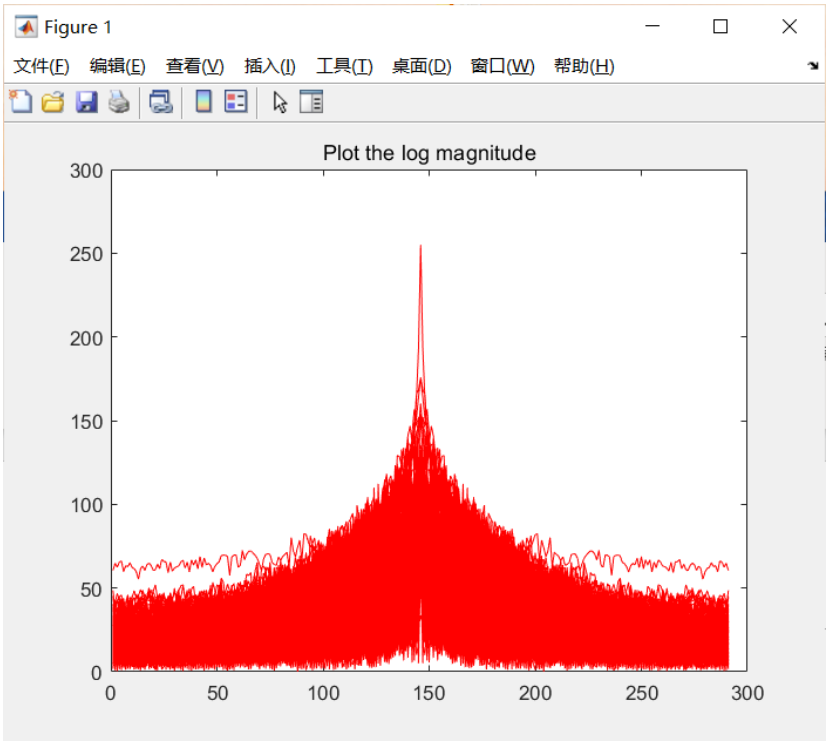
TS Logbook

Yibing Liu 01939400

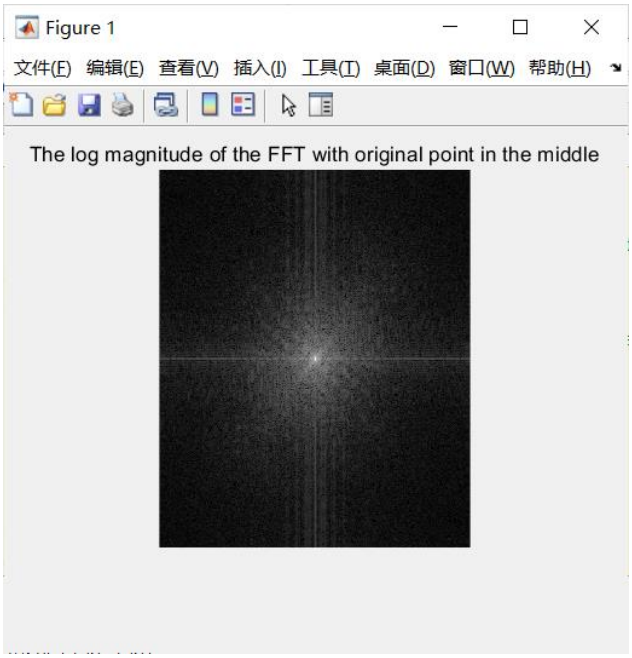
Part1. Image Transforms

1.

a. We take the picture ‘pout.tif’ as an example, firstly, we can plot the log magnitude of the FFT :

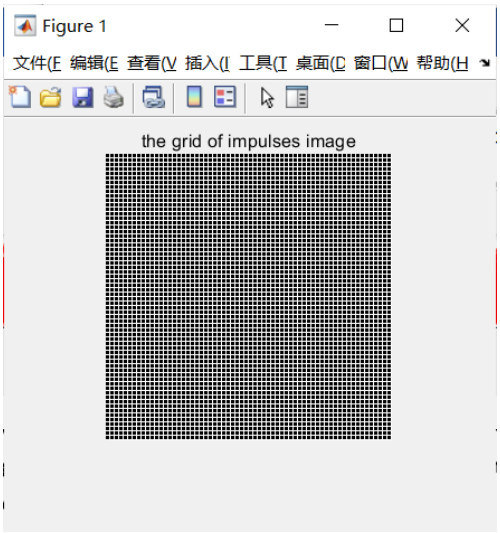


And we can just *imshow* the picture with the original point is in the middle

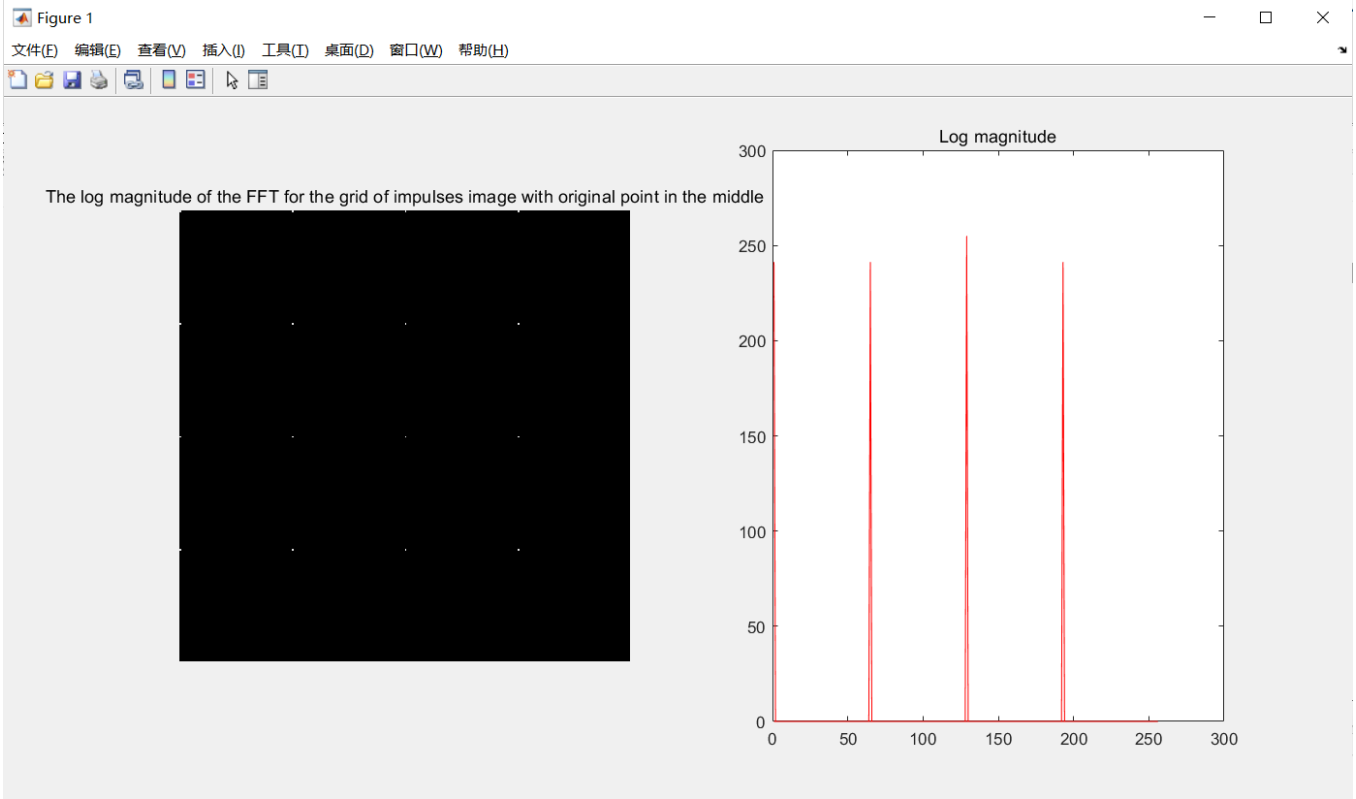


We can see that with the low index, FFT coefficients are always higher than those with high index. The reason is that the natural images are often smooth, the low-frequency components are always the main component and there is a very small part of high-frequency component.

b. Following the guidance, we can get a 256*256 grid of impulses: by



And repeat the process in a, we can get:



To get the periodic and non-periodic signals, we can use the *cos* and *rand*:

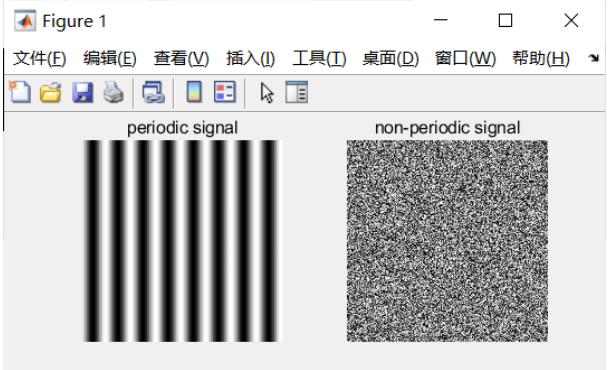
The way we used to produce periodic signal:

```
%Periodic image
image_cos = ones(256,256);
%Generate the grid of impluses image
for i = 1:256
    for n = 1:256
        image_cos(i,n) = 255*cos((pi/16)*((i-1)*256+n)+pi/8);
    end
end
```

the way we used to produce nonperiodic signal:

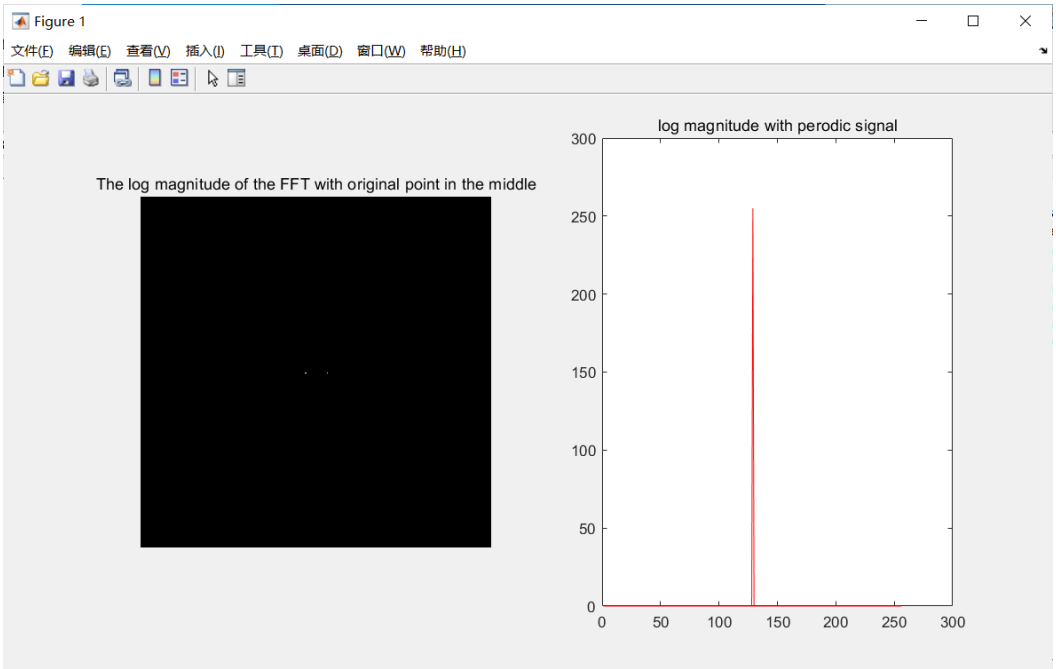
```
image_rand = 255*rand(256);
```

plot the signals we can get:



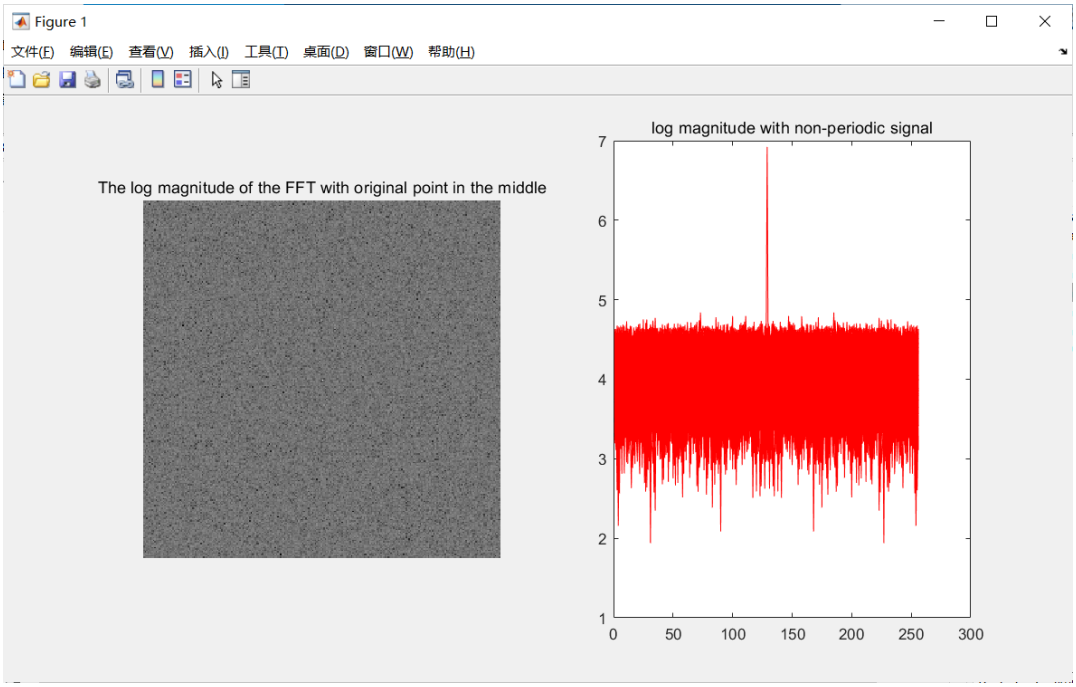
Then we can repeat the steps in a:

For periodic signal:



For the periodic image, because it is periodic, it only has one frequency components, hence in the log magnitude, it only has one peak.

For non-periodic signal:



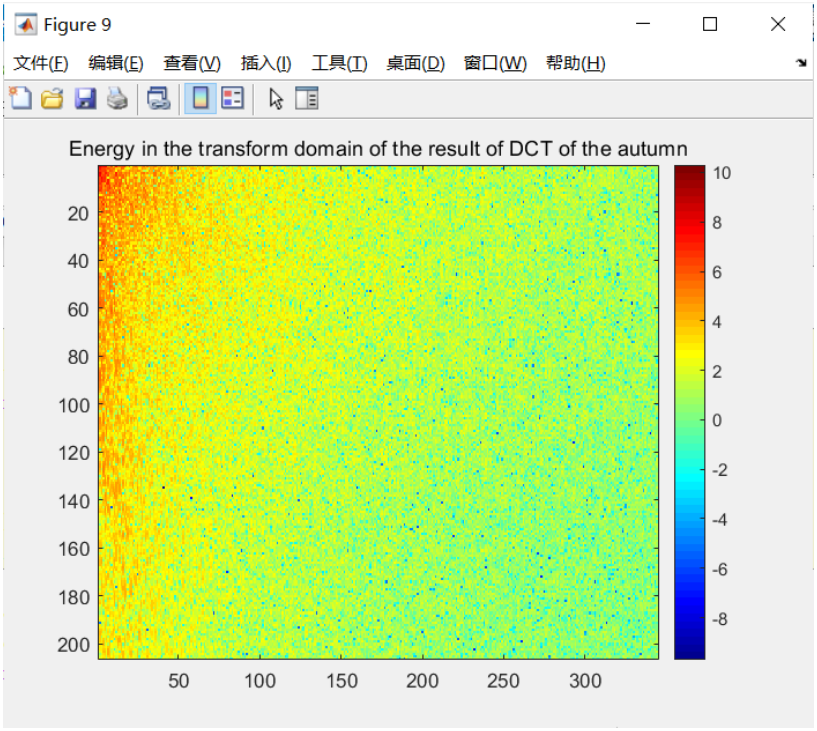
For the non-periodic image, it doesn't have any periodicity, so in the log magnitude, there are many different frequency components.

c. In 1.c, we use the picture 'cameraman' as A, and the picture 'sevilla' as B, combine them, we can get:

In this part, we can see that the picture combined mainly shows the information of picture A.

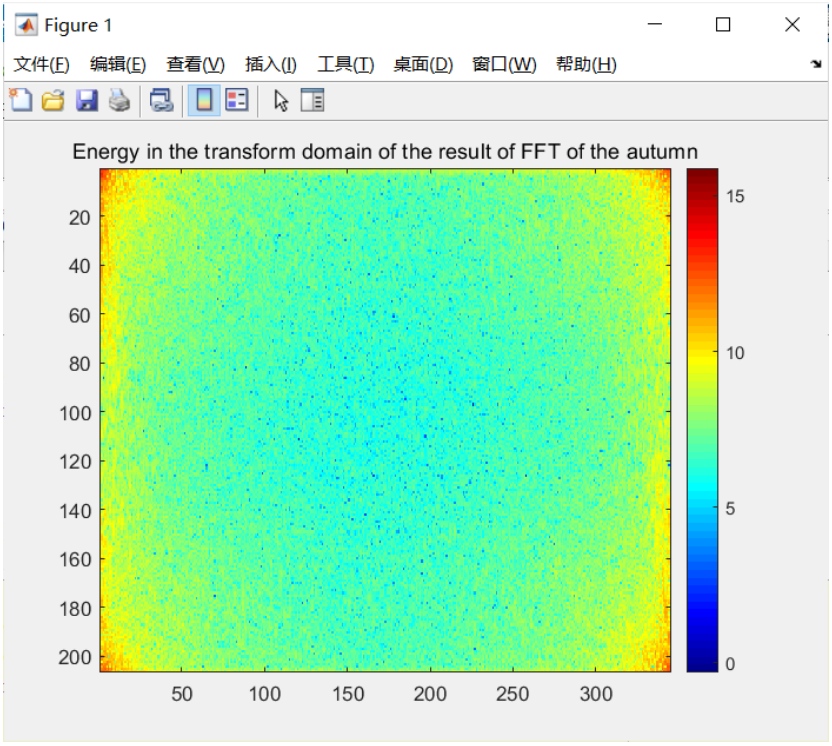
2. DCT

a. Plot the DCT:



The figure is just as the note said, most of the energy in the transform domain is concentrated within the upper left corner.

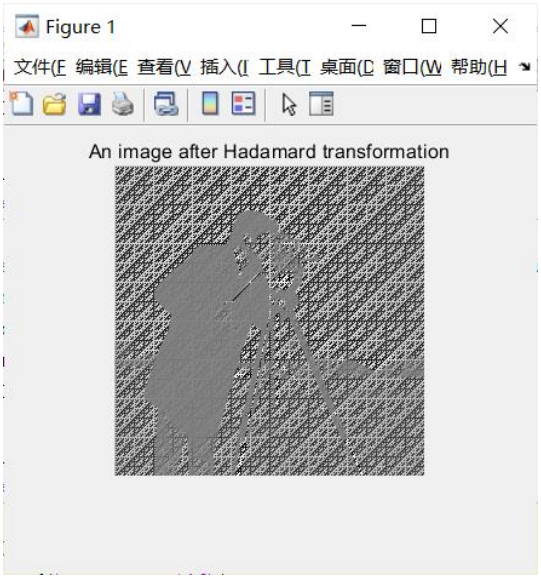
b. FFT:



The figure shows that the energy in FFT is concentrated in the four corners, which is mainly due to the symmetry.

3. Hadamard Transform

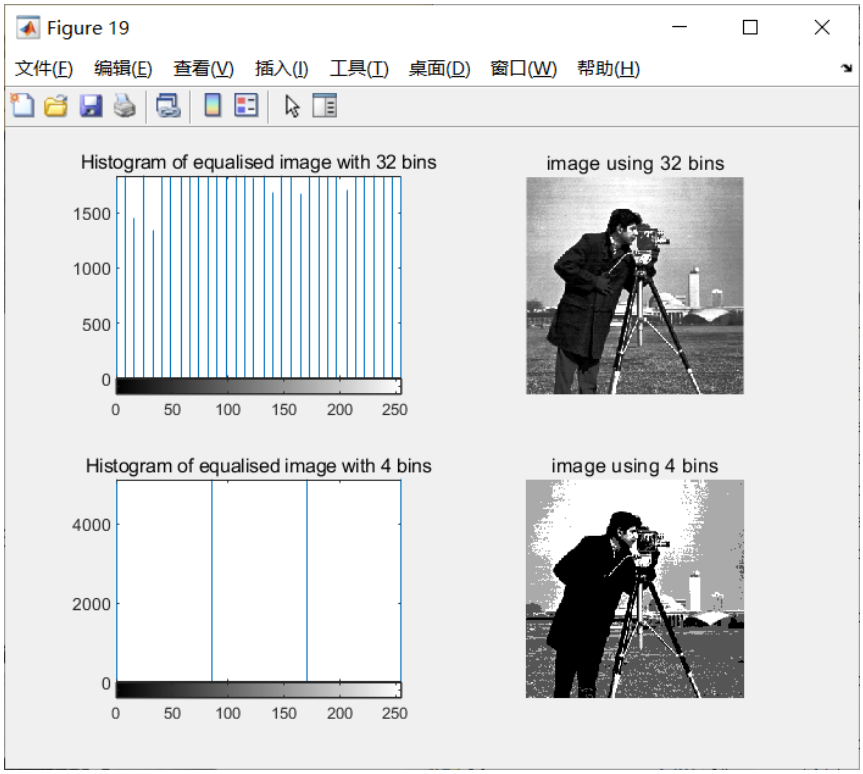
Here we use the 'cameraman.tif' as example:



For the advantages: The function of Hadamard transform is more efficiently in a digital environment than exponential basis functions of Fourier transform, because It only need real operation so that the complexity of computation can be reduced.

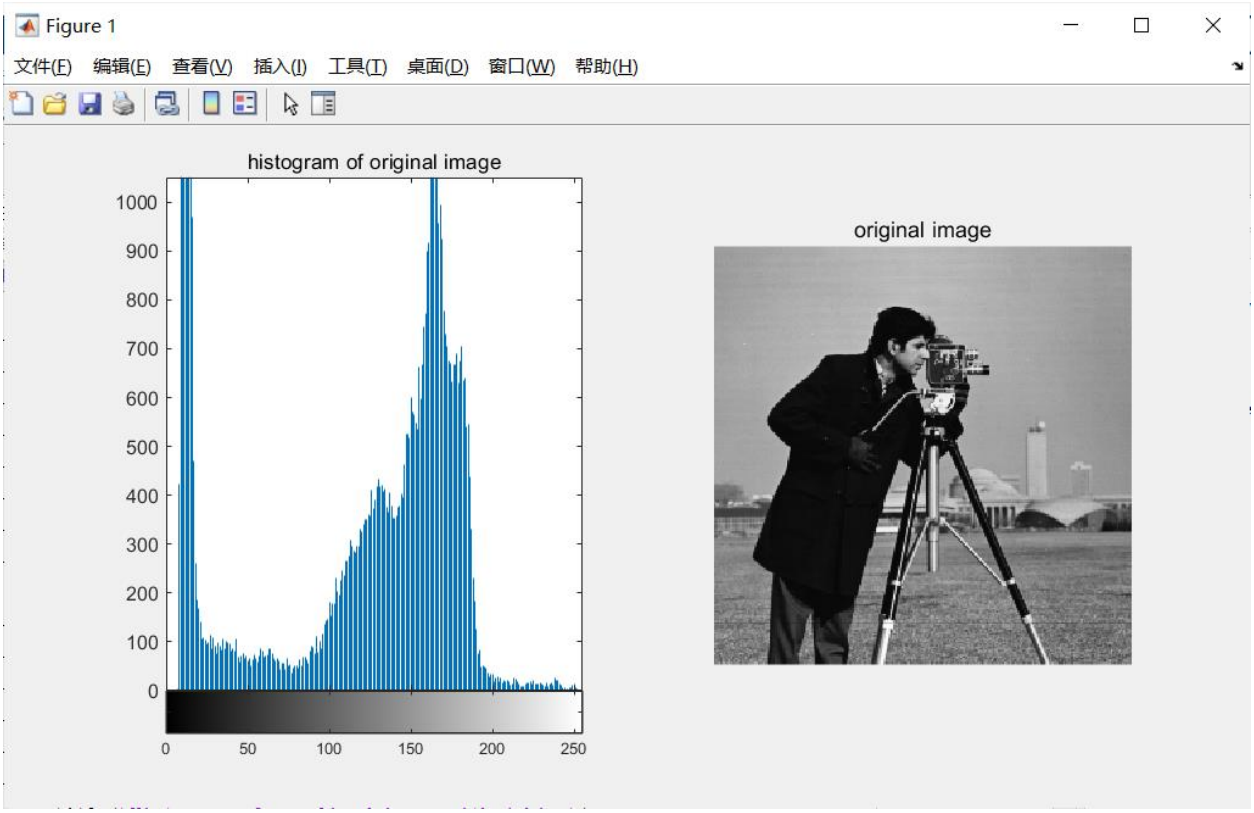
Part2. Image Enhancement

1. In this section, *"histeq"* function would be used and two histograms of equalized images would be created. The image "cameraman.tif" is used in this experiment.

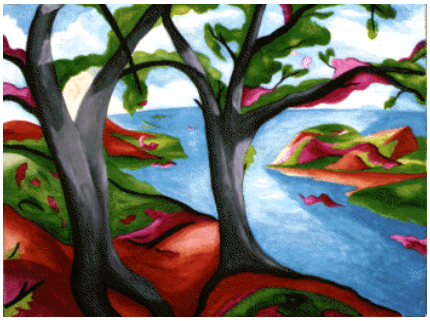


With the increasing of bin, the color contained in the picture increases, so the picture is clearer.

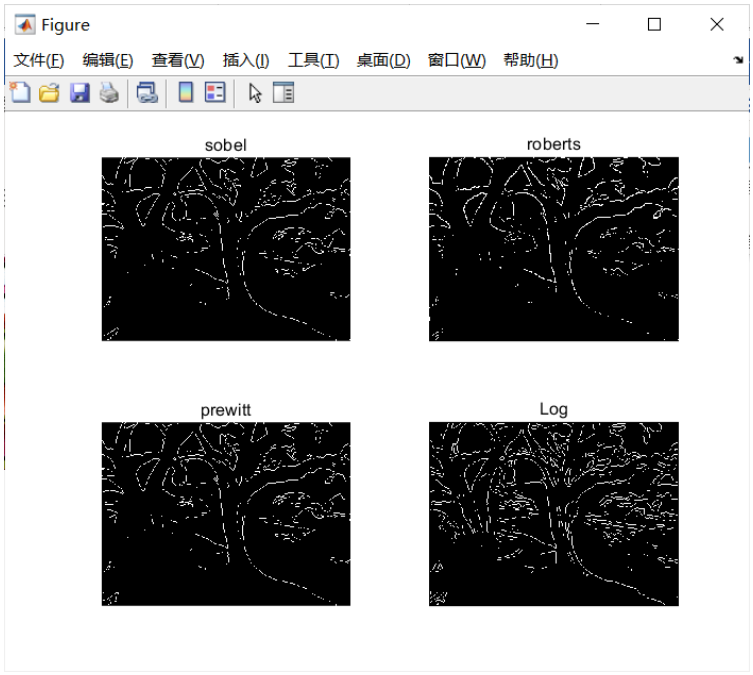
2. We use the cameraman as original image.



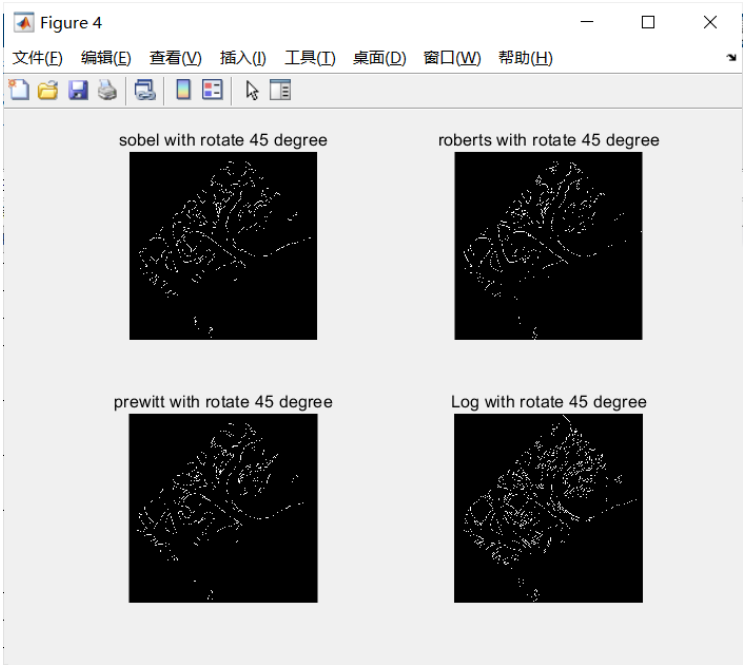
3. We use the image tree:



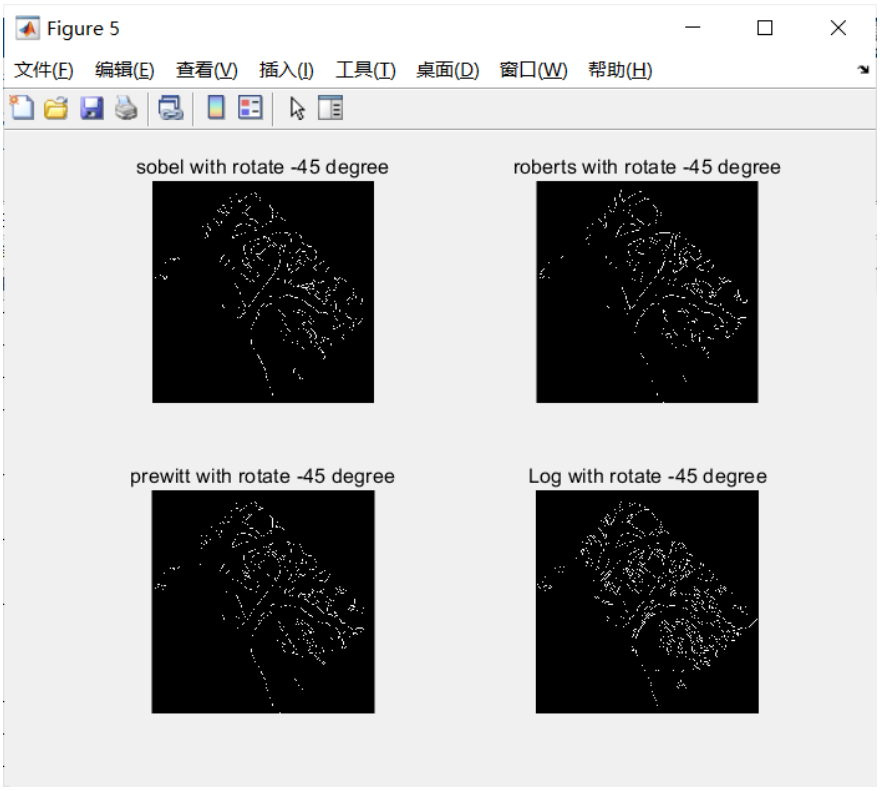
different methods to do edge detection:
we can see that, compared with other three methods, the ‘Log’ method can generate more edges and the white lines are denser



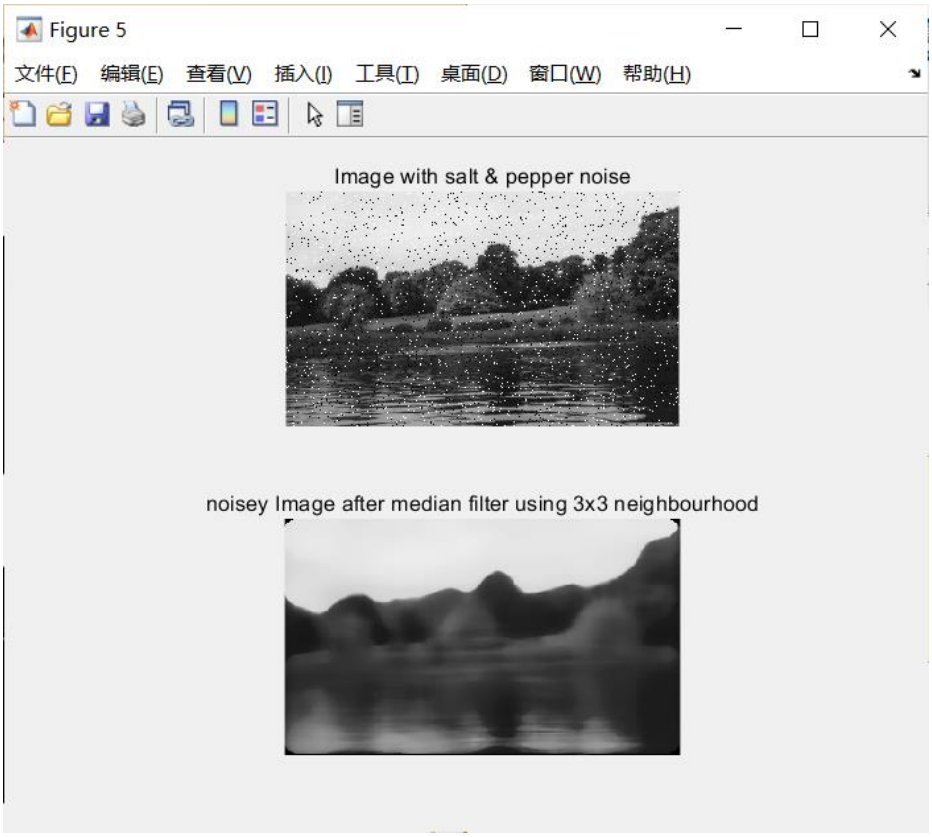
Rotate 45 degree:



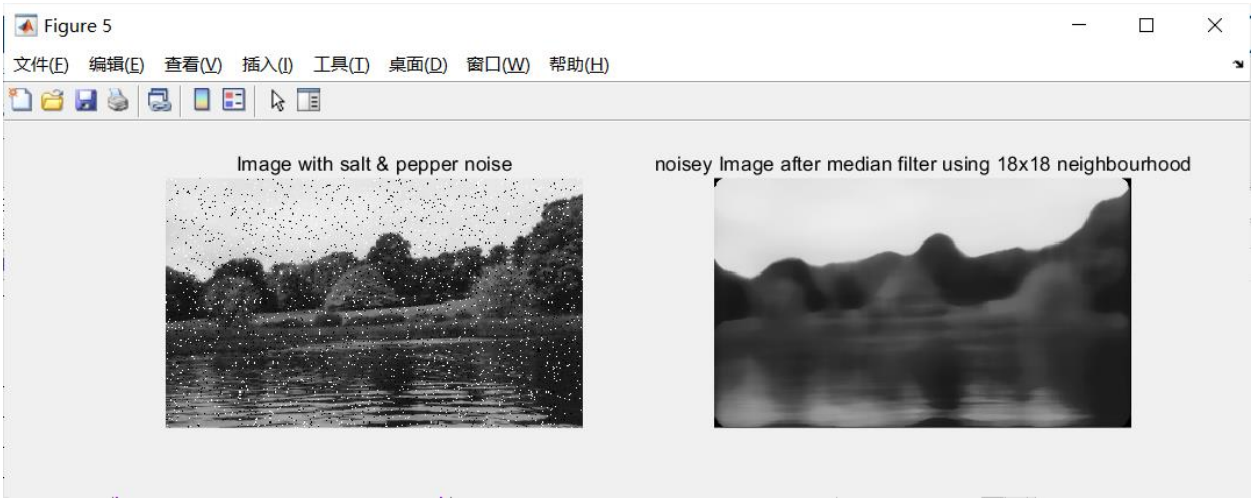
Rotate -45 degree:



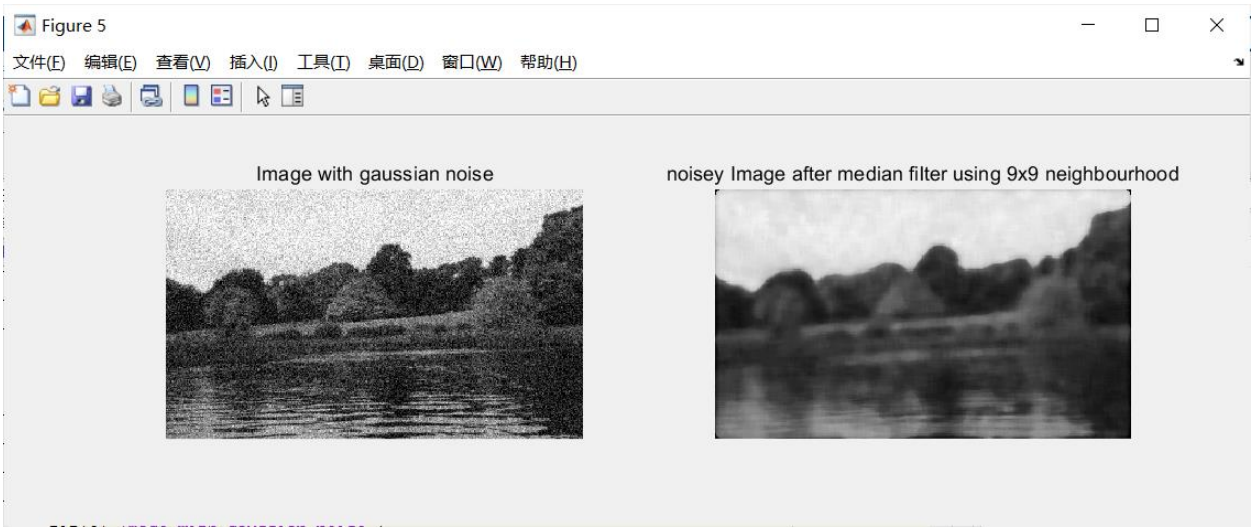
4. we use “autumn” image and add “salt and pepper” noise:
   image with “salt and pepper” noise and 3x3 neighbourhood median filter:



Use the same method as above to filter image but using 9x9 neighbourhood:



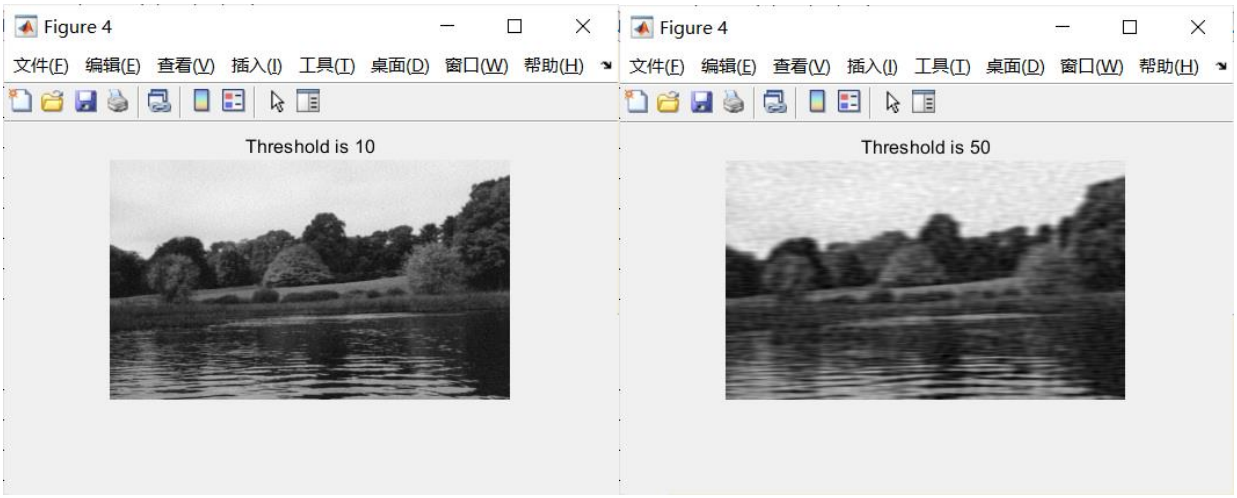
Then we use different noise characteristics, we use gaussian noise:



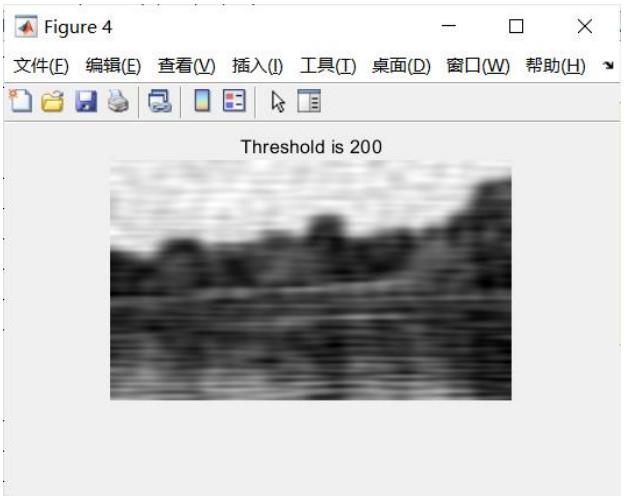
With the increasing of the neighbourhood of median filter, the filtered images become more and more blurred. Because The high frequencies of the original image are lost more with the increasing of the size of neighbourhood.

Part 3 Image compression

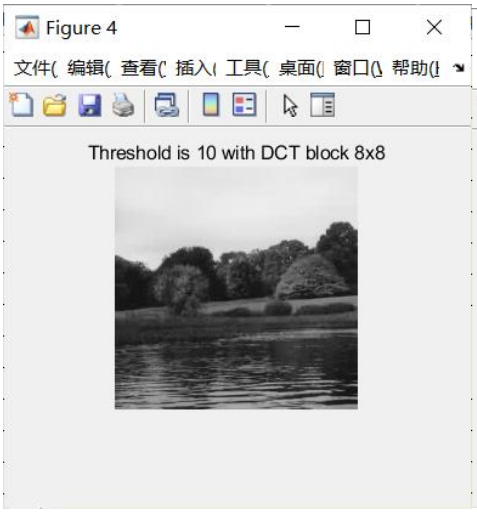
Result:

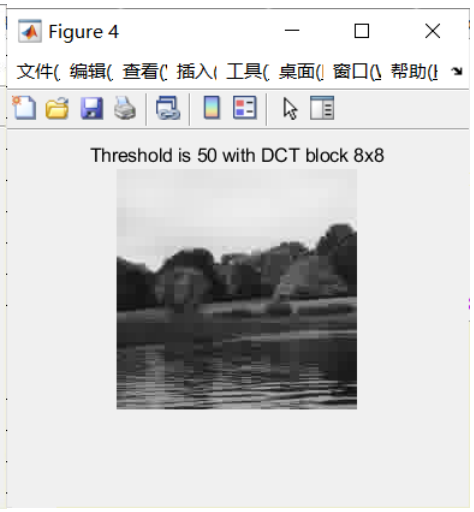


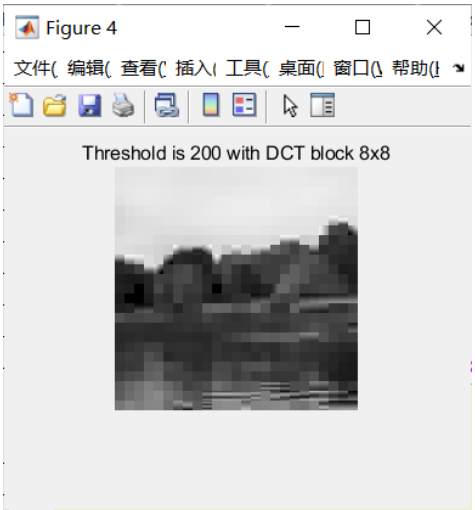
We can see that with the increasing of the threshold, we will lose more information of the image, so the image is getting fuzzy.



We use 8*8 block:





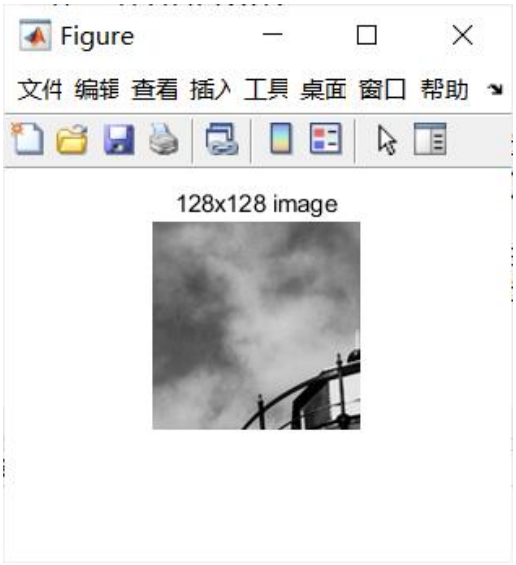


The size of block increased from the upper left corner to the lower right corner.

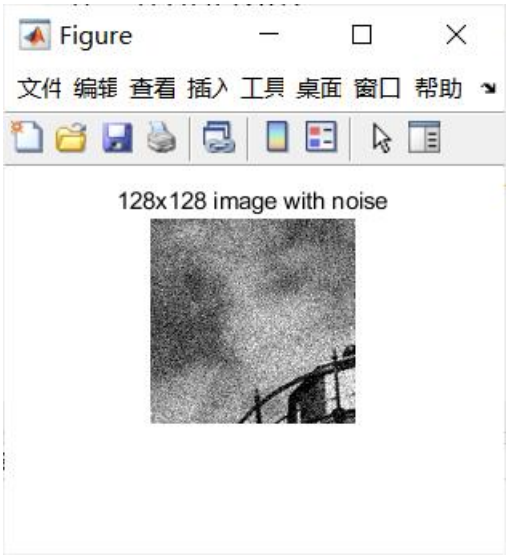
Part 4 Design Exercise

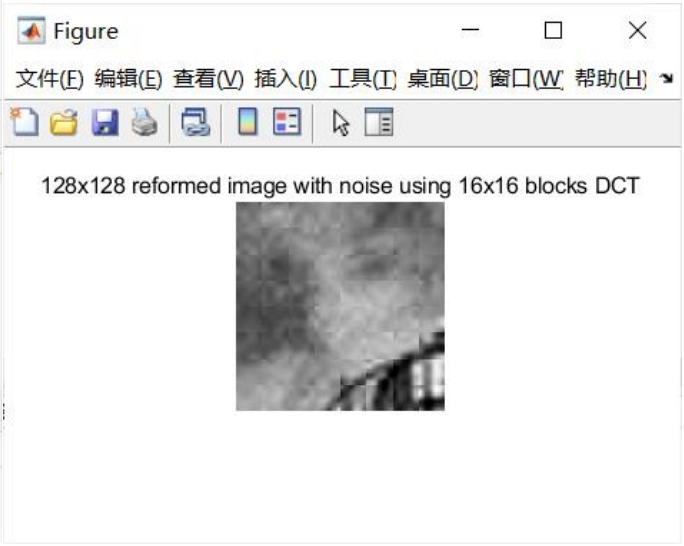
We use ‘lighthouse.png’ as example, and change it from RGB to gray:

Cut and adding noise:



Adding noise and then cut the image:



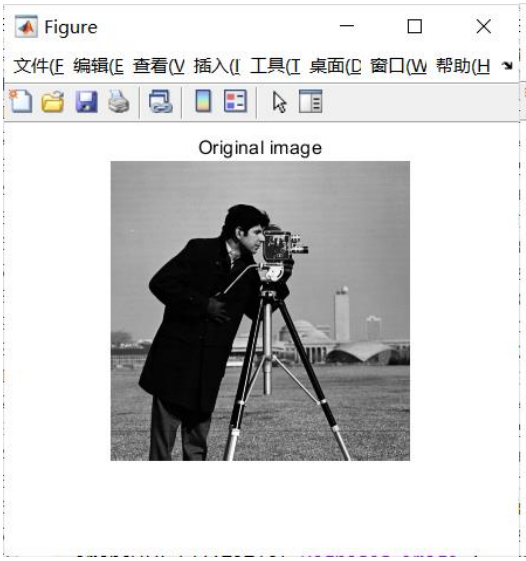


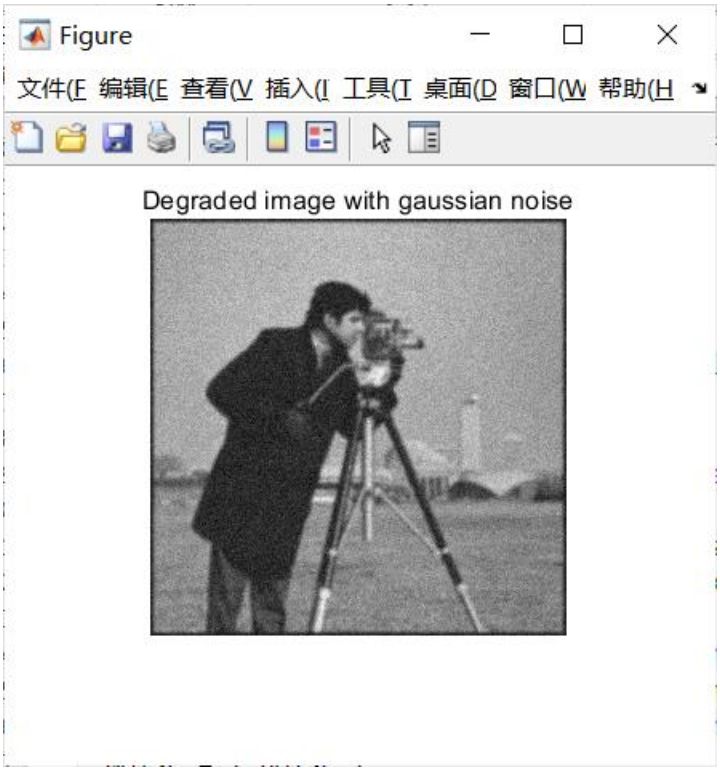
Part 5 Image Restoration

We take the ‘cameraman.tif’ as example.

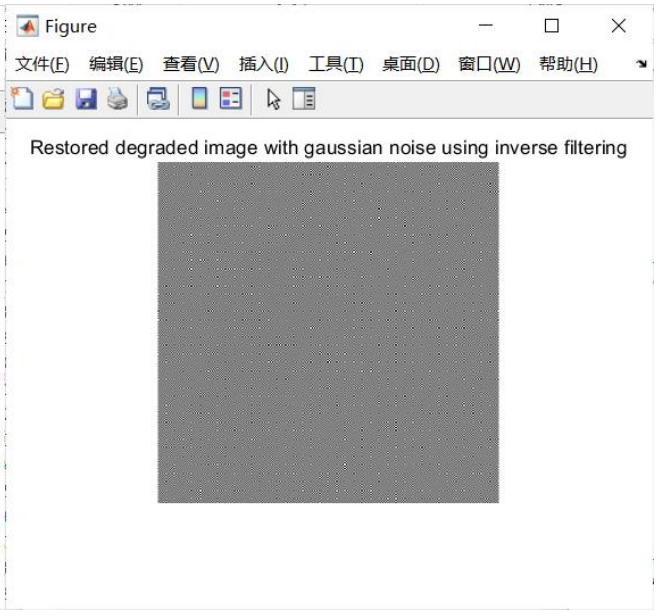
The first part is to degrade the image and add Gaussian noise.

(5* 5 Gaussian degradation)

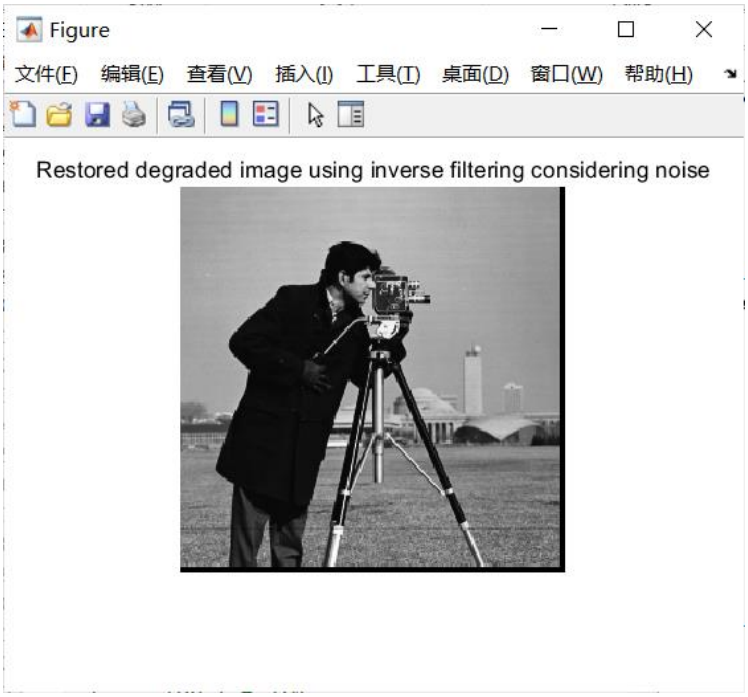




Then we use inverse Filter in both noiseless and noisy cases:



When we consider the noise in above right, we can get:



For Wiener Filter:

This below degraded image(7x7 gaussian degradation) with noise is our original image.

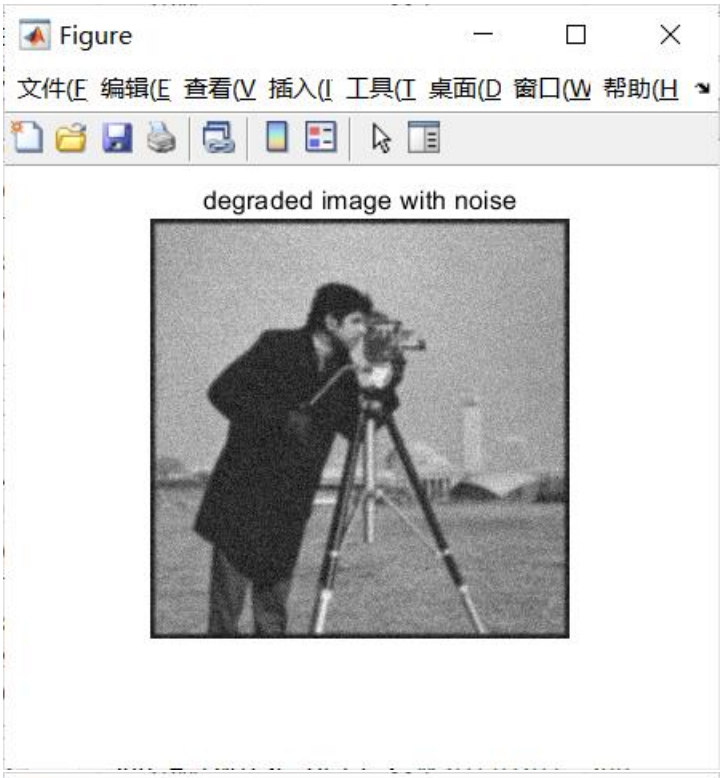


Figure
文件(F 编辑(E 查看(V 插入(I 工具(T 桌面(D 窗口(W 帮助(H
degraded image with noise

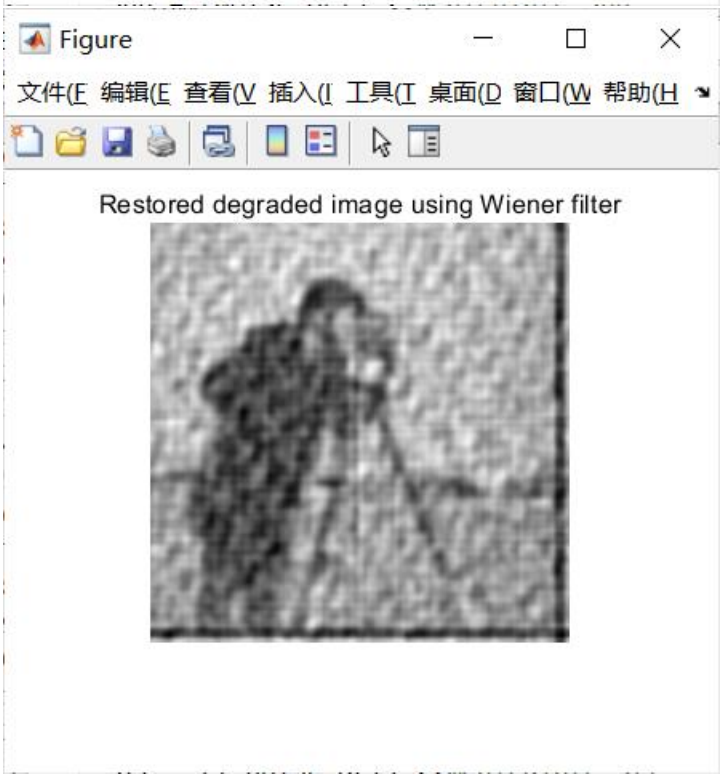


Figure
文件(F 编辑(E 查看(V 插入(I 工具(T 桌面(D 窗口(W 帮助(H
Restored degraded image using Wiener filter