

# 《数字图像处理》实验报告

实验名称 : 实验 10 图像复原

实验日期 : 2022.11.4

姓 名 : 傅康

学 号 : 084520126

班 级 : 医信 20

成 绩 :

信息技术学院

南京中医药大学

## 实验目的：

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。

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

建立一个名为“xxxxx 实验 10”的解决方案（xxxxx 为自己的学号）

1. 熟悉 `imnoise2()`，`imnoise3()` 函数，实践各种噪声模型的模拟；
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. 为图像添加噪声，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. 为图像添加周期噪声，并验证带阻滤波器或带通滤波器实现有周期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维纳滤波复原。（注：实验中使用的图像数据量最好小一些）

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）

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。

1. 熟悉 `imnoise2()`，`imnoise3()` 函数，实践各种噪声模型的模拟；

`imnoise2()` 可以实现 uniform, gaussian, salt & pepper, lognormal, rayleigh, exponential, erlang 七种不同分布的噪音

代码

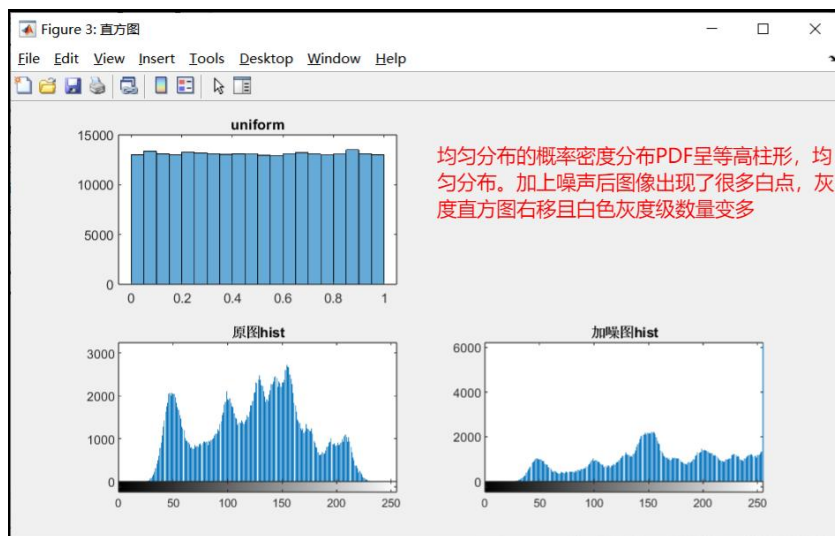
```

2      [M,N]=size(I);
3      figure,subplot(121),imshow(I),title('原图');
4      % imnoise2生成噪声模式本身,imnoise输出一副带噪声的图像
5
6      % uniform noise "均匀"区间 (A, B) 中的均匀随机数
7      noise1=imnoise2('uniform',M,N,0,1);
8      subplot(122),imshow(I+uint8(noise1)*100),title('加噪图');
9      figure("Name","直方图"),subplot(221),histogram(noise1,20),title('uniform');
10     subplot(223),imhist(I),title('原图hist');
11     subplot(224),imhist(I+uint8(noise1)*100),title('加噪图hist');
12     % gaussian noise 具有平均值A和标准偏差B的高斯随机数。默认值为A=0, B=1。
13     noise2=imnoise2('gaussian',M,N,0,1);
14     subplot(122),imshow(I+uint8(noise2)*50),title('加噪图');
15     figure("Name","直方图"),subplot(221),histogram(noise2,20),title('gaussian');
16     subplot(223),imhist(I),title('原图hist');
17     subplot(224),imhist(I+uint8(noise2)*50),title('加噪图hist');
18     % salt & pepper noise 椒盐噪声,默认Pa(盐振幅)=Pb(胡椒振幅)=A=B=0.05,
19     % 噪声值为0,Pa=A;噪声值为1, PB=B;均为0.5则没有分配噪声
20     noise3=imnoise2('salt & pepper',M,N,0.05,0.05);
21     subplot(122),imshow(I+uint8(noise3)*50),title('salt & pepper加噪图');
22     figure("Name","直方图"),subplot(221),histogram(noise3,10),title('salt & pepper');
23     subplot(223),imhist(I),title('原图hist');
24     subplot(224),imhist(I+uint8(noise3)*50),title('加噪图hist');
25     % lognormal noise 具有偏移A和形状参数B的对数正态数。默认值为A=1和B=0.25。
26     noise4=imnoise2('lognormal',M,N,1,0.25);
27     subplot(122),imshow(I+uint8(noise4)*20),title('lognormal加噪图');
28     figure("Name","直方图"),subplot(221),histogram(noise4,10),title('lognormal');
29     subplot(223),imhist(I),title('原图hist');
30     subplot(224),imhist(I+uint8(noise4)*20),title('加噪图hist');
31     % rayleigh noise 参数为A和B的瑞利噪声,服从瑞利分布,默认值为A=0和B=1。
32     noise5=imnoise2('rayleigh',M,N,0,1);
33     subplot(122),imshow(I+uint8(noise5)*50),title('rayleigh加噪图');
34     figure("Name","直方图"),subplot(221),histogram(noise5,10),title('rayleigh');
35     subplot(223),imhist(I),title('原图hist');
36     subplot(224),imhist(I+uint8(noise5)*50),title('加噪图hist');
37     % exponential noise 带参数A的指数随机数。默认值为A=1。
38     noise6=imnoise2('exponential',M,N,1);
39     subplot(122),imshow(I+uint8(noise6)*50),title('exponential加噪图');
40     figure("Name","直方图"),subplot(221),histogram(noise6,10),title('exponential');
41     subplot(223),imhist(I),title('原图hist');
42     subplot(224),imhist(I+uint8(noise6)*50),title('加噪图hist');
43     % erlang noise B必须是正整数。默认值为A=2和B=5。|
44     % Erlang随机数近似为B个指数随机数之和。
45     noise7=imnoise2('erlang',M,N,2,5);
46     subplot(122),imshow(I+uint8(noise7)*20),title('erlang加噪图');
47     figure("Name","直方图"),subplot(221),histogram(noise7,10),title('erlang');
48     subplot(223),imhist(I),title('原图hist');
49     subplot(224),imhist(I+uint8(noise7)*20),title('加噪图hist');
50
51
52
53
54

```

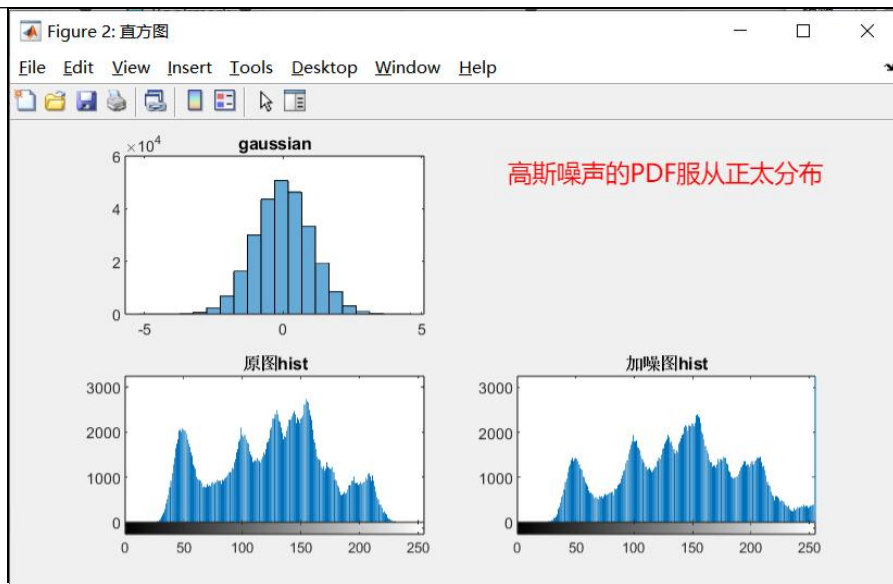
噪声类型模拟效果

## ①Uniform“均匀”区间（A，B）中的均匀噪声

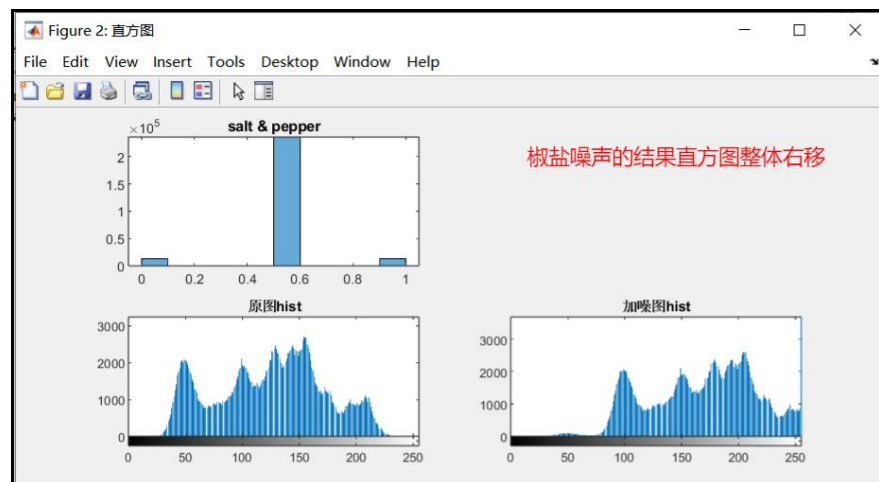
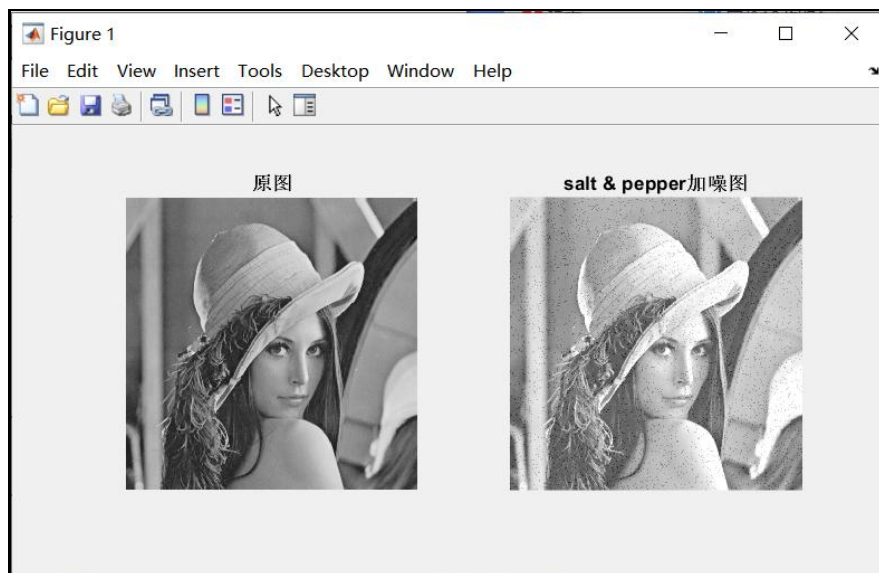


## ②Gaussian



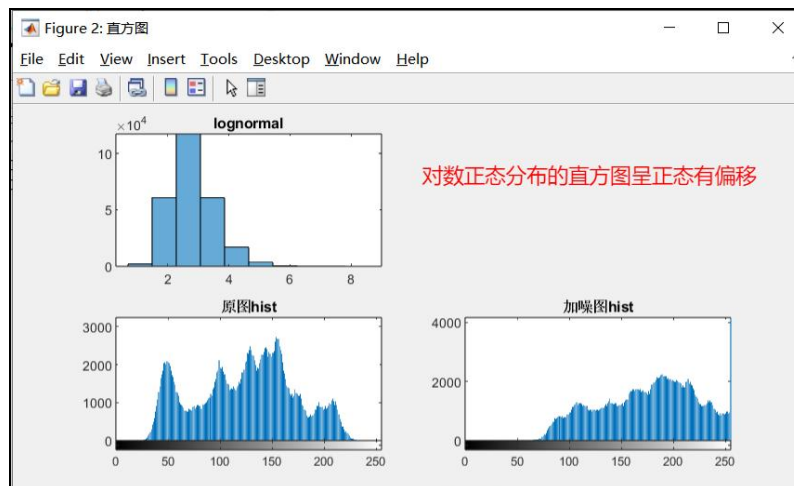
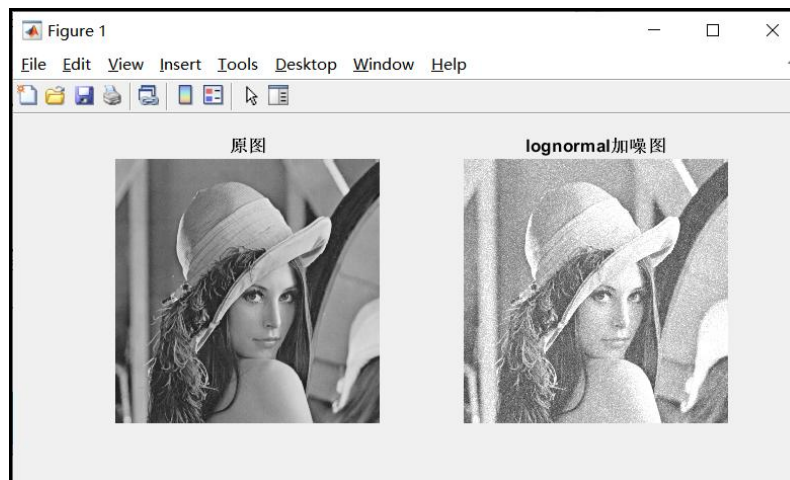


### ③Salt & pepper

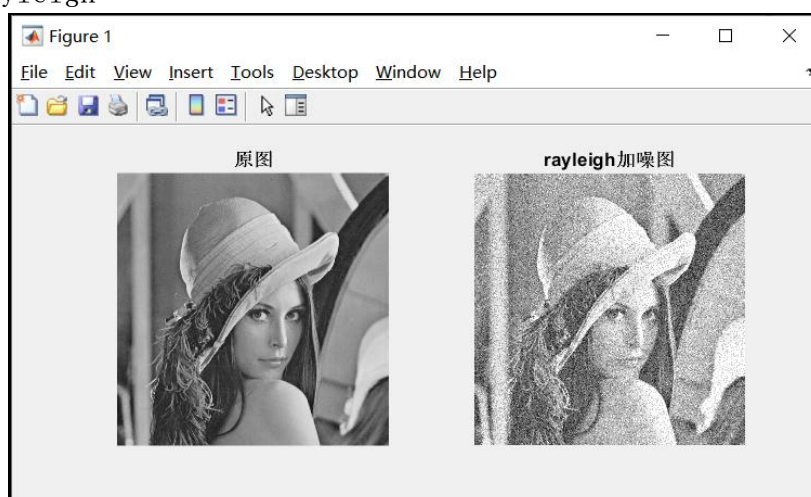


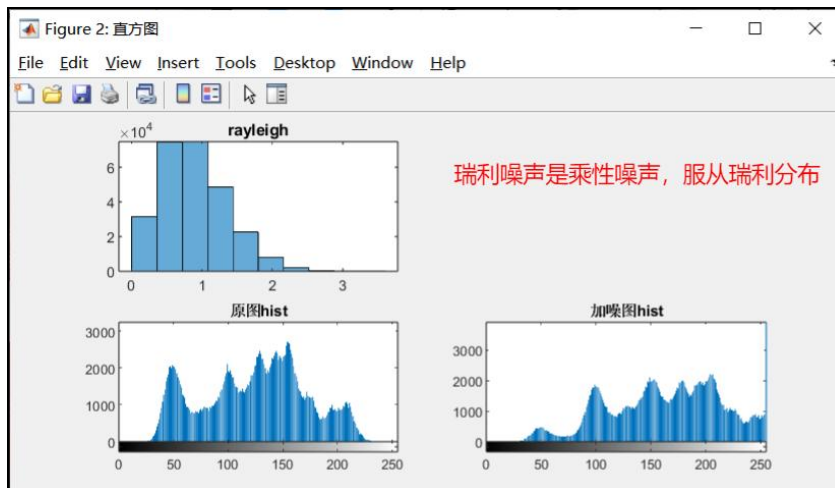


#### ④Lognormal

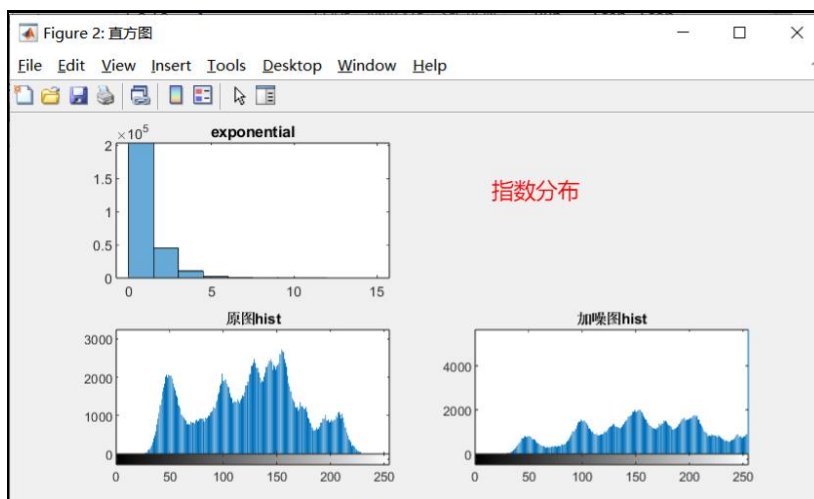
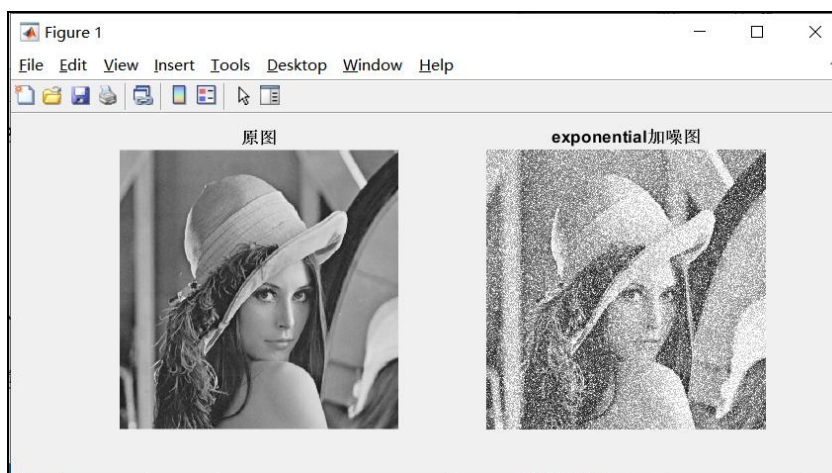


#### ⑤Rayleigh

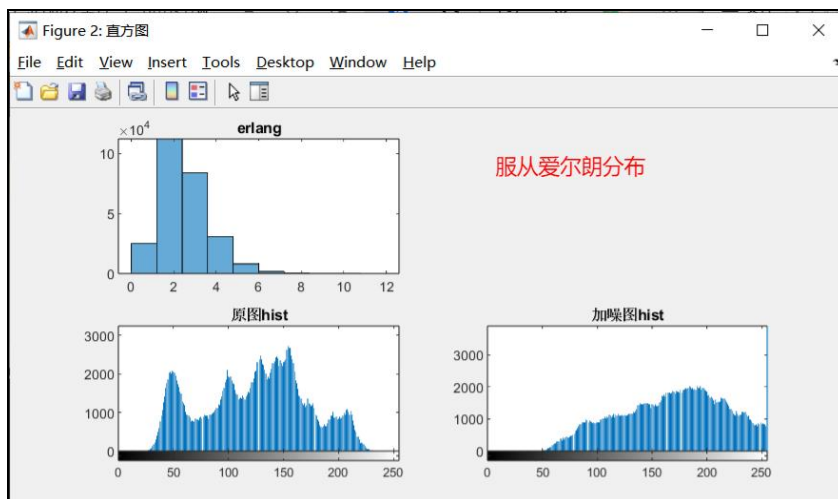
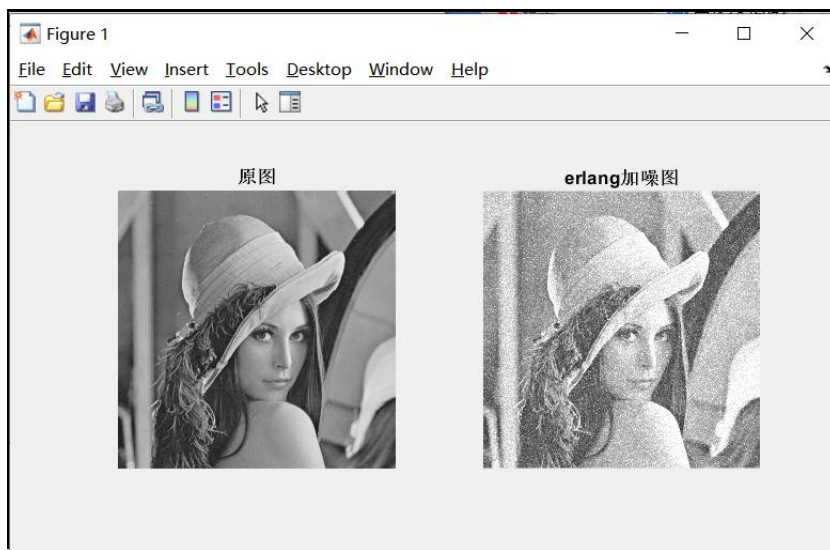




## ⑥Exponential



## ⑦Erlang



### 几种噪声的运用

- 高斯噪声源于电子电路噪声和由低照明度或高温带来的传感器噪声
- 瑞利噪声描述图像范围内特征化噪声
- 伽马分布和指数分布用于激光成像噪声
- 均匀密度分布作为模拟随机数产生器的基础
- 脉冲噪声生成于成像中的短暂停留中，如错误的开关操作

**imnoise3()** 函数 周期噪声

## 函数imnoise3的定义

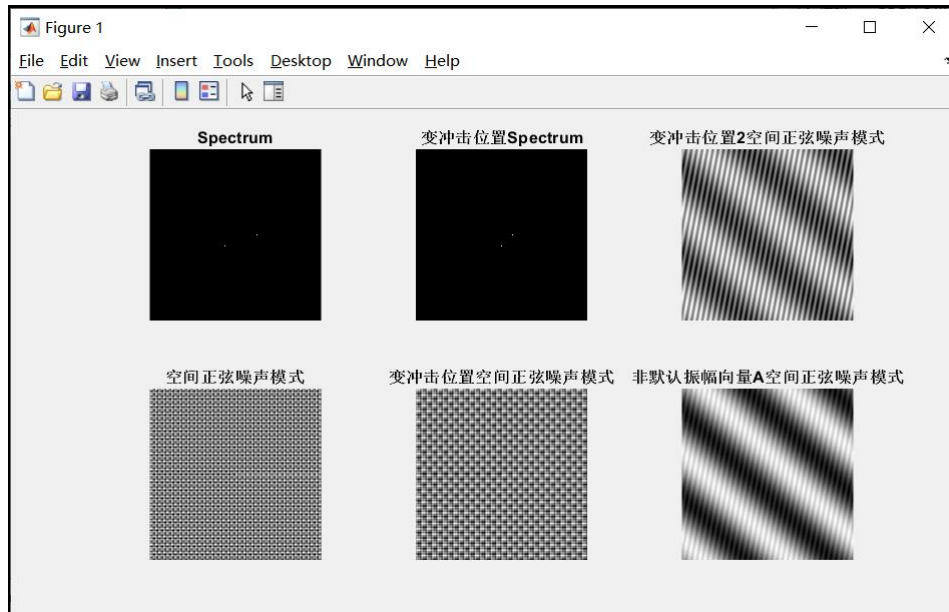
- Function  $[r, R, S] = \text{imnoise3}(M, N, C, A, B)$
- 产生一个大小为  $M \times N$  的正弦噪声模型， $r$  代表所得到的周期噪声模型， $R$  代表所得周期噪声模型的Fourier变换， $S$  代表所得周期噪声模型的Fourier的频谱
- 所得周期噪声模型的大小为  $M \times N$ 。  $C$  为冲击位置的坐标， $A$  是  $1 \times k$  维向量包含振幅的冲击对， $B$  是由  $k \times 2$  矩阵组成的冲击对。



```

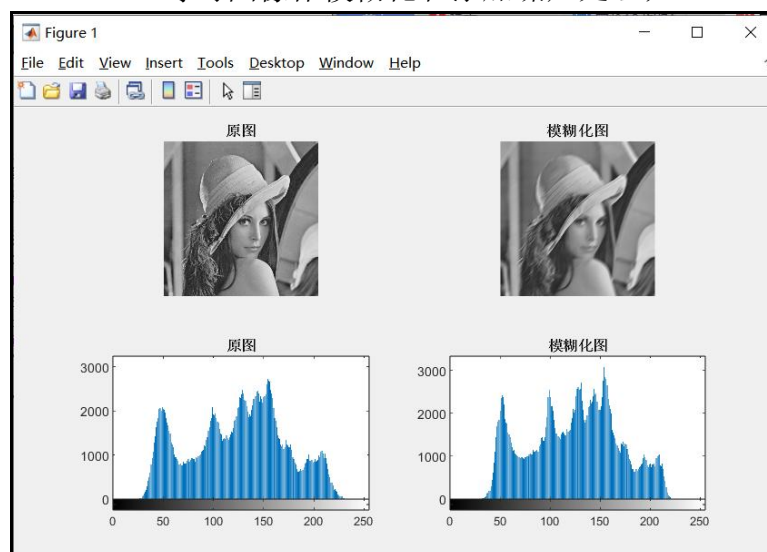
1 C=[0 64;0 128;32 32;64 0;128 0;-32 32];
2 [r,R,S]=imnoise3(512,512,C);
3 subplot(231),imshow(S,[]),title('Spectrum');%显示频谱
4 subplot(234),imshow(r,[]),title('空间正弦噪声模式');%显示空间正弦噪声模式
5 C=[0 32;0 64;16 16;32 0;64 0;-16 16];%改变冲击位置，观察频谱和空间正弦噪声模式变化
6 [r,R,S]=imnoise3(512,512,C);
7 subplot(2,3,2),imshow(S,[]),title('变冲击位置Spectrum');
8 subplot(2,3,5),imshow(r,[]),title('变冲击位置空间正弦噪声模式');
9 C=[6 32;-2 2];%改变冲击位置，观察空间正弦噪声模式变化
10 [r,R,S]=imnoise3(512,512,C);
11 subplot(2,3,3),imshow(r,[]),title('变冲击位置2空间正弦噪声模式');
12 A=[1 5]%使用非默认振幅向量A，观察imA(:, :,1);噪声模式变化
13 [r,R,S]=imnoise3(512,512,C,A);
14 subplot(2,3,6),imshow(r,[]),title('非默认振幅向量A空间正弦噪声模式');

```

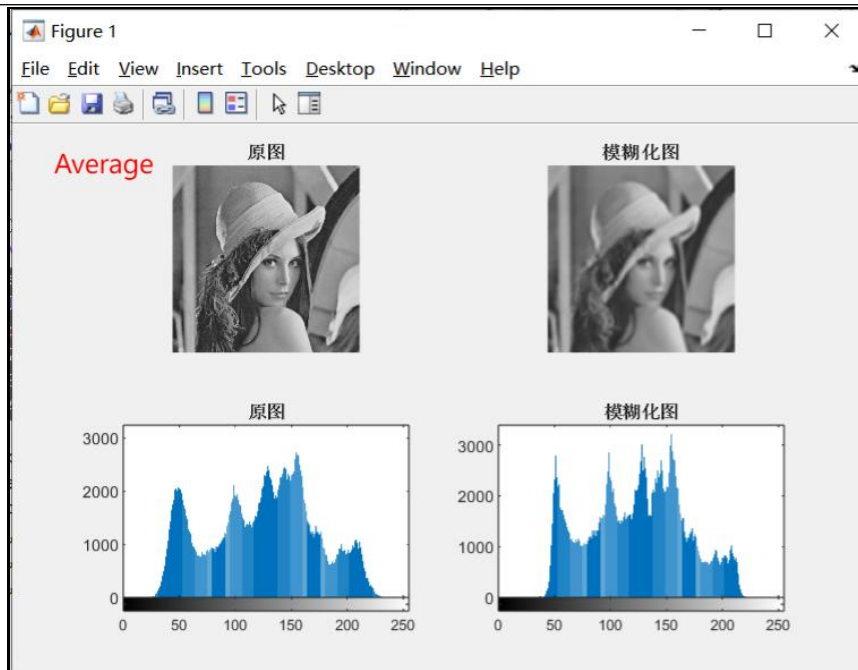


频谱显示不明显，改变冲击位置，频谱有规律变化，空间噪声模式变密。非默认振幅 A 空间正弦噪声频率更高。

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；



使用 `motion` 滤波器返回模糊化图片，其直方图变得更尖锐。



Average 滤波器效果类似。

代码

```

1 % 2. 打开一幅图像，应用 MATLAB 中工具箱函数
2 % fspecial()、imfilter()、imnoise2() 等对图像作模糊化和添加噪声处理；
3 I=imread('C:\Users\Knight6\Pictures\matlabimage\Fig4.bmp');
4 motion_fspecial=fspecial('motion',14,45);
5 IBlur=imfilter(I,motion_fspecial,'replicate');
6 figure,subplot(221),imshow(I),title('原图');
7 subplot(222),imshow(IBLur),title('模糊化图');
8 subplot(223),imhist(I),title('原图');
9 subplot(224),imhist(IBLur),title('模糊化图');
10
11 % average_fspecial=fspecial('average',12);
12 % IBlur=imfilter(I,average_fspecial,'replicate');
13 % figure,subplot(221),imshow(I),title('原图');
14 % subplot(222),imshow(IBLur),title('模糊化图');
15 % subplot(223),imhist(I),title('原图');
16 % subplot(224),imhist(IBLur),title('模糊化图');

```

3. 为图像添加噪声，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）

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

算术均值滤波器

$$\hat{f}(x,y) = \left[ \prod_{(p,q) \in W} g(p,q) \right]^{\frac{1}{mn}}$$

几何均值滤波器

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

谐波均值滤波器

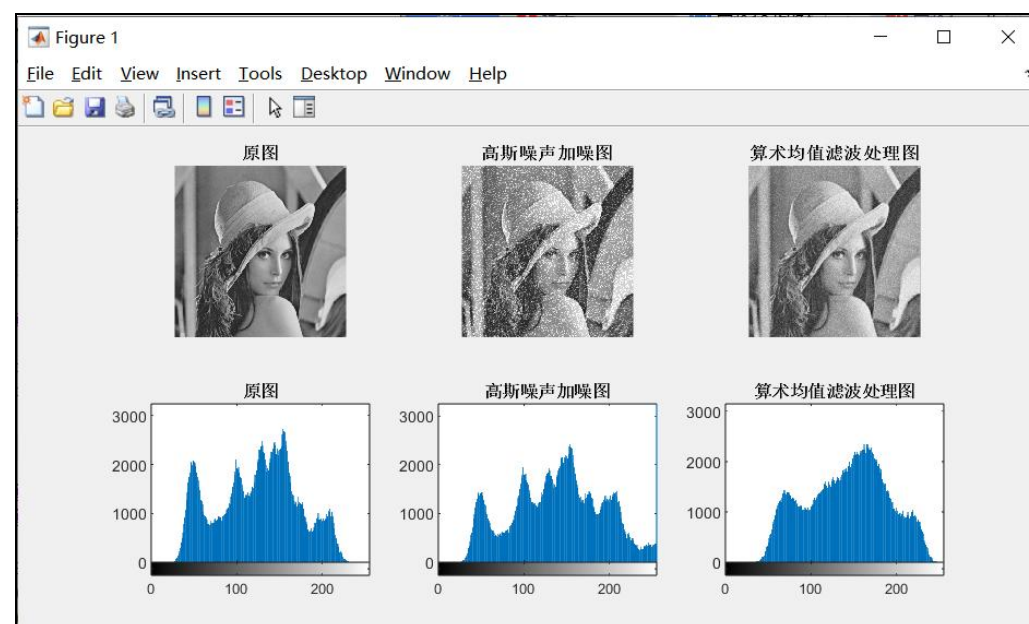
$$\hat{f}(x,y) = \frac{\sum_{(p,q) \in W} g(p,q)^{k+1}}{\sum_{(p,q) \in W} g(p,q)^k}$$

逆谐波均值滤波器

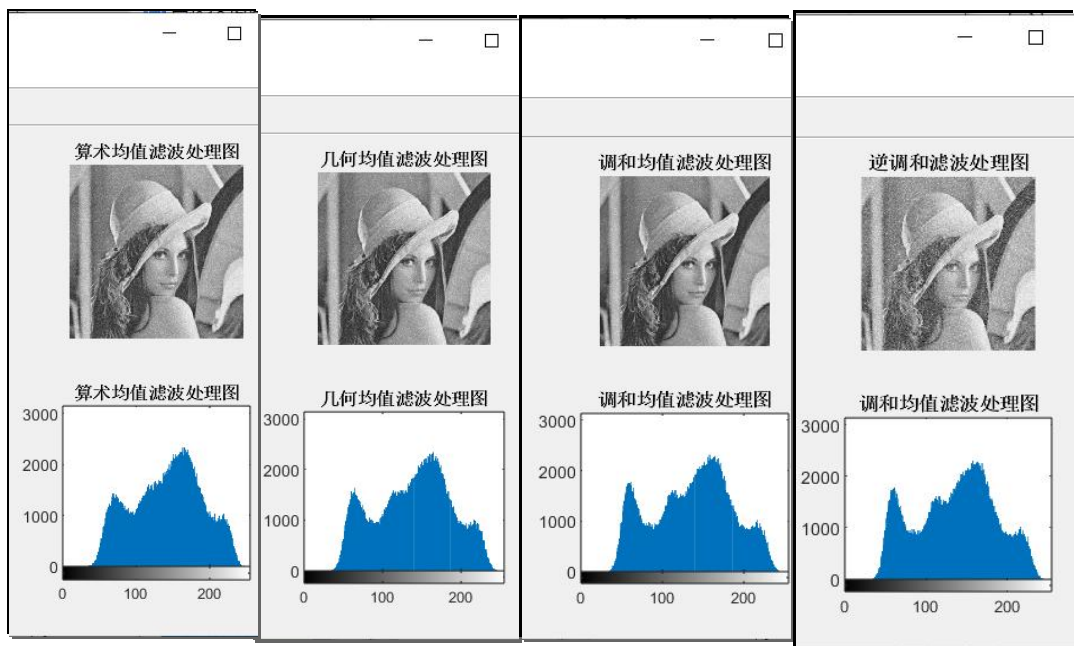
## 代码

```
1 % 3. 为图像添加噪声，验证算术均值、几何均值、调和均值、逆调和均值等
2 % 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）
3 I=imread('C:\Users\Knight6\Pictures\matlabimage\Fig4.bmp');%原图
4 [M,N]=size(I);
5 % Imgnoise=I+uint8(noise2)*50;%高斯噪声的加噪图
6 noise3=imnoise2('salt & pepper',M,N,0.1,0);
7 Imgnoise=I+uint8(noise3)*50;%胡椒噪声的加噪图
8 % Imgnoise=I+uint8(noise5)*50;%瑞利噪声的加噪图
9 figure,subplot(231),imshow(I),title('原图');
10 subplot(234),imhist(I),title('原图');
11 subplot(232),imshow(Imgnoise),title('胡椒噪声加噪图');
12 subplot(235),imhist(Imgnoise),title('胡椒噪声加噪图');
13 % %算术均值滤波
14 % w=fspecial('average',3);
15 % IRestore1=imfilter(Imgnoise,w);
16 % subplot(233),imshow(IRestore1),title('算术均值滤波处理图');
17 % subplot(236),imhist(IRestore1),title('算术均值滤波处理图');
18 % %几何均值滤波
19 % IRestore2=spfilt(Imgnoise,'gmean',3,3);
20 % subplot(233),imshow(IRestore2),title('几何均值滤波处理图');
21 % subplot(236),imhist(IRestore2),title('几何均值滤波处理图');
22 % %谐波(调和)均值滤波
23 % IRestore3=spfilt(Imgnoise,'hmean',3,3);
24 % subplot(233),imshow(IRestore3),title('调和均值滤波处理图');
25 % subplot(236),imhist(IRestore3),title('调和均值滤波处理图');
26 % %逆谐波(逆调和)均值滤波
27 IRestore4=spfilt(Imgnoise,'chmean',3,3,1.5);
28 subplot(233),imshow(IRestore4),title('逆调和滤波处理图');
29 subplot(236),imhist(IRestore3),title('调和均值滤波处理图');
```

## 高斯噪声加噪图

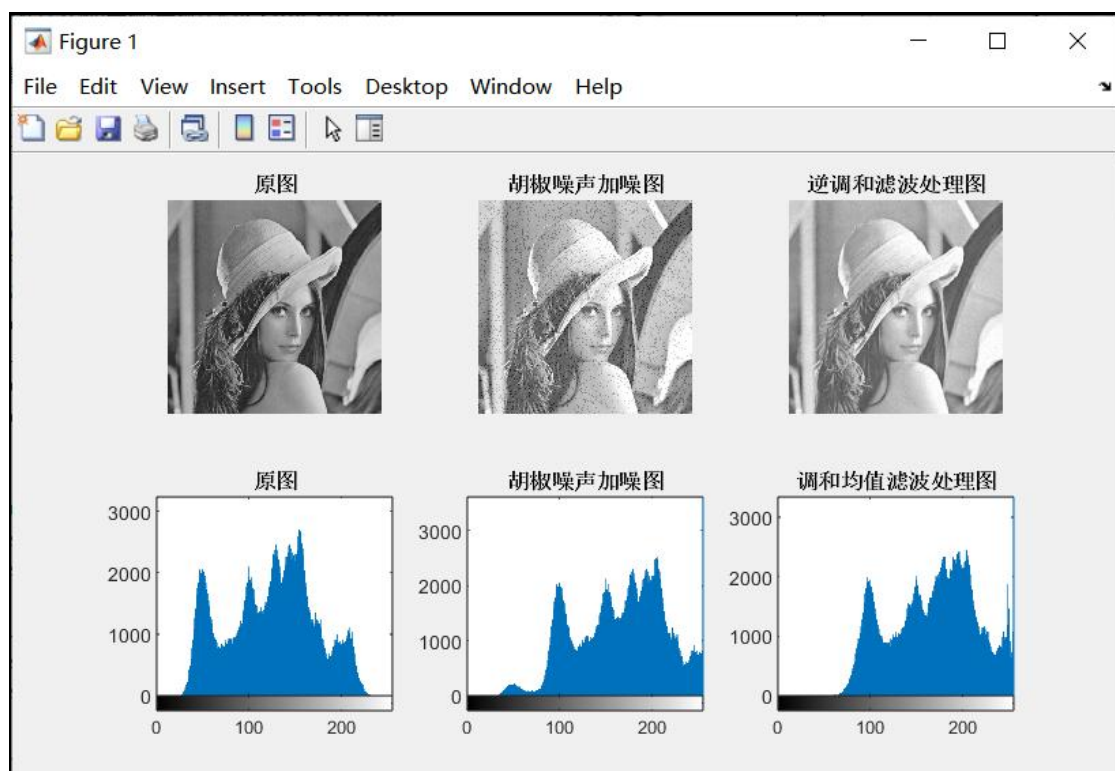


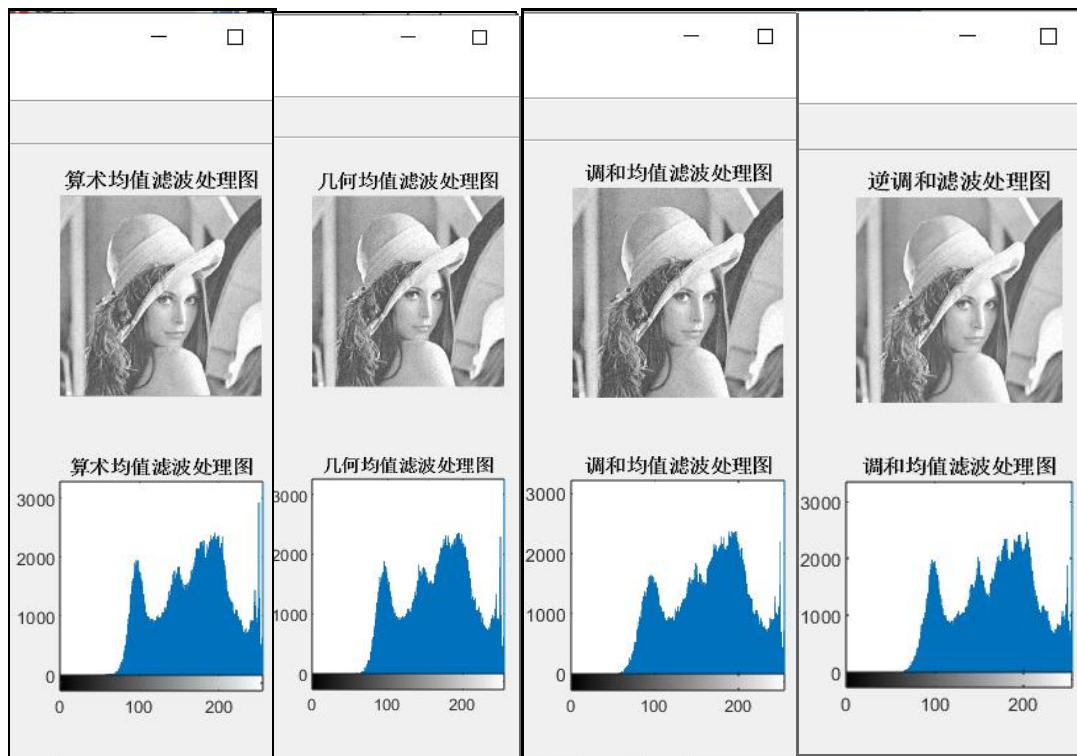




对于高斯噪声污染退化的图像，不同滤波器的复原效果不同，逆调和对图像的复原效果较差，调和和几何均值看起来效果都不错，且灰度直方图处理后更接近原图。

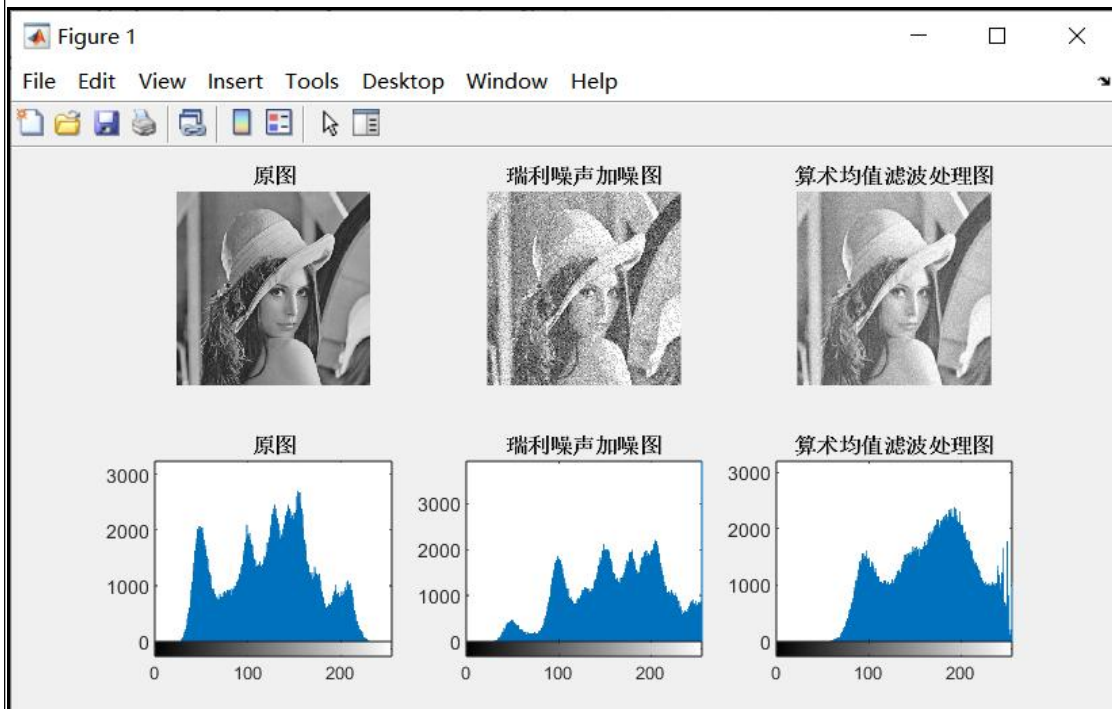
胡椒噪声加噪图



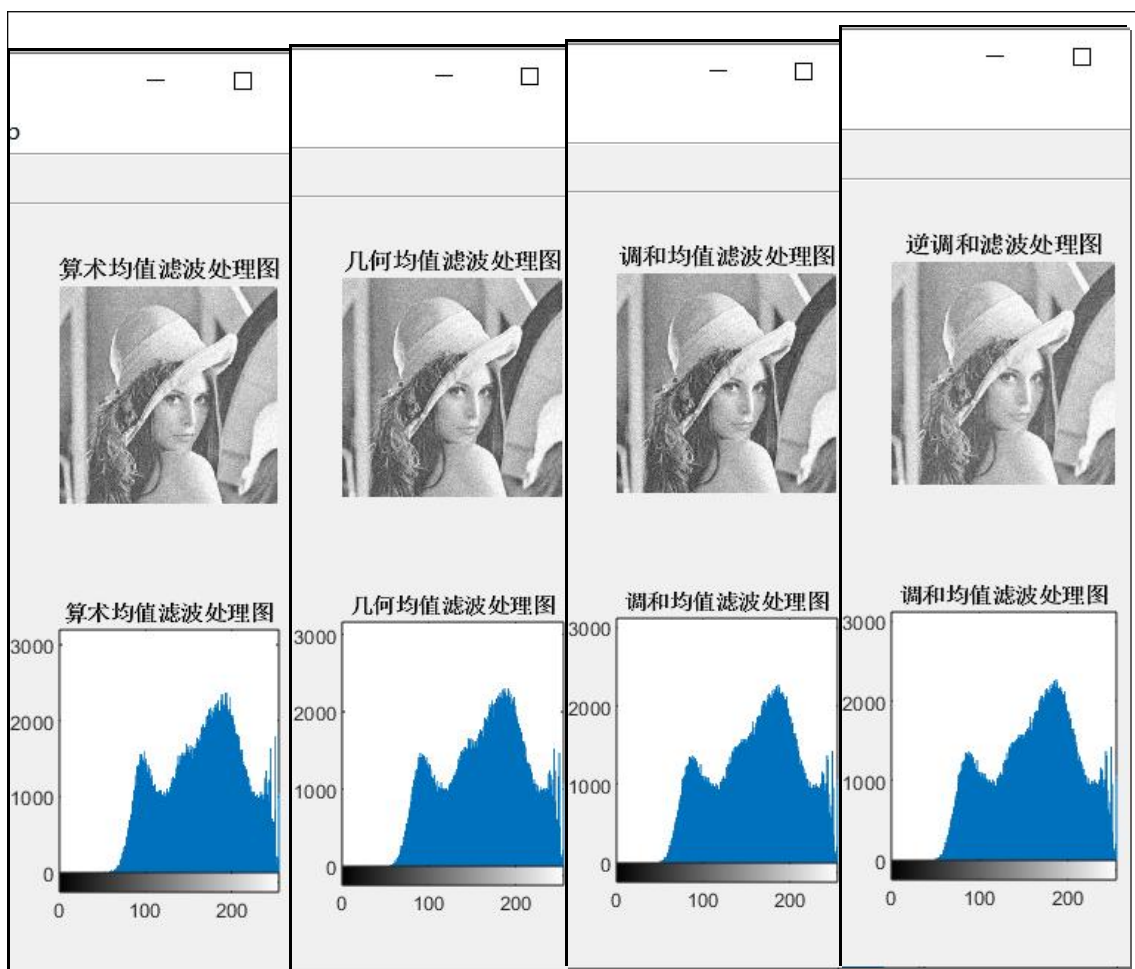


胡椒噪声属于脉冲噪声，由图和看出 Q 为正值的逆调和滤波器处理的效果更好，图像的直方图也更接近原图。

瑞利噪声加噪图







对于瑞丽噪声，处理之后都有不同程度的模糊，但均能实现基础的图像复原。

## 均值滤波器总结

- 算术均值滤波器和几何均值滤波器适合于处理高斯或均匀等随机噪声
- 谐波均值滤波器适合于处理脉冲噪声，但是必须事先知道噪声是暗噪声还是亮噪声，以便于选择合适的滤波阶数符号。

4. 为图像添加周期噪声，并验证带阻滤波器或带通滤波器实现有周期噪声退化的图像复原。

### 带阻滤波器

阻止一定频率范围内的信号通过而允许其它频率范围内的信号通过，消除或衰减傅里叶变换原点处的频段

- 理想带阻滤波器
- 巴特沃思带阻滤波器
- 高斯带阻滤波器

### 带通滤波器

- 允许一定频率范围内的信号通过而阻止其它频率范围内的信号通过

$$H_{bp} = 1 - H_{br}(u, v)$$

以带阻滤波器为例，带通滤波器实质上就是带阻滤波器的对立。

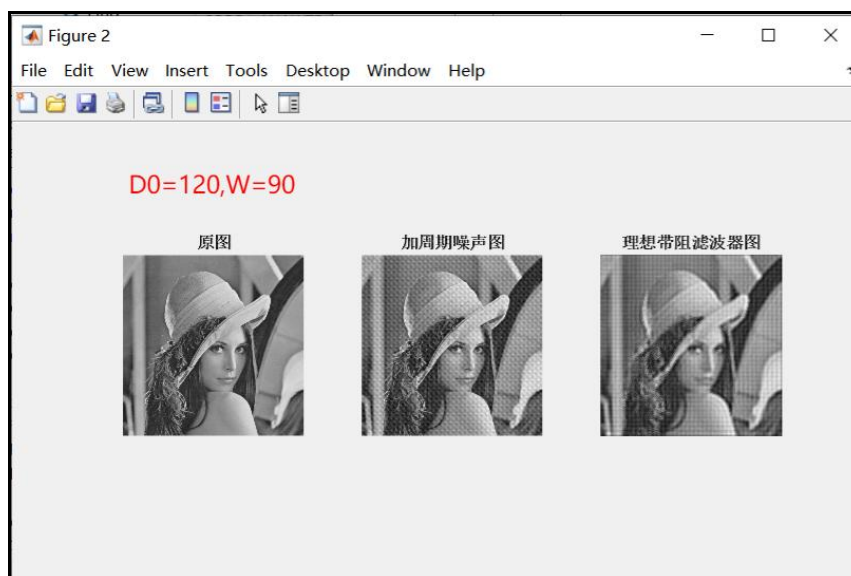


### ①理想带阻滤波器

**带阻滤波器--理想带阻滤波器**

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

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



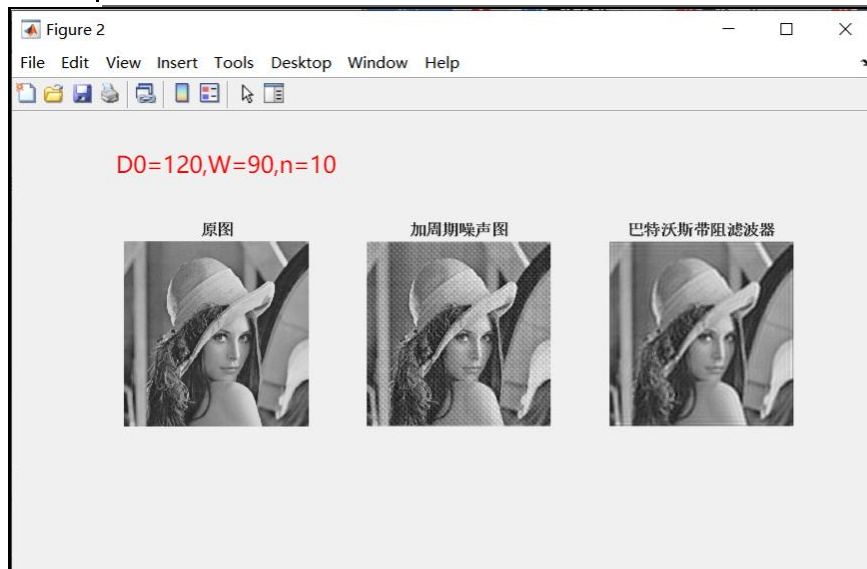
W 是带宽（频带的宽度），D0 是频带的中心半径。

D0 为 120，W 为 90 的情况下，图像模糊较为严重，复原效果不佳。

## ②n 阶巴特沃思带阻滤波器

### 带阻滤波器--n阶的巴特沃思带阻滤波器

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

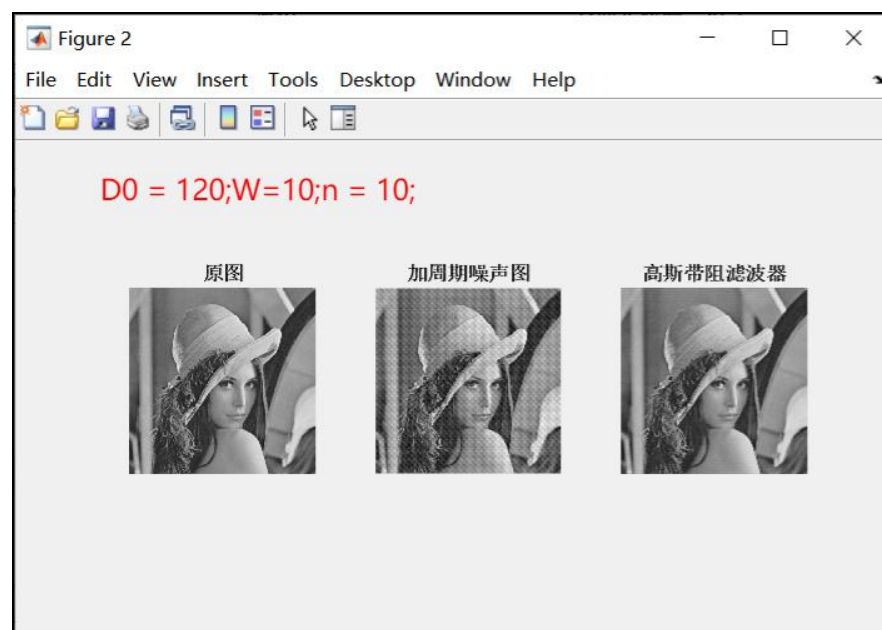
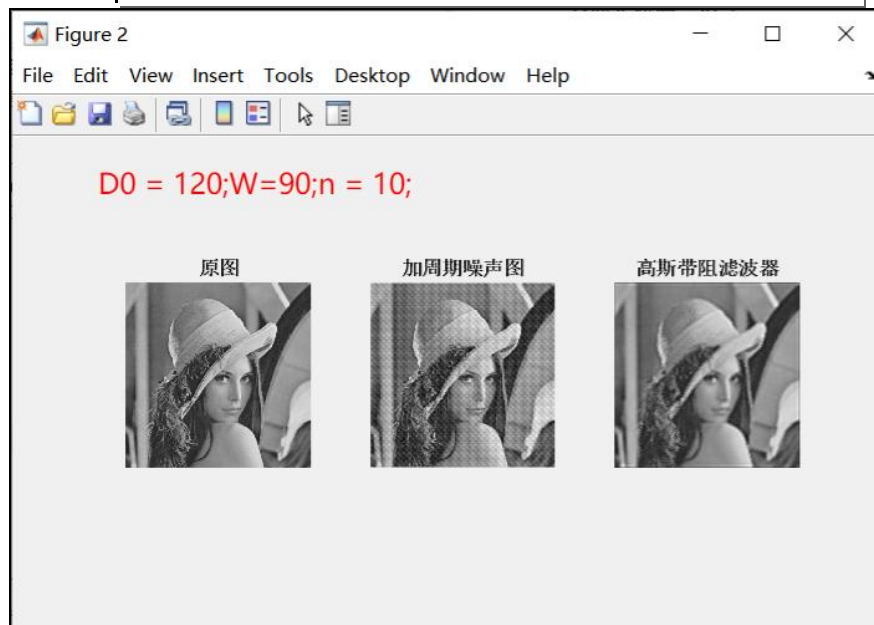


相比理想带阻滤波器，n 阶巴特沃思带阻滤波器明显效果更好。且调整 W 值进行对比，可以发现，另两者不变的情况下，缩小 W 能提升复原效果。

### ③高斯带阻滤波器

#### 带阻滤波器--高斯带阻滤波器

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



和 n 阶巴特沃思带阻滤波器相似，高斯带阻滤波器也能获得较好的复原效果。



## 代码

```
1 % 4. 为图像添加周期噪声，并验证带阻滤波器或带通滤波器
2 % 实现有周期噪声退化的图像复原。
3 I=imread('C:\Users\Knight6\Pictures\matlabimage\Fig4.bmp');
4 [MM,NN]=size(I);
5
6 C=[0 64;0 128;32 32;64 0;128 0;-32 32];
7 A=[0 10 6 3 6 12];
8 [r,R,S]=imnoise3(MM,NN,C,A);
9
10 Imgnoise=I;
11 for i=1:MM
12     for j=1:NN
13         Imgnoise(i,j)=I(i,j)+r(i,j);
14     end
15 end
16 figure,subplot(221),imshow(I),title('原图');
17 subplot(222),imshow(r,[,]),title('空间正弦噪声模式');
18 subplot(223),imshow(S,[,]),title('频谱');
19 subplot(224),imshow(Imgnoise),title('加周期噪声图');
20
21 Imgnoise=im2double(Imgnoise);
22 M = 2*size(I,1); N = 2*size(I,2);
23 u = -M/2:(M/2-1);
24 v = -N/2:(N/2-1);
25 [U,V] = meshgrid(u, v);
26 D = sqrt(U.^2+V.^2);
27 D0 = 120;
28 W=10;
29 n = 10;
30
31 % 理想低通滤波器
32 % H1 = double(D<(D0-W/2))+double(D>(D0+W/2))+0;
33 % J1 = fftshift(fft2(Imgnoise, size(H1, 1), size(H1, 2)));
34 % K1 = J1.*H1; % 时域图像转换为频域
35 % L1 = ifft2(ifftshift(K1));
36 % L1 = L1(1:size(Imgnoise,1), 1:size(Imgnoise, 2)); % 频域图像转换为时频
37 % figure(2),subplot(131),imshow(I),title('原图');
38 % subplot(132),imshow(Imgnoise),title('加周期噪声图');
39 % subplot(133),imshow(L1),title('理想带阻滤波器图');
40
41 % n阶巴特沃斯带阻滤波器
42 % H2 = 1./(1+(D*W./(D.^2-D0.^2)).^(2*n));
43 % J2 = fftshift(fft2(I, size(H2, 1), size(H2, 2))); % 时域图像转换为频域
44 % K2 = J2.*H2;
45 % L2 = ifft2(ifftshift(K2)); % 频域图像转换为时频
46 % L2 = L2(1:size(I,1), 1:size(I, 2));
47 % figure(2),subplot(131),imshow(I),title('原图');
48 % subplot(132),imshow(Imgnoise),title('加周期噪声图');
49 % subplot(133),imshow(uint8(L2)),title('巴特沃斯带阻滤波器');
50
51 % 高斯带阻滤波器
52 H3 = 1-exp(-1/2*((D.^2-D0.^2)./(D*W)).^2);
53 J3 = fftshift(fft2(I, size(H3, 1), size(H3, 2))); % 时域图像转换为频域
54 K3 = J3.*H3;
55 L3 = ifft2(ifftshift(K3));
56 L3=L3(1:size(I,1), 1:size(I, 2)); % 频域图像转换为时频
57 figure(2),subplot(131),imshow(I),title('原图');
58 subplot(132),imshow(Imgnoise),title('加周期噪声图');
59 subplot(133),imshow(uint8(L3)),title('高斯带阻滤波器');
```

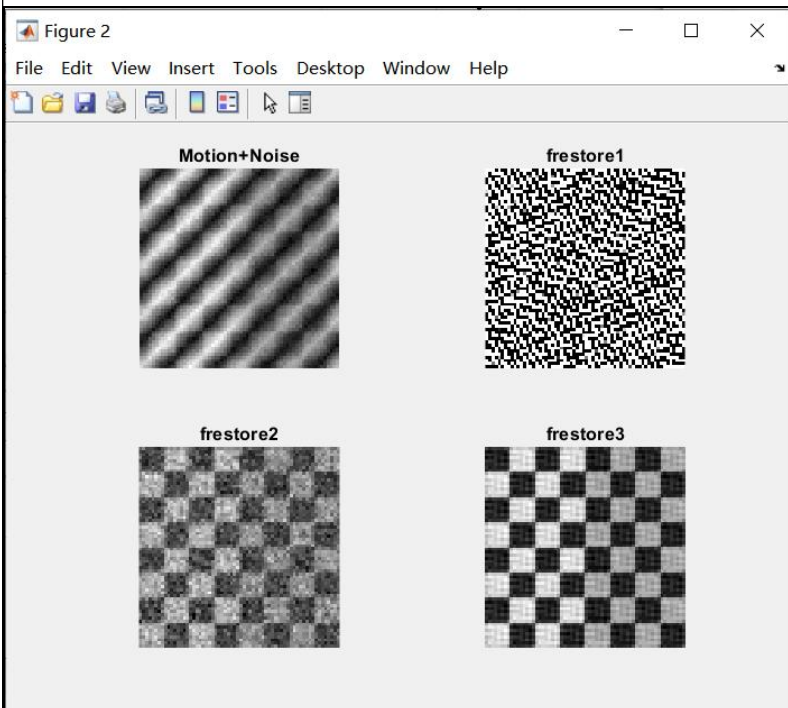
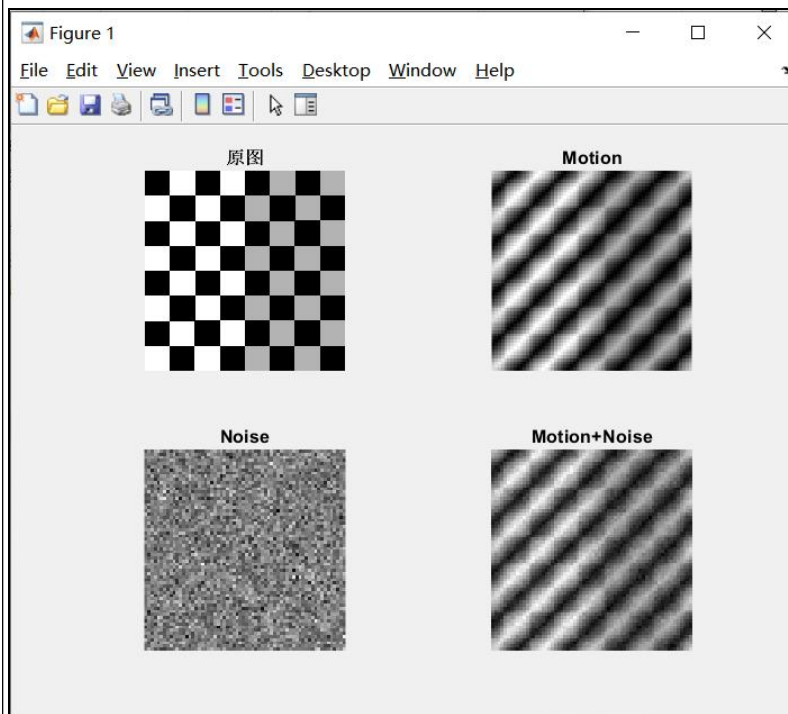


5. 参照课本和课堂讲解的例题中的方法对由退化函数加噪声共同作用下的退化图像进行维纳滤波复原。（注：实验中使用的图像数据量最好小一些）

代码

```
1 % 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
2 % 的退化图像进行维纳滤波复原。
3 % （注：实验中使用的图像数据量最好小一些）
4 f=checkerboard(8);
5 PSF=fspecial('motion',14,45);
6 gb=imfilter(f,PSF,'circular');
7 noise=imnoise2('Gaussian',size(f,1),size(f,2),0,sqrt(0.001));
8 g=gb+noise;
9 figure,
10 subplot(221),imshow(pixeldup(f,8),[]),title('原图');
11 subplot(222),imshow(pixeldup(gb,8),[]),title('Motion');
12 subplot(223),imshow(pixeldup(noise,8),[]),title('Noise');
13 subplot(224),imshow(pixeldup(g,8),[]),title('Motion+Noise');
14 %使用维纳滤波对图像进行去模糊处理
15 %frestore1
16 frestore1=deconvwnr(g,PSF);
17 %frestore2
18 Sn=abs(fft2(noise)).^2;
19 nA=sum(Sn(:))/numel(noise);
20 Sf=abs(fft2(f)).^2;
21 fA=sum(Sf(:))/numel(f);
22 R=nA/fA;
23 frestore2=deconvwnr(g,PSF,R);
24 %frestore3
25 NCORR=fftshift(real(ifft2(Sn)));
26 ICORR=fftshift(real(ifft2(Sf)));
27 frestore3 =deconvwnr(g,PSF,NCORR,ICORR);
28
29 figure(2);
30 subplot(221),imshow(pixeldup(g,8),[]),title('Motion+Noise');
31 subplot(222),imshow(frestore1),title('frestore1');
32 subplot(223),imshow(pixeldup(frestore2,8),[]),title('frestore2');
33 subplot(224),imshow(pixeldup(frestore3,8),[]),title('frestore3');
```

## 处理效果



- ①方法 1 直接利用 `deconvwnr` 函数直接对图像进行维纳逆滤波，效果很差；
- ②方法 2 根据原图像与噪音图像，计算信噪比  $R$  信息，再利用  $R$  和维纳滤波复原图像，效果相对变好；
- ③方法 3 用自相关函数  $S, S_f$  进行维纳滤波，复原效果有了很大提高，只有较少噪声存在。我们根据一定的先验知识（已知的原始图像和噪声图像）对参数进行估计，再将估计值参数带入通过维纳滤波器进行图像复原，效果更好。

## 实验的体会与思考题

### 1. 常见的图像退化模型有哪些种类？

从图像信号的产生形式来看，可将退化模型分为连续图像退化模型和离散图像退化模型。

(1) 连续图像退化的一般模型如图所示。输入图像  $f(x, y)$  经过一个退化系统或退化算子  $H(x, y)$  后考虑加性噪声的影响产生的退化图像  $g(x, y)$  可以表示为：
$$g(x, y) = h(x, y) * f(x, y) + n(x, y)。$$

(2) 数字图像处理系统处理的图像是离散图像，所以对连续退化模型的离散化即形成离散图像退化模型。这种模型通常用矩阵代数求解。

根据降质系统的传递函数主要有：

- (1) 空间非相干成像系统由于衍射限制造成的图像退化模型。
- (2) 照相机与被摄景物之间的相对运动造成的图像退化模型。
- (3) 大气湍流造成的图像退化模型。
- (4) 由于成像系统的非线性、飞行器的姿态、高度和速度变化等引起的不稳定与不可预测的几何失真，造成的几何畸变模型。

### 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。