Lab 2 Report

Ruijie Song

Feb.04.2021

Section 1 Report:

1.

Figure 1. The gray scale image img04g.tif.

2.

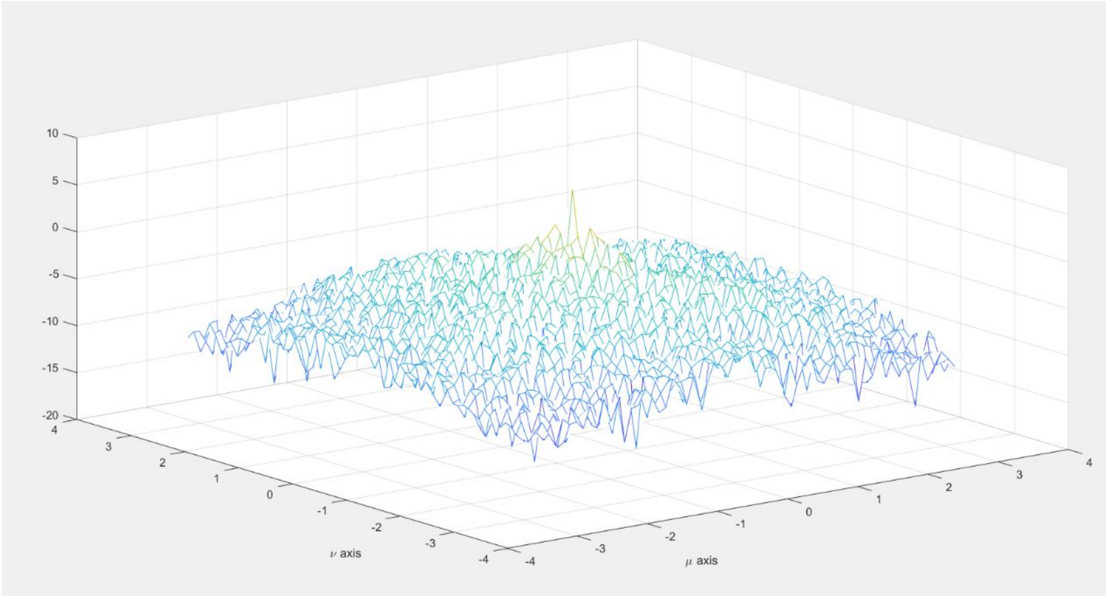


Figure 2. The power spectral density plots for block sizes of 64 × 64

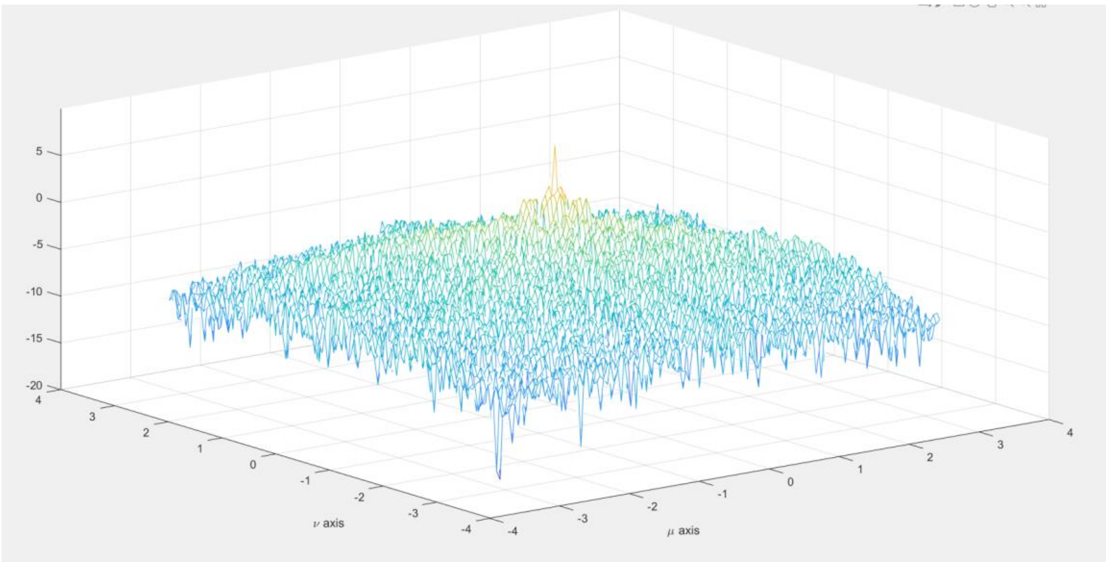


Figure 3. The power spectral density plots for block sizes of 128 × 128

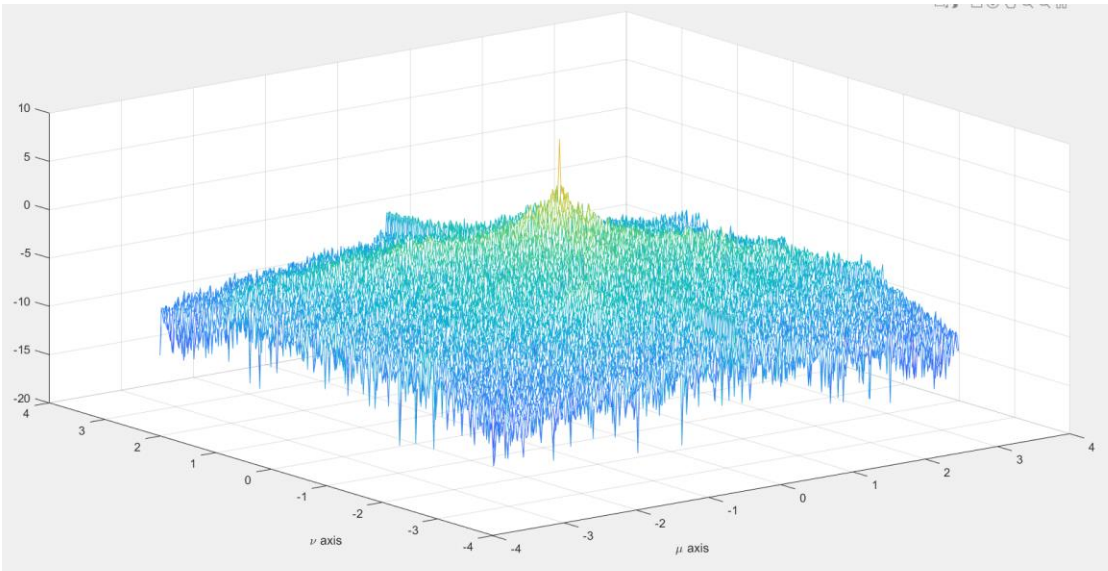


Figure 4. The power spectral density plots for block sizes of 256 × 256

3.

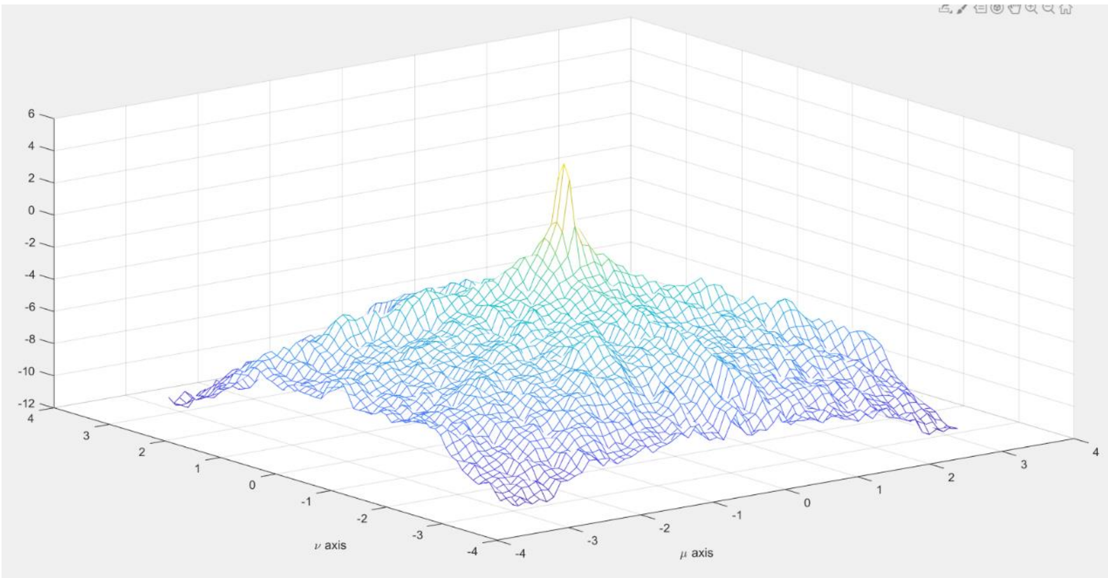


Figure 5. The improved power spectral density estimate.

2 Power Spectral Density of a 2-D AR Process

1.

Figure 6. The image 255 ∗ (x + 0.5).

2.

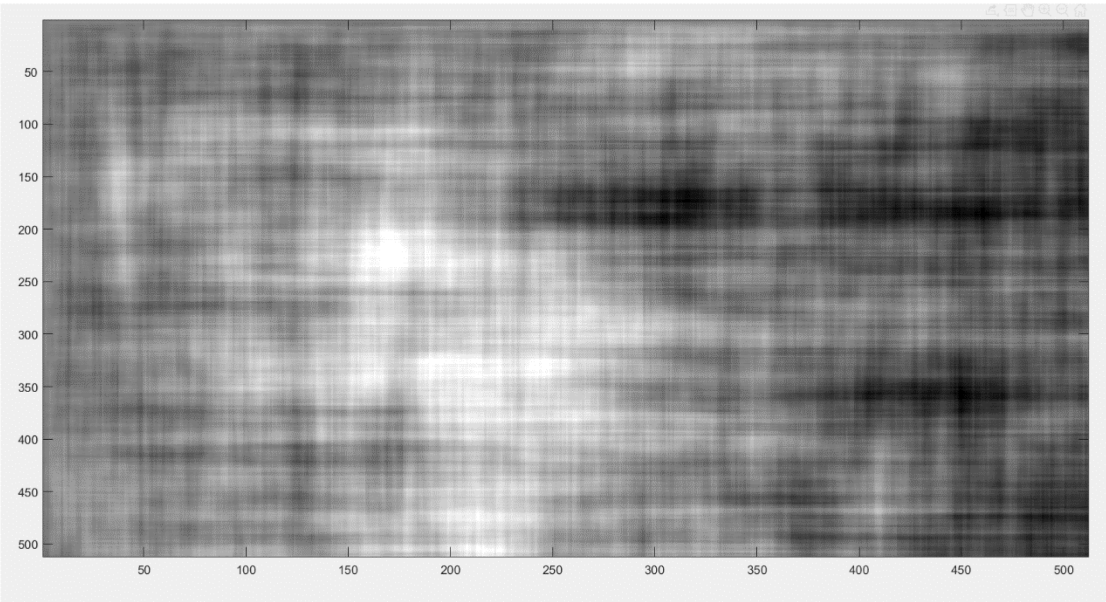
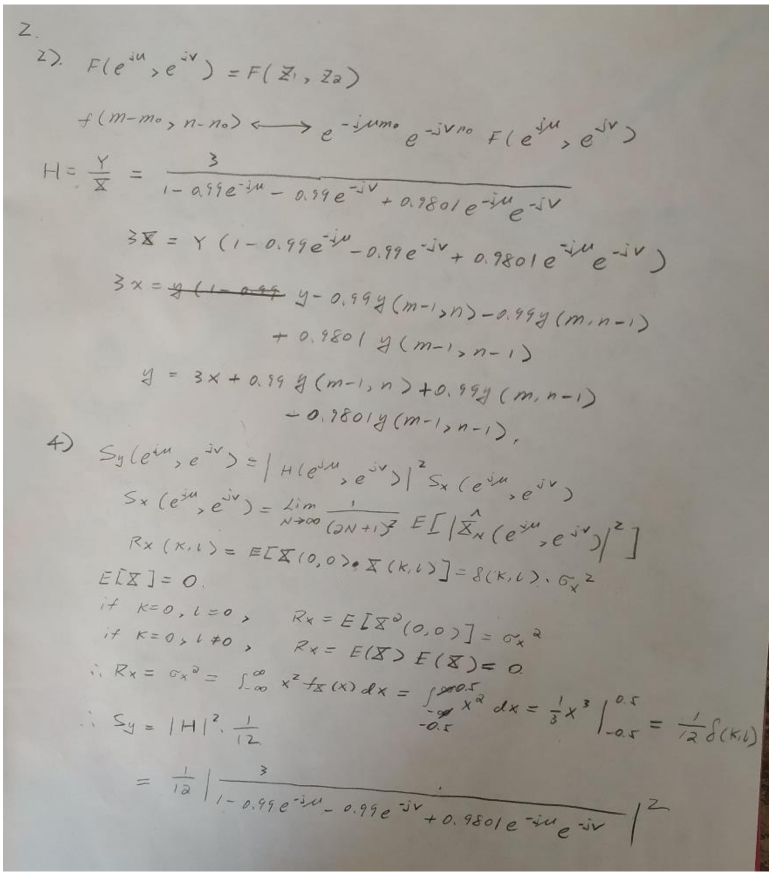


Figure 7. The image y + 127.

3.



2.

2). $F(e^{ju}, e^{jv}) = F(z_1, z_2)$

$$f(m-m_0, n-n_0) \longleftrightarrow e^{-jum_0} e^{-jvn_0} F(e^{ju}, e^{jv})$$

$$H = \frac{Y}{X} = \frac{3}{1-0.99e^{-ju}-0.99e^{-jv}+0.9801e^{-ju}e^{-jv}}$$

$$3X = Y(1-0.99e^{-ju}-0.99e^{-jv}+0.9801e^{-ju}e^{-jv})$$

$$3x = \cancel{y(1-0.99} \; y - 0.99y(m-1,n) - 0.99y(m,n-1) + 0.9801y(m-1,n-1)$$

$$y = 3x + 0.99y(m-1,n) + 0.99y(m,n-1) - 0.9801y(m-1,n-1).$$

4) $S_y(e^{ju}, e^{jv}) = |H(e^{ju}, e^{jv})|^2 S_x(e^{ju}, e^{jv})$

$$S_x(e^{ju}, e^{jv}) = \lim_{N\to\infty} \frac{1}{(2N+1)^2} E\left[|\hat{X}_N(e^{ju}, e^{jv})|^2\right]$$

$$R_x(k,l) = E[X(0,0) \cdot X(k,l)] = \delta(k,l) \cdot \sigma_x^2$$

$E[X] = 0$.

if $k=0, l=0$, $R_x = E[X^2(0,0)] = \sigma_x^2$

if $k=0, l\neq 0$, $R_x = E(X)E(X) = 0$

$$\therefore R_x = \sigma_x^2 = \int_{-\infty}^{\infty} x^2 f_X(x)\,dx = \int_{-0.5}^{0.5} x^2\,dx = \frac{1}{3}x^3 \Big|_{-0.5}^{0.5} = \frac{1}{12}\delta(k,l)$$

$$\therefore S_y = |H|^2 \cdot \frac{1}{12}$$

$$= \frac{1}{12}\left|\frac{3}{1-0.99e^{-ju}-0.99e^{-jv}+0.9801e^{-ju}e^{-jv}}\right|^2$$

Figure 8. Theoretically calculate Sy

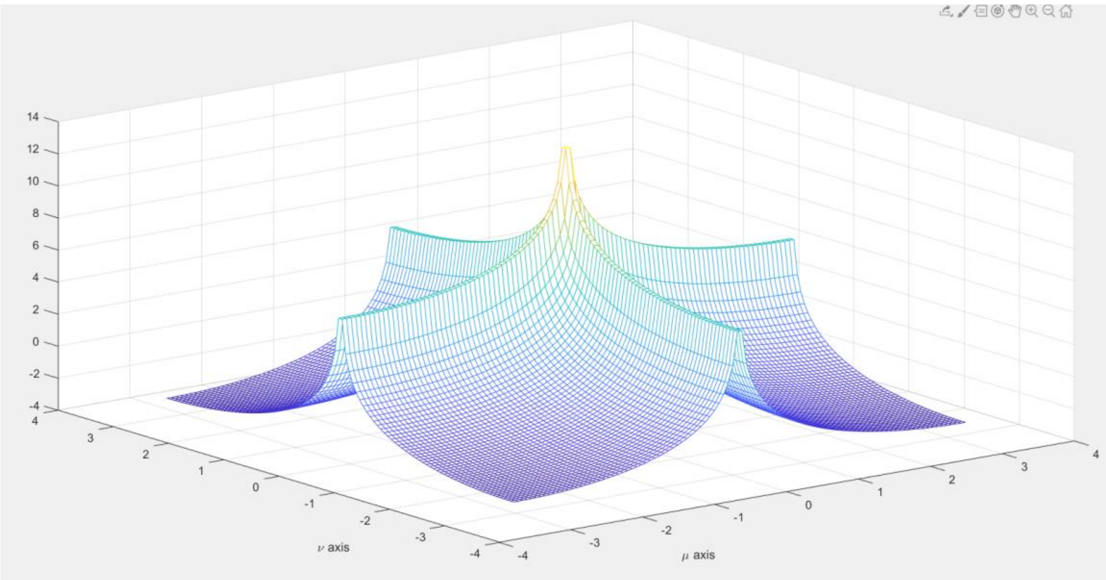


Figure 9. A mesh plot of the function log Sy

4.

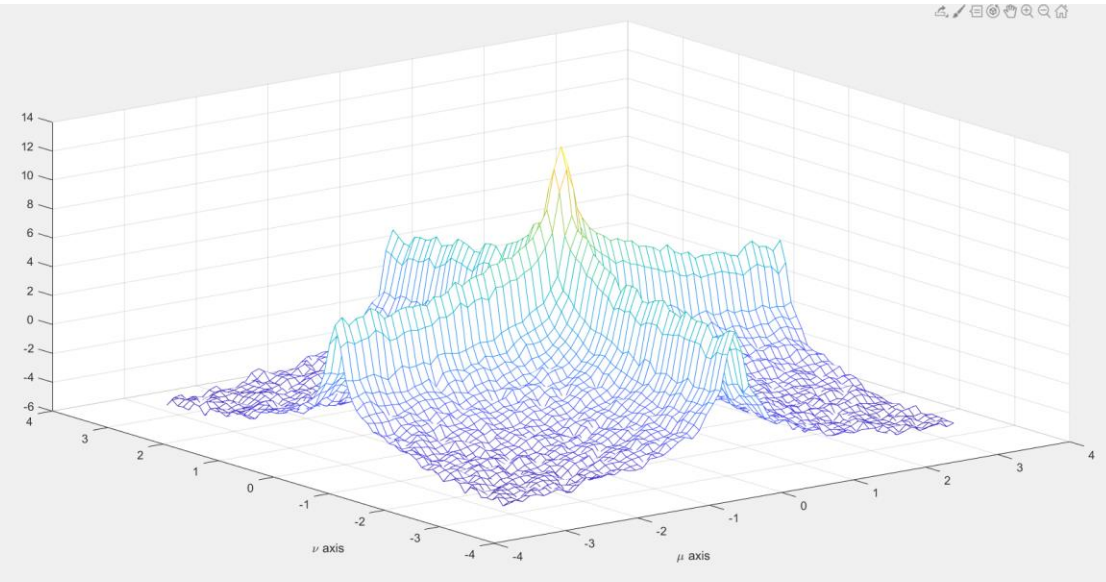


Figure 10. A mesh plot of the log of the estimated power spectral density of y using BetterSpecAnal(y)

```
clear all



%% 1 Power Spectral Density of an Image
clear all

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

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

X = double(img)/255;

BetterSpecAnal(X);



%% 2 Power Spectral Density of a 2-D AR Process
x = rand(512,512) - 0.5;
x_scaled=255*(x+0.5);
figure(1)
colormap(gray(256));
image(uint8(x_scaled))

x = padarray(x,[1 1]);
y = zeros(514,514);
for i = 2:513
    for j = 2:513
        y(i,j) = 3*x(i,j) + 0.99*y(i-1,j) + 0.99*y(i,j-1) - 0.9801*y(i-1,j-1);
    end
end
y = y(2:513,2:513);
y127 = y+127;
figure(2)
colormap(gray(256));
image(uint8(y127))

S = zeros(100,100);
K = 1; L = 1;
for u = linspace(-pi,pi,100)
    for v = linspace(-pi,pi,100)
        S(K,L) = 1/12 * (abs(3/(1-0.99*exp(-sqrt(-1)*u)-0.99*exp(-sqrt(-1)*v)+0.9801*exp(-sqrt(-1)
*u)*exp(-sqrt(-1)*v))))^2;
        L = L+1;
    end
```

```
    L = 1;
    K = K+1;
end
S = log( S );
figure(3)
mesh(linspace(-pi,pi,100),linspace(-pi,pi,100),S)
xlabel('\mu axis')
ylabel('\nu axis')


%%  Use BetterSpecAnal(y), your Matlab function from the previous exercise, to estimate the power
spectral density of y. Plot the estimated power spectral density and export the result.
BetterSpecAnal(y);
```

```
function BetterSpecAnal(x)

% initialize the e 25 non-overlapping image windows of size 64 × 64
windows = zeros(64,64,25);

% find the center of x
[r,c] = size(x);
r = r/2;
c = c/2;
N = 64;

% obtain the 64*64 windows
k = 1;
for i = linspace(-2,2,5)
    for j = linspace(-2,2,5)
        windows(:,:,k) = x(((r+i*N)-(N/2-1)):((r+i*N)+N/2),((c+j*N)-(N/2-1)):((c+j*N)+N/2));
        k = k + 1;
    end
end

% create the 2-D Hamming window
W=hamming(64)*hamming(64)';

% Multiply each 64 × 64 window by a 2-D separable Hamming window
for k = 1:1:25
    windows(:,:,k) = windows(:,:,k) .* W;
end

% Compute the power spectrum for the NxN region
windows = (1/N^2)*abs(fft2(windows)).^2;

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

% Average this power spectral density across the 25 windows
A_windows = mean(windows,3);

% Display a mesh plot of the log of the estimated power spectral density
psd = log( A_windows );
x = 2*pi*((0:(N-1)) - N/2)/N;
y = 2*pi*((0:(N-1)) - N/2)/N;
figure
mesh(x,y,psd)
xlabel('\mu axis')
ylabel('\nu axis')

end
```