

**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3 | | | | |
| 实验名称： | 图像空域滤波增强 | | | | |
| 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： | 2021.5.28 | | | | |
| 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．理解图像空间滤波的基本原理；

2．掌握常见空间滤波器的设计及使用。

二、实验要求

1．在MATLAB环境下实现图像的空域增强算法；

2．应用设计的算法实现对图像的增强。

三、实验内容

1．平滑空间滤波器的设计及实现；

2．锐化空间滤波器的设计及实现；

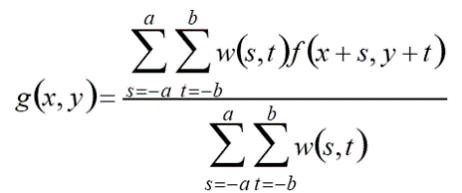
3．图像空域滤波增强。

四、实验结果及分析

实验素材见实验四附件材料。

**1．平滑空间滤波器的设计及实现**

**（1）线性滤波器**



1）实验代码

% 线性滤波器验证实验1

**T1\_1**

ImagePath ='D:\大三（下）专业课\数字图像处理\Test4\image\11.tif';

InputImage = imread(ImagePath);

InputImageGray = double(InputImage);

figure(1);

subplot(2,3,1);

imshow(uint8(InputImageGray));

title('原图');

model = [1 2 1;2 4 2;1 2 1];

image = func1(InputImageGray,model);

subplot(2,3,2);

imshow(image);

title('3 X 3');

model = [2 4 5 4 2;4 9 12 9 4;5 12 15 12 5;4 9 12 9 4;2 4 5 4 2;];

image = func1(InputImageGray,model);

subplot(2,3,3);

imshow(image);

title('5 X 5');

model = ones(9,9); %3X3模板

image = func1(InputImageGray,model);

subplot(2,3,4);

imshow(image);

title('9 X 9');

model = ones(15,15);

image = func1(InputImageGray,model);

subplot(2,3,5);

imshow(image);

title('15 X 15');

model = ones(35,35);

image = func1(InputImageGray,model);

subplot(2,3,6);

imshow(image);

title('35 X 35');

**func1.m**

function res = func1(image,model)

[x,y] = size(image);

[~,m] = size(model);

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

res = image;

for i = 1+a:x-a

for j = 1+a:y-a

res(i,j)=sum(image(i-a:i+a,j-a:j+a).\*model)/sum(model);

end

end

res = uint8(res);

end

**% 线性滤波器验证实验2**

ImagePath ='D:\大三（下）专业课\数字图像处理\Test4\image\22.tif';

InputImage = imread(ImagePath);

InputImageGray = double(InputImage);

figure(2);

subplot(1,3,1);

imshow(uint8(InputImageGray));

title('原图');

model = ones(15,15);

image = func1(InputImageGray,model);

subplot(1,3,2);

imshow(image);

title('15 X 15');

maxImg = max(image)\*0.25;

for i = 1:size(image,1)

for j = 1:size(image,2)

if image(i,j)>=maxImg;

image(i,j) = 255;

else

image(i,j) = 0;

end

end

end

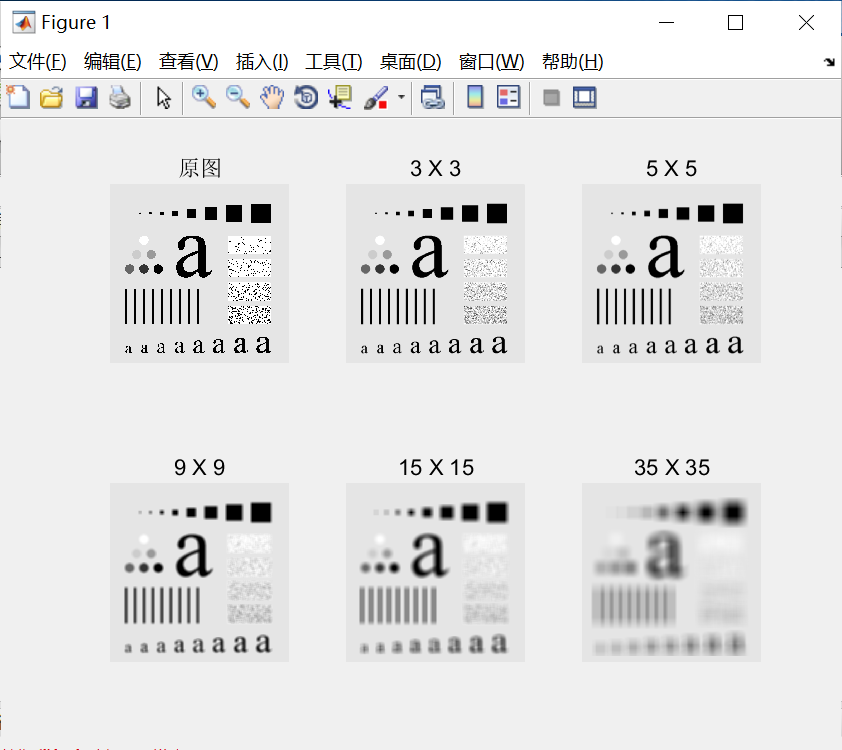
subplot(1,3,3);

imshow(image);

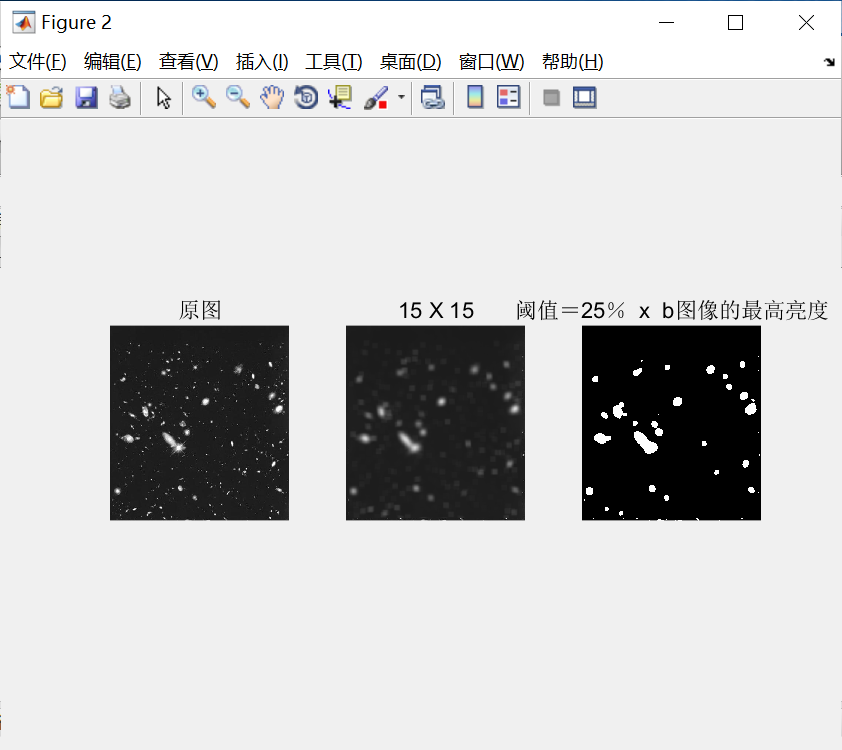
title('阈值＝25％ x b图像的最高亮度');

2）实验结果（图示）

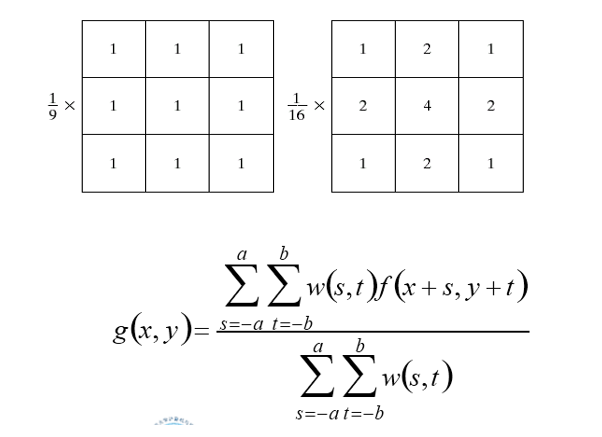
验证实验一



验证试验二



3）实验结果分析



平滑滤波器用于模糊处理和减小噪声。模糊处理经常用于预处理。平滑线性空间滤波器的输出(响应)是包含在滤波掩膜领域内像素的简单平均值。因此这些滤波器也称为均值滤波器，也指的是低通滤波器。它用滤波掩膜确定的领域内像素的平均灰度值去代替图像每个像素点的值，这种处理减小了图像灰度的“尖锐”变化。由于典型的随机噪声由灰度级的尖锐变化组成，因此，常见的平滑处理应用就是减噪。然而，由于图像边缘也是由图像灰度尖锐变化带来的特性，所以均值滤波器处理还是存在着不希望的边缘模糊的负面效应。均值滤波器的主要应用是去除图像中的不相干的细节，其中“不相干”是指与滤波掩膜尺寸相比，较小的像素区域。

**（2）统计排序滤波器**

* 中值滤波器；
* 最大值滤波器；
* 最小值滤波器；

1）实验代码

**验证实验一**

**statistic\_1.m**

close all;

clear all;

clc;

image = imread('D:\大三（下）专业课\数字图像处理\Test4\image\33.tif');

[m, n] = size(image);

%模板大小

k = 3;

myFilter = zeros(k, k, 'uint8');

for i = 1 : k

for j = 1 : k

myFilter(i, j) = 1;

end

end

image2=spacelFilter(image, myFilter);

image3=spacelFilter\_zz(image, myFilter);

image4 = medfilt2(image, [3 3]);

figure;

subplot(2, 2, 1), imshow(image), title('原图像');

subplot(2, 2, 2), imshow(image2), title('3\*3均值滤波图像');

subplot(2, 2, 3), imshow(image3), title('3\*3中值滤波图像');

**spacelFilter.m**

function [image\_out] = spacelFilter(image\_in, filter)

% 图像边缘的填充为最近的像素值，目的是消除填充0时会出现的黑框

% 滤波器的大小为 n \* n, n = 2 \* k + 1, k为整数

% 输入图像大小为 m \* n, 灰度图像，像素值范围为0-255，L = 256

[m, n] = size(image\_in);

[mf, nf] = size(filter);

k = (mf - 1) / 2;

image2 = zeros(m+2\*k, n+2\*k, 'double');

image\_out = zeros(m, n, 'uint8');

coeff = sum(filter(:));

% 填充部分

% 内部直接复制

for i = 1+k : m+k

for j = 1+k : n+k

image2(i, j) = image\_in(i-k, j-k);

end

end

% 填充上下边缘

for i = 1 : k

for j = 1 : n

image2(i, j+k) = image\_in(1, j);

image2(m+k+i, j+k) = image\_in(m, j);

end

end

% 填充左右边缘

for i = 1 : m

for j = 1 : k

image2(i+k, j) = image\_in(i, 1);

image2(i+k, n+k+j) = image\_in(i, n);

end

end

% 填充四个角

for i = 1 : k

for j = 1 : k

image2(i, j) = image\_in(1, 1); %填充左上角

image2(i, j+n+k) = image\_in(1, n); %填充右上角

image2(i+n+k, j) = image\_in(m, 1); %填充左下角

image2(i+n+k, j+n+k) = image\_in(m, n); %填充右下角

end

end

% 滤波部分

for i = 1+k : m+k

for j = 1+k : n+k

sub\_image = image2(i-k:i+k, j-k:j+k);

temp1 = double(filter') .\* double(sub\_image);

temp2 = sum(temp1(:)) / coeff;

image\_out(i-k, j-k) = uint8(temp2);

end

end

end

**中值滤波器的滤波部分**

% 滤波部分

for i = 1+k : m+k

for j = 1+k : n+k

sub\_image = image2(i-k:i+k, j-k:j+k);

temp = median(sub\_image(:));

image\_out(i-k, j-k) = uint8(temp);

end

end

**% 统计滤波器验证实验2**

ImagePath ='D:\大三（下）专业课\数字图像处理\Test4\image\44.jpg'; %图片路径

InputImage = imread(ImagePath);

InputImageGray = double(InputImage);

figure(4);

subplot(2,2,1);

imshow(uint8(InputImageGray));

title('原图');

image = func2(InputImageGray,3,1);

subplot(2,2,2);

imshow(image);

title('Ymed 3 X 3');

image = func2(InputImageGray,3,3);

subplot(2,2,3);

imshow(image);

title('Ymin 3 X 3');

image = func2(InputImageGray,3,2);

subplot(2,2,4);

imshow(image);

title('Ymax 3 X 3');

**func2.m**

function res = func2(image,m,flag)

[x,y] = size(image);

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

res = image;

for i = 1+a:x-a

for j = 1+a:y-a

if flag ==1

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

elseif flag == 2

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

elseif flag == 3

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

end

end

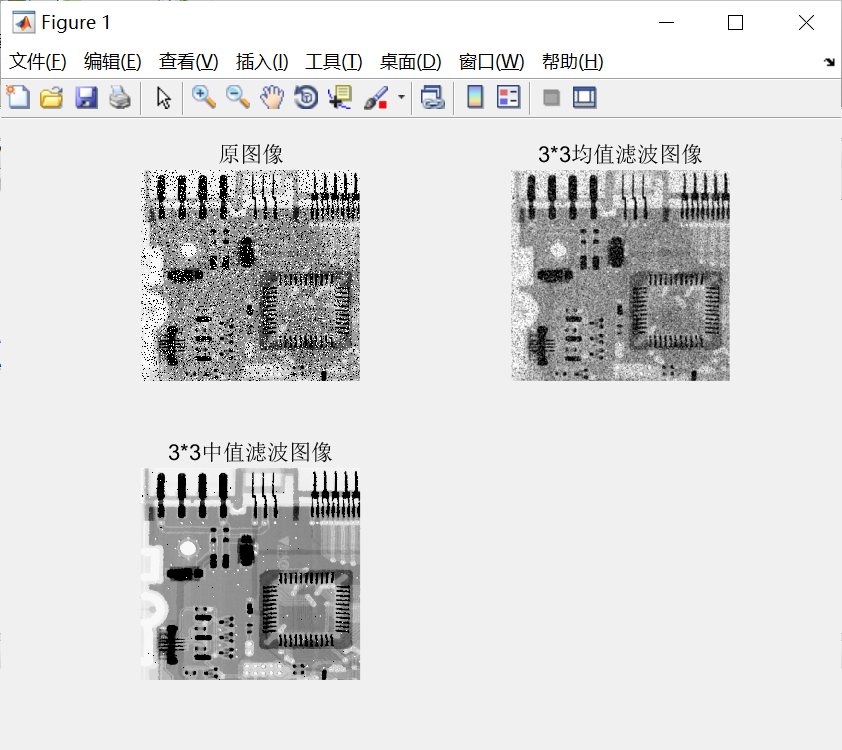
end

res = uint8(res);

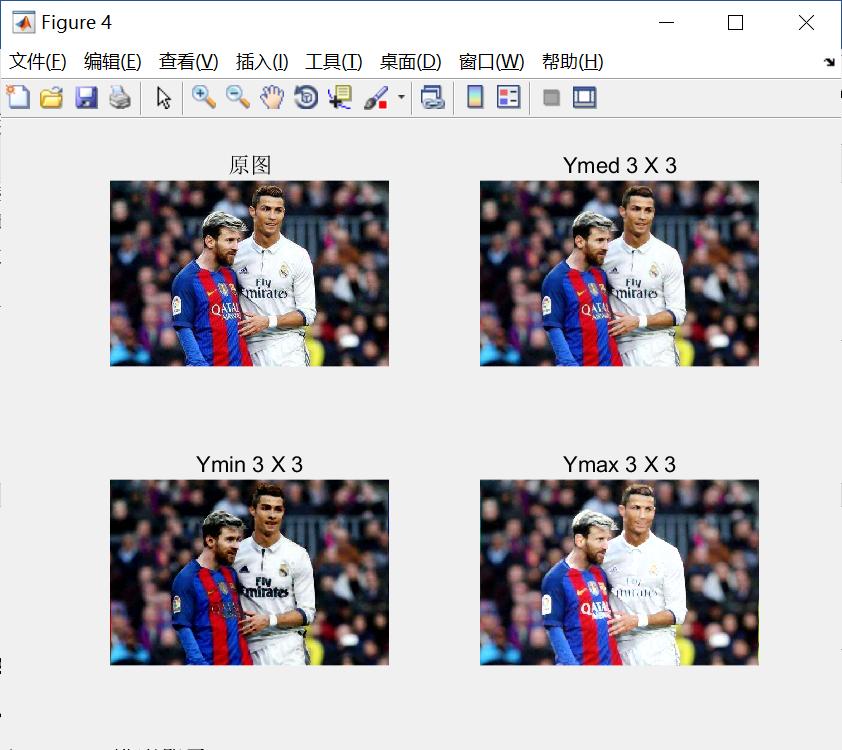
end

2）实验结果（图示）

**验证实验一**



验证实验二



3）实验结果分析

中值滤波器：用像素邻域内的中间值代替该像素

（线性）均值滤波函数， 输入为需要进行空间滤波的灰度图像，线性方形滤波器，输出为经过滤波之后的图像，图像边缘的填充为最近的像素值，目的是消除填充0时会出现的黑框，滤波器的大小为 n \* n, n = 2 \* k + 1, k为整数。输入图像大小为 m \* n, 灰度图像，像素值范围为0-255，L = 256

最大最小值滤波是一种比较保守的图像处理手段，与中值滤波类似，首先要排序周围像素和中心像素值，然后将中心像素值与最小和最大像素值比较，如果比最小值小，则替换中心像素为最小值，如果中心像素比最大值大，则替换中心像素为最大值。

**2．锐化空间滤波器**

* 拉普拉斯算子
* Prewitt 梯度算子
* Sobel 梯度算子

（1）实验代码

**验证实验一：**

%锐化滤波器验证实验一

ImagePath ='D:\大三（下）专业课\数字图像处理\Test4\image\55.tif';

InputImage = imread(ImagePath);

InputImageGray = double(InputImage);

figure(5);

subplot(1,3,1);

imshow(uint8(InputImageGray));

title('原图');

image = Laplace(InputImageGray,1);

subplot(1,3,2);

imshow(image);

title('Laplace');

image = uint8(InputImageGray)-image;

subplot(1,3,3);

imshow(image);

title('空间域锐化后图像');

**Laplace.m**

function res = Laplace(image,flag)

if flag == 1

model = ones(3,3);

model(2,2) = -8;

elseif flag == 2

model = [-1 -1 -1; 0 0 0; 1 1 1];

elseif flag == 3

model = [-1 0 1; -1 0 1; -1 0 1];

end

[x,y] = size(image);

[a,b] = size(model);

% 边界填充

imageT = zeros(x+a-1,y+b-1);

imageT(a/2+0.5:a/2-0.5+x,b/2+0.5:b/2-0.5+y) = image(:,:);

res = zeros(x+a-1,y+b-1);

for i=a/2+0.5:(a/2-0.5+x)

for j=b/2+0.5:(b/2-0.5+y)

res(i,j) = sum(sum(imageT(i-a/2+0.5:i+a/2-0.5,j-b/2+0.5:j+b/2-0.5).\*model));

end

end

% 去除边缘

res = res(a/2+0.5:a/2-0.5+x,b/2+0.5:b/2-0.5+y);

res = uint8(res);

end

**%锐化滤波器验证实验2**

ImagePath ='D:\大三（下）专业课\数字图像处理\Test4\image\66.tif';

InputImage = imread(ImagePath);

InputImageGray = double(InputImage);

figure(6);

subplot(2,3,1);

imshow(uint8(InputImageGray));

title('Original');

image = Laplace(InputImageGray,1);

subplot(2,3,2);

imshow(image);

title('Laplace');

image = uint8(InputImageGray)-image;

subplot(2,3,3);

imshow(image);

title('空间域锐化后');

image1 = Laplace(InputImageGray,2);

subplot(2,3,4);

imshow(image1);

title('PremageX');

image2 = Laplace(InputImageGray,3);

subplot(2,3,5);

imshow(image2);

title('PremageY');

image = image1+image2;

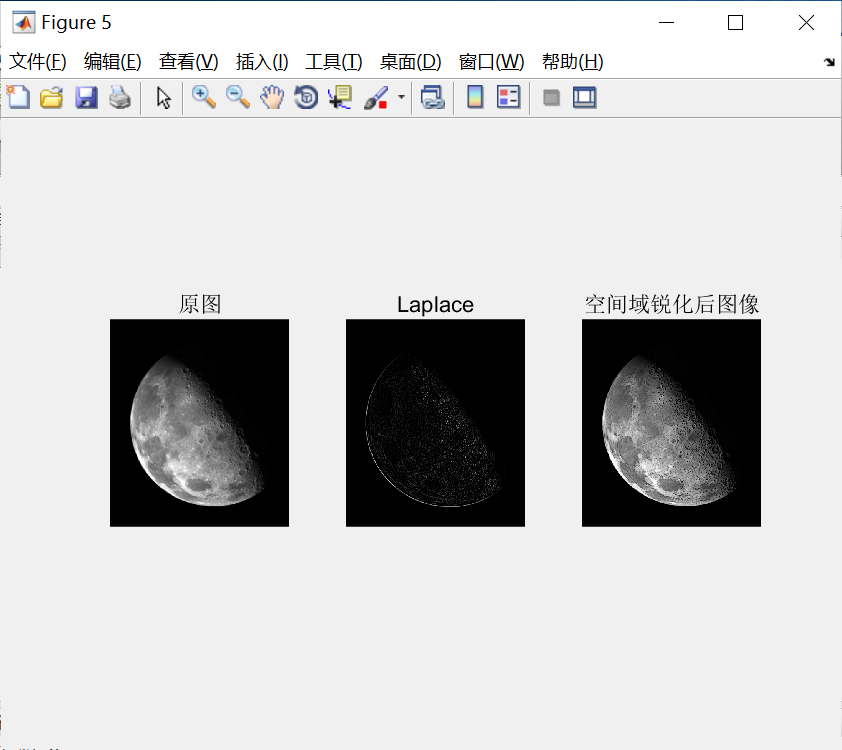
subplot(2,3,6);

imshow(image);

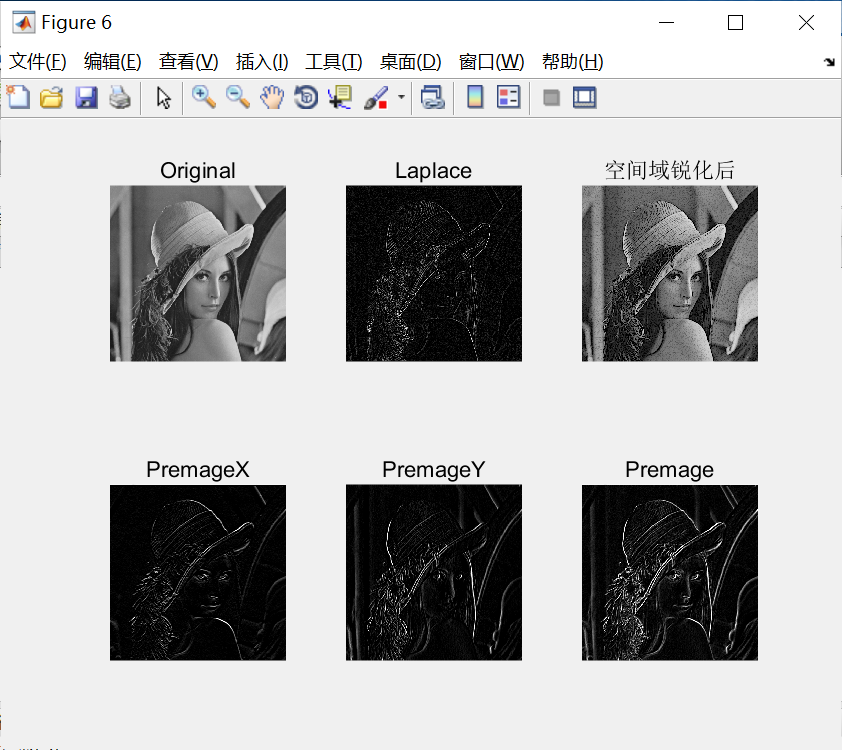
title('Premage');

（2）实验结果（图示）

验证实验一：拉普拉斯算子



验证实验二



（3）实验结果分析

锐化处理的主要目的是，突出灰度的过渡部分，增强图像中的细节。空间域像素邻域平均法可以使图像变模糊，均值处理与积分类似，所以锐化处理可以用空间微分（差分）来完成。

对比模糊：

模糊（平滑）是去除图像的细节，均值处理。

锐化是突出图像的细节，微分（差分）处理。

锐化滤波器主要有两种锐化方法：

1. 使用二阶微分的图像锐化：拉普拉斯锐化

2. 使用一阶微分的图像锐化：梯度锐化

拉普拉斯滤波处理后图像中既有正值又有负值，所有负值被修剪为0，很多经过拉普拉斯滤波后的图像大部分是黑色的。典型的标定拉普拉斯图像的方法是对他的最小值加一个新的代替0的最小值。