

《数字图像处理》实验报告

实验名称 : 实验十二 图像复原

实验日期 : 2021.12.2

姓 名 : 王译

学 号 : 084519117

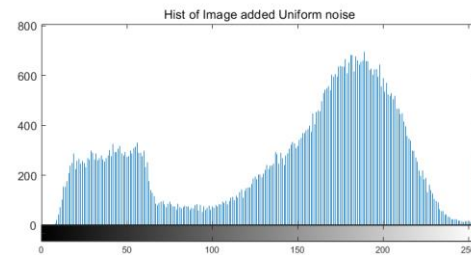
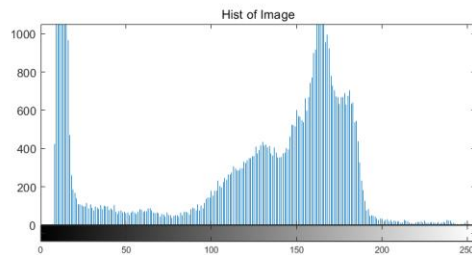
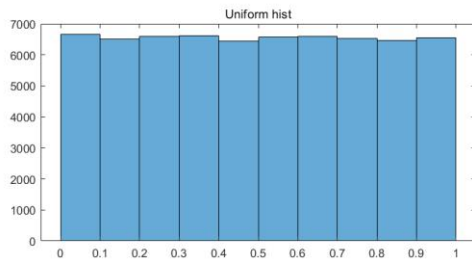
班 级 : 19 医信

成 绩 : _____

信息技术学院

南京中医药大学

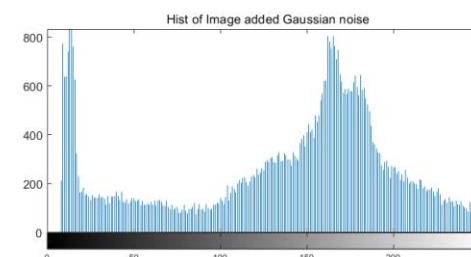
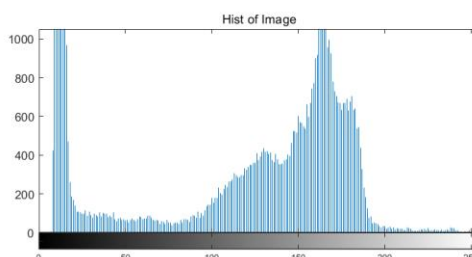
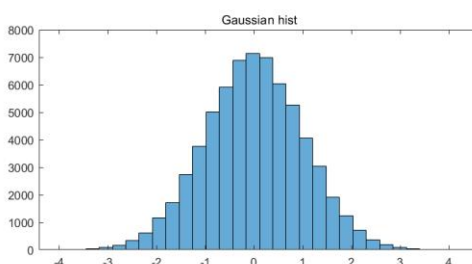
实验目的：
<p>数字图像恢复是数字图像处理的一个基本的和重要的课题，它是后期图像处理（分析和理解）的前提。图像在摄取、传输、储存的过程中不可避免地引起图像质量的下降（图像退化），图像恢复就是试图利用退化过程的先验知识使已退化的图像恢复本来面貌，即根据退化的原因，分析引起退化的环境因素，建立相应的数学模型，并沿着使图像降质的逆过程恢复图像。</p> <ol style="list-style-type: none"> 1. 了解图像复原的原理； 2. 掌握常用图像复原方法。
实验内容和要求
<p>建立一个名为“xxxxx 实验 12”的解决方案（xxxxx 为自己的学号）</p> <ol style="list-style-type: none"> 1. 应用 imnoise2（），imnoise3（）模拟各种噪声模型； 2. 打开一幅图像，应用 MATLAB 中工具箱函数 fspecial（）、imfilter（）、imnoise2（）等对图像作模糊化和添加噪声处理； 3. 验证算术均值、几何均值、调和均值、逆调和均值等空间域滤波法实现仅有噪声退化的图像复原（至少尝试三种不同噪声） 4. 验证带阻滤波器等频域滤波器实现仅有噪声退化的图像复原。 5. 参照课本和课堂讲解的例题中的方法对由退化函数加噪声共同作用下的退化图像进行复原。（注：实验中使用的图像数据量最好小一些） 6. 以上在进行图像复原操作之前请先对图像进行退化或加噪处理，记录和整理实验报告
运行结果（写清题号）
<p>描述实验的基本步骤，用数据和图片给出各个步骤中取得的实验结果和源代码，并进行必要的讨论，必须包括原始图像及其计算/处理后的图像。</p> <div data-bbox="252 1240 1353 1317" style="border: 1px solid black; padding: 5px;"> <pre>clc,clear I = imread('Fig0.tif'); sizeofI = size(I)</pre> </div> <p>1. 应用 imnoise2（），imnoise3（）模拟各种噪声模型； imnoise2（）可以实现 uniform，gaussian，salt & pepper，lognormal，rayleigh，exponential，erlang 七种不同分布的噪音，下面分别展示。</p> <p>(1) Uniform noise</p> <div data-bbox="252 1458 1353 1601" style="border: 1px solid black; padding: 5px;"> <pre>close all noise1 = imnoise2('uniform',sizeofI(1),sizeofI(2),0,1); subplot(2,2,1),histogram(noise1,10);title('Uniform hist'); subplot(2,2,2),imshow(I+uint8(noise1*50));title('Image added Uniform noise'); subplot(2,2,3),imhist(I),title('Hist of Image'); subplot(2,2,4),imhist(I+uint8(noise1*50)),title('Hist of Image added Uniform noise');</pre> </div>



均匀分布的直方图呈等高柱形，加噪后图像出现很多白点，在直方图上表示就是整体右移了。

(2) Gaussian noise

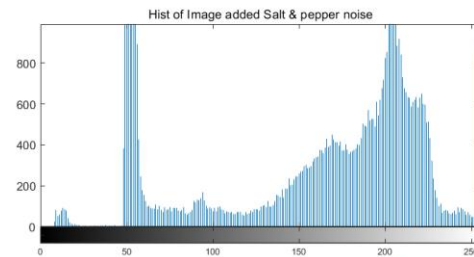
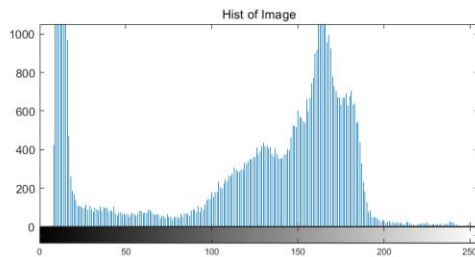
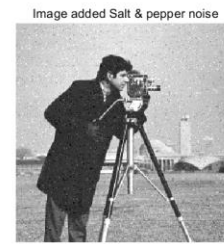
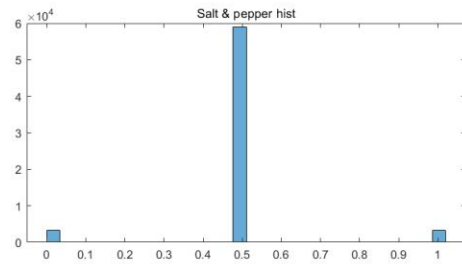
```
close all
noise2 = imnoise2('gaussian',sizeofl(1),sizeofl(2),0,1);
subplot(2,2,1),histogram(noise2,30);title('Gaussian hist');
subplot(2,2,2),imshow(1+uint8(noise2*20));title('Image added Gaussian noise');
subplot(2,2,3),imhist(I),title('Hist of Image');
subplot(2,2,4),imhist(1+uint8(noise2*50)),title('Hist of Image added Gaussian noise');
```



高斯分布的直方图呈中高外低，加噪后图像出现很多白点，直方图上像素值高的点累计变多。

(3) Salt & pepper

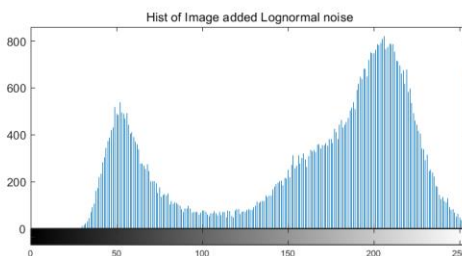
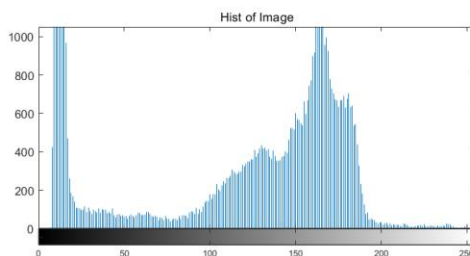
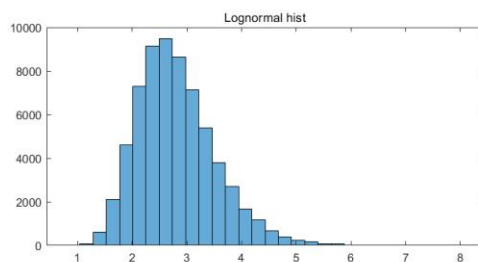
```
close all
noise3 = imnoise2('salt & pepper',sizeofl(1),sizeofl(2),0.05,0.05);
subplot(2,2,1),histogram(noise3,30);title('Salt & pepper hist');
subplot(2,2,2),imshow(1+uint8(noise3*80));title('Image added Salt & pepper noise');
subplot(2,2,3),imhist(I),title('Hist of Image');
subplot(2,2,4),imhist(1+uint8(noise3*80)),title('Hist of Image added Salt & pepper noise');
```



椒盐噪声的直方图呈针状，加噪后图像出现很多白点，直方图上整体右移。

(4) Lognormal

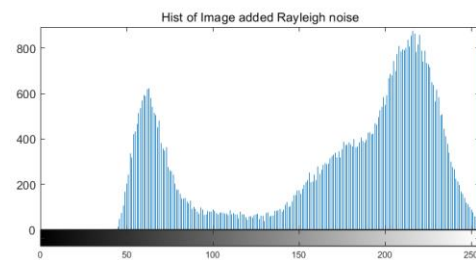
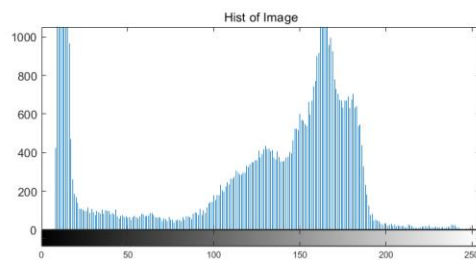
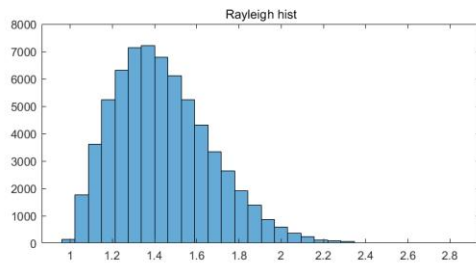
```
close all
noise4 = imnoise2('lognormal',sizeofl(1),sizeofl(2),1,0.25);
subplot(2,2,1),histogram(noise4,30);title('Lognormal hist');
subplot(2,2,2),imshow(I+uint8(noise4*15));title('Image added Lognormal noise');
subplot(2,2,3),imhist(I),title('Hist of Image');
subplot(2,2,4),imhist(I+uint8(noise4*15)),title('Hist of Image added Lognormal noise');
```



对数正太分布的直方图呈正太，但偏置，加噪后图像出现很多白点，直方图上整体右移。

(5) Rayleigh

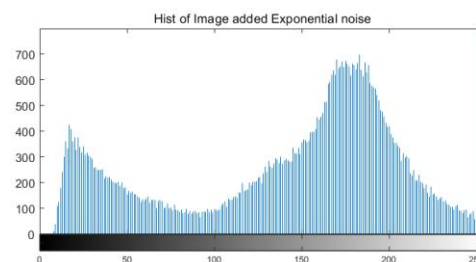
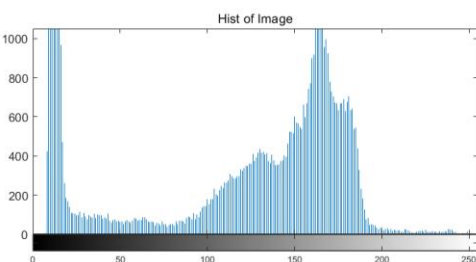
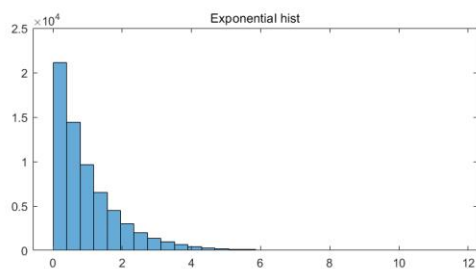
```
close all
noise5 = imnoise2('rayleigh',sizeofl(1),sizeofl(2),1,0.25);
subplot(2,2,1),histogram(noise5,30);title('Rayleigh hist');
subplot(2,2,2),imshow(I+uint8(noise5*35));title('Image added Rayleigh noise');
subplot(2,2,3),imhist(I),title('Hist of Image');
subplot(2,2,4),imhist(I+uint8(noise5*35)),title('Hist of Image added Rayleigh noise');
```



瑞利分布的直方图和对数正态差不多，加噪后图像出现很多白点，直方图上整体右移。

(6) Exponential

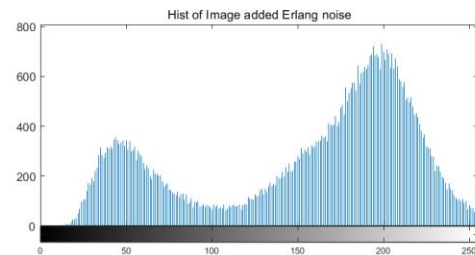
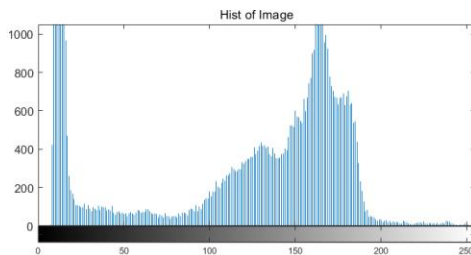
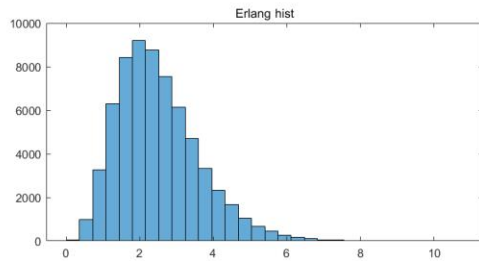
```
close all
noise6 = imnoise2('exponential',sizeofl(1),sizeofl(2),1,0.25);
subplot(2,2,1),histogram(noise6,30);title('Exponential hist');
subplot(2,2,2),imshow(1+uint8(noise6*25));title('Image added Exponential noise');
subplot(2,2,3),imhist(1),title('Hist of Image');
subplot(2,2,4),imhist(1+uint8(noise6*25)),title('Hist of Image added Exponential noise');
```



瑞利分布的直方图逐渐递减，加噪后图像出现很多白点，直方图上整体右移很明显。

(7) Erlang

```
close all
noise7 = imnoise2('erlang',sizeofl(1),sizeofl(2),2,5);
subplot(2,2,1),histogram(noise7,30);title('Erlang hist');
subplot(2,2,2),imshow(1+uint8(noise7*15));title('Image added Erlang noise');
subplot(2,2,3),imhist(1),title('Hist of Image');
subplot(2,2,4),imhist(1+uint8(noise7*15)),title('Hist of Image added Erlang noise');
```



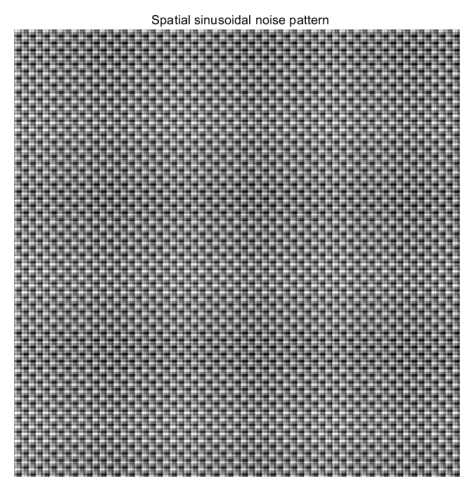
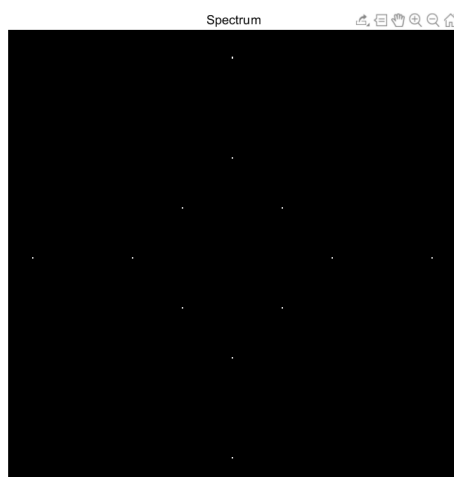
爱尔兰分布的直方图与瑞利分布相似,加噪后图像出现很多白点,直方图上整体右移很明显。

Function `[r, R, S]=imnoise3(M, N, C, A, B)`

产生一个大小为 $M \times N$ 的正弦噪声模型, r 代表所得到的周期噪声模型, R 代表所得周期噪声模型的 Fourier 变换, S 代表所得周期噪声模型的 Fourier 的频谱

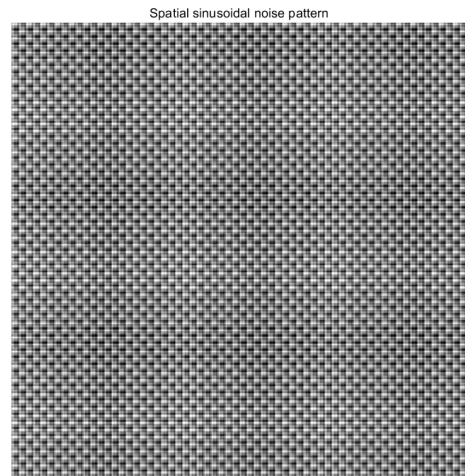
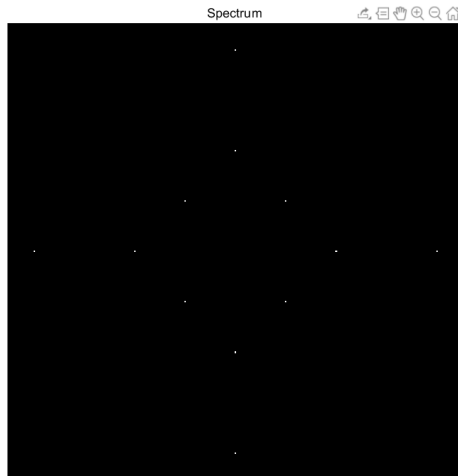
所得周期噪声模型的大小为 $M \times N$ 。 C 为冲击位置的坐标, A 是 $1 \times k$ 维向量包含振幅的冲击对, B 是由 $k \times 2$ 矩阵组成的冲击对。

```
clc,clear,close all
C = [0 64;0 128;32 32;64 0;128 0;-32 32];
[r,R,S]=imnoise3(512,512,C);
figure
subplot(1,2,1),imshow(S,[]),title('Spectrum') % 显示频谱
subplot(1,2,2),imshow(r,[]),title('Spatial sinusoidal noise pattern') % 显示空间正弦噪声模式
```



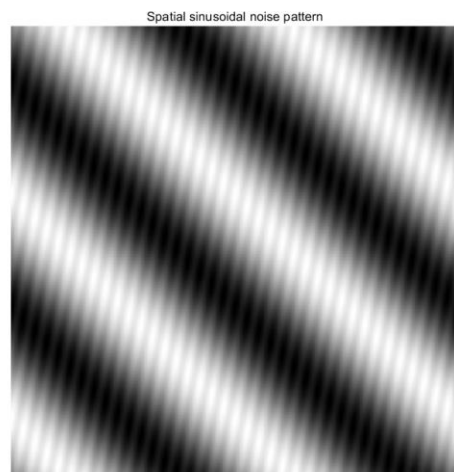
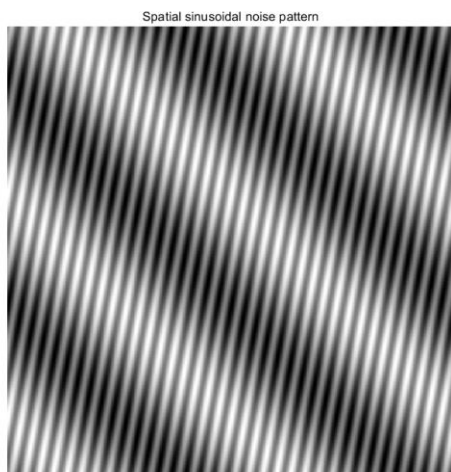
改变了冲击位置

```
C = [0 32;0 64;16 16;32 0;64 0;-16 16]; % 改变冲击位置, 观察频谱和空间正弦噪声模式变化
figure
subplot(1,2,1),imshow(S,[]),title('Spectrum')
subplot(1,2,2),imshow(r,[]),title('Spatial sinusoidal noise pattern')
```



使用非默认振幅向量 **A**, 观察空间正弦噪声模式变化

```
C = [6 32;-2 2]; % 改变冲击位置, 观察空间正弦噪声模式变化
[r,R,S]=imnoise3(512,512,C);
figure
subplot(1,2,1),imshow(r,[],),title('Spatial sinusoidal noise pattern')
A=[1 10]; % 使用非默认振幅向量 A, 观察空间正弦噪声模式变化
[r,R,S]=imnoise3(512,512,C,A);
subplot(1,2,2),imshow(r,[],),title('Spatial sinusoidal noise pattern')
```

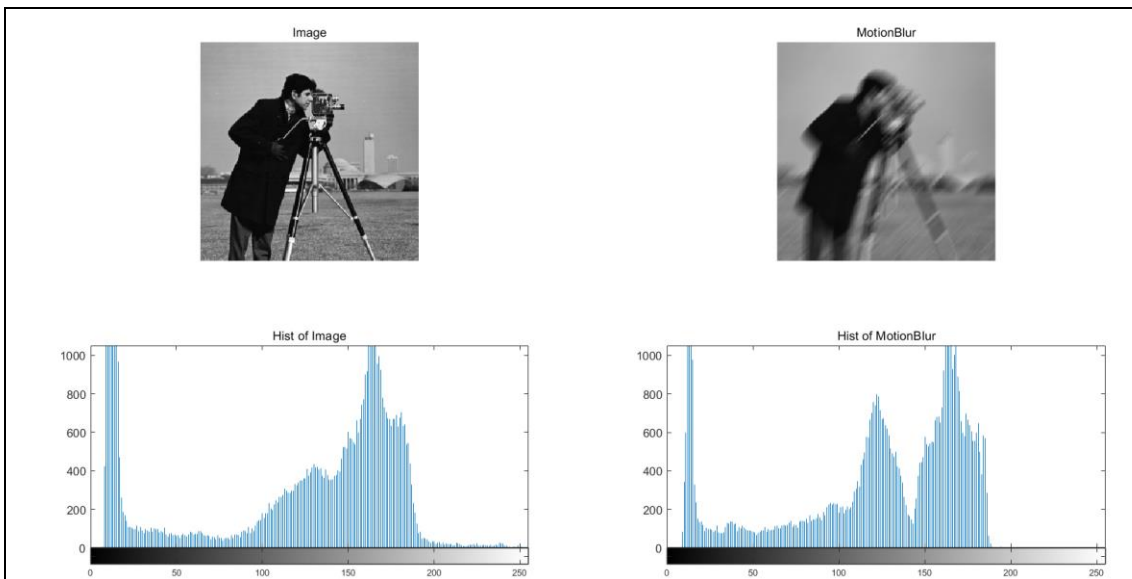


右图的频率更高

2. 打开一幅图像, 应用 MATLAB 中工具箱函数 `fspecial()`、`imfilter()`、`imnoise2()` 等对图像作模糊化和添加噪声处理;

(1) Motion

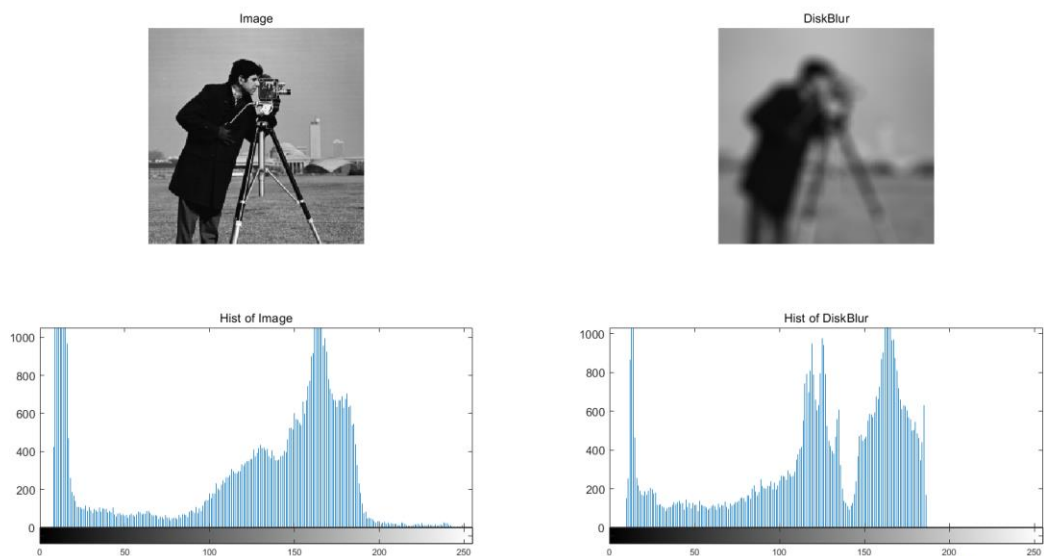
```
clc,clear,close all
I = imread('Fig0.tif');
hMotion = fspecial('motion',20,45);
MotionBlur = imfilter(I,hMotion,'replicate');
subplot(2,2,1),imshow(I),title('Image')
subplot(2,2,2),imshow(MotionBlur),title('MotionBlur')
subplot(2,2,3),imhist(I),title('Hist of Image');
subplot(2,2,4),imhist(MotionBlur),title('Hist of MotionBlur');
```



图像出现了明显的运动模糊，直方图也变化显著。

(2) Disk

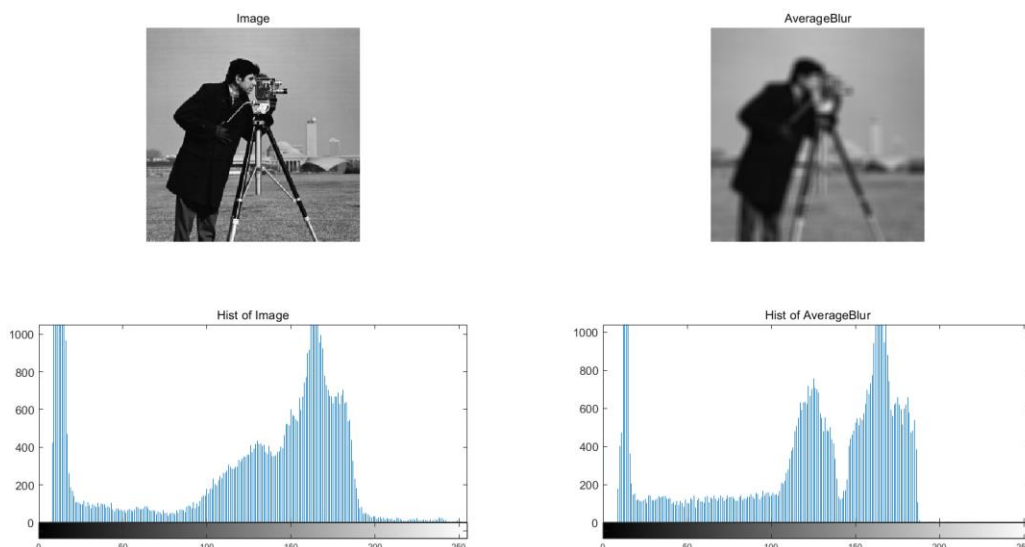
```
close all
hDisk = fspecial('disk',10);
DiskBlur = imfilter(I,hDisk,'replicate');
subplot(2,2,1),imshow(I),title('Image')
subplot(2,2,2),imshow(DiskBlur),title('DiskBlur')
subplot(2,2,3),imhist(I),title('Hist of Image');
subplot(2,2,4),imhist(DiskBlur),title('Hist of DiskBlur');
```



图像被明显模糊，直方图大值消失。

(3) Average

```
close all
hAverage = fspecial('average',10);
AverageBlur = imfilter(I,hAverage,'replicate');
subplot(2,2,1),imshow(I),title('Image')
subplot(2,2,2),imshow(AverageBlur),title('AverageBlur')
subplot(2,2,3),imhist(I),title('Hist of Image');
subplot(2,2,4),imhist(AverageBlur),title('Hist of AverageBlur');
```

图像被明显模糊，直方图大值消失。

`imnoise2()` 为图像添加噪声已经在问题 1 中得到解决。

3. 验证算术均值、几何均值、调和均值、逆调和均值等空间域滤波法实现仅有噪声退化的图像复原（至少尝试三种不同噪声）

根据算术均值公式：
$$\bar{f}(x,y) = \frac{1}{mn} \sum_{(p,q) \in W} f(p,q)$$
，几何均值公式：
$$\hat{f}(x,y) = \left[\prod_{(p,q) \in W} g(p,q) \right]^{\frac{1}{mn}},$$

$$\hat{f}(x,y) = \frac{mn}{\sum_{(p,q) \in W} \frac{1}{g(p,q)}} \quad \hat{f}(x,y) = \frac{\sum_{(p,q) \in W} g(p,q)^{k+1}}{\sum_{(p,q) \in W} g(p,q)^k}$$

调和均值公式：
$$\hat{f}(x,y) = \frac{mn}{\sum_{(p,q) \in W} \frac{1}{g(p,q)}}$$
，逆调和均值公式：

我自定义了一个图像复现函数，针对噪声退化，窗口大小为 3*3，调用该函数实现下述功能非常的便利：

```
function restored = ImgRestore(img, method, M, N, Q)
img = double(img);
restored = img;
imgSize = size(img);
% filter is a MxN matrix
offset = (M-1)/2;
switch(method)
case 'arithmetic mean'
for i = 2:imgSize(1)-1
for j = 2:imgSize(2)-1
temp = img(i-offset:i+offset,j-offset:j+offset); % Get the MxN matrix
restored(i,j) = sum(sum(temp))/(M*N);
end
end
case 'geometric mean'
for i = 2:imgSize(1)-1
for j = 2:imgSize(2)-1
temp = img(i-offset:i+offset,j-offset:j+offset); % Get the MxN matrix
restored(i,j) = prod(prod(temp))^(1/(M*N));
end
end
case 'harmonic mean'
for i = 2:imgSize(1)-1
for j = 2:imgSize(2)-1
temp = img(i-offset:i+offset,j-offset:j+offset); % Get the MxN matrix
restored(i,j) = (M*N)/(sum(sum(1./temp)));
end
end
case 'inverse harmonic mean'
for i = 2:imgSize(1)-1
for j = 2:imgSize(2)-1
```

```

        temp = img(i-offset:i+offset,j-offset:j+offset); % Get the MxN matrice
        up = sum(sum(temp.^(Q+1)));
        down = sum(sum(temp.^Q));
        restored(i,j) = up/down;
    end
end
end
restored = uint8(restored);
end

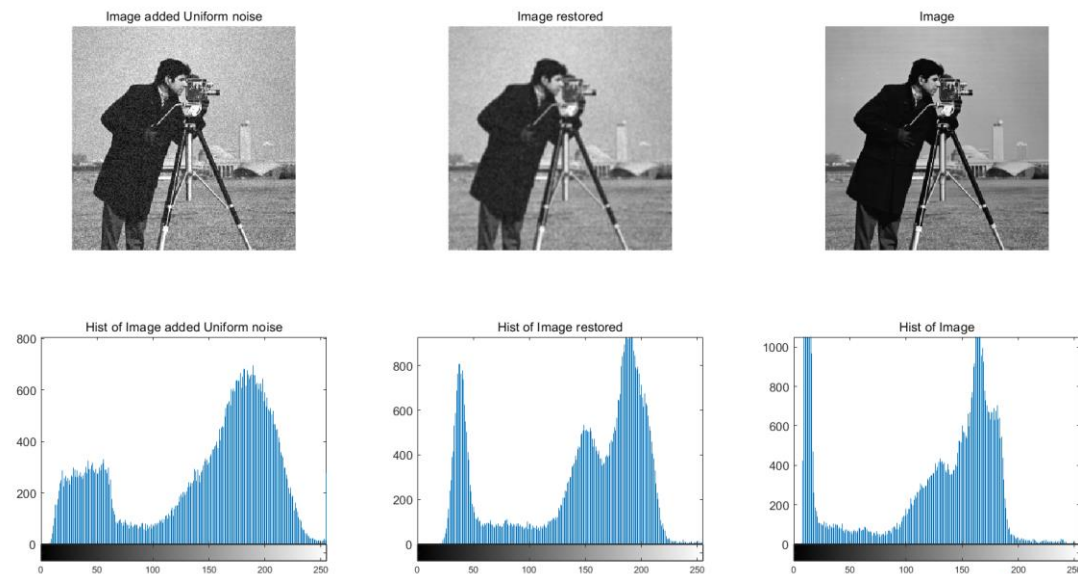
```

(1) 算数均值滤波，以均匀分布噪声图像为例

```

close all
imgNoise1 = I+uint8(noise1*50);
imgRes1 = ImgRestore(imgNoise1,'arithmetic mean',3,3);
subplot(2,3,1),imshow(imgNoise1),title('Image added Uniform noise')
subplot(2,3,2),imshow(imgRes1),title('Image restored')
subplot(2,3,3),imshow(I),title('Image');
subplot(2,3,4),imhist(imgNoise1),title('Hist of Image added Uniform noise');
subplot(2,3,5),imhist(imgRes1),title('Hist of Image restored');
subplot(2,3,6),imhist(I),title('Hist of Image');

```



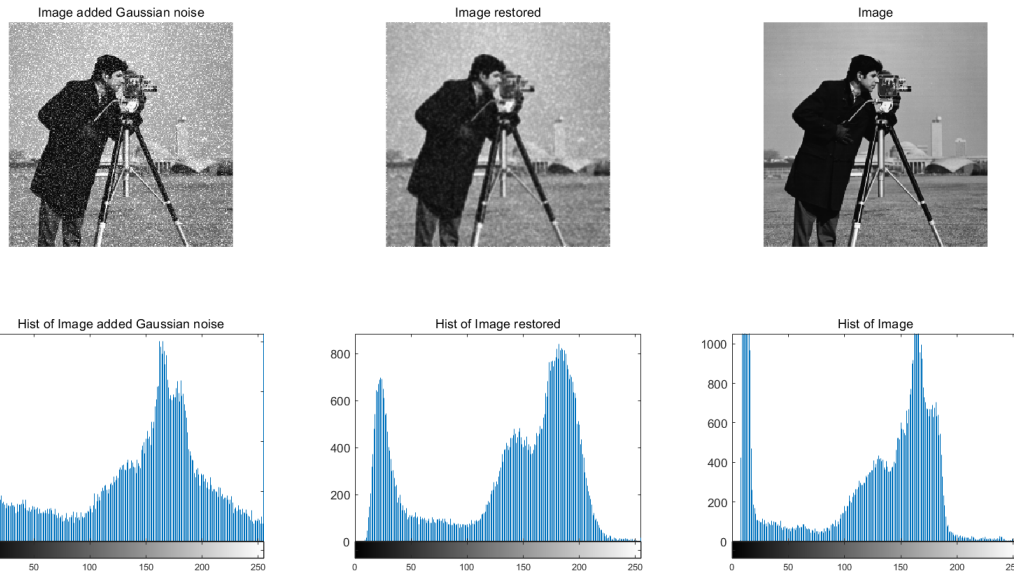
噪声图像在滤波后很明显的噪音点变少，但图像同时变得模糊，并且直方图与原图像的直方图较为相似。

(2) 几何均值滤波，以高斯分布噪声图像为例

```

close all
imgNoise2 = I+uint8(noise2*50);
imgRes2 = ImgRestore(imgNoise2,'geometric mean',3,3);
subplot(2,3,1),imshow(imgNoise2),title('Image added Gaussian noise')
subplot(2,3,2),imshow(imgRes2),title('Image restored')
subplot(2,3,3),imshow(I),title('Image');
subplot(2,3,4),imhist(imgNoise2),title('Hist of Image added Gaussian noise');
subplot(2,3,5),imhist(imgRes2),title('Hist of Image restored');
subplot(2,3,6),imhist(I),title('Hist of Image');

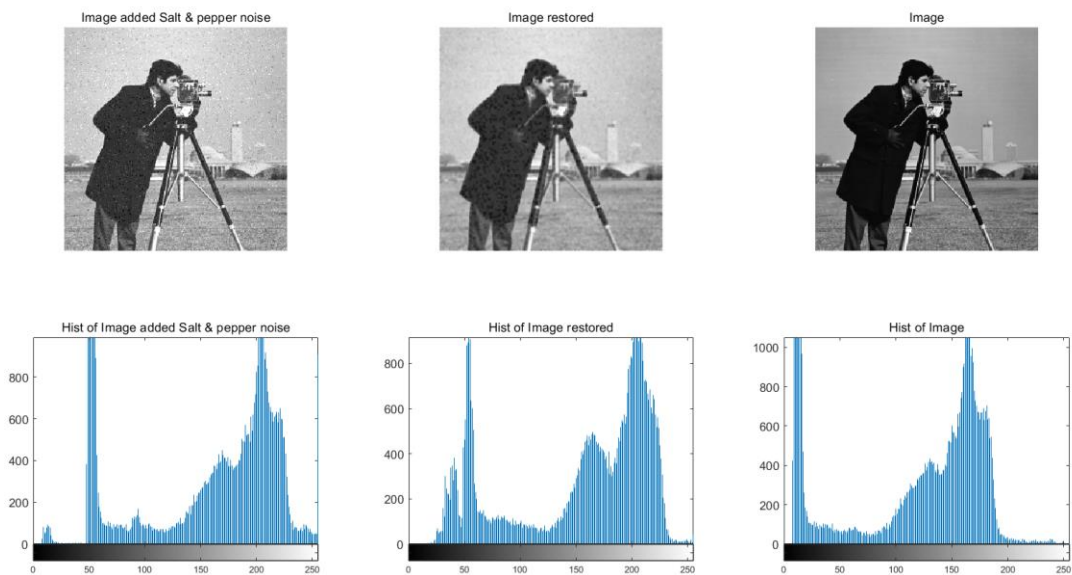
```



噪音图像在滤波后很明显的噪音点变少，但图像同时变得模糊，并且直方图与原图像的直方图较为相似。

(3) 调和均值滤波，以椒盐噪声图像为例

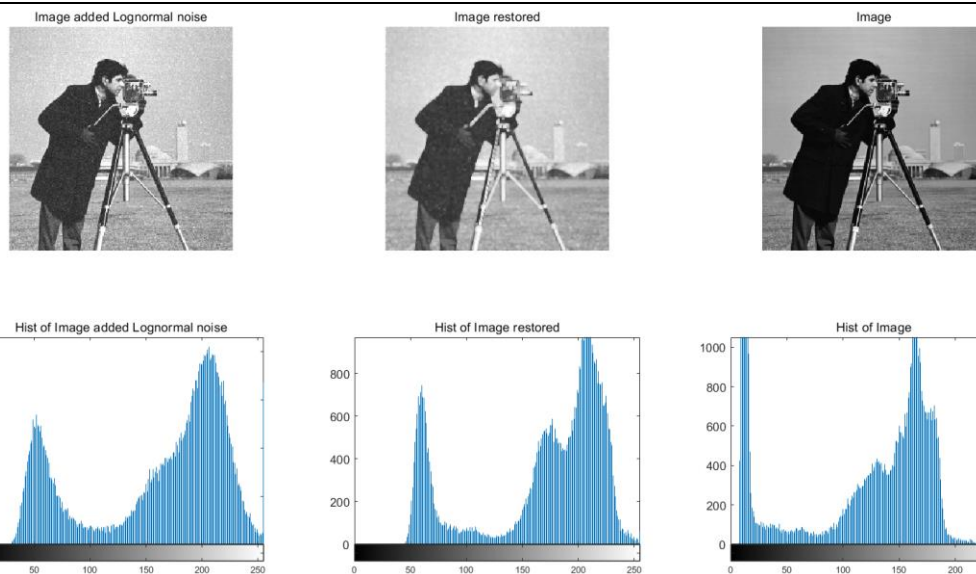
```
close all
imgNoise3 = I+uint8(noise3*80);
imgRes3 = ImgRestore(imgNoise3,'harmonic mean',3,3);
subplot(2,3,1),imshow(imgNoise3),title('Image added Salt & pepper noise')
subplot(2,3,2),imshow(imgRes3),title('Image restored')
subplot(2,3,3),imshow(I),title('Image');
subplot(2,3,4),imhist(imgNoise3),title('Hist of Image added Salt & pepper noise');
subplot(2,3,5),imhist(imgRes3),title('Hist of Image restored');
subplot(2,3,6),imhist(I),title('Hist of Image');
```



椒盐噪声很明显的被去除了，而且图像也没有模糊太多，效果较佳。

(4) 逆调和均值滤波，以对数正态噪声图像为例

```
close all
imgNoise4 = I+uint8(noise4*15);
imgRes4 = ImgRestore(imgNoise4,'inverse harmonic mean',3,3,3);
subplot(2,3,1),imshow(imgNoise4),title('Image added Lognormal noise')
subplot(2,3,2),imshow(imgRes4),title('Image restored')
subplot(2,3,3),imshow(I),title('Image');
subplot(2,3,4),imhist(imgNoise4),title('Hist of Image added Lognormal noise');
subplot(2,3,5),imhist(imgRes4),title('Hist of Image restored');
subplot(2,3,6),imhist(I),title('Hist of Image');
```



噪音图像在滤波后很明显的噪音点变少，但图像同时变得模糊，并且直方图与原图像的直方图较为相似。

4. 验证带阻滤波器等频域滤波器实现仅有噪声退化的图像复原。

(1) 理想带阻滤波器，以瑞利分布为例

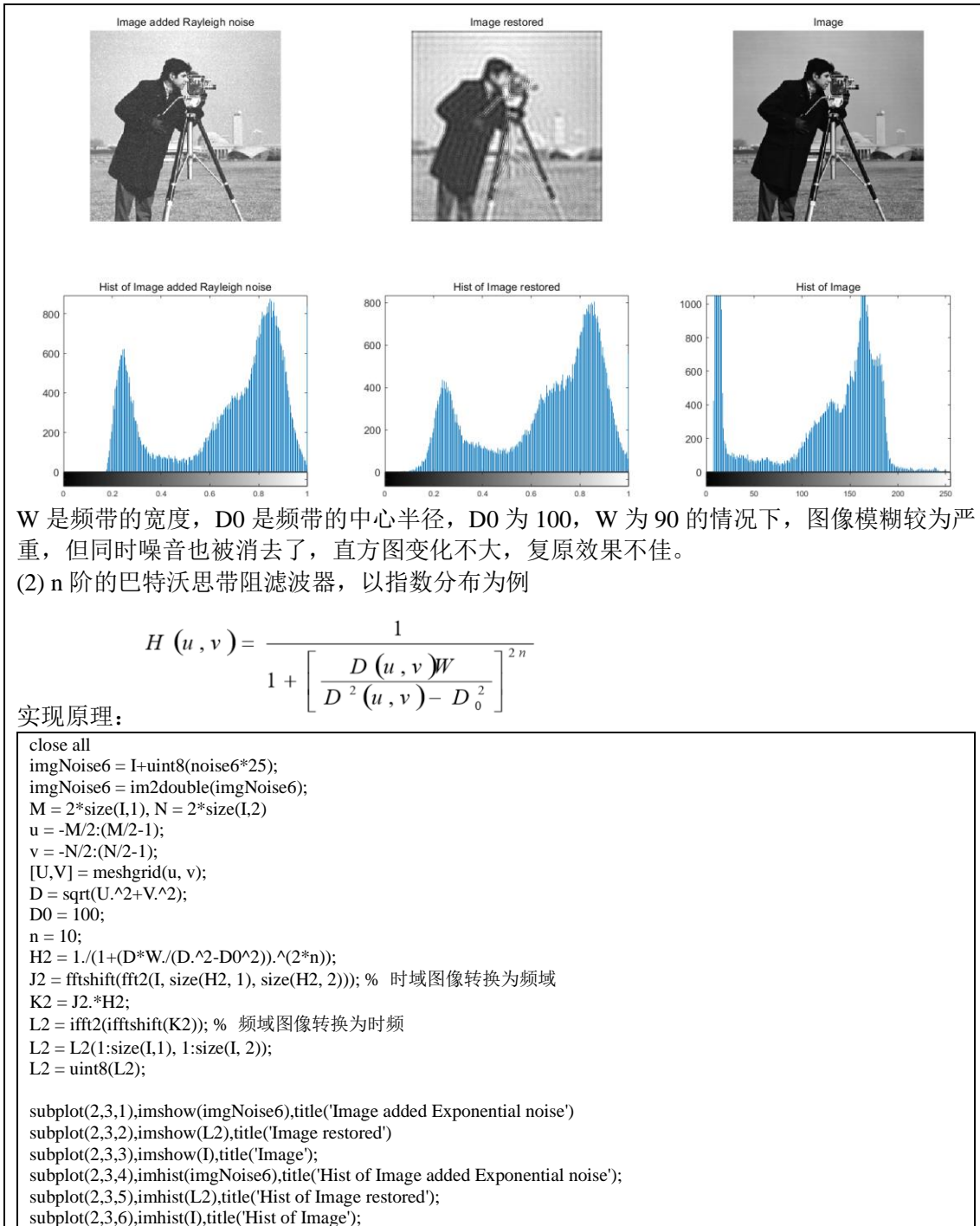
$$H(u, v) = \begin{cases} 1, D(u, v) < D_0 - \frac{W}{2} \\ 0, D_0 - \frac{W}{2} \leq D(u, v) \leq D_0 + \frac{W}{2} \\ 1, D(u, v) > D_0 + \frac{W}{2} \end{cases}$$

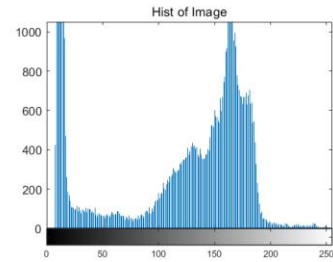
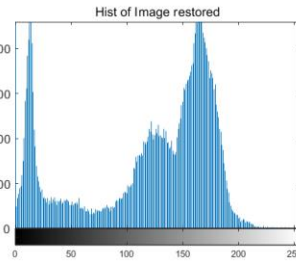
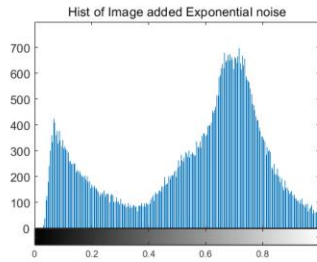
$$D(u, v) = \left[\left(u - \frac{M}{2} \right)^2 + \left(v - \frac{N}{2} \right)^2 \right]^{\frac{1}{2}}$$

实现原理：

```
close all
imgNoise5 = I+uint8(noise5*35);
imgNoise5 = im2double(imgNoise5);
M = 2*size(I,1), N = 2*size(I,2)
u = -M/2:(M/2-1);
v = -N/2:(N/2-1);
[U,V] = meshgrid(u, v);
D = sqrt(U.^2+V.^2);
D0 = 100;
W = 90;
Htemp1 = double(D<(D0-W/2));
Htemp2 = double(D>(D0+W/2));
H1 = Htemp1 + Htemp2;
J1 = fftshift(fft2(imgNoise5, size(H1, 1), size(H1, 2)));
K1 = J1.*H1; % 时域图像转换为频域
L1 = ifft2(ifftshift(K1));
L1 = L1(1:size(imgNoise5,1), 1:size(imgNoise5, 2)); % 频域图像转换为时域

subplot(2,3,1),imshow(imgNoise5),title('Image added Rayleigh noise')
subplot(2,3,2),imshow(L1),title('Image restored')
subplot(2,3,3),imshow(I),title('Image');
subplot(2,3,4),imhist(imgNoise5),title('Hist of Image added Rayleigh noise');
subplot(2,3,5),imhist(L1),title('Hist of Image restored');
subplot(2,3,6),imhist(I),title('Hist of Image');
```





可以看到几乎所有的噪音都被消去，复原效果极佳，模糊程度在可以接收的范围内，直方图非常相似，可以得出结论，巴特沃斯效果很好，阶为 10.

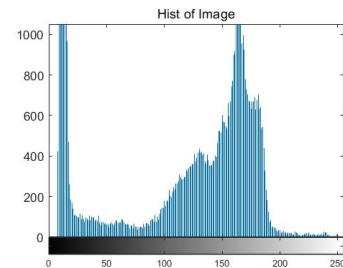
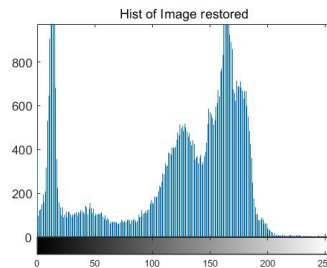
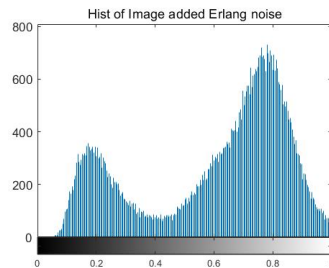
(3) 高斯带阻滤波器，以爱尔兰分布为例

$$H(u, v) = 1 - e^{-\frac{1}{2} \left[\frac{D^2(u, v) - D_0^2}{D(u, v)W} \right]^2}$$

实现原理:

```
close all
imgNoise7 = I+uint8(noise7*15);
imgNoise7 = im2double(imgNoise7);
M = 2*size(I,1), N = 2*size(I,2)
u = -M/2:(M/2-1);
v = -N/2:(N/2-1);
[U,V] = meshgrid(u, v);
D = sqrt(U.^2+V.^2);
D0 = 120;
W = 90;
H3 = 1 - exp(-1/2*((D.^2-D0^2)./(D*W)).^2);
J3 = fftshift(fft2(I, size(H3, 1), size(H3, 2))); % 时域图像转换为频域
K3 = J3.*H3;
L3 = ifft2(ifftshift(K3)); % 频域图像转换为时域
L3 = L3(1:size(I,1), 1:size(I, 2));
L3 = uint8(L3);

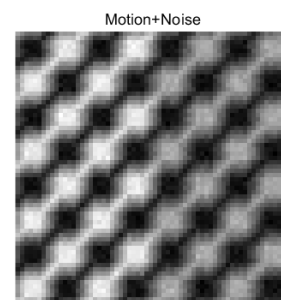
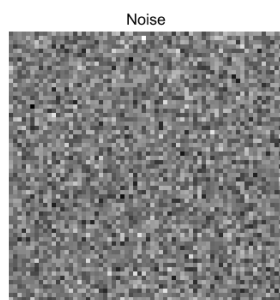
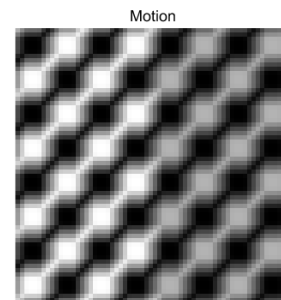
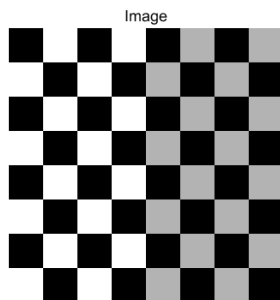
subplot(2,3,1),imshow(imgNoise7),title('Image added Erlang noise')
subplot(2,3,2),imshow(L3),title('Image restored')
subplot(2,3,3),imshow(I),title('Image');
subplot(2,3,4),imhist(imgNoise7),title('Hist of Image added Erlang noise');
subplot(2,3,5),imhist(L3),title('Hist of Image restored');
subplot(2,3,6),imhist(I),title('Hist of Image');
```

高斯带阻滤波器的效果甚至比巴特沃思滤波器的效果更好，图像几乎没有被模糊，噪音点几乎都已消失，直方图非常相似，效果极佳。

5. 参照课本和课堂讲解的例题中的方法对由退化函数加噪声共同作用下的退化图像进行复原。（注：实验中使用的图像数据量最好小一些）

```
clc,clear,close all
f=checkerboard(8);
PSF=fspecial('motion',7,45);
gb=imfilter(f,PSF,'circular');
noise=imnoise2('Gaussian',size(f,1),size(f,2),0,sqrt(0.001));
g=gb+noise;
subplot(2,2,1),imshow(pixeldup(f,8),[]),title('Image')
subplot(2,2,2),imshow(pixeldup(gb,8),[]),title('Motion')
subplot(2,2,3),imshow(pixeldup(noise,8),[]),title('Noise')
subplot(2,2,4),imshow(pixeldup(g,8),[]),title('Motion+Noise')
```



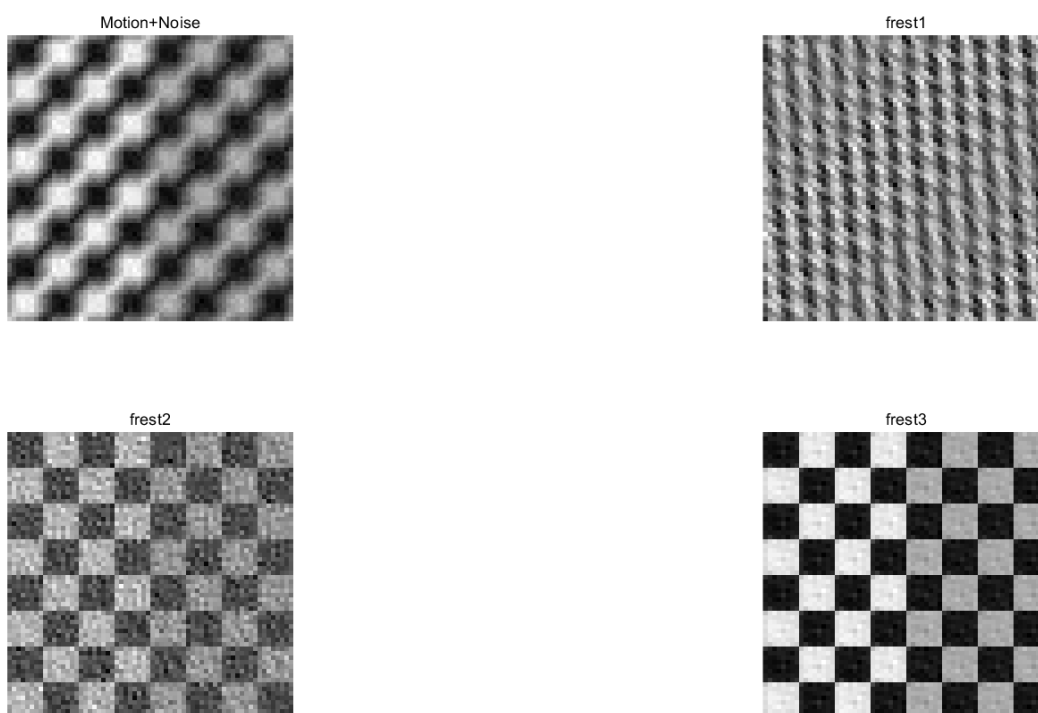
利用 motion 生成 PSF，对棋盘滤波后再加噪，得到第 4 图，即退化又有噪音的图像。

```
% Method 1
frest1 = deconvwnr(g,PSF);
% Method 2
Sn = abs(fft2(noise)).^2;
```

```

nA = sum(Sn(:))/numel(noise);
Sf = abs(fft2(f)).^2;
fA = sum(Sf(:))/numel(f);
R = nA/fA;
frest2 = deconvwnr(g,PSF,R);
% Method 3
NCORR = fftshift(real(ifft2(Sn)));
ICORR = fftshift(real(ifft2(Sf)));
frest3 = deconvwnr(g,PSF,NCORR,ICORR);
subplot(2,2,1),imshow(pixeldup(g,8),[]),title('Motion+Noise')
subplot(2,2,2),imshow(frest1,[],),title('frest1');
subplot(2,2,3),imshow(frest2,[],),title('frest2');
subplot(2,2,4),imshow(frest3,[],),title('frest3');

```



尝试用三种方法对图像进行复原，分别得到 frest1,2,3

方法 1 是根据得到的 PSF 点扩散函数利用 deconvwnr 函数直接对图像进行维纳逆滤波，如结果所见效果很差；

方法 2 是根据原图像与噪音图像，计算信噪比 R，再利用这一个比例恢复图像，可以看到补充了信噪比信息后，维纳滤波复原图像效果变得可以接收；

方法 3 是使用自相关函数 Sn，Sf 进行维纳滤波，可以看到结果很接近原始图像，但仍有一些噪声存在，这是因为原始图像和噪声图像都是已知的，因此我们可以估计到正确的参数，再通过维纳滤波器。

实验的体会与思考题

1. 常见的图像退化模型有哪些种类？
运动模糊退化模型,离焦模糊退化模型
2. 不同的 PSF 对复原效果有什么影响？
用维纳滤波的方法进行图像复原，不同的 PSF 参数值对复原效果影响较大。模糊函数可能是高斯函数或运动模糊函数等。教材中以运动模糊为例，对不同的 PSF 参数值产生的复原效果进行比较。实际应用过程中，真实的 PSF 通常是未知的，需要根据一定的先验知识对它进行估计，再将估计值作为参数进行图像复原。