ECE 637 Digital Image Processing Laboratory: 2-D Random Processes

Yang Wang

January 26, 2016

1 Power Spectral Density of an Image

In this section, a gray scale image is analyzed for its power spectral density content. Different window size is used to calculate the normailzed energy spectrum. In the last section, a better spectral density analysis implemetation is realized by applying multiple non-overlapping windows of the same size to the image.

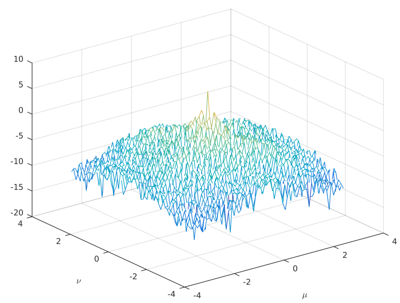
1.1 Plot grayscale image

Plot the given gray scale image in MATLAB.

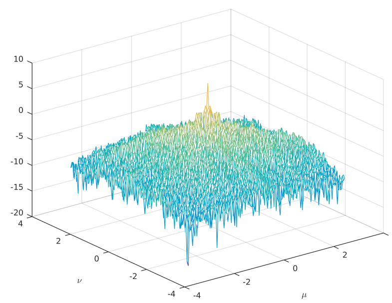
Figure 1: img04g.tif

1.2 Plot power spectral density for different block sizes

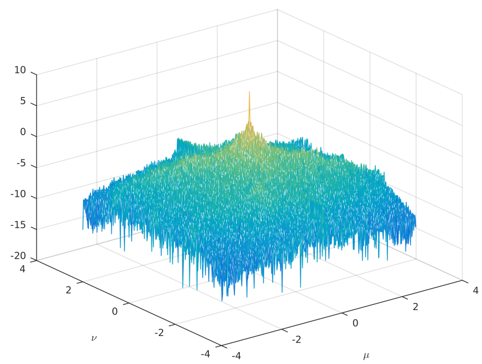
From Figure 2b) and 2c), even with increased block sizes, the power spectrum estimates main noisy.



(a) PSD with block size 64×64



(b) PSD with block size 128×128



(c) PSD with block size 256×256

Figure 2: PSD with different block sizes

1.3 Plot improved power spectral density estimate

From Figure 3, the noise from the original PSD is eliminated.

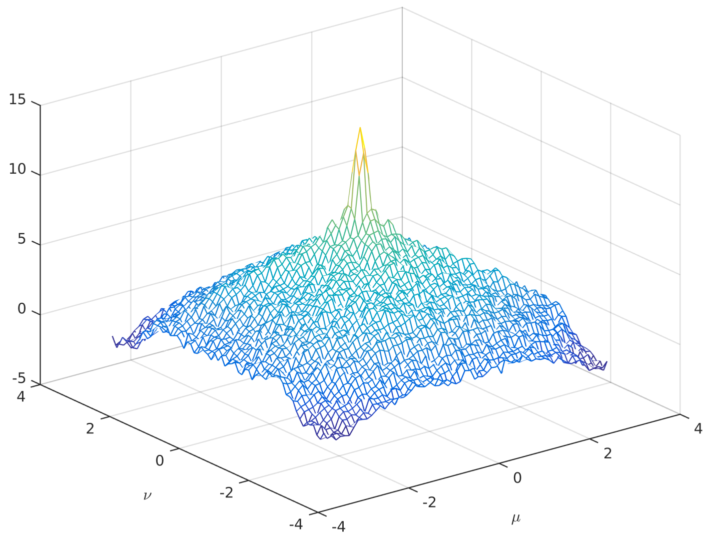


Figure 3: Improved PSD Estimate

1.4 Code listing

1.4.1 BetterSpecAnal.m

```
function [Z, fig] = BetterSpecAnal(img)
    W = hamming(64) * hamming(64)';
    N = 64;
    w_start = floor((size(img) - 5*N)/2);
    X = double(img);

    Z = zeros(N, N);
    for i = 1:5
        for j = 1:5
            z = X(w_start(1) + (i-1)*N:w_start(1) + i*N - 1,
             ↪  w_start(2) + (j-1)*N:w_start(2) + j*N - 1);
            Z = Z + log((1/N^2)*abs(fftshift(fft2(z.*W)).^2));
        end
    end
```

```
    Z = Z/25;

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

2 Power Spectral Density of a 2-D AR Process

In this section, a synthetic 2-D autoregressive (AR) process is generated and its power spectral density is analyzed.

2.1 Plot the scaled image generated using rand()

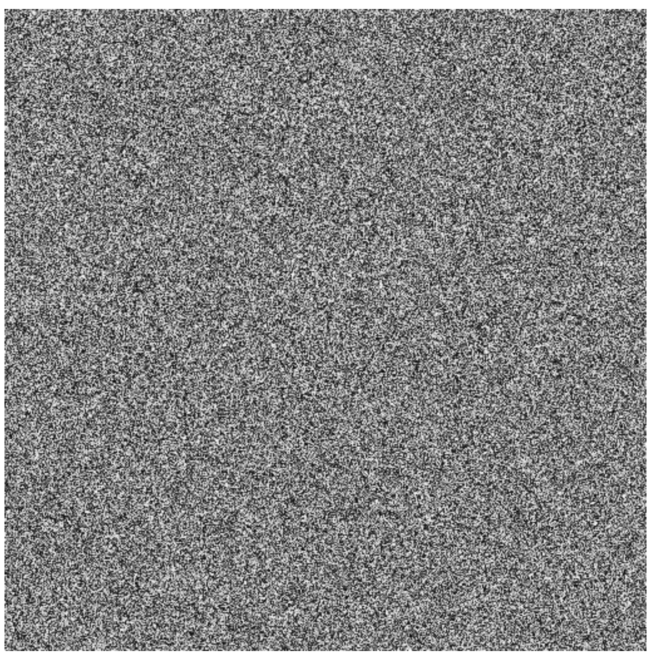


Figure 4: Image generated using rand() uniformly distributed on [-0.5,0.5]

2.2 Find the difference equation to the given filter

We are given:

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

We know,

$$H(z_1, z_2) = \frac{Y(z_1, z_2)}{X(z_1, z_2)}$$

Hence,

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

Taking the inverse Z-transform,

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

2.3 Plot the filtered image

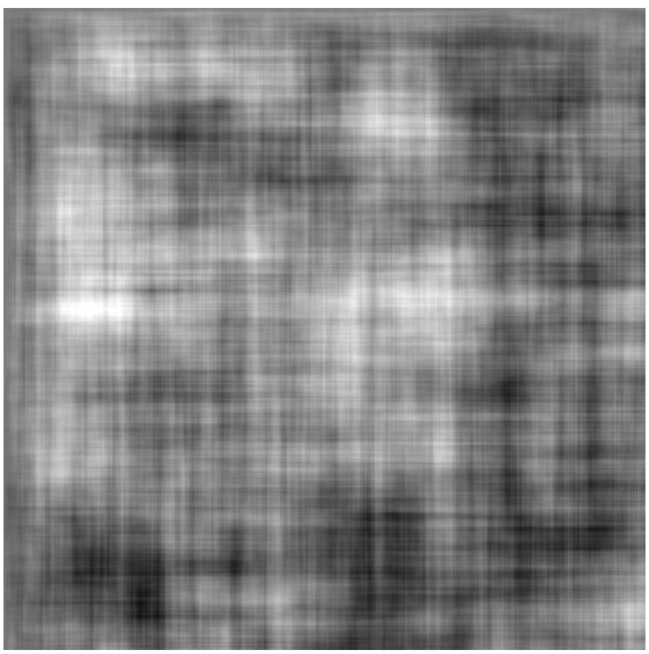


Figure 5: Filtered results of noisy image

2.4 Plot the theoretical result of log(PSD)

2.4.1 Derivation of PSD

We know the relation:

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

We need to derive the PSD of x. We also know:

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

We have the relation:

$$Var[X] = E[X^2] - (E[X])^2 \tag{3}$$

For uniform distribution on interval [-0.5,0.5], $(E[X])^2 = 0$, hence,

$$Var[X] = E[X^2]$$

Further,

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

Therefore,

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

2.4.2 Plot of theoretical PSD

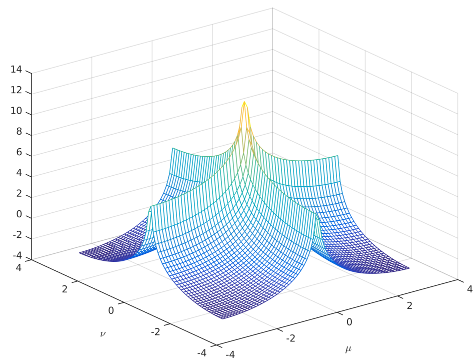


Figure 6: Theoretical plot of PSD of 2-D AR Process

2.5 Plot the estimated result of log(PSD)

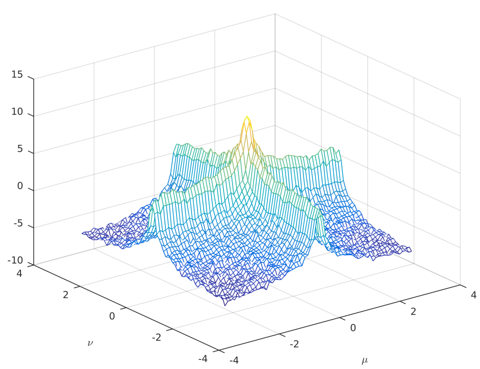


Figure 7: Estimated plot of PSDi of 2-D AR Process

3 Additonal code listing

3.1 SpecAnal.m

```
%% Section 1 %%
```

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

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

map = gray(256);
fig4 = figure(4);
colormap(fig4, map);
image(uint8(img));
axis('image');
imwrite(uint8(img), '../report/gs_img_4g.png');

X = double(img)/255;

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

p=100;
q=100;

N = 64;
z = X(p:(N+p-1), q:(N+q-1));

% Compute the power spectrum for the NxN region
Z = (1/N^2)*abs(fft2(z)).^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.
x = 2*pi*((0:(N-1)) - N/2)/N;
y = 2*pi*((0:(N-1)) - N/2)/N;
fig1 = figure(1);
mesh(x,y,Zabs);
xlabel('\mu');
ylabel('\nu');
print('-dpng','-r300','../report/psd_64x64.png');

% N = 128 %
N = 128;
```

```
z = X(p:(N+p-1), q:(N+q-1));
Z = (1/N^2)*abs(fft2(z)).^2;
Z = fftshift(Z);
Zabs = log(Z);
x = 2*pi*((0:(N-1)) - N/2)/N;
y = 2*pi*((0:(N-1)) - N/2)/N;
fig2 = figure(2);
mesh(x,y,Zabs);
xlabel('\mu');
ylabel('\nu');
print('-dpng','-r300','../report/psd_128x128.png');

% N = 256 %
N = 256;
z = X(p:(N+p-1), q:(N+q-1));
Z = (1/N^2)*abs(fft2(z)).^2;
Z = fftshift(Z);
Zabs = log(Z);
x = 2*pi*((0:(N-1)) - N/2)/N;
y = 2*pi*((0:(N-1)) - N/2)/N;
fig3 = figure(3);
mesh(x,y,Zabs);
xlabel('\mu');
ylabel('\nu');
print('-dpng','-r300','../report/psd_256x256.png');

% Use BetterSpecAnal.m %
[Z_better, fig5] = BetterSpecAnal(img);
print('-dpng','-r300','../report/psd_better.png');

%% Section 2 %%

img_rand = -0.5 + (0.5+0.5)*rand(512);
img_rand_s = 255 * (img_rand+0.5);
fig6 = figure(6);
colormap(fig6, map);
image(uint8(img_rand_s));
axis('image');
imwrite(uint8(img_rand_s), '../report/randimg.png');

img_f = zeros(512, 512);
for m = 1:512
    for n = 1:512
        img_f(m,n) = 3 * img_rand(m,n);
        if (m > 1)
            img_f(m,n) = img_f(m,n) + 0.99*img_f(m-1,n);
```

```
        end
        if (n > 1)
            img_f(m,n) = img_f(m,n) + 0.99*img_f(m,n-1);
        end
        if (m > 1 && n > 1)
            img_f(m,n) = img_f(m,n) - 0.9801*img_f(m-1,n-1);
        end
    end
end

fig7 = figure(7);
colormap(fig7, map);
image(uint8(img_f+127));
axis('image');
imwrite(uint8(img_f+127), '../report/randimg_f.png');

% Theoretical PSD %
u = -pi:0.1:pi;
v = -pi:0.1:pi;
[U,V] = meshgrid(u,v);
sigma = 1/12;
S_y =
 ↪ abs(3./((1-0.99*exp(-1i*U)).*(1-0.99*exp(-1i*V)))).^2*sigma;
fig8 = figure(8);
mesh(U,V,log(S_y));
xlabel('\mu');
ylabel('\nu');
print('-dpng','-r300','../report/psd_theo.png');

% Estimated PSD %
[est_S_y, fig9] = BetterSpecAnal(img_f);
print('-dpng','-r300','../report/psd_esti.png');
```