

# 数字图像处理

## 第六次作业

姓名：回彭杨

班级：自动化 66 班

学号：2160504148

1、在测试图像上产生高斯噪声 lena 图-需能指定均值和方差；并用多种滤波器恢复图像，分析各自优缺点；

1) 添加高斯噪声

lena.bmp 原始图像



lena 添加高斯噪声 ( $\mu=0$ ,  $\sigma^2=0.01$ )



lena 添加高斯噪声 ( $\mu=0$ ,  $\sigma^2=0.05$ )



lena 添加高斯噪声 ( $\mu=0$ ,  $\sigma^2=0.1$ )



lena 添加高斯噪声 ( $\mu=0.1$ ,  $\sigma^2=0.01$ )



lena 添加高斯噪声 ( $\mu=0.5$ ,  $\sigma^2=0.01$ )



## 2) 图像恢复(选取被均值为 0, 方差为 0.01 的高斯噪声污染的图像为例)

利用算术均值滤波器恢复图像(5x5 模板)

lena 添加高斯噪声 ( $\mu=0$ ,  $\sigma^2=0.01$ )



算术均值滤波的结果



利用中值滤波器恢复图像 (5x5 模板)

lena 添加高斯噪声 ( $\mu=0$ ,  $\sigma^2=0.01$ )



中值滤波的结果



### (4) 结果分析及总结:

①首先通过 `imnoise` 函数分别产生了被不同均值和方差的高斯噪声污染的图像。当高斯噪声均值不变为 0 时,随着方差增加,图像噪声越严重;当高斯噪声方差不变时,均值会影响到整个图像的灰度值,使整个图像变亮。与理论上均值和方差对图像的影响一致。

②分别使用算术均值滤波器和中值滤波器对加噪图像进行恢复。两种方法在一定程度上都可以降低噪声。算术均值滤波器降低噪声的同时也模糊了图像。

### 2.推导维纳滤波器并实现下边要求;

(a) 实现模糊滤波器如方程 Eq. (5.6-11).

(b) 模糊 lena 图像: 45 度方向,  $T=1$ ;

(c) 再模糊的 lena 图像中增加高斯噪声, 均值=0, 方差=10 pixels 以产生模糊图像;

(d) 分别利用方程 Eq. (5.8-6)和(5.9-4), 恢复图像; 并分析算法的优缺点. (1) 问题分析:

解: (a) 实现模糊滤波器如方程 Eq. (5.6-11).

模糊滤波器的频域表达式为:

$$H(u,v) = \frac{T}{\pi(ua+vb)} \sin[\pi(ua+vb)] e^{-j\pi(ua+vb)}$$

故实现该滤波器，只需先将输入图像进行傅里叶变换并移至图像中心，之后将图像的傅里叶变换和模糊滤波器的傅里叶变换进行阵列相乘，将得到的结果经过傅里叶反变换返回到空间域即可实现该滤波器。具体程序见附件 `mohu_filter.m`

(b) 模糊 lena 图像：45 度方向， $T=1$ ；（ $a=0.1$ ， $b=0.1$ ； $T=1$ ）

lena.bmp 原始图像



lena 运动模糊的结果（ $a=0.1$ ， $b=0.1$ ； $T=1$ ）



lena.bmp 原始图像



lena 运动模糊的结果(调用 MATLAB 中的函数)



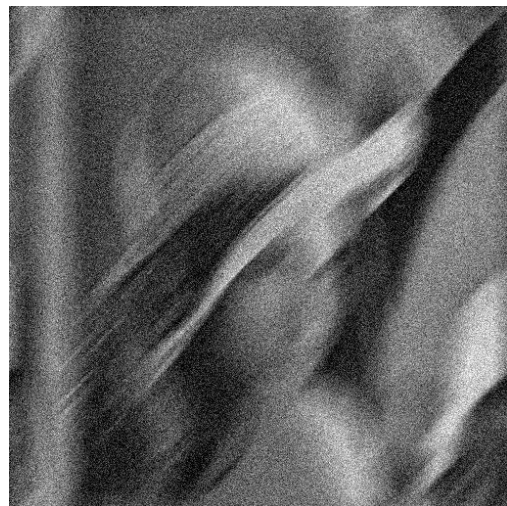
(c) 再模糊的 lena 图像中增加高斯噪声，均值=0，方差=10 pixels 以产生模糊图像；  
lena 运动模糊的结果（ $a=0.1$ ， $b=0.1$ ； $T=1$ ） 添加高斯噪声的结果（均值为0，方差为0.01）



lena 运动模糊的结果 (MATLAB 版)



添加高斯噪声的结果 (MATLAB 版)



(d) 分别利用方程 Eq. (5.8-6)和(5.9-4), 恢复图像。  
维纳滤波的结果:

lena 运动模糊+高斯噪声.bmp

维纳滤波结果 ( $K=0.06$ )





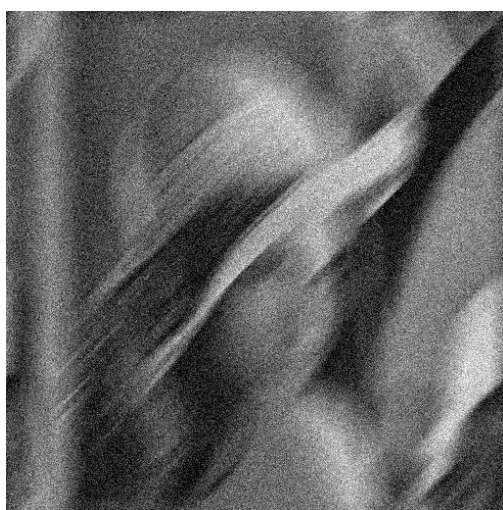
维纳滤波结果 ( $K=0.01$ )



维纳滤波结果 ( $K=0.5$ )



lena 运动模糊+高斯噪声 (MATLAB 版)

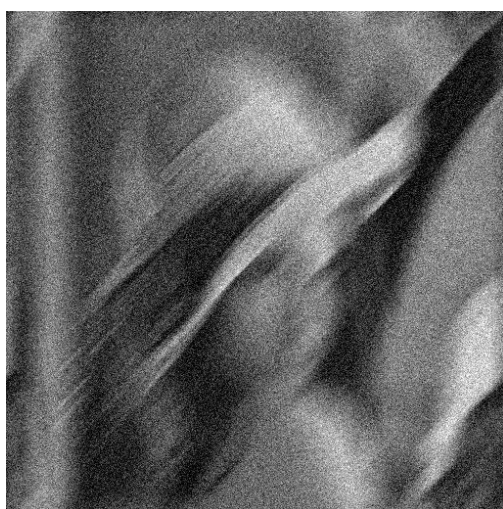


维纳滤波结果 (MATLAB 版)



约束最小二乘方滤波的结果:

lena 运动模糊+高斯噪声 (MATLAB 版)



约束最小二乘滤波结果 (MATLAB 版)



### 结果分析:

①首先分别通过自己编写的模糊函数和 MATLAB 中提供的 `imfilter` 和 `fspecial` 函数配合使用对图像 `lena` 进行了模糊滤波。发现套用书上的公式图像是斜向下 45 度运动模糊，而 MATLAB 中函数模糊的结果是斜向上 45 度运动模糊，不过这不是重点可以接受，模糊的基本效果还是一致的。之后调用 `imnoise` 函数对两幅图像加入高斯噪声，得到第二问的结果。

②之后分别使用自己编写的函数和 MATLAB 中提供的 `deconvwnr` 函数进行维纳滤波。调用 MATLAB 中函数滤波后的图像得到了一定的改善，运动模糊的影响基本被消除，但噪声的影响仍然较大，导致图像质量下降；对于自己编写的维纳滤波函数，难点在于寻找令信噪比最大的  $K$  值，报告中显示了部分  $K$  值对应的滤波结果，其中  $K=0.06$ ，为信噪比最大时的滤波结果，从结果看，视觉上的效果并不是很理想，要想达到更好的效果可能需要寻找更加合适的  $K$  值。

## 附录

### 【参考文献】

- [1] 冈萨雷斯.数字图像处理（第三版）北京：电子工业出版社，2011
- [2] 周品.MATLAB 数字图像处理 北京：清华大学出版社，2012
- [3] 杨杰.数字图像处理及 MATLAB 实现 北京：电子工业出版社，2010

### 【源代码】

#### **add\_gaussian\_noise.m**

```
I=imread('lena.bmp');
figure(1);
imshow(I);
title('源图像 lena.bmp');
imwrite(I, 'lena 原始图像.bmp');
I2=imnoise(I, 'gaussian', 0.5, 0.01);
figure(2);
imshow(I2);
title('加入 gaussian 噪声后的 lena.bmp');
imwrite(I2, 'lena 加入 gaussian 噪声后(u=0.5, s^2=0.01).bmp');
```

#### **suanshujunzhi\_filter.m**

```
I=imread('lena_gaussian_noise.bmp');
figure(1);
imshow(I);
title('lena 加入 gaussian 噪声后(u=0, s^2=0.01).bmp');
imwrite(I, 'lena 加入 gaussian 噪声后(u=0, s^2=0.01).bmp');
n=5;
h=1/n^2.*ones(n,n);
I2=conv2(I,h, 'same');
I2=uint8(I2);
figure(2);
imshow(I2);
title('算术均值滤波的结果(5x5)');
imwrite(I2, '算术均值滤波的结果(5x5).bmp');
```

#### **median\_filter.m**

```
I=imread('lena_gaussian_noise.bmp');
figure(1);
imshow(I);
title('lena 加入 gaussian 噪声后(u=0, s^2=0.01).bmp');
imwrite(I, 'lena 加入 gaussian 噪声后(u=0, s^2=0.01).bmp');
figure(2);
n=5;
a=ones(n,n);
p=size(I);
x1=double(I);
```



```

x2=x1;
for i=1:p(1)-n+1
    for j=1:p(2)-n+1
        c=x1(i:i+(n-1),j:j+(n-1));
        e=c(1,:);
        for u=2:n
            e=[e,c(u,:)];
        end
        mm=median(e);
        x2(i+(n-1)/2,j+(n-1)/2)=mm;
    end
end
I2=uint8(x2);
imshow(I2);
title('中值滤波的结果 (5x5) ');
imwrite(I2, '中值滤波的结果 (5x5) .bmp');

```

### **mohu\_filter.m**

```

I=imread('lena.bmp');
figure(1);
imshow(I);
title('lena.bmp 原始图像');
imwrite(I, 'lena 原始图像.bmp');
f=double(I);
F=fft2(f);
F=fftshift(F);
[M,N]=size(F);
a=0.1;b=0.1;T=1;
for u=1:M
    for v=1:N
        H(u,v)=(T/(pi*(u*a+v*b)))*sin(pi*(u*a+v*b))*exp(-sqrt(-1)*pi*(u*a+v*b));
        G(u,v)=H(u,v)*F(u,v);
    end
end
G=ifftshift(G);
g=ifft2(G);
g=256.*g./max(max(g));
g=uint8(real(g));
figure(2);
imshow(g);
title('运动模糊化 lena.bmp');
imwrite(g, 'lena 运动模糊的结果.bmp');

```

### Wiener\_filter.m

```
I=imread('lena 运动模糊+高斯噪声.bmp');
figure(1);
imshow(I);
title('lena 运动模糊+高斯噪声');
imwrite(I, 'lena 运动模糊+高斯噪声.bmp');
g=double(I);
G=fft2(g);
G=fftshift(G);
[M,N]=size(G);
a=0.1;b=0.1;T=1;K=0.0259;
for u=1:M
    for v=1:N
        H(u,v)=(T/(pi*(u*a+v*b)))*sin(pi*(u*a+v*b))*exp(-sqrt(-1)*pi*(u*a+v*b));
        F(u,v)=1/H(u,v)*(abs(H(u,v)))^2/((abs(H(u,v)))^2+K)*G(u,v);
    end
end
F=ifftshift(F);
f=ifft2(F);
f=256.*f./max(max(f));
f=uint8(real(f));
figure(2);
imshow(f);
title('维纳滤波的结果');
imwrite(f, '维纳滤波的结果(K=0.0259).bmp');
```

### wiener\_filter\_k.m

```
I=imread('lena 运动模糊+高斯噪声.bmp');
figure(1);
imshow(I);
title('lena 运动模糊+高斯噪声');
imwrite(I, 'lena 运动模糊+高斯噪声.bmp');
g=double(I);
G=fft2(g);
G=fftshift(G);
[M,N]=size(G);
a=0.1;b=0.1;T=1;i=1;
format long
for k=0.01:0.01:0.11
    for u=1:M
        for v=1:N
            H(u,v)=(T/(pi*(u*a+v*b)))*sin(pi*(u*a+v*b))*exp(-sqrt(-1)*pi*(u*a+v*b));
```

```

F(u,v)=(1/H(u,v))*(abs(H(u,v))^2/(abs(H(u,v))^2+k))*G(u,v);
    end
end
F=ifftshift(F);
f=ifft2(F);
f=256.*f./max(max(f));
f=uint8(real(f));
figure;
imshow(f);
title('维纳滤波的结果');
e=f-uint8(g);
SNR(i)=sum(sum(g.^2))/sum(sum(e.^2));
i=i+1;
end
idx=find(max(SNR))

```

#### **wiener\_filter\_matlab.m**

```

I=imread('lena.bmp');
H=fspecial('motion',50,45);
I1=imfilter(I,H,'circular','conv');
figure(1);
imshow(I1);
title('运动模糊后的 lena.bmp(角度为 45 度)');
imwrite(I1,'lena 运动模糊（调用 matlab 中的函数）.bmp');
I2=imnoise(I1,'gaussian',0,0.01);
figure(2);
imshow(I2);
title('加噪并模糊的 lena.bmp');
imwrite(I2,'lena 运动模糊+高斯噪声（调用 matlab 中的函数 0.bmp');
figure(3);
noise=imnoise(zeros(size(I)),'gaussian',0,0.01);
NSR=sum(noise(:).^2)/sum(im2double(I(:)).^2);
I3=deconvwnr(I2,H,NSR);
imshow(I3);
title('维纳滤波的结果');
imwrite(I3,'lena 维纳滤波的结果（调用 MATLAB 中的函数）.bmp');

```

#### **yueshuzuixiaercheng\_filter\_matlab.m**

```

I=imread('lena.bmp');
h=fspecial('motion',50,45);
I1=imfilter(I,h,'circular','conv');
I2=imnoise(I1,'gaussian',0,0.01);
figure(1);
imshow(I2);

```

```
title('lena 运动模糊+高斯噪声');  
imwrite(I2, 'lena 运动模糊+高斯噪声 (MATLAB 版) .bmp');  
V=0.0001;  
NoisePower=V*prod(size(I));  
[g,LAGRA]=deconvreg(I1,h,NoisePower);  
figure(2);  
imshow(g);  
title('约束最小二乘滤波的结果 (MATLAB 版)');  
imwrite(g, '约束最小二乘滤波的结果 (MATLAB 版) .bmp');
```