obtain low noise power spectral density.

1.1 Gray Scale Image *img04.tif*

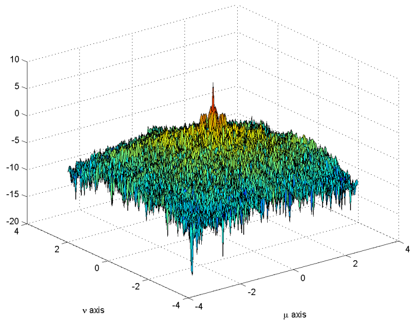
As it shows in Figure 1.1

Figure 1.1: *img04g.tif*

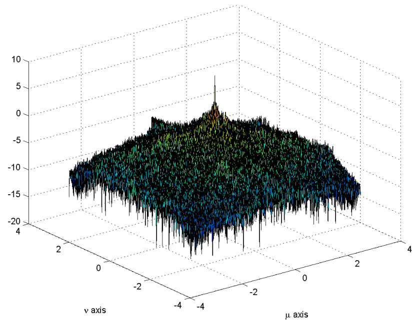
1.2 Power Spectral Density for Different Windows



(a) Block Size 64*64



(b) Block Size 128*128



(c) Block Size 256*256

Figure 1.2: PSD with Different Block size

According to the plots, the noise cannot be eliminated by increasing block size.

1.3 The Improved Power Spectral Density

In this section, we use get 25 different image window and utilize hamming window and averaging method to obtain improved power spectral density.

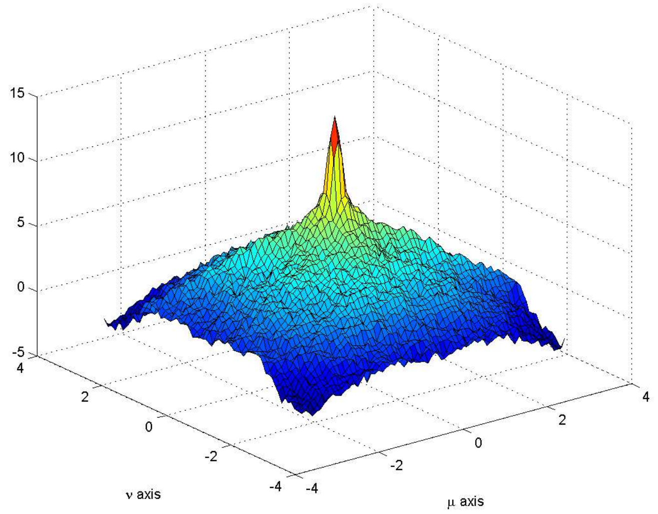


Figure 1.3: Improved Power Spectral Density

1.4 CODE LIST

1.4.1 BETTERSPECANAL.M

```
function y = BetterSpecAnal(img)
N = 64;
windows = zeros(64,64,25);
center = size(img)/2;
t = 1;
for i = -2:1:2
for j = -2:1:2
windows(:,:,t) = img(center + (-0.5+i)*N:center + (0.5+i)*N - 1,center + (-0.5+j)*N:center + (0.5+j)*N - 1);
t = t + 1;
end
end
w = hamming(64)*hamming(64)';
spec = zeros(64,64,25);
for i = 1:25
hamminged = windows(:,:,i).*w;
Z = (1/N^2)*abs(fft2(hamminged)).^2;
Z = fftshift(Z);
spec(:,:,i) = log(Z);
end
y = mean(spec,3);
```

```
end
```

2 Power Spectral Density of a 2-D AR Process

In this section, we create 2-D AR process and analyze its power spectral density.

2.1 Random image using *rand()*

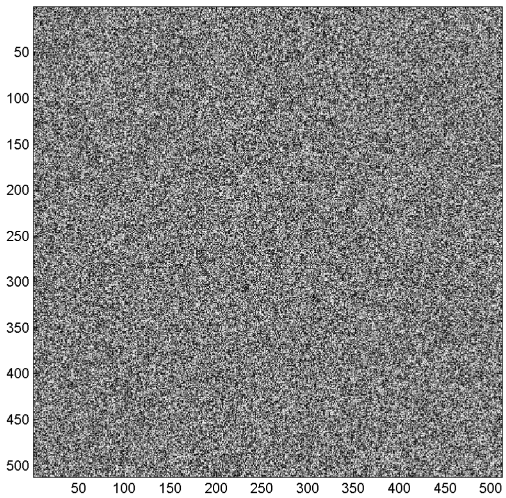


Figure 2.1: Random Image(Distributed in [-0.5,0.5])

2.2 Filtered Image with IIR filter

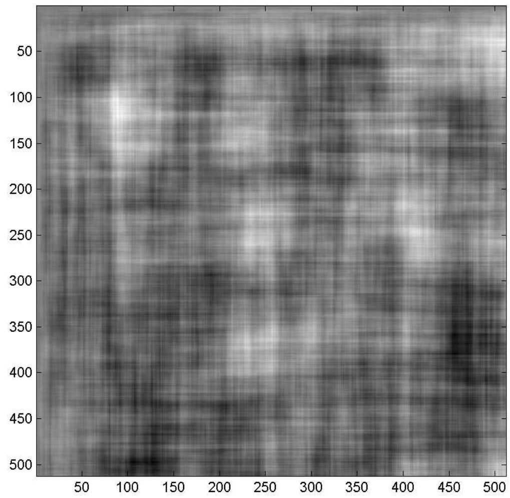


Figure 2.2: Filtered Result of the Noisy Image

2.3 Plot the Theoretical Result of the PSD

2.3.1 Theoretical Derivation of the PSD

First of all, we have the equation from the lecture:

$$S_y(e^{j\mu}, e^{j\nu}) = |H(e^{j\mu}, e^{j\nu})|^2 S_x(e^{j\mu}, e^{j\nu}) \tag{2.1}$$

And we also have the formula:

$$S_y(e^{j\mu}, e^{j\nu}) = \lim_{N\to\infty} \frac{1}{N} E(|X_N(e^{j\mu}, e^{j\nu})|^2) \tag{2.2}$$

For the uniform distribution X in [-0.5,0.5]: we have:

$$E[X] = 0 \tag{2.3}$$

$$\begin{aligned} Var[X] &= (E[X])^2 - E[X^2] \\ &= E[X^2] \\ &= \frac{1}{12} \end{aligned} \tag{2.4}$$

So,normalized power spectral density would be

$$
\begin{aligned}
S_x(e^{j\mu}, e^{j\nu}) &= \lim_{N\to\infty} \frac{1}{N} E(|X_N(e^{j\mu}, e^{j\nu})|^2). \\
&= \lim_{N\to\infty} \frac{1}{N} NVar[X] \\
&= E[X^2] \\
&= \frac{1}{12}
\end{aligned} \tag{2.5}
$$

And, then:

$$
\begin{aligned}
S_y(e^{j\mu}, e^{j\nu}) &= \frac{1}{12} |H(e^{j\mu}, e^{j\nu})|^2 \\
&= \frac{1}{12} \left| \frac{3}{1 - 0.99 z_1^{-1} - 0.99 z_2^{-1} + 0.9801 z_1^{-1} z_2^{-1}} \right|^2
\end{aligned} \tag{2.6}
$$

2.3.2 Plot of the Theoretical PSD

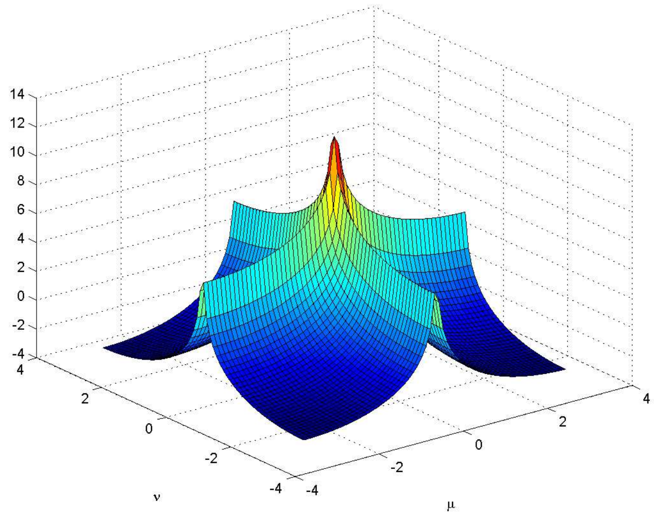


Figure 2.3: Plot of the Theoretical PSD

2.4 Plot of the estimated PSD

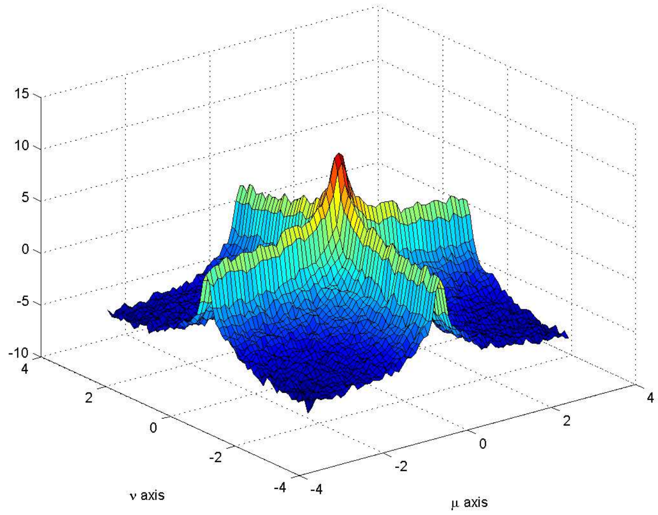


Figure 2.4: Estimated PSD for the 2-D AR Process

3 MATLAB code List

```
clc
clear

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

%%%%%%%%% block size 64
X = double(img)/255;

i=100;
j=100;

N = 64;
z = X(i:(N+i-1), j:(N+j-1));

```

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

Z = fftshift(Z);

Z_64 = log( Z );

x = 2*pi*((0:(N-1)) - N/2)/N;
y = 2*pi*((0:(N-1)) - N/2)/N;
figure(1)
surf(x,y,Z_64)
xlabel('\mu axis')
ylabel('\nu axis')
print(1,'-dpng','block64.png')

%%%%%%%%%% block size 128

X = double(img)/255;

i=100;
j=100;

N = 128;
z = X(i:(N+i-1), j:(N+j-1));

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

Z = fftshift(Z);

Z_128 = log( Z );

x = 2*pi*((0:(N-1)) - N/2)/N;
y = 2*pi*((0:(N-1)) - N/2)/N;
figure(2)
surf(x,y,Z_128)
xlabel('\mu axis')
ylabel('\nu axis')
print(2,'-dpng','block128.png')

%%%%%%%%%%%%%%%% block size 256

X = double(img)/255;

i=100;
j=100;
```

```
N = 256;
z = X(i:(N+i-1), j:(N+j-1));

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

Z = fftshift(Z);

Z_256 = log( Z );

x = 2*pi*((0:(N-1)) - N/2)/N;
y = 2*pi*((0:(N-1)) - N/2)/N;
figure(3)
surf(x,y,Z_256)
xlabel('\mu axis')
ylabel('\nu axis')
print(3,'-dpng','block256.png')

%%%%%%% Improved Power Spectral Density

N = 64;
Improved_PSD = BetterSpecAnal(img);
x = 2*pi*((0:(N-1)) - N/2)/N;
y = 2*pi*((0:(N-1)) - N/2)/N;
figure(4)
surf(x,y,Improved_PSD)
xlabel('\mu axis')
ylabel('\nu axis')
print(4,'-dpng','Improved_PSD.png')

%%%%%%% random img

A = rand(512) - 0.5;

img = (A+0.5)*255;
figure(5)
map=gray(256);
colormap(gray(256));
image(img)
axis('image')
print(5,'-dpng','Random_img.png')

output_img = zeros(512,512);
```

```
for i = 1:512
for j = 1:512
temp = 0;
temp = temp + 3*A(i,j);
if(i>2)
temp = temp + 0.99*output_img(i-1,j);
end
if(j>2)
temp = temp + 0.99*output_img(i,j-1);
end
if(i>2&&j>2)
temp = temp - 0.9801*output_img(i-1,j-1);
end
output_img(i,j) = temp;
end
end

fig = figure(6);
map = gray(256);
colormap(fig,map);
image(uint8(output_img+127))
axis('image')
print(6,'-dpng','filtered_img.png')
N = 64;

u = -pi:0.1:pi;
v = -pi:0.1:pi;
[U,V] = meshgrid(u,v);
sigma = 1/12;
PSD_t = sigma*abs(3./((1-0.99*exp(-1i*U)).*(1-0.99*exp(-1i*V)))).^2;
figure(7)
surf(U,V,log(PSD_t));
xlabel('\mu');
ylabel('\nu');
print(7,'-dpng','theoryPSD.png')

result = BetterSpecAnal(output_img);
x = 2*pi*((0:(N-1)) - N/2)/N;
y = 2*pi*((0:(N-1)) - N/2)/N;
figure(8)
surf(x,y,result)
xlabel('\mu axis')
ylabel('\nu axis')
print(8,'-dpng','estimatedPSD.png')
```