



华北电力大学  
NORTH CHINA ELECTRIC POWER UNIVERSITY

# 3 空间域滤波



# 主要内容

- 空间域滤波概述
  - ✓ 空间域滤波器
  - ✓ 模板卷积
  - ✓ 空间域滤波器分类
- 平滑滤波器
  - ✓ 均值滤波器
  - ✓ 加权均值滤波器
  - ✓ 中值滤波器
  - ✓ 最值滤波器



# 主要内容

- 锐化滤波器
  - ✓ 梯度滤波器
  - ✓ Roberts滤波器
  - ✓ Prewitt滤波器
  - ✓ Sobel滤波器
  - ✓ 拉普拉斯滤波器



## 3.3 锐化滤波器

- 锐化滤波器的主要用途
  - ✓ 突出图像中的细节，增强被模糊了的细节
  - ✓ 印刷中的细微层次强调，弥补扫描对图像的钝化
  - ✓ 超声探测成像，分辨率低，边缘模糊，通过锐化来改善
  - ✓ 图像识别中，分割前的边缘提取
  - ✓ 锐化处理恢复过度钝化、暴光不足的图像





## 3.3 锐化滤波器

- 锐化滤波器的工作原理

邻域平均方法—积分过程—结果使图像的边缘模糊

锐化方法—**微分**过程—结果使图像的边缘突出

**先去噪，再锐化操作**



## 3.3 锐化滤波器

- 锐化滤波器的分类

- ✓ 一阶微分滤波器

$$\nabla f = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}$$

- ✓ 二阶微分滤波器

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$



### 3.3.1 一阶微分滤波器

- 梯度滤波器

✓ 梯度用一个二维列向量来定义

$$\begin{aligned}\nabla f &= \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \\ \nabla f &= \text{mag}(\nabla f) \\ &= \left[ G_x^2 + G_y^2 \right]^{\frac{1}{2}} \\ &= \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{\frac{1}{2}}\end{aligned}$$

✓ 实际中往往用梯度模值代替梯度为减少计算量

$$\nabla f \approx |G_x| + |G_y|$$



### 3.3.1 一阶微分滤波器

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

- 梯度滤波器

✓ 点 $z_5$ 的 $\nabla f$ 值可用数字方式近似。

$$\nabla f = |z_6 - z_5| + |z_8 - z_5|$$

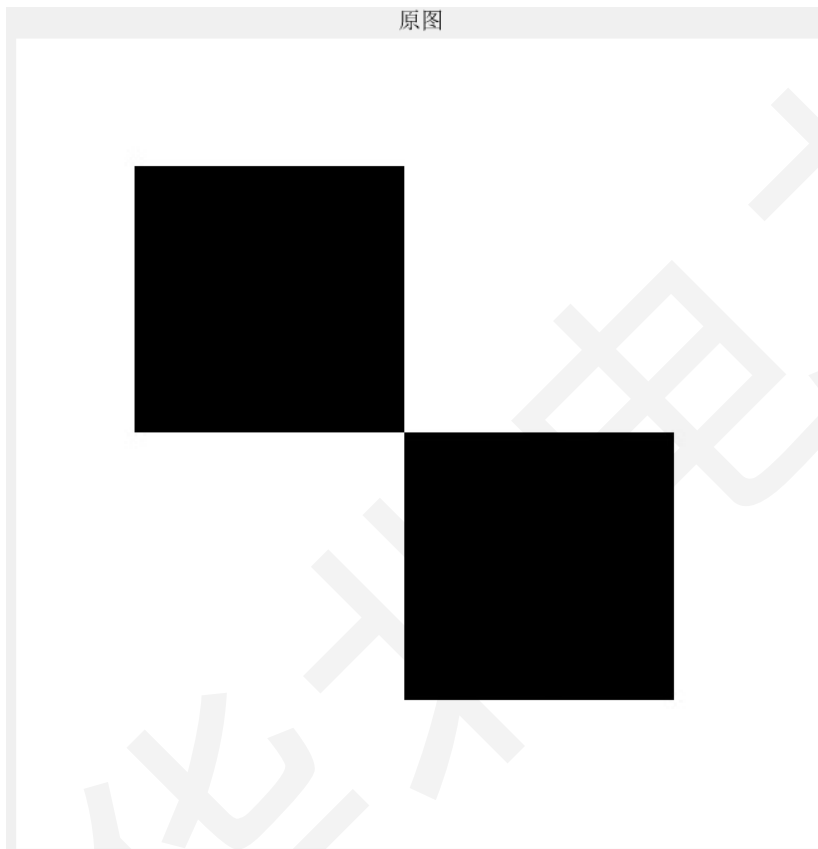




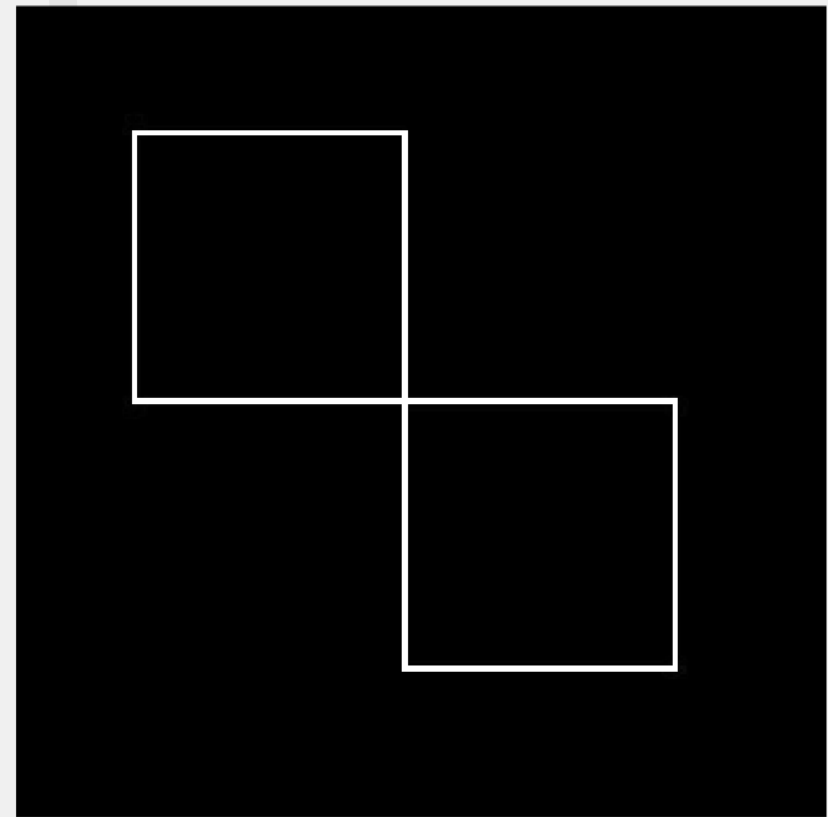
### 3.3.1 一阶微分滤波器

- 梯度滤波器

原图



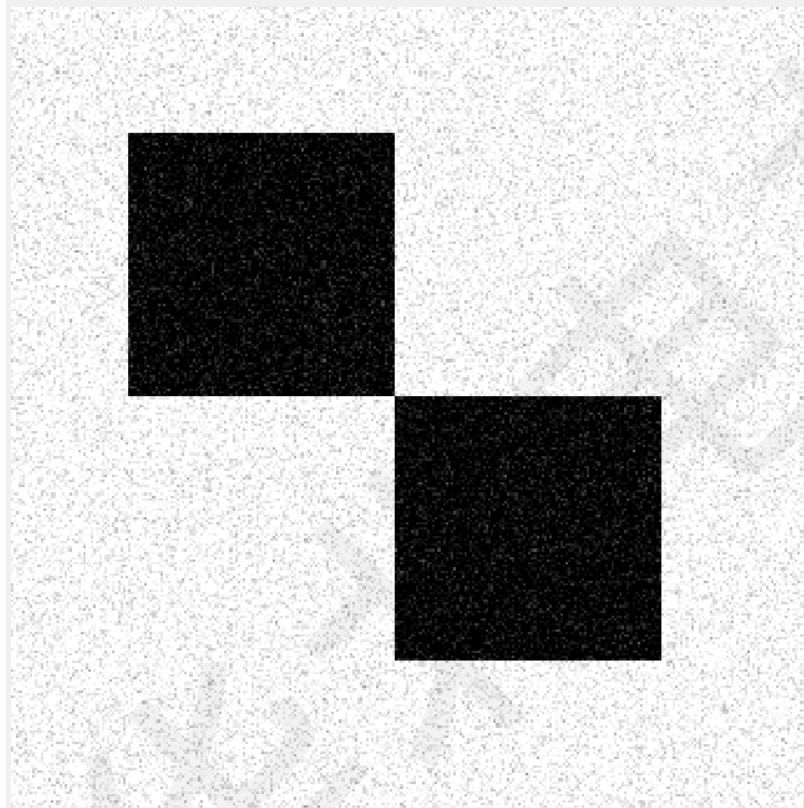
梯度滤波



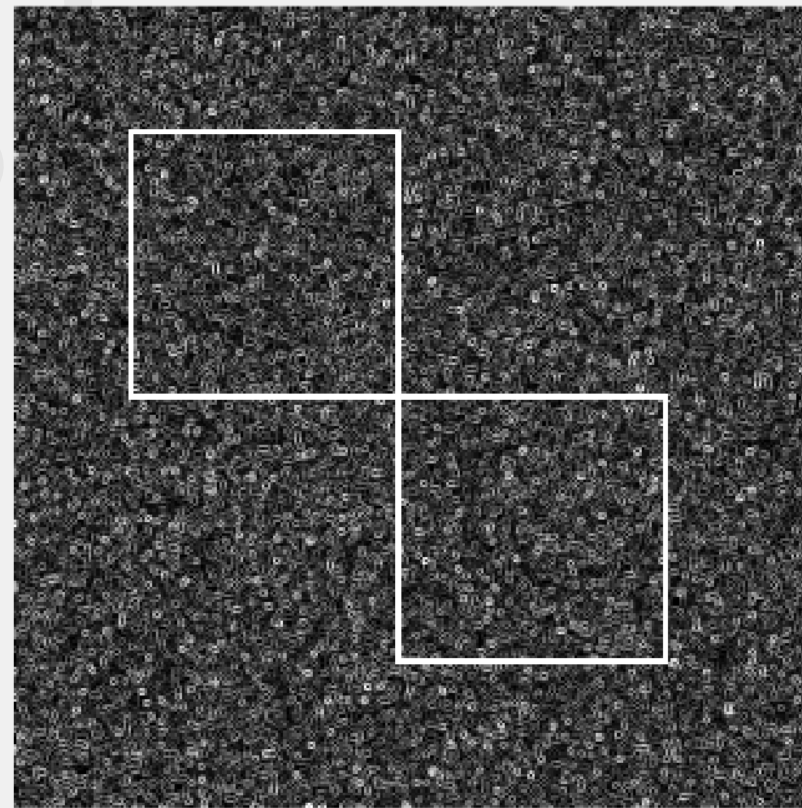
## 3.3.1 一阶微分滤波器

- 梯度滤波器

原图



梯度滤波





## 3.3.1 一阶微分滤波器

- 梯度滤波器

原图



梯度滤波





### 3.3.1 一阶微分滤波器

```
I1 = imread('blackblock.jpg');  
I1 = rgb2gray(I1); %必须转为灰度图像  
I1 = im2double(I1);  
  
I2 = imgradient(I1); %梯度滤波  
  
%创建图形窗口  
figure(1);  
subplot(1,2,1);imshow(I1);title('原图');  
subplot(1,2,2);imshow(I2);title('梯度滤波');
```





### 3.3.1 一阶微分滤波器

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

- Roberts 交叉梯度滤波器

- ✓ 点 $z_5$ 的 $\nabla f$ 值可用数字方式近似

$$\nabla f = |z_9 - z_5| + |z_8 - z_6|$$

- ✓ 梯度计算由两个模板组成，第一个求得梯度的第一项，第二个求得梯度的第二项，然后求和，得到梯度。

- ✓ 常用来处理具有陡峭的低噪声图像，当图像边缘接近于正45度或负45度时，该算法处理效果更理想。

- ✓ 其缺点是对边缘的定位不太准确。



### 3.3.1 一阶微分滤波器

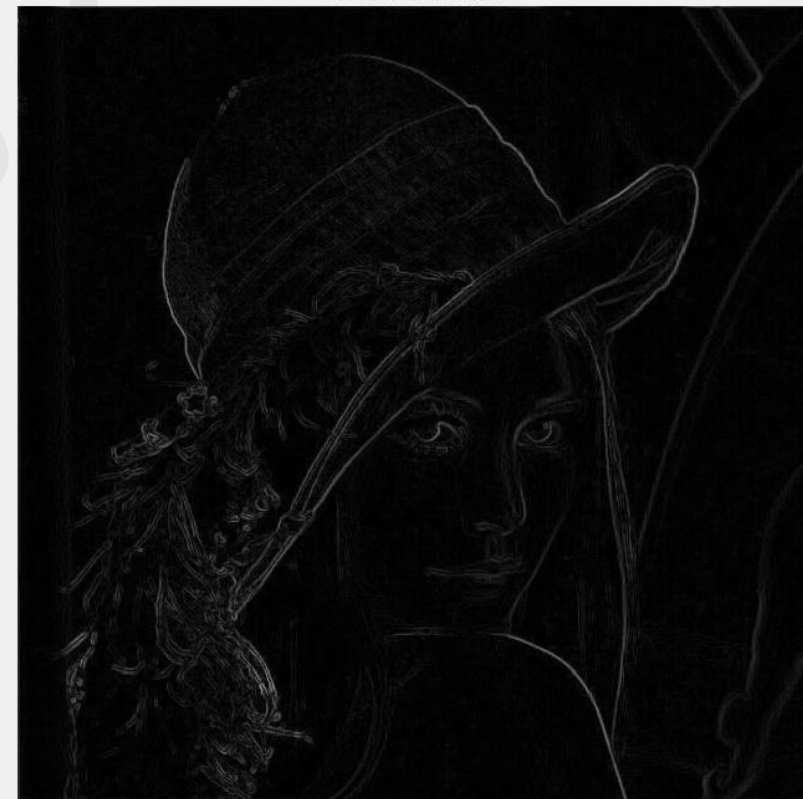
$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

- Roberts交叉梯度滤波器

原图



Roberts梯度滤波





### 3.3.1 一阶微分滤波器

```
I1 = imread('lena.jpg');  
I1 = rgb2gray(I1); %必须转为灰度图像  
I1 = im2double(I1);  
  
I2 = imgradient(I1, 'roberts'); %Roberts梯度滤波  
  
%创建图形窗口  
figure(1);  
subplot(1,2,1);imshow(I1);title('原图');  
subplot(1,2,2);imshow(I2);title('Roberts梯度滤波');
```





### 3.3.1 一阶微分滤波器

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

- Prewitt梯度滤波器

✓ 点 $z_5$ 的 $\nabla f$ 值可用数字方式近似

$$\nabla f = |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| \\ + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)|$$

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

✓ 边缘检测结果在水平方向和垂直方向均比Robert算子更加明显。

✓ 适合用来识别噪声较多、灰度渐变的图像。





## 3.3.1 一阶微分滤波器

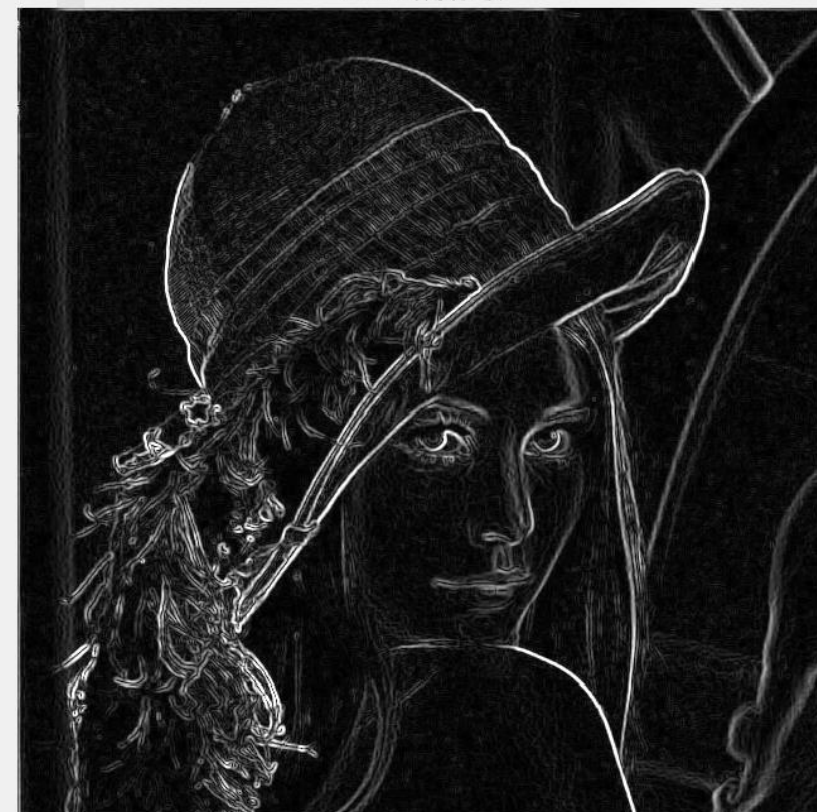
- Prewitt梯度滤波器

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

原图



Prewitt梯度滤波





### 3.3.1 一阶微分滤波器

```
I1 = imread('lena.jpg');  
I1 = rgb2gray(I1); %必须转为灰度图像  
I1 = im2double(I1);  
  
I2 = imgradient(I1, 'prewitt'); %Prewitt梯度滤波  
  
%创建图形窗口  
figure(1);  
subplot(1,2,1);imshow(I1);title('原图');  
subplot(1,2,2);imshow(I2);title('Prewitt梯度滤波');
```



### 3.3.1 一阶微分滤波器

- Sobel梯度滤波器

✓ 点 $z_5$ 的 $\nabla f$ 值可用数字方式近似

$$\nabla f = |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| \\ + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

✓ 边缘定位更准确，常用于噪声较多、灰度渐变的图像。





## 3.3.1 一阶微分滤波器

- Sobel梯度滤波器

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

原图



Sobel梯度滤波







### 3.3.1 一阶微分滤波器

```
I1 = imread('lena.jpg');  
I1 = rgb2gray(I1); %必须转为灰度图像  
I1 = im2double(I1);  
  
I2 = imgradient(I1, 'sobel'); %Sobel梯度滤波  
  
%创建图形窗口  
figure(1);  
subplot(1,2,1);imshow(I1);title('原图');  
subplot(1,2,2);imshow(I2);title('Sobel梯度滤波');
```

## 3.3.1 一阶微分滤波器

原图



梯度滤波



Roberts梯度滤波



Prewitt梯度滤波



Sobel梯度滤波







## 3.3.1 一阶微分滤波器

原图



梯度滤波



Roberts梯度滤波



Prewitt梯度滤波



Sobel梯度滤波



## 3.3.1 一阶微分滤波器

原图



梯度滤波



Roberts梯度滤波



Prewitt梯度滤波



Sobel梯度滤波

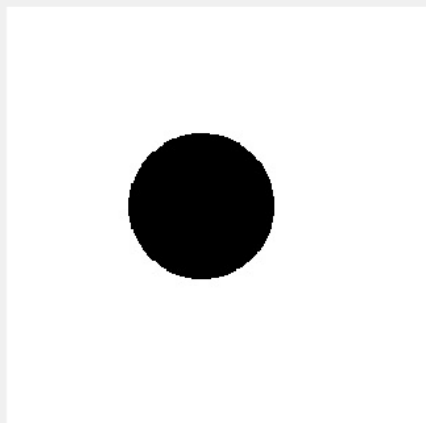




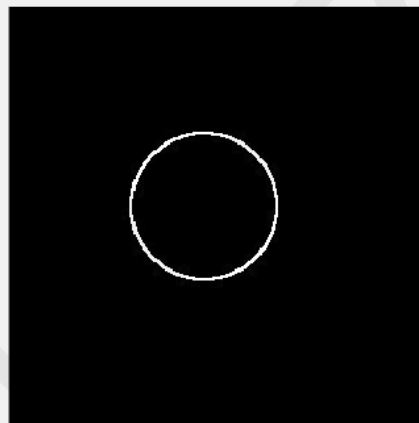


### 3.3.1 一阶微分滤波器

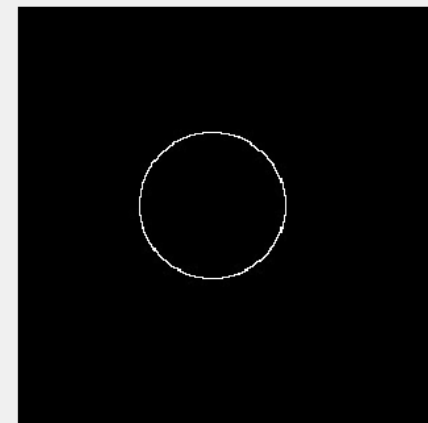
原图



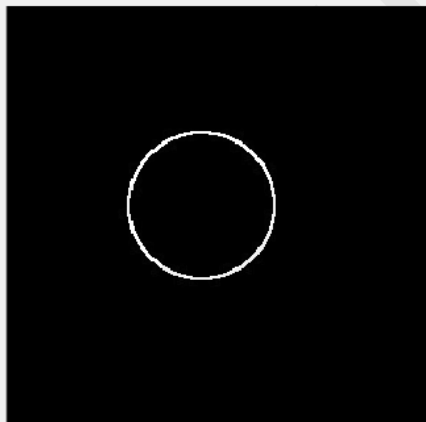
梯度滤波



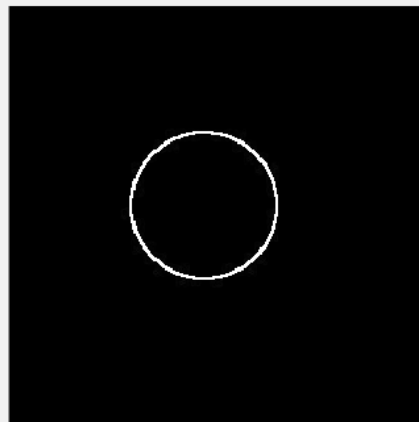
Roberts梯度滤波



Prewitt梯度滤波



Sobel梯度滤波





### 3.3.2 二阶微分滤波器

- 拉普拉斯滤波器  $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$

✓ 在x方向上

$$\frac{\partial^2 f}{\partial x^2} = [f(x+1, y) - f(x, y)] - [f(x, y) - f(x-1, y)]$$

✓ 在y方向上

$$\frac{\partial^2 f}{\partial y^2} = [f(x, y+1) - f(x, y)] - [f(x, y) - f(x, y-1)]$$

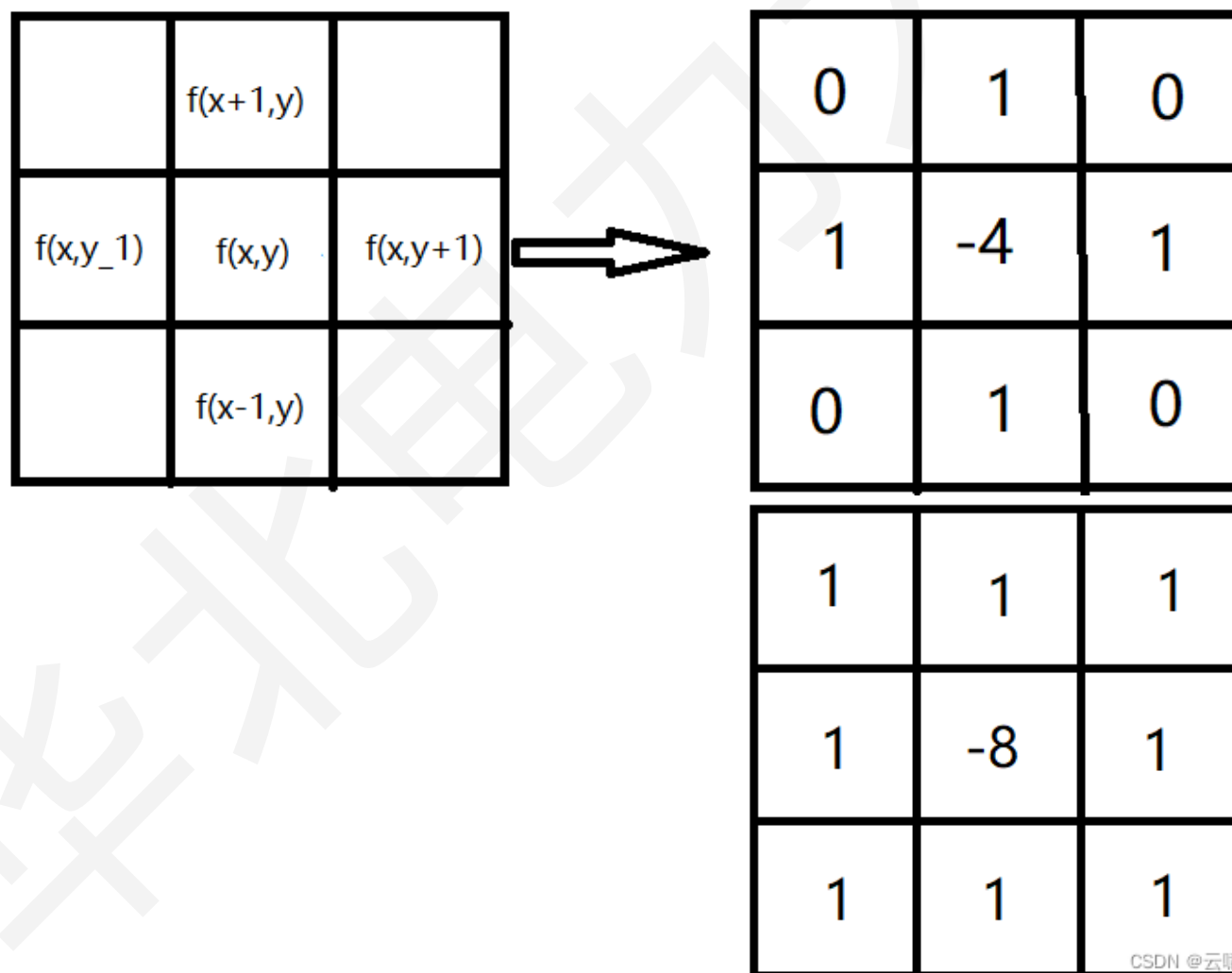
✓ 综合

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$



### 3.3.2 二阶微分滤波器

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$







### 3.3.2 二阶微分滤波器

- 拉普拉斯滤波器

- ✓ 拉普拉斯算子计算的是二阶微分，所以对图像中的噪声相当敏感。
- ✓ 常产生双像素宽的边缘，也不能提供边缘方向的信息。
- ✓ 很少直接用于检测边缘，而主要用于已知边缘像素后确定该像素是在图像的暗区或明区一边。

## 3.3.2 二阶微分滤波器

- 拉普拉斯滤波器

原图



中心为-8的拉普拉斯滤波



中心为-4的拉普拉斯滤波



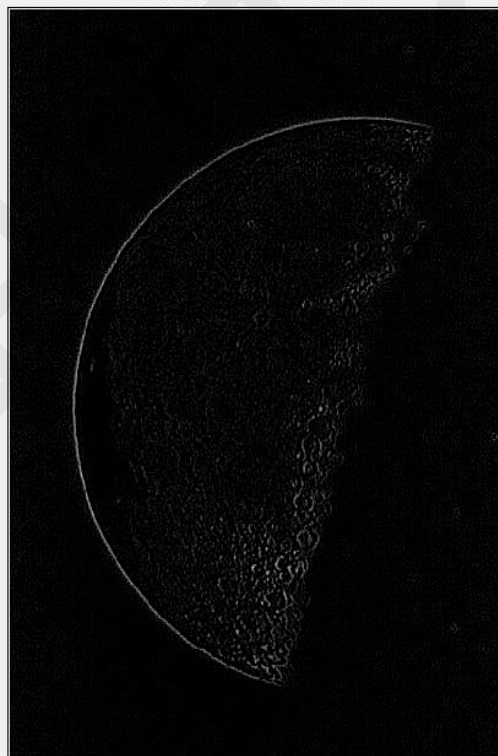
## 3.3.2 二阶微分滤波器

- 拉普拉斯滤波器

原图



中心为-8的拉普拉斯滤波



中心为-4的拉普拉斯滤波







### 3.3.2 二阶微分滤波器

```
I1 = imread('lena.jpg');  
I1 = rgb2gray(I1);  
I1 = im2double(I1);  
laplacian1 = [1,1,1;1,-8,1;1,1,1]; %扩展的拉普拉斯滤波器  
I2 = imfilter(I1, laplacian1);  
  
laplacian2 = [0,1,0;1,-4,1;0,1,0]; %拉普拉斯滤波器  
I3 = imfilter(I1, laplacian2);  
  
figure(1);  
subplot(1,3,1);imshow(I1);title('原图');  
subplot(1,3,2);imshow(I2);title('中心为-8的拉普拉斯滤波');  
subplot(1,3,3);imshow(I3);title('中心为-4的拉普拉斯滤波');
```



### 3.3.2 二阶微分滤波器

- 拉普拉斯滤波器对图像增强
  - ✓ 拉普拉斯算子是强调灰度突变而减加重灰度慢变化的区域。
  - ✓ 把原图像拉普拉斯图像叠加在一起，这样既能保护拉氏锐化效果，同时又能复原背景信息。

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{中心轴为负} \\ f(x, y) + \nabla^2 f(x, y) & \text{中心轴为正} \end{cases}$$

## 3.3.2 二阶微分滤波器

- 拉普拉斯滤波器

原图



叠加中心为-8的拉普拉斯增强



叠加中心为-4的拉普拉斯增强





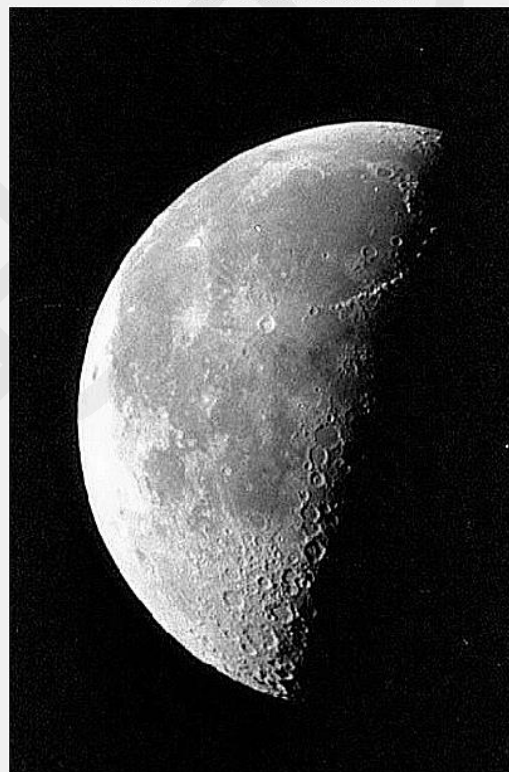
## 3.3.2 二阶微分滤波器

- 拉普拉斯滤波器

原图



叠加中心为-8的拉普拉斯增强



叠加中心为-4的拉普拉斯增强





## 3.3.2 二阶微分滤波器

```
I1 = imread('lena.jpg');  
I1 = rgb2gray(I1);  
I1 = im2double(I1);  
laplacian1 = [1,1,1;1,-8,1;1,1,1]; %扩展的拉普拉斯滤波器  
I2 = imfilter(I1, laplacian1);  
  
laplacian2 = [0,1,0;1,-4,1;0,1,0]; %拉普拉斯滤波器  
I3 = imfilter(I1, laplacian2);  
  
figure(1);  
subplot(1,3,1);imshow(I1);title('原图');  
subplot(1,3,2);imshow(I1-I2);title('叠加中心为-8的拉普拉斯增强');  
Subplot(1,3,3);imshow(I1-I3);title('叠加中心为-4的拉普拉斯增强');
```