

**计算机与信息学院实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验课程： | 数字图像处理实验 | | | | |
| 实验编号： | 实验6 | | | | |
| 实验名称： | 图像的复原与重建 | | | | |
| 实验人员： | 学号 | 18111207248 | | | |
| 姓名 | 吴钰 | | | |
| 班级 | 计算机科学与技术（18创新班） | | | |
| 实验日期： | 2021/6/11 | | | | |
| 实验室： | 2070404 | | | | |
|  |  | | | | |
| 实验评价： |  | | | | |
| 实验成绩： | |  | 评价日期： |  |
|  | 指导教师： | |  | | |

# 图像空域滤波增强

一、实验目的

1．掌握在MATLAB中使用函数添加噪声的方法；

2．掌握常见图像复原方法的实现及应用。

二、实验要求

1．在MATLAB环境下实现图像的复原算法；

2．应用设计的算法实现对图像的复原。

三、实验内容

1．图像复原的空间滤波；

2．图像复原的频率域滤。

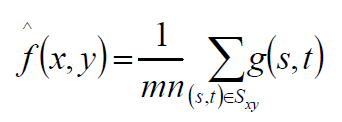
四、实验结果及分析

实验素材见实验六附件材料。

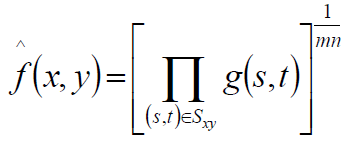
**1．图像复原的空间滤波**

**1）均值滤波复原**

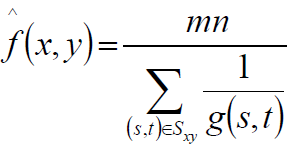
* **算术均值滤波复原**



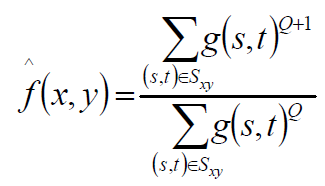
* **几何均值滤波复原**



* **谐波均值滤波复原**



* **逆谐波均值滤波复原**



（1）实验代码

**T1.m**

%图像复原的空间滤波

clc,clear

close all

ImagePath ='D:\大三（下）专业课\数字图像处理\Test6\image\11-orig.tif';

ImageGray = imread(ImagePath);

InputImagegray = im2double(ImageGray);

ImagePath ='D:\大三（下）专业课\数字图像处理\Test6\image\11-gauss-var.tif';

ImageGray = imread(ImagePath);

InputImage\_guass = im2double(ImageGray);

figure(1);

hold on

subplot(221);

imshow(InputImagegray);

title('原图');

subplot(222);

imshow(InputImage\_guass);

title('加入高斯噪声后的图像');

m=3;n=3;

image = MeanFilter\_1(InputImage\_guass,m,n);

subplot(223);

imshow(image);

title('算数均值滤波后图像');

image = MeanFilter\_2(InputImage\_guass,m,n);

subplot(224);

imshow(image);

title('几何均值滤波后图像');

hold off

figure(2);

hold on

subplot(221);

imshow(InputImagegray);

title('原图');

subplot(222);

imshow(InputImage\_guass);

title('加入高斯噪声后的图像');

m=9;n=9;

image = MeanFilter\_1(InputImage\_guass,m,n);

subplot(223);

imshow(image);

title('算数均值滤波后图像');

image = MeanFilter\_2(InputImage\_guass,m,n);

subplot(224);

imshow(image);

title('几何均值滤波后图像');

hold off

figure(3);

hold on

subplot(221);

imshow(InputImagegray);

title('原图');

subplot(222);

imshow(InputImage\_guass);

title('加入高斯噪声后的图像');

m=17;n=17;

image = MeanFilter\_1(InputImage\_guass,m,n);

subplot(223);

imshow(image);

title('算数均值滤波后图像');

image = MeanFilter\_2(InputImage\_guass,m,n);

subplot(224);

imshow(image);

title('几何均值滤波后图像');

hold off

%逆谐波均值滤波复原

ImagePath ='D:\大三（下）专业课\数字图像处理\Test6\image\22-pepper.tif';

ImageGray = imread(ImagePath);

InputImagegray = im2double(ImageGray);

figure(4);

hold on

subplot(232);

imshow(InputImagegray);

title('含有椒噪声的图像');

Q = 1.5;

image = MeanFilter\_3(InputImagegray,Q);

subplot(233);

imshow(image);

title('Q为正的逆谐滤波');

Q = -1.5;

image = MeanFilter\_3(InputImagegray,Q);

subplot(234);

imshow(image);

title('Q为负的逆谐滤波');

Q = 0;

image = MeanFilter\_3(InputImagegray,Q);

subplot(235);

imshow(image);

title('Q为0的逆谐滤波 => 算数均值');

Q = -1;

image = MeanFilter\_3(InputImagegray,Q);

subplot(236);

imshow(image);

title('Q为-1的逆谐滤波 => 谐波均值');

hold off

ImagePath ='D:\大三（下）专业课\数字图像处理\Test6\image\22-salt.tif';

ImageGray = imread(ImagePath);

InputImagegray = im2double(ImageGray);

figure(5);

hold on

subplot(232);

imshow(InputImagegray);

title('含有盐噪声的图像');

Q = 1.5;

image = MeanFilter\_3(InputImagegray,Q);

subplot(233);

imshow(image);

title('Q为正的逆谐滤波');

Q = -Q;

image = MeanFilter\_3(InputImagegray,Q);

subplot(234);

imshow(image);

title('Q为负的逆谐滤波');

Q = 0;

image = MeanFilter\_3(InputImagegray,Q);

subplot(235);

imshow(image);

title('Q为0的逆谐滤波 => 算数均值');

Q = -1;

image = MeanFilter\_3(InputImagegray,Q);

subplot(236);

imshow(image);

title('Q为-1的逆谐滤波 => 谐波均值');

hold off

**MeanFilter\_1.m**

%算术均值滤波器

function res = MeanFilter\_1(image,m,n)

%确定要扩展的行列数

lm=floor(m/2);

ln=floor(n/2);

%将原始图像进行扩展，这里采用了镜像扩展，以进行图像边缘计算

image\_pad=padarray(image,[lm,ln],'symmetric');

%获得扩展后的图像尺寸

[M,N]=size(image\_pad);

%存放处理后的图像

[MM,NN] = size(image);

res=zeros(MM,NN);

%逐点计算子窗口的算术平均

for i=1+lm:M-lm

for j=1+ln:N-ln

%从扩展图像中取出子图像

Block=image\_pad(i-lm:i+lm,j-ln:j+ln);

%求子窗口的算术平均

res(i-lm,j-ln)=mean(mean(Block));

end

end

**MeanFilter\_2.m**

function res = MeanFilter\_2(image,m,n)

%几何均值滤波器

%确定要扩展的行列数

lm=floor(m/2);

ln=floor(n/2);

%将原始图像进行扩展，这里采用了镜像扩展，以进行图像边缘计算

image\_pad=padarray(image,[lm,ln],'symmetric');

%获得扩展后的图像尺寸

[M,N]=size(image\_pad);

%存放处理后的图像

[MM,NN] = size(image);

res=zeros(MM,NN);

%逐点计算子窗口的几何平均

for i=1+lm:M-lm

for j=1+ln:N-ln

%从扩展图像中取出子图像

Block=image\_pad(i-lm:i+lm,j-ln:j+ln);

%求子窗口的几何均值

res(i-lm,j-ln)=(prod(prod(Block)))^(1/(m\*n));

end

end

**MeanFilter\_3.m**

%逆谐波均值滤波器

function res = MeanFilter\_3(image,Q)

%确定要扩展的行列数

m=3;n=3;

lm=floor(m/2);

ln=floor(n/2);

%将原始图像进行扩展，这里采用了镜像扩展，以进行图像边缘计算

image\_pad=padarray(image,[lm,ln],'symmetric');

%获得扩展后的图像尺寸

[M,N]=size(image\_pad);

%存放处理后的图像

[MM,NN] = size(image);

res=zeros(MM,NN);

%逐点计算子窗口的谐波平均

for i=1+lm:M-lm

for j=1+ln:N-ln

%从扩展图像中取出子图像

Block=image\_pad(i-lm:i+lm,j-ln:j+ln);

%求子窗口的谐波平均

s1=sum(Block(:).^(Q+1));

s2=sum(Block(:).^Q);

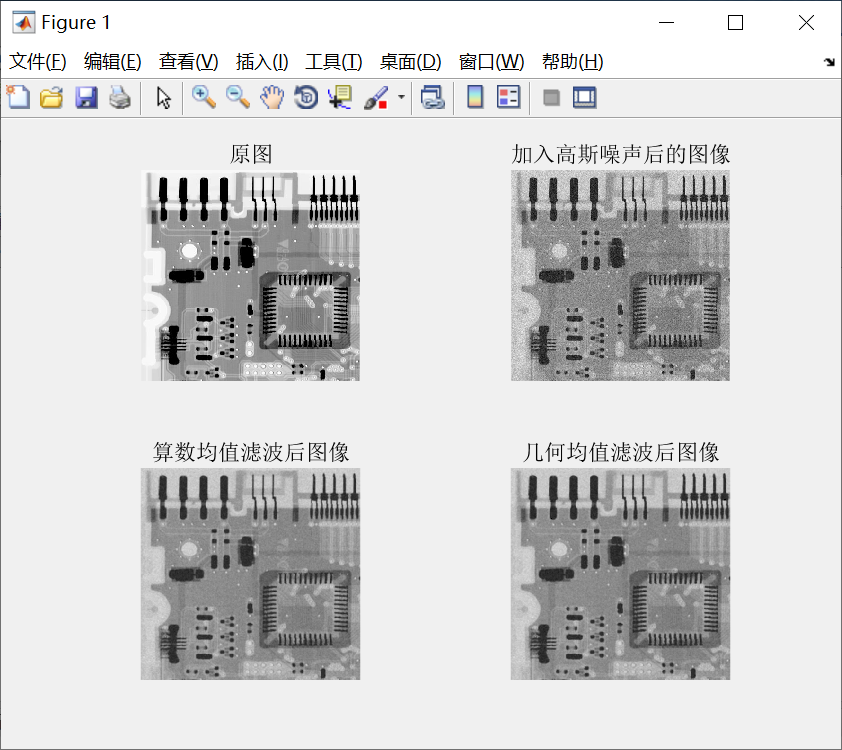
res(i-lm,j-ln)=s1/s2;

end

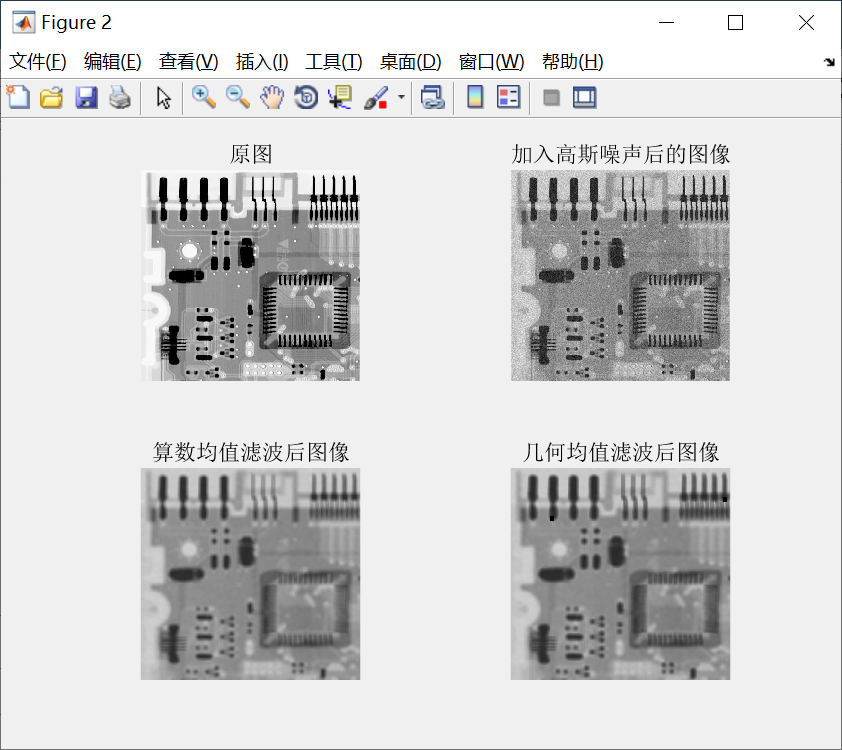
end

（2）实验结果（图示）

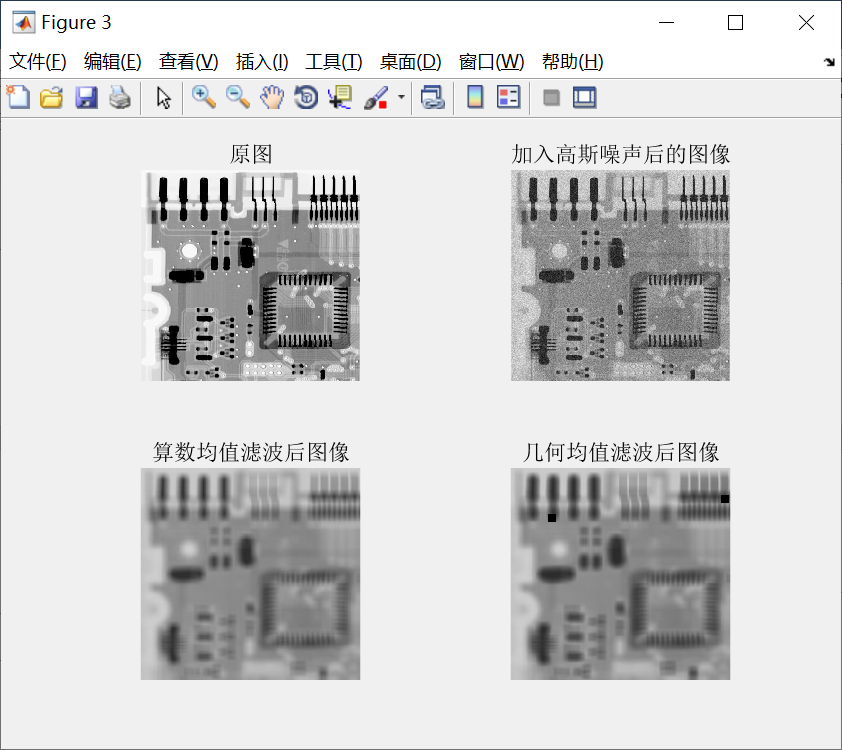
①m = 3; n = 3



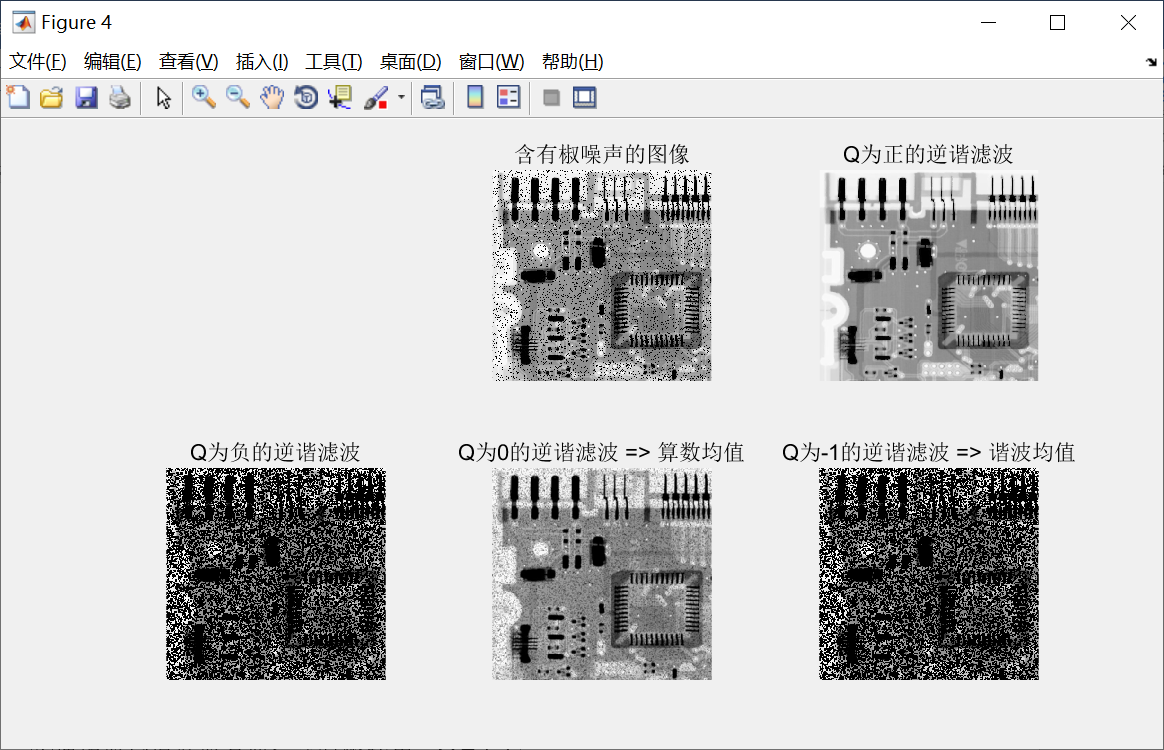
②m = 9; n = 9



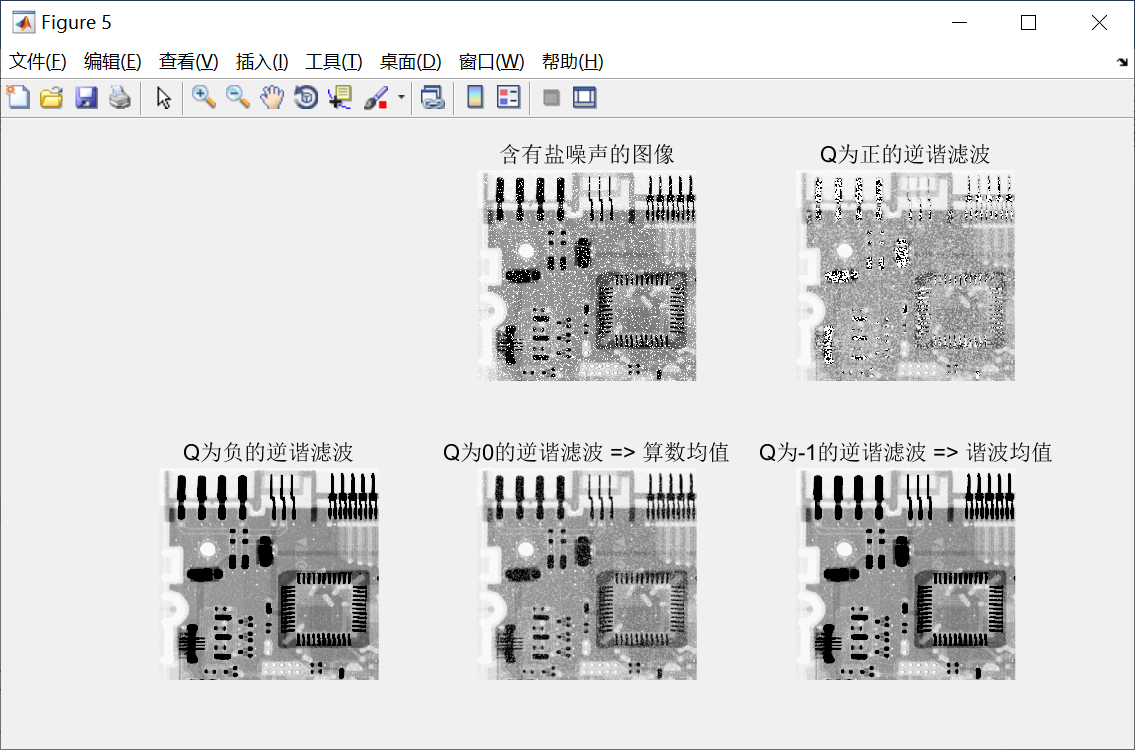
③m = 17; n = 17



④逆谐波均值滤波复原：含有椒噪声，Q = 1.5



⑤逆谐波均值滤波复原：含有盐噪声，Q = 1.5



（3）实验结果分析

Q 称为滤波器的阶数。

这种滤波器适合减少或是在实际中消除椒盐噪声的影响。

当 Q 值为正数时，滤波器用于消除“胡椒”噪声；

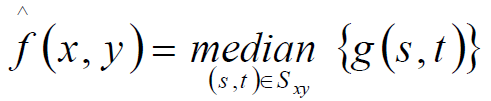
当 Q 值为负数时，滤波器用于消除“盐”噪声；

当 Q=0 时，逆谐波均值滤波器退化为算术均值滤波器；

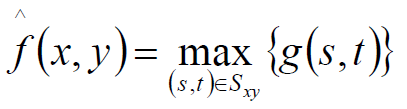
当 Q=-1 时，逆谐波均值滤波器退化为谐波均值滤波器。

**2）顺序统计滤波复原**

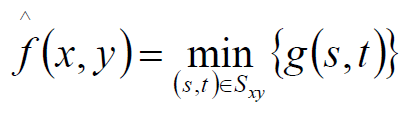
* **中值滤波复原**



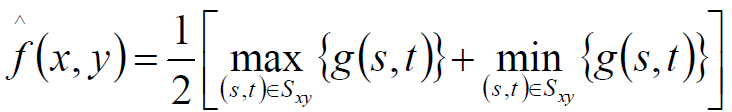
* **最大值滤波复原**



* **最小值滤波复原**



* **中点滤波复原**



（1）实验代码

**T1\_2.m**

%顺序统计滤波复原

%中值滤波复原1

ImagePath ='D:\大三（下）专业课\数字图像处理\Test6\image\cameraman.tif';

ImageGray = imread(ImagePath);

InputImagegray = rgb2gray(ImageGray);

figure(1);

hold on

subplot(231);

imshow(InputImagegray);

title('原图');

InputImagegray\_temp = imnoise(InputImagegray,'salt & pepper',0.01);

subplot(232);

imshow(InputImagegray\_temp);

title('添加椒盐噪声的图像');

image = SequentialFilter(InputImagegray\_temp,1);

subplot(233);

imshow(image);

title('中值滤波复原');

image = SequentialFilter(InputImagegray\_temp,2);

subplot(234);

imshow(image);

title('中点滤波复原');

image = SequentialFilter(InputImagegray\_temp,3);

subplot(235);

imshow(image);

title('最小值滤波复原');

image = SequentialFilter(InputImagegray\_temp,4);

subplot(236);

imshow(image);

title('最大值滤波复原');

%中值滤波复原2

ImagePath ='D:\大三（下）专业课\数字图像处理\Test6\image\cameraman.tif';

ImageGray = imread(ImagePath);

InputImagegray = rgb2gray(ImageGray);

figure(2);

hold on

subplot(231);

imshow(InputImagegray);

title('原图');

InputImagegray\_temp = imnoise(InputImagegray,'gauss',0.01);

subplot(232);

imshow(InputImagegray\_temp);

title('添加高斯噪声的图像');

image = SequentialFilter(InputImagegray\_temp,1);

subplot(233);

imshow(image);

title('中值滤波复原');

image = SequentialFilter(InputImagegray\_temp,2);

subplot(234);

imshow(image);

title('中点滤波复原');

image = SequentialFilter(InputImagegray\_temp,3);

subplot(235);

imshow(image);

title('最小值滤波复原');

image = SequentialFilter(InputImagegray\_temp,4);

subplot(236);

imshow(image);

title('最大值滤波复原');

%中值滤波复原3

ImagePath ='D:\大三（下）专业课\数字图像处理\Test6\image\11-orig.tif';

ImageGray = imread(ImagePath);

InputImagegray = ImageGray;

figure(3);

hold on

subplot(231);

imshow(InputImagegray);

title('原图');

InputImagegray\_temp = imread('D:\大三（下）专业课\数字图像处理\Test6\image\22-salt.tif');

subplot(232);

imshow(InputImagegray\_temp);

title('添加盐噪声的图像');

image = SequentialFilter(InputImagegray\_temp,1);

subplot(233);

imshow(image);

title('中值滤波复原');

image = SequentialFilter(InputImagegray\_temp,2);

subplot(234);

imshow(image);

title('中点滤波复原');

image = SequentialFilter(InputImagegray\_temp,3);

subplot(235);

imshow(image);

title('最小值滤波复原');

image = SequentialFilter(InputImagegray\_temp,4);

subplot(236);

imshow(image);

title('最大值滤波复原');

%中值滤波复原4

ImagePath ='D:\大三（下）专业课\数字图像处理\Test6\image\11-orig.tif';

ImageGray = imread(ImagePath);

InputImagegray = ImageGray;

figure(4);

hold on

subplot(231);

imshow(InputImagegray);

title('原图');

InputImagegray\_temp = imread('D:\大三（下）专业课\数字图像处理\Test6\image\22-pepper.tif');

subplot(232);

imshow(InputImagegray\_temp);

title('添加椒噪声的图像');

image = SequentialFilter(InputImagegray\_temp,1);

subplot(233);

imshow(image);

title('中值滤波复原');

image = SequentialFilter(InputImagegray\_temp,2);

subplot(234);

imshow(image);

title('中点滤波复原');

image = SequentialFilter(InputImagegray\_temp,3);

subplot(235);

imshow(image);

title('最小值滤波复原');

image = SequentialFilter(InputImagegray\_temp,4);

subplot(236);

imshow(image);

title('最大值滤波复原');

%中值滤波复原5

ImagePath ='D:\大三（下）专业课\数字图像处理\Test6\image\11-orig.tif';

ImageGray = imread(ImagePath);

InputImagegray = ImageGray;

figure(5);

hold on

subplot(231);

imshow(InputImagegray);

title('原图');

InputImagegray\_temp = imnoise(InputImagegray,'salt & pepper',0.1);

subplot(232);

imshow(InputImagegray\_temp);

title('添加椒盐噪声的图像');

image = SequentialFilter(InputImagegray\_temp,1);

subplot(233);

imshow(image);

title('中值滤波复原');

image = SequentialFilter(InputImagegray\_temp,2);

subplot(234);

imshow(image);

title('中点滤波复原');

image = SequentialFilter(InputImagegray\_temp,3);

subplot(235);

imshow(image);

title('最小值滤波复原');

image = SequentialFilter(InputImagegray\_temp,4);

subplot(236);

imshow(image);

title('最大值滤波复原');

**SequentialFilter.m**

function res = SequentialFilter(image,flag)

m=3;

[x,y] = size(image);

a = floor((m-1)/2);

res = image;

for i = 1+a:x-a

for j = 1+a:y-a

if flag ==1 %中值滤波

temp = image(i-a:i+a,j-a:j+a);

temp = sort(temp(:));

res(i,j) = temp((m^2-1)/2,1);

elseif flag == 2 %中点滤波

temp = image(i-a:i+a,j-a:j+a);

res(i,j) = (max(temp(:))+min(temp(:)))/2;

elseif flag == 3 %最小值滤波

temp = image(i-a:i+a,j-a:j+a);

res(i,j) = min(temp(:));

elseif flag == 4 %最大值滤波

temp = image(i-a:i+a,j-a:j+a);

res(i,j) = max(temp(:));

end

end

end

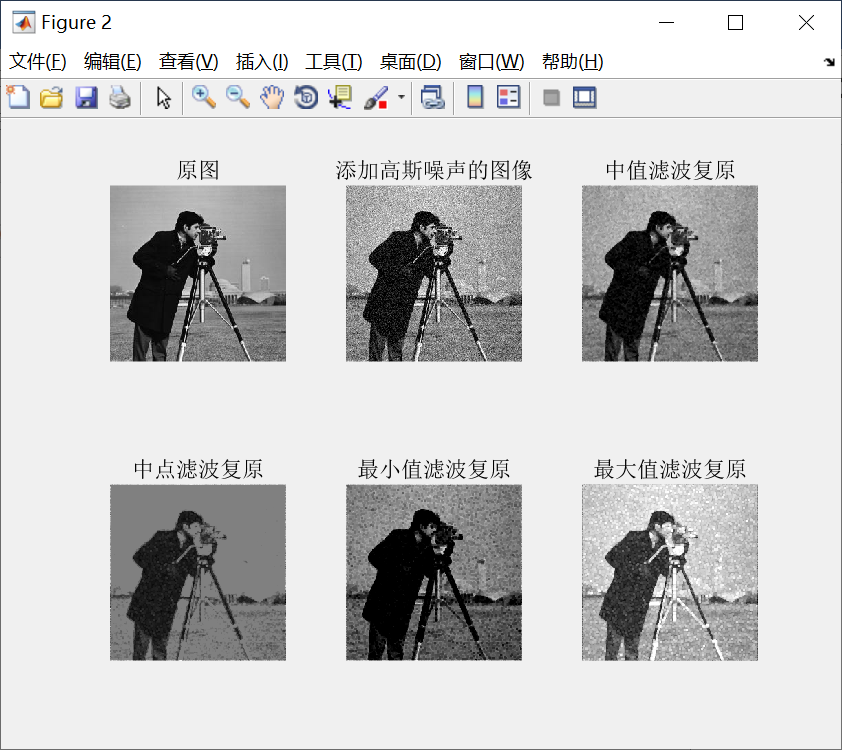
end

（2）实验结果（图示）

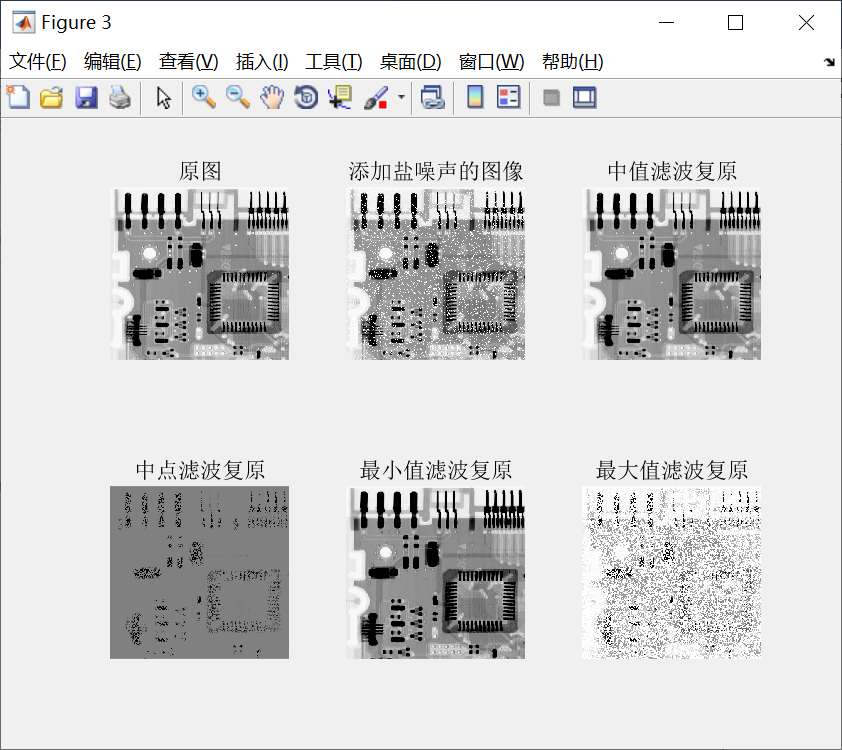
①对图例cameraman.tif添加椒盐噪声后的顺序统计滤波复原



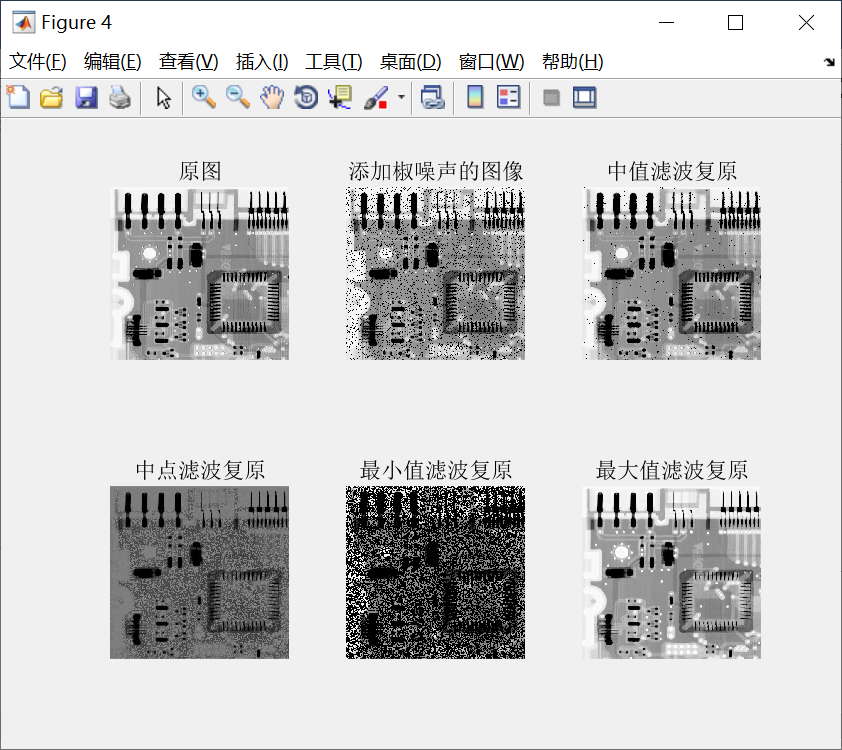
②对图例cameraman.tif添加高斯噪声后的顺序统计滤波复原



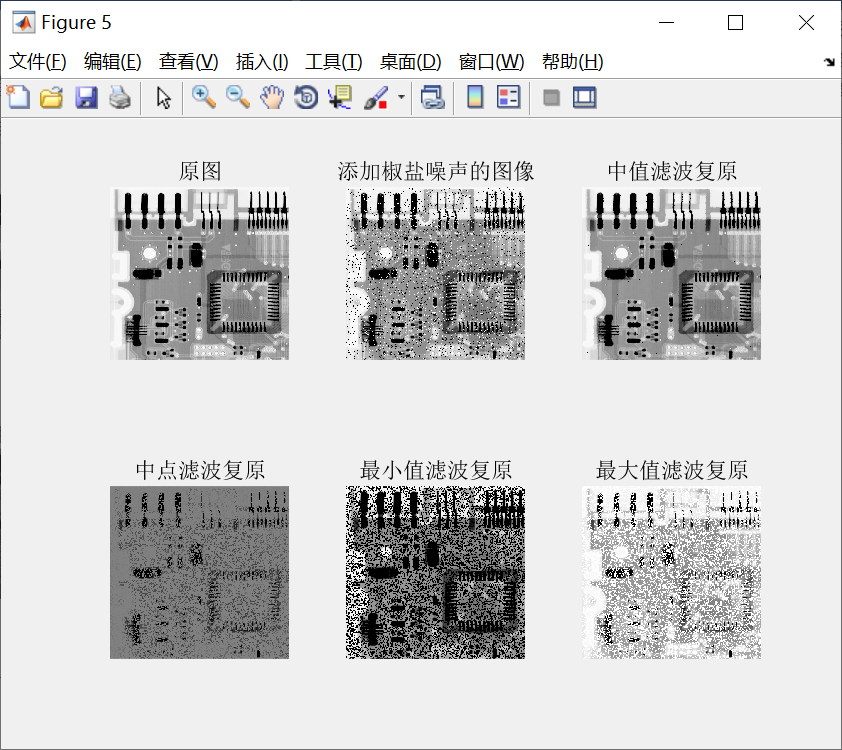
③对图例22-salt.tif的顺序统计滤波复原



④对图例22-pepper.tif的顺序统计滤波复原



⑤对图例22-orig.tif添加椒盐噪声后的顺序统计滤波复原



（3）实验结果分析

①中点滤波器在滤波器涉及范围内计算最大值和最小值之间的中点，这种滤波器结合了顺序统计和求平均，对于高斯和均匀随机分布这类噪声有最好的结果。

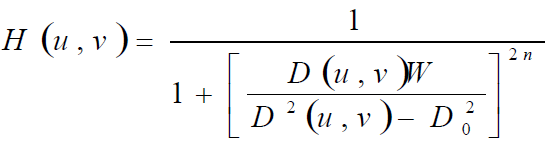
②修正后的阿尔法均值滤波器d 值可取 0 到 mn-1 之间的任意数。

当 d=0 时，修正的阿尔法均值滤波器退变为算术均值滤波器。

d=(mn-1)/2，修正后的阿尔法均值滤波器退变为中值滤波器。

d 取其他值时，修正后的阿尔法均值滤波器在包括多种噪声的情况非常适用，如高斯噪声和椒盐噪声混合的情况下。

**2．图像复原的频率域滤-** **n阶的巴特沃思带阻滤波器**



（1）实验代码

**T2.m**

%图像复原的频率域滤

ImagePath ='D:\大三（下）专业课\数字图像处理\实验6\_yy\image\33-orig.bmp';

ImageGray = imread(ImagePath);

Image=im2double(ImageGray);

ImagePath ='D:\大三（下）专业课\数字图像处理\实验6\_yy\image\33-noise.bmp';

ImageGray = imread(ImagePath);

InputImagegray=im2double(ImageGray);

[x,y] = size(InputImagegray);

imageF=fft2(Image); %二维傅立叶变换

imageF=fftshift(imageF); %频移

%设计巴特沃斯滤波器

n = 4; %巴特沃斯滤波器阶数为4

D0 = 0.2 \* x; %截止频率20%的图像宽度，内圈

W = 80; %归一化通带截止频率，外圈

[Z,image] = ButterworthFilter(InputImagegray,D0,n,W);

[y,x] = size(image);

figure(1);

hold on

subplot(221);

imshow(Image);

title('原图');

subplot(222);

imshow(InputImagegray);

title('加入正弦噪声后的图像');

subplot(223);

imshow(log(abs(imageF)),[]);

title('傅里叶频谱图');

subplot(224);

imshow(image);

title('降噪后的图像');

hold off

figure(2);

subplot(121);

[X,Y] = meshgrid(1:x,1:y); %设置三维图像的X,Y坐标

surf(X,Y,Z);

shading interp

set(gca,'ZTick',0:0.2:1);

xlabel('X'),ylabel('Y'),zlabel('Z')

title('三维图');

subplot(122);

surf(X,Y,Z);

shading interp

axis([0,x,0,y]);

xlabel('X'),ylabel('Y')

title('二维图');

**ButterworthFilter.m**

function [H,res] = ButterworthFilter(image,D0,n,W)

image=fft2(image); %二维傅立叶变换

image=fftshift(image); %频移

[M,N]=size(image);

H = zeros(M,N);

res = zeros(M,N);

a=fix(M/2);

b=fix(N/2);

for i=1:M

for j=1:N

D=sqrt((i-a)^2+(j-b)^2);

H(i,j)=1/(1+(D\*W/(D^2-D0^2))^(2\*n)); %计算带阻滤波器传递函数

res(i,j)=H(i,j)\*image(i,j); %用设计的滤波器处理原图像

end

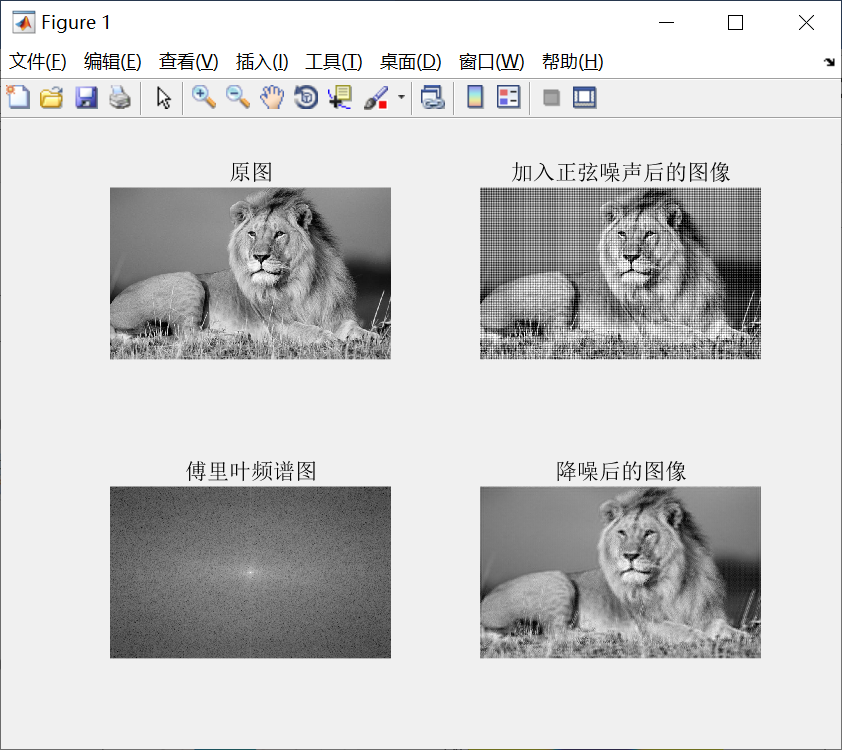
end

%傅里叶反变换

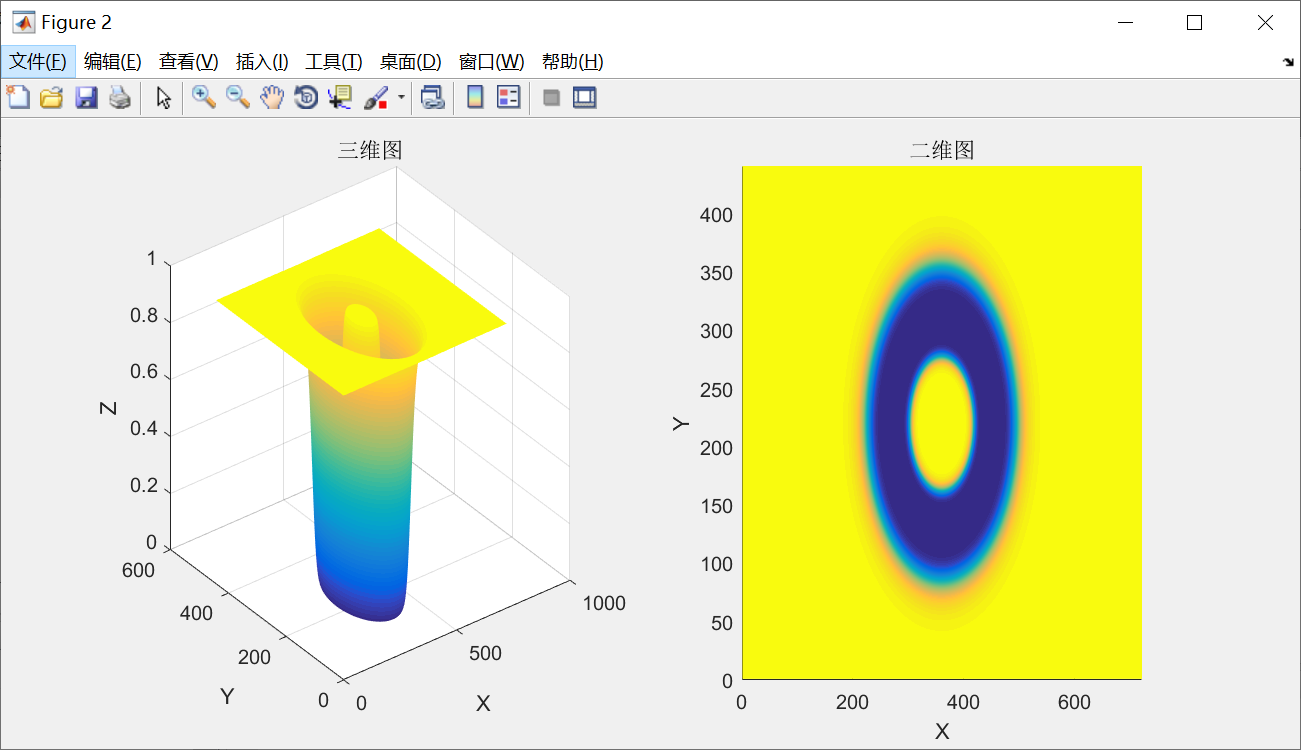
res=im2uint8(real(ifft2(ifftshift(res))));

（2）实验结果（图示）

①对图例33-orig.bmp添加正弦噪声后的巴特沃思带阻滤波复原



②4阶的巴特沃思带阻滤波器，D0 = 20%\*图像宽度，W = 80



（3）实验结果分析

巴特沃斯滤波器的特点是通频带内的频率响应曲线最大限度平坦，没有起伏，而在阻频带则逐渐下降为零。在振幅的对数对角频率的波特图上，从某一边界角频率开始，振幅随着角频率的增加而逐步减少，趋向负无穷大。

一阶巴特沃斯滤波器的衰减率为每倍频6分贝，每十倍频20分贝。二阶巴特沃斯滤波器的衰减率为每倍频12分贝、三阶巴特沃斯滤波器的衰减率为每倍频18分贝、如此类推。巴特沃斯滤波器的振幅对角频率单调下降，并且也是唯一的无论阶数，振幅对角频率曲线都保持同样的形状的滤波器。只不过滤波器阶数越高，在阻频带振幅衰减速度越快。其他滤波器高阶的振幅对角频率图和低级数的振幅对角频率有不同的形状。可见阶数n越高，其幅频特性越好，低频检测信号保真度越高。