EE 440 Autumn 2018
Homework 5 Report

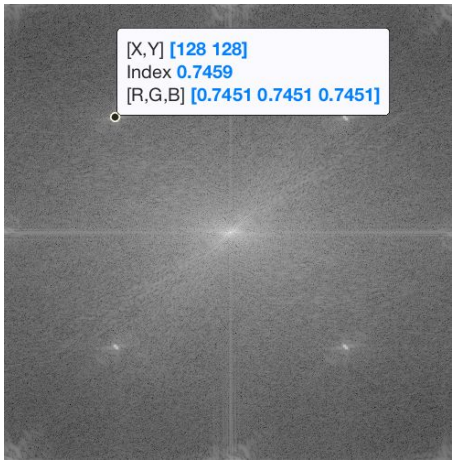
Professor: Ming-Ting Sun
October 29th, 2018

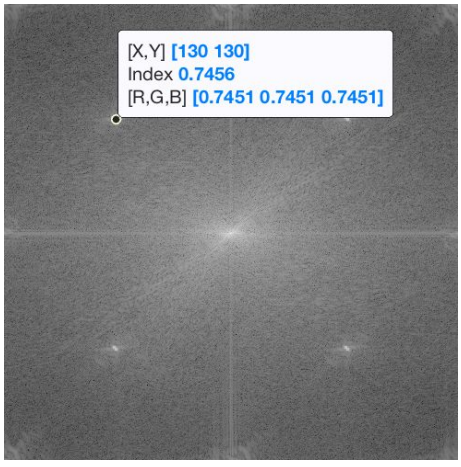
He Feng 1427841

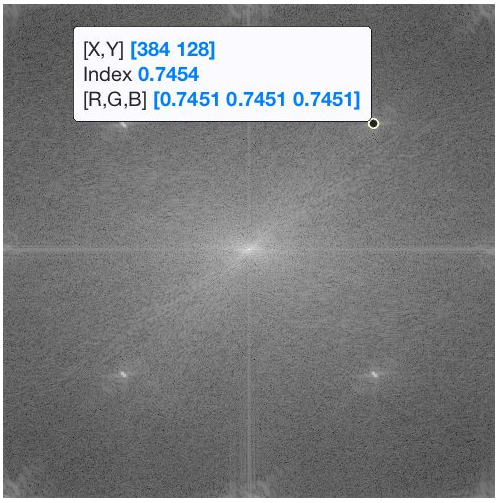
In this problem, we aimed to remove the noise of the image by filtering in the frequency domain. I initially load the image and display the original image into the first subplot of the first figure. Then I convert the image into double type such that I can apply FFT to it. After this step I used log function and mat2gray function to determine the magnitude of the image in frequency domain. Then I used angle function to get the phase, and I display the magnitude and phase into the second and third subplot of the first figure.

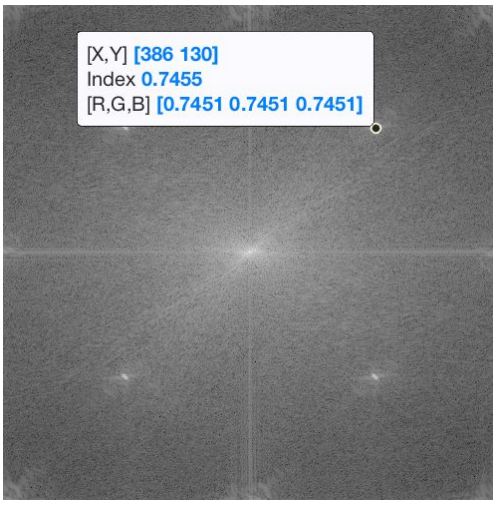
After drawing the original image and its magnitude and phase, I used the data cursor to find the coordinates of the noise points in the original magnitude graph. Even though it seems that there are only four white dots on the screen, but if we zoom out the image we can see each white space contains more than one pixel which is white. Hence I found eight pixels intotal and create a two dimensional variable to represent the x-coordinate and y-coordinate of the pixel. After doing this step, I found the size of the variable above, and I used the for loop to discover each pixel one by one. I set the pixel and the pixels around them into zero value, such that we can see them as color black instead of white. In this way we can cover the white color pixel in the image better. Then I set the fixed image back to time domain, and change its type to uint8. I draw the above image into the first subplot of the second figure, and this image is my denoised image. Then I used the same method to get the magnitude graph and phase graph of the fixed image into the second and third subplots of the second figure such that I can compare the difference. After comparing I can see the white dots in the original magnitude graph have already changed to color black, hence I can say I did the process correctly.

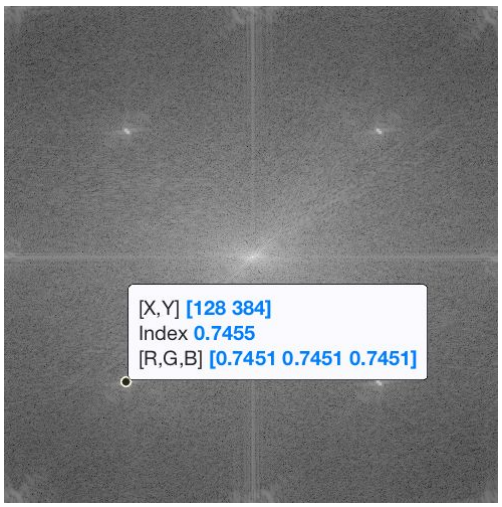
Here are the screenshot of the coordinates of each white dots in the original magnitude graph by using the data cursor:

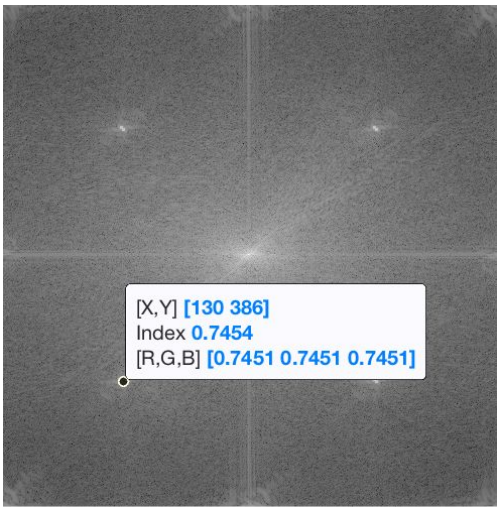


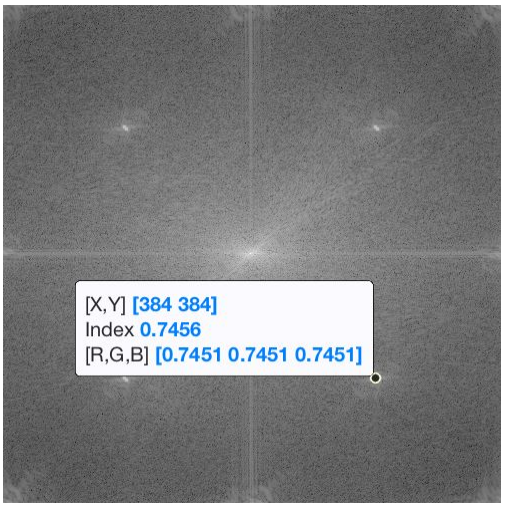


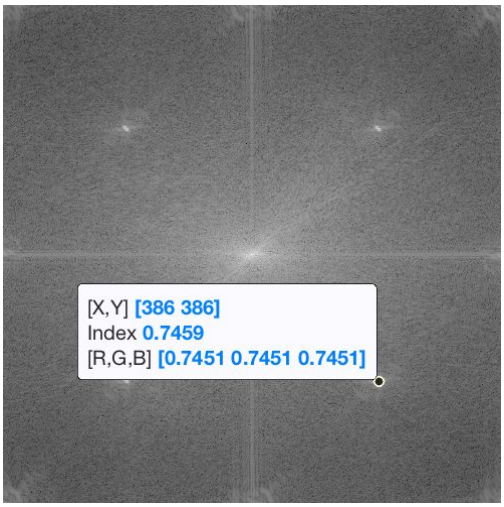

[X,Y] [384 128]
Index 0.7454
[R,G,B] [0.7451 0.7451 0.7451]


[X,Y] [386 130]
Index 0.7455
[R,G,B] [0.7451 0.7451 0.7451]

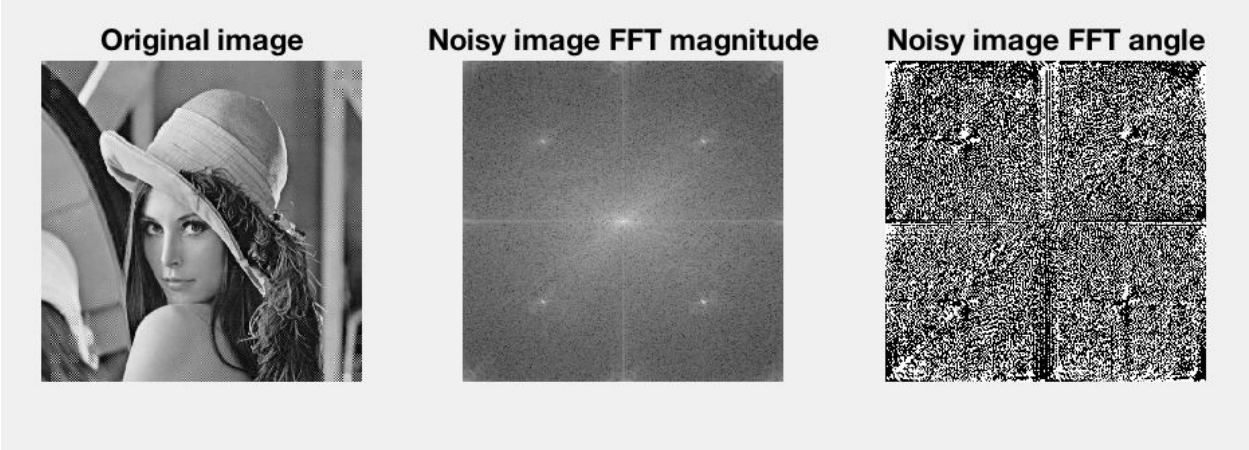

[X,Y] [128 384]
Index 0.7455
[R,G,B] [0.7451 0.7451 0.7451]

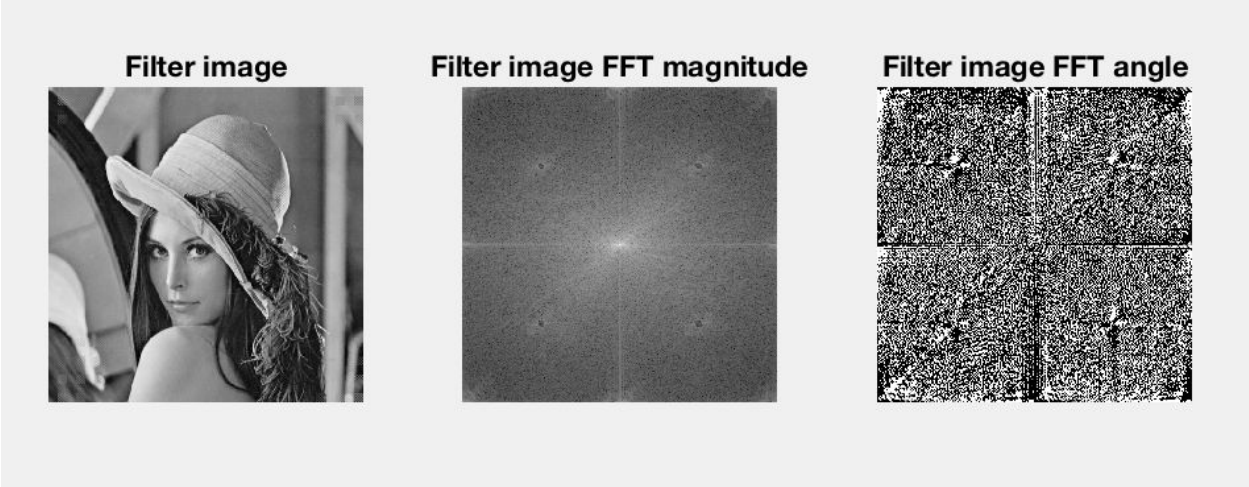

[X,Y] [130 386]
Index 0.7454
[R,G,B] [0.7451 0.7451 0.7451]


[X,Y] [384 384]
Index 0.7456
[R,G,B] [0.7451 0.7451 0.7451]


[X,Y] [386 386]
Index 0.7459
[R,G,B] [0.7451 0.7451 0.7451]

Here are the output of my Matlab code. The first figure represents the original image, and the second figure represents the fixed image.





Here are the Matlab code HW5.m

```
% He Feng
% EE 440 HW 5

clear all;
close all;

im = imread('5_1.bmp');
figure(1);
subplot(1,3,1)
imshow(im)
title('Original image');
im = double(im);

%% Apply FFT
f = fftshift(fft2(im));

magnitudeF = log(abs(f));
magnitudeF = mat2gray(magnitudeF);
angleF = angle(f);

subplot(1,3,2)
imshow(magnitudeF);
title('Noisy image FFT magnitude');
subplot(1,3,3)
imshow(angleF);
title('Noisy image FFT angle');

%% Fixing the noise.
% Using the Data Cursor to get the coordinate of each noise point in Noisy
% Image FFT Magnitude graph.
xy(:,1) = [128 130 384 386 128 130 384 386];
xy(:,2) = [128 130 128 130 384 386 384 386];

% Locate each point of noise and make the pixels arround them also to black
% such that the noise will be removed.
k=size(xy);
for i=1:k
    f(xy(i,1)-3:xy(i,1)+3,xy(i,2)-3:xy(i,2)+3) = 0;
end

%% Print out the fixed images.
im1 =real(ifft2(ifftshift(f)));
im1 = im2uint8(mat2gray(im1));
figure(2);
subplot(1,3,1)
imshow(im1);
title('Filter image');

f1 = fftshift(fft2(im1));

magnitudeF1 = log(abs(f1));
magnitudeF1 = mat2gray(magnitudeF1);
angleF1 = angle(f1);
subplot(1,3,2)
imshow(magnitudeF1);
title('Filter image FFT magnitude');
subplot(1,3,3)
imshow(angleF1);
title('Filter image FFT angle');
```