实验5 图像复原

Matlab自带图像所在目录，不同的安装路径前面的目录可能会有差异，请自行查阅安装目录后查找自带图像位置 C:\Program Files\MATLAB\R2016a\toolbox\images\imdata

**实验目的：**

（1）掌握频率带阻滤波器的原理和基本应用方法；

（2）掌握图像退化模型的时域和频域复原原理配对关系；

（3）掌握加性噪声退化模型的原理，及复原加性噪声退化图像的方法；

（4）掌握自适应维纳滤波，自适应中值滤波器的设计和基本应用。

（5）掌握逆滤波，维纳滤波的运动模糊图像复原原理及实现算法。

**实验要求：**

（1）以自己的学号+姓名创建一个文件夹，并将该文件夹添加到MATLAB路径管理器中；

（2）掌握函数*fft*2( ), *fftshift*( )，*ifftshift*( )，*ifft*2( ) 等基本函数的调用。

（3）掌握函数wiener2(I,[M N])在复原高浓度高斯噪声图像的技术和方法；

（4）掌握自适应中值滤波算法在复原高浓度椒盐噪声图像的技术和方法。

（5）掌握函数deconvwnr(K, PSF, 0);%inverse filter，

SR2=noise\_var/var(I(:)); L2=deconvwnr(K, PSF, NSR2);%wiener filter

在运动模糊复原中的应用。

**实验内容**：

（1）读入实验提供的彩色copper.png图，将其转换为灰度图，变换到频谱图后观察图中左右两个高频点，设计带阻滤波器，将左右两个高频两点去除，并将处理后的频谱逆变换回空域，观察处理后的图像与原图的效果异同。

附代码：

clear

clc

im=imread('C:\Users\86181\Desktop\copper.png');

figure(1)

imshow(im)

title('原图')

im=rgb2gray(im);

[M,N]=size(im);

im1=fftshift(fft2(im));

figure(2)

imshow(log(1+abs(im1)),[])

title('原图频谱图')

freq=9;

width=6;

ff = ones(M,N);

for i=1:M

for j=1:N

ff(i,j)=1-exp(-0.5\*((((i-M/2)^2+(j-N/2)^2)-freq^2)/(sqrt((i-M/2)^2+(j-N/2)^2)\*width))^2);

end

end

figure(3)

mesh(ff);

title('带阻滤波器三维图');

out=im1.\*ff;

figure(4)

imshow(log(1+abs(out)),[])

title('原图频谱经过带阻滤波后的频谱图')

out = ifftshift(out);

out = ifft2(out);

out = abs(out);

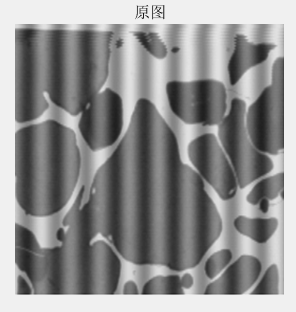
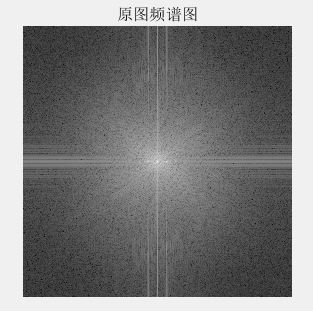
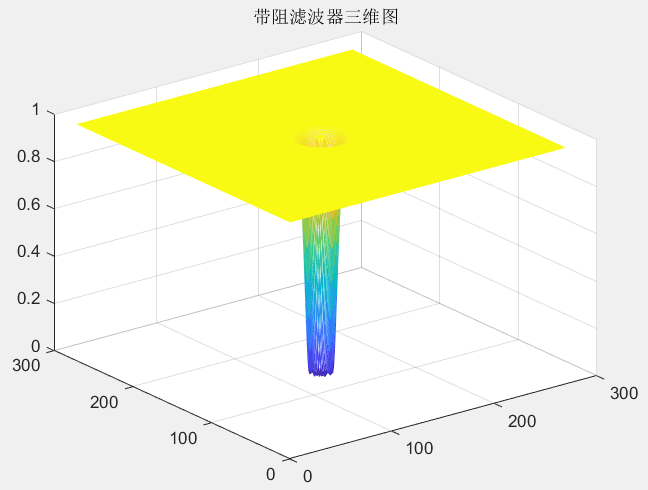
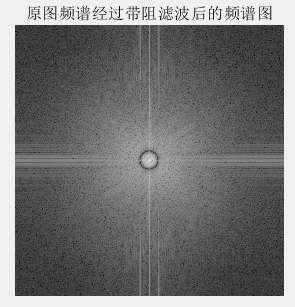
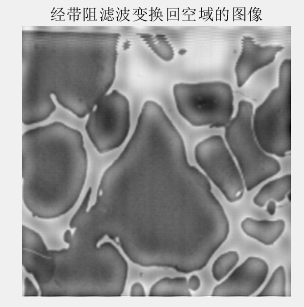
out = out/max(out(:));

figure(5)

imshow(out,[])

title('经带阻滤波变换回空域的图像')

结果图：

（2）（选做）读入matlab自带彩色图像’ saturn.png’，先将此图转为灰度图，并在灰度图添加均值为0，方差为0.3的高斯噪声，用均值滤波器和自适应维纳滤波器wiener2(I,[M N])对图像进行复原，并对比复原结果

附代码：

clear

clc

im=imread('saturn.png');

figure(1)

imshow(im)

title('原图')

im=rgb2gray(im);

im1=imnoise(im,'gaussian',0,0.3);

figure(2)

imshow(im1)

title('噪声图')

PSF=fspecial('average', 3);

im2=imfilter(im1,PSF);

figure(3)

imshow(im2)

title('均值滤波图')

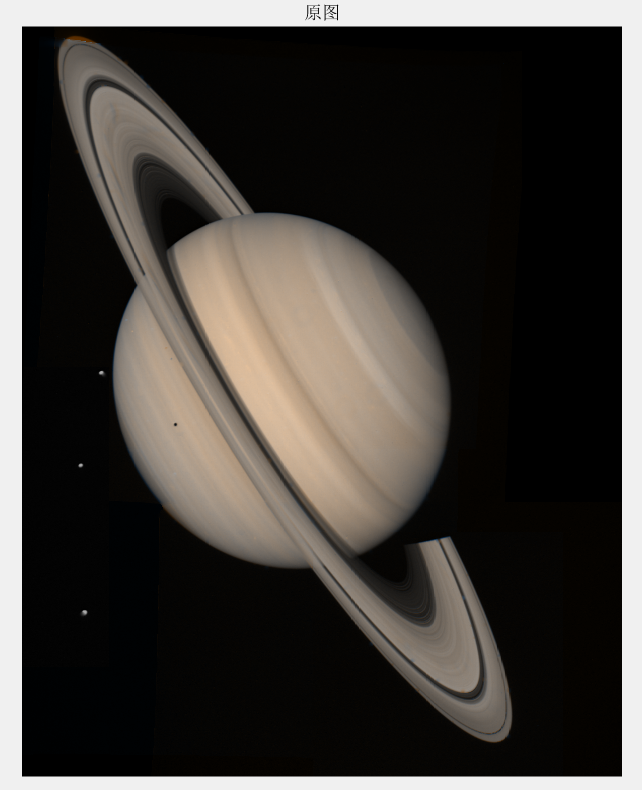
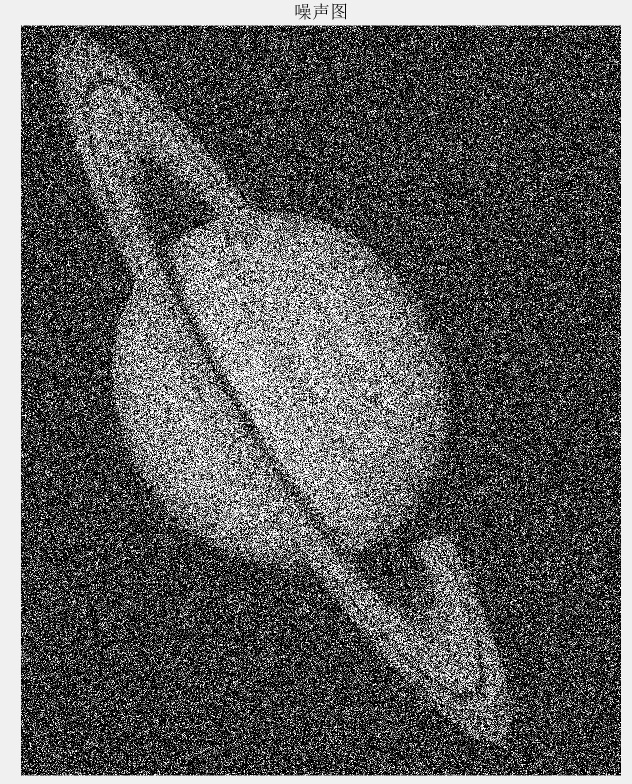
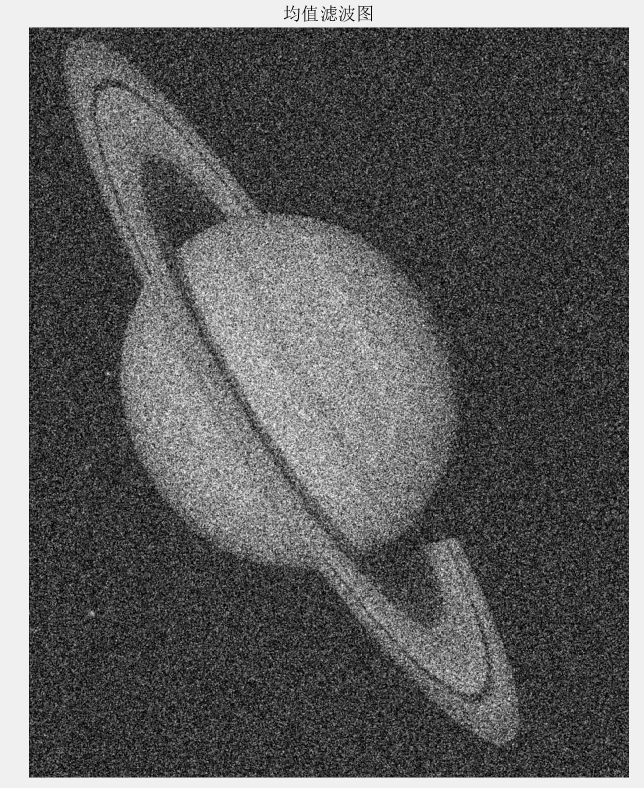
im3=wiener2(im1,[3 3]);

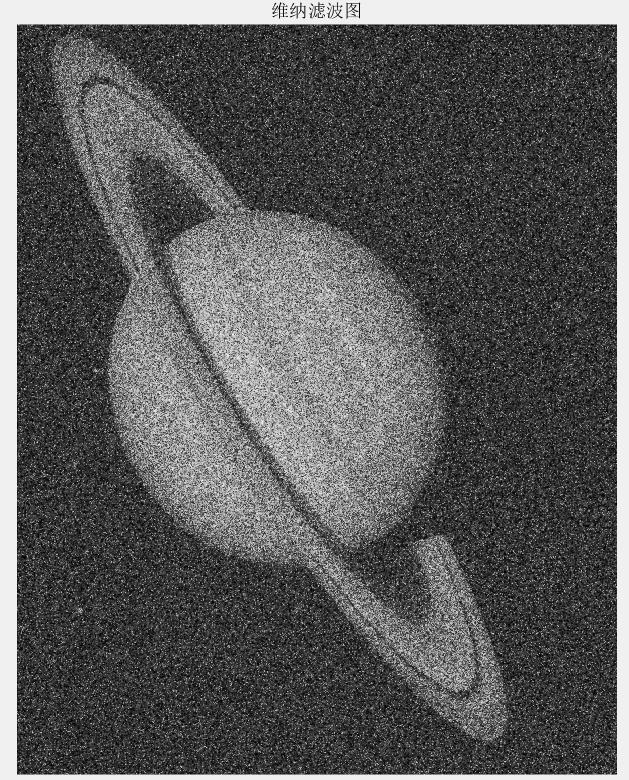
figure(4)

imshow(im3)

title('维纳滤波图')

结果图：



（3）（选做）读入matlab自带彩色图像’ saturn.png’，先将此图转为灰度图，并在灰度图添加均值30%的椒盐噪声，用中值滤波器和自适应中值滤波器对图像进行复原，并对比复原结果

附代码：

Main：

clear

clc

gray=imread('saturn.png');

figure(1)

imshow(gray)

title('原图')

gray=rgb2gray(gray);

gray\_noise=imnoise(gray,'salt & pepper',0.5);

M1=medfilt2(gray\_noise,[3,3]);

f1=adaptive\_median\_filter(gray\_noise,11);

figure("color",[1,1,1]);

figure(2)

imshow(gray\_noise)

title('噪声图')

figure(3)

imshow(M1)

title("中值滤波图")

figure(4)

imshow(f1)

title("自适应中值滤波图")

fuction：

function f = adaptive\_median\_filter (g, Smax)

if (Smax <= 1) || (Smax/2 == round(Smax/2)) || (Smax ~= round(Smax))

error ('SMAX must be an odd integer > 1.')

end

f = g;

f(:) = 0;

alreadyProcessed = false (size(g));

for k = 3:2:Smax

zmin = ordfilt2(g, 1, ones(k, k),'symmetric');

zmax = ordfilt2(g, k \* k, ones(k, k), 'symmetric');

zmed = medfilt2(g, [k k], 'symmetric');

processUsingLevelB = (zmed > zmin) & (zmax > zmed) & ~alreadyProcessed;

zB = (g > zmin) & (zmax > g);

outputZxy = processUsingLevelB & zB;

outputZmed = processUsingLevelB & ~zB;

f (outputZxy) = g(outputZxy);

f (outputZmed) = zmed(outputZmed);

alreadyProcessed = alreadyProcessed | processUsingLevelB;

if all (alreadyProcessed (:))

break;

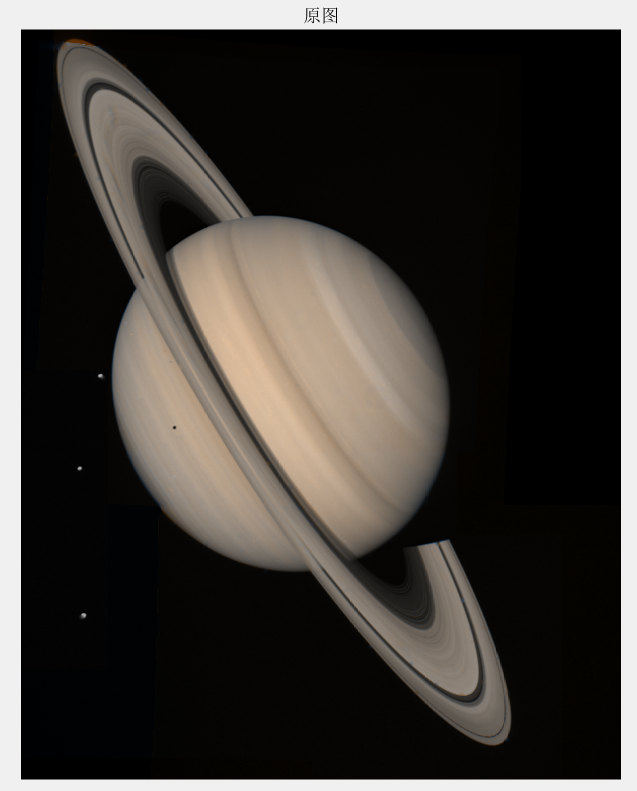
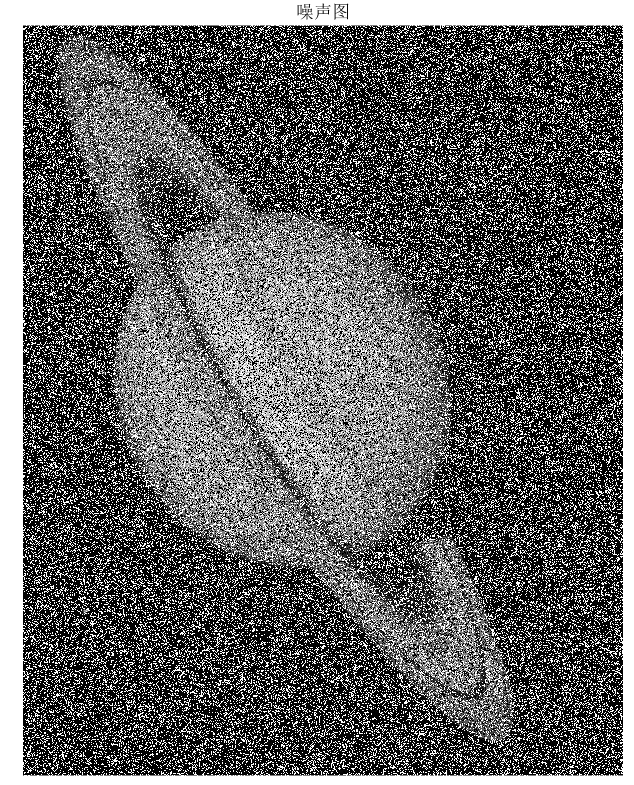
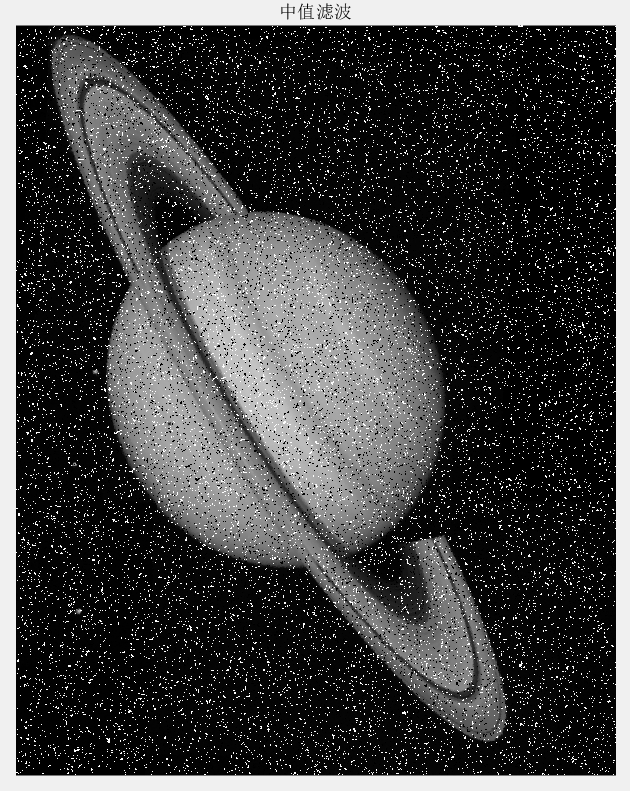
end

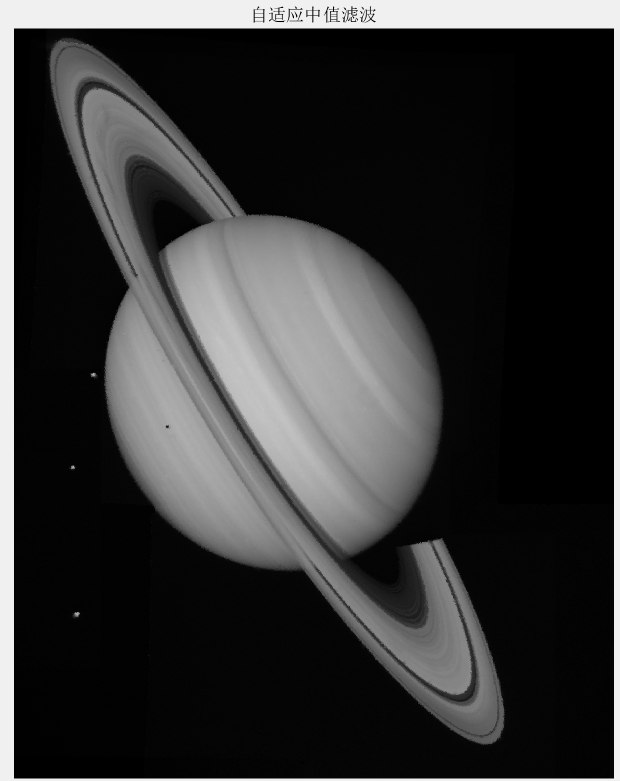
end

f (~alreadyProcessed) = zmed (~alreadyProcessed);

end

结果图：



（4）读入matlab自带的灰度图像“'cameraman.tif”，并对此图像进行len=21pixel， theta=45度的运动模糊。用逆滤波deconvwnr(K, PSF, 0)和维纳滤波deconvwnr(K, PSF, NSR2)分别进行复原，并比较复原效果。

附代码

Main：

clear

clc

im=im2double(imread('cameraman.tif') );

figure(1)

imshow(im)

title('原图')

LEN=21;

THETA=45;

PSF=fspecial('motion',LEN,THETA);

Blurred=imfilter(im,PSF,'circular','conv');

figure(2)

imshow(Blurred);

title('运动模糊图')

[m, n]=size(im);

u=-m/2:m/2-1;

v=-n/2:n/2-1;

[U, V]=meshgrid(u, v);

D=sqrt(U.^2+V.^2);

D0=35;

H=exp(-(D.^2)./(2\*(D0^2)));

N=0.01\*ones(size(im,1), size(im,2));

N=imnoise(N, 'gaussian', 0, 0.002);

J=fftfilter(im, H)+N;

HC=zeros(m, n);

M1=H>0.1;

HC(M1)=1./H(M1);

K=fftfilter(J, HC);

figure(3)

imshow(mat2gray(K),[])

title('逆滤波结果')

Q=0.05;

W=((H.^2)./(H.^2+Q))./H;

Q=fftfilter(J, W);

figure(4)

imshow(Q, [ ]);

title('维纳滤波结果');

fuction：

function Z=fftfilter(X,H)

F=fft2(X,size(H,1),size(H,2));

F=ifftshift(F);

Z=H.\*F;

Z=abs(ifft2(Z));

Z=Z(1:size(X,1),1:size(X,2));

end

结果图：



（5）读入matlab自带的灰度图像“'cameraman.tif”，并对此图像进行len=21pixel， theta=45度的运动模糊，并添加均值为0，方差较小0.0001的高斯噪声。用逆滤波deconvwnr(K, PSF, 0)和维纳滤波deconvwnr(K, PSF, NSR2)分别进行复原，并比较复原效果。

附代码

Main：

clear

clc

im=im2double(imread('cameraman.tif') );

figure(1)

imshow(im)

title('原图')

LEN=21;

THETA=45;

PSF=fspecial('motion', LEN, THETA);

J=imfilter(im, PSF, 'conv', 'circular');

noise\_mean=0;

noise\_var=0.0001;

K=imnoise(J, 'gaussian', noise\_mean, noise\_var);

figure(2)

imshow(K);

title('运动模糊噪声图')

NSR1=0;

L1=deconvwnr(K, PSF, NSR1);

NSR2=noise\_var/var(im(:));

L2=deconvwnr(K, PSF, NSR2);

figure(3)

imshow(L1)

title('逆滤波结果')

figure(4)

imshow(L2)

title('维纳滤波结果')

fuction：

function Z=fftfilter(X,H)

F=fft2(X,size(H,1),size(H,2));

F=ifftshift(F);

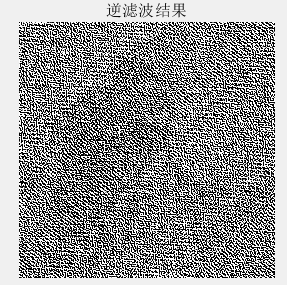
Z=H.\*F;

Z=abs(ifft2(Z));

Z=Z(1:size(X,1),1:size(X,2));

end

结果图：



（6）请比较（4）和（5）的贴图效果，如有不同，请说明贴图效果不同的原因。

回答：逆滤波对无噪声的图像具有很好的恢复能力，但是不稳定，如果加上各种噪声，会导致图像变得更加模糊。维纳滤波是效果最好的，并且对噪声的抑制能力最强，但是需要知道图像的信噪比，才能达到最好的效果