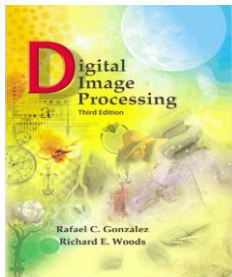


## 第三章

# 空间域图像增强

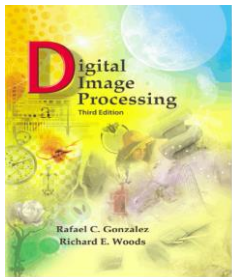
## 第三部分--空间滤波



# 主要内容

---

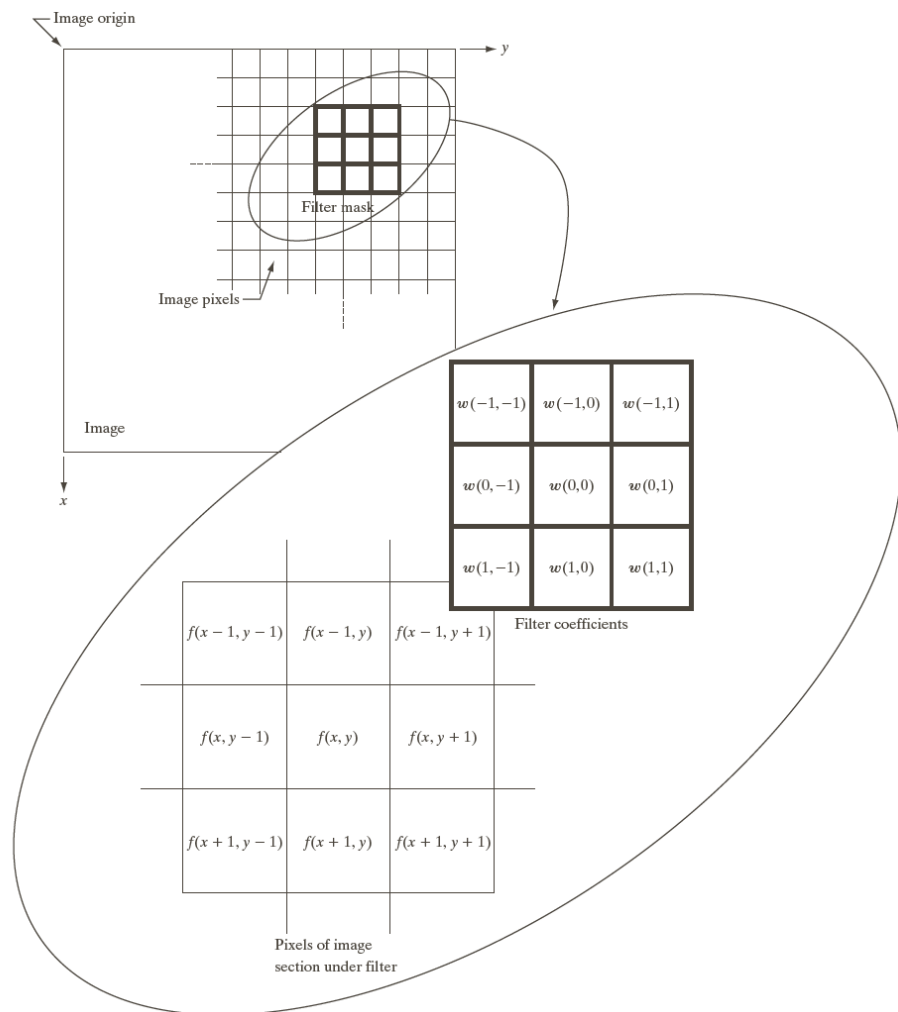
- 3.5 空间域滤波基础
- 3.6 平滑空间滤波
- 3.7 锐化空间滤波

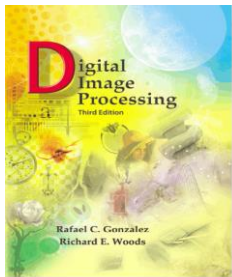


## 3.5 空间滤波基础

图像的平滑、锐化都是利用掩模操作来完成的.通过**掩模操作**实现一种邻域运算,待处理像素点的结果由邻域的图像像素以及相应的与邻域有相同维数的子图像得到.这些子图像被称为**滤波器、掩模、核、模板或窗口**.

掩模运算的数学含义是**卷积（或互相关）运算**。

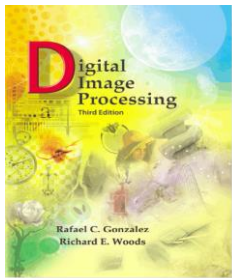




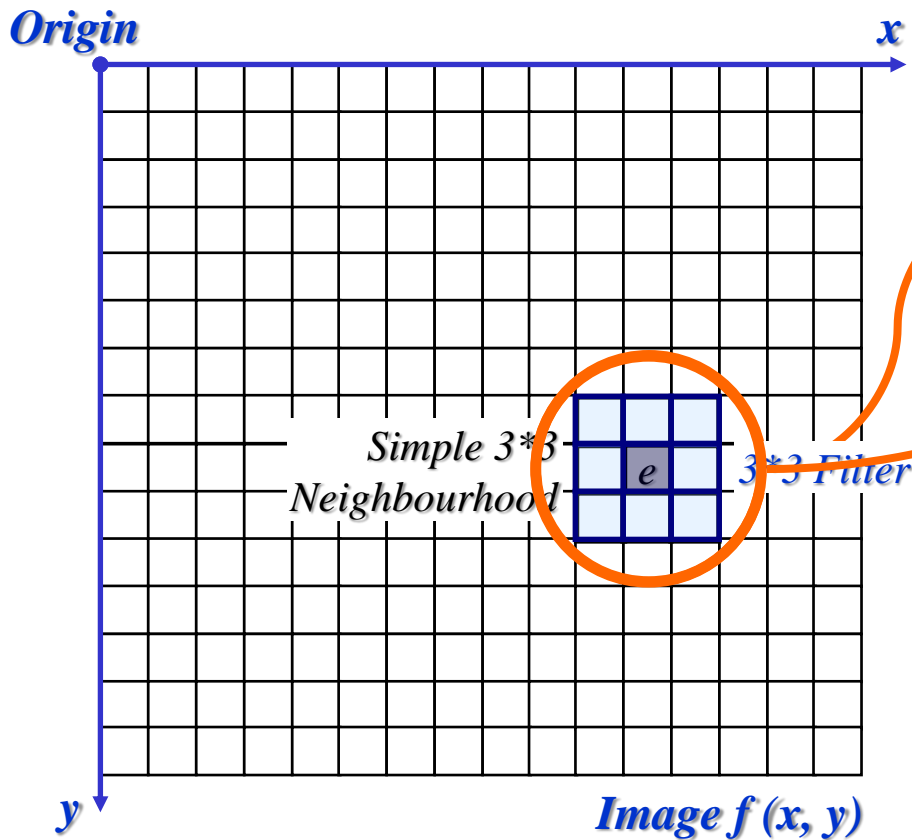
## 3.5 空间滤波基础

- 直接对图像像素处理的操作。
- 该处理就是在待处理图像中逐点地移动掩模。在每一点 $(x,y)$ 处，滤波器在该点的响应通过事先定义的关系来计算。对于线性空间滤波，其响应由滤波器系数与滤波掩模扫过区域的相应像素值的乘积之和给出。例如— $3 \times 3$ 的掩模，该掩模线性滤波的响应 $R$ 为：

$$R = w(-1,-1)f(x-1, y-1) + w(-1,0)f(x-1, y) + \dots + w(0,0)f(x, y) + \dots + w(1,1)f(x+1, y+1)$$



# 3.5 空间滤波基础



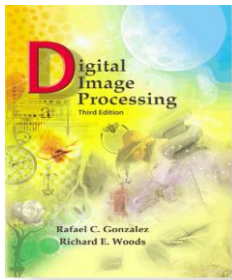
$a$	$b$	$c$
$d$	$e$	$f$
$g$	$h$	$i$

Original Image  
Pixels

$r$	$s$	$t$
$u$	$v$	$w$
$x$	$y$	$z$

Filter

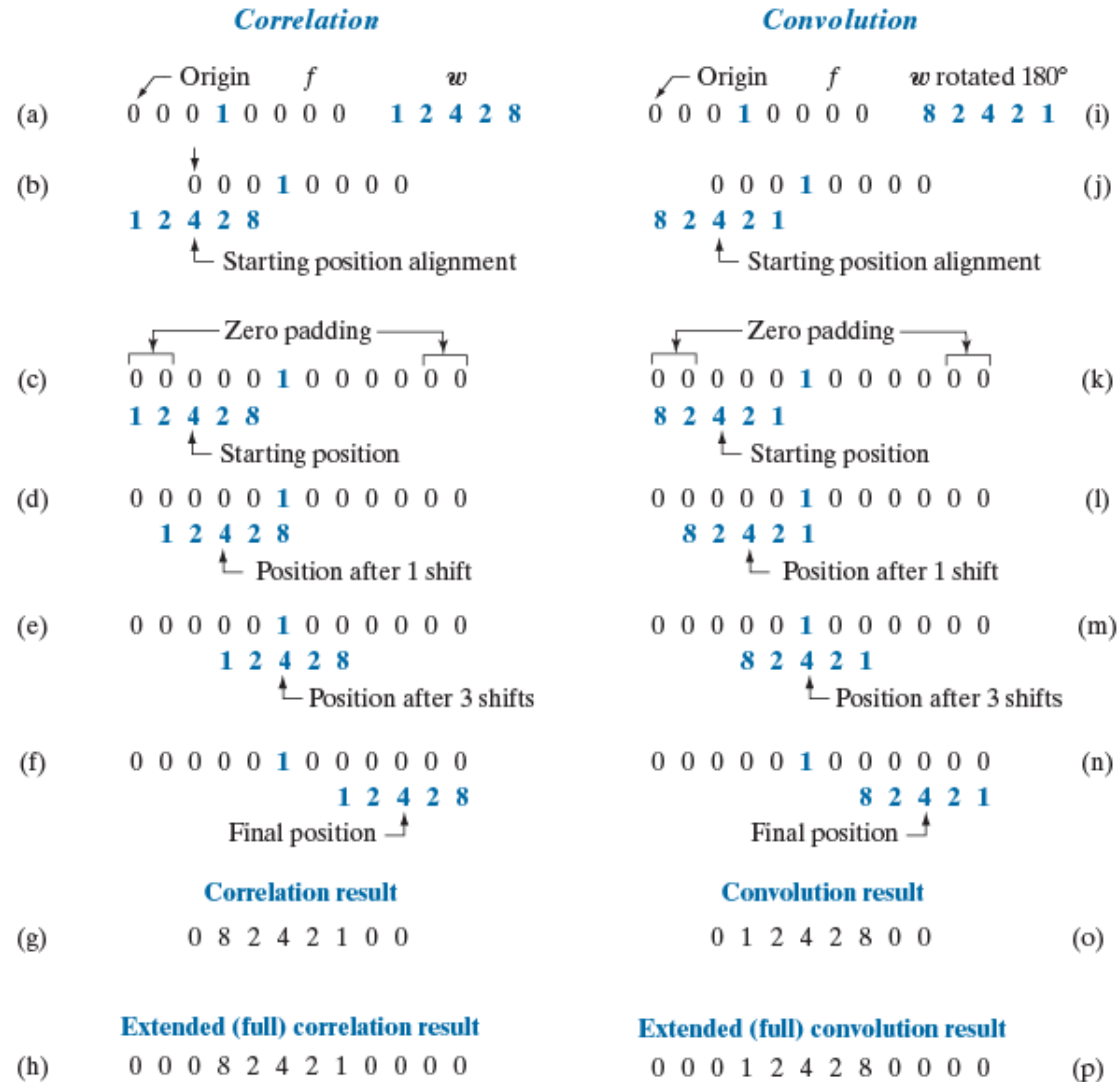
$$e_{processed} = v * e + r * a + s * b + t * c + u * d + w * f + x * g + y * h + z * i$$

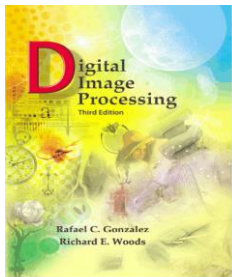


# 空间相关与卷积

FIGURE 3.29

Illustration of 1-D correlation and convolution of a kernel,  $w$ , with a function  $f$  consisting of a discrete unit impulse. Note that correlation and convolution are functions of the variable  $x$ , which acts to *displace* one function with respect to the other. For the extended correlation and convolution results, the starting configuration places the right-most element of the kernel to be coincident with the origin of  $f$ . Additional padding must be used.

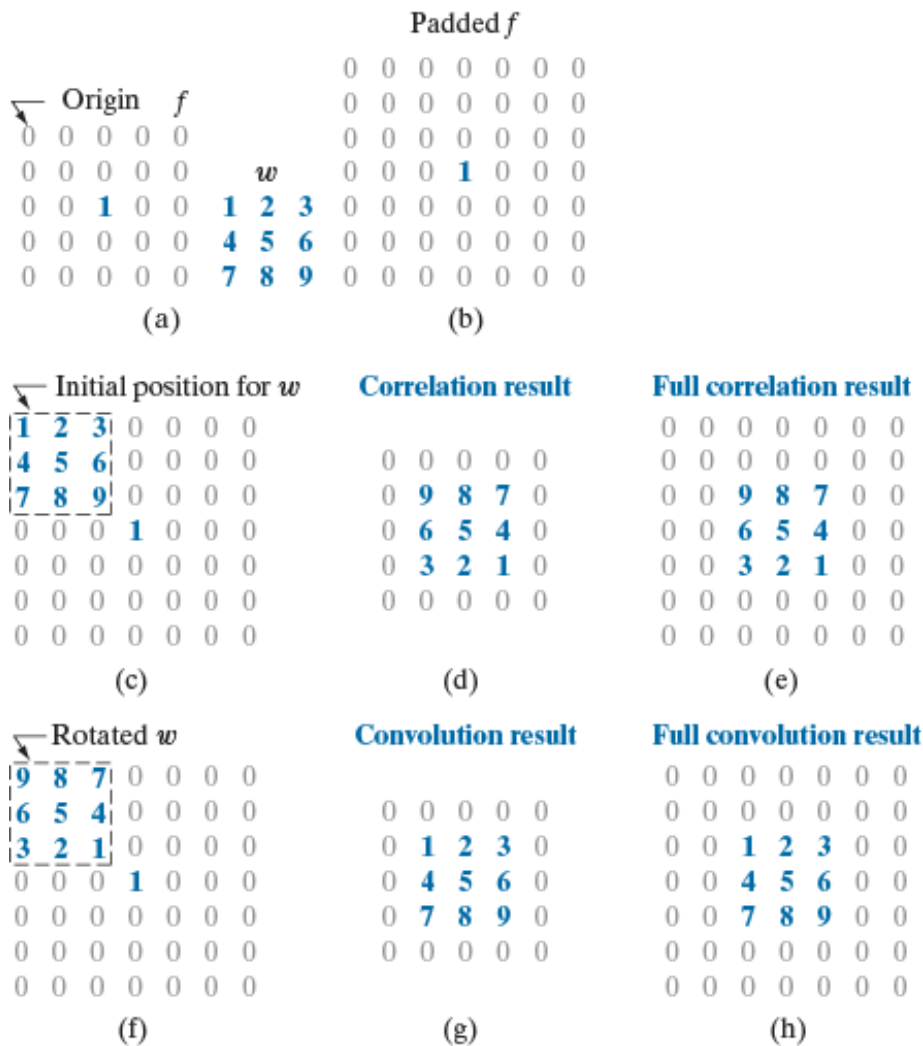


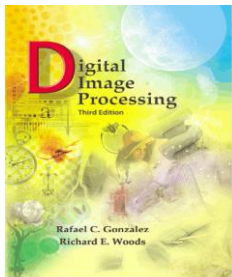


# 空间相关与卷积

FIGURE 3.30

Correlation (middle row) and convolution (last row) of a 2-D kernel with an image consisting of a discrete unit impulse. The 0's are shown in gray to simplify visual analysis. Note that correlation and convolution are functions of  $x$  and  $y$ . As these variable change, they *displace* one function with respect to the other. See the discussion of Eqs. (3-36) and (3-37) regarding full correlation and convolution.

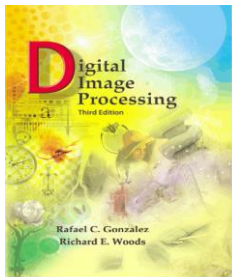




## 3.6平滑空间滤波器

- 平滑滤波器用于模糊处理和减少噪声。  
模糊处理经常用于预处理，例如，在提取大的目标之前去除图像中一些琐碎的细节、桥接直线或曲线的缝隙。
- 通过线性滤波器和非线性滤波器的模糊处理可以减小噪声。





## 3.6.1 平滑线性滤波器模板

$$\frac{1}{9} \times$$

1	1	1
1	1	1
1	1	1

box filter

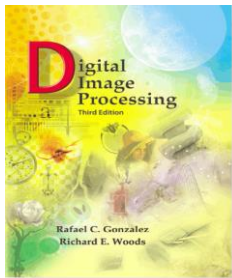
所有系数都相等的  
空间均值滤波器也  
叫盒滤波器

$$\frac{1}{16} \times$$

1	2	1
2	4	2
1	2	1

weighted  
average

中心像素权重较大，边缘  
权重较小

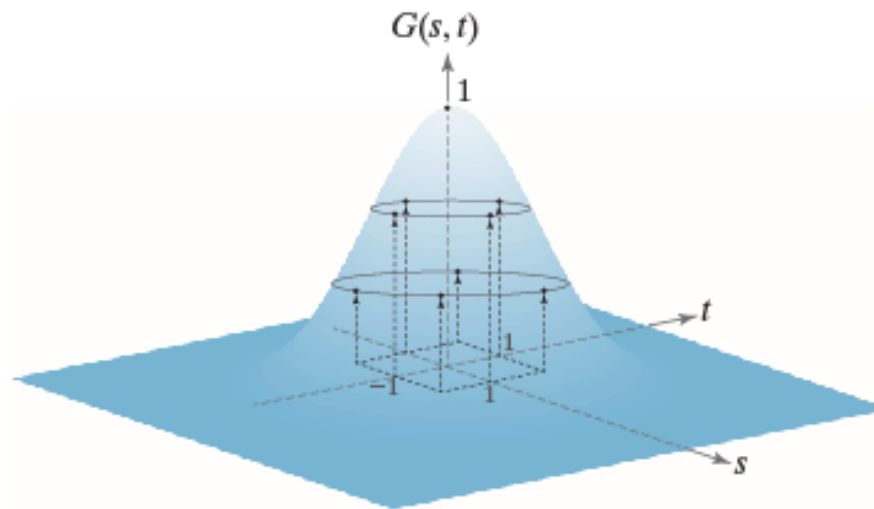


# 高斯核

a b

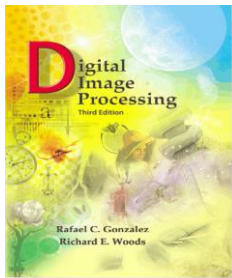
**FIGURE 3.35**

(a) Sampling a Gaussian function to obtain a discrete Gaussian kernel. The values shown are for  $K = 1$  and  $\sigma = 1$ . (b) Resulting  $3 \times 3$  kernel [this is the same as Fig. 3.31(b)].



$$\frac{1}{4.8976} \times$$

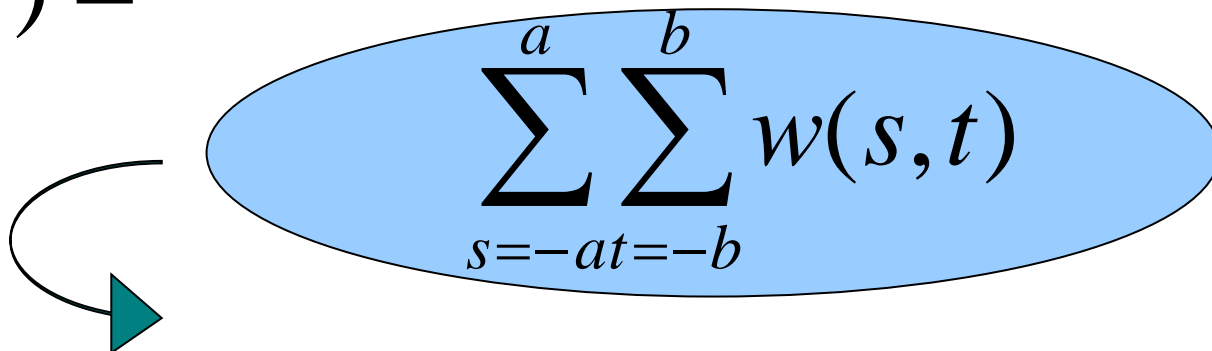
0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679



## 3.6.1平滑线性滤波器

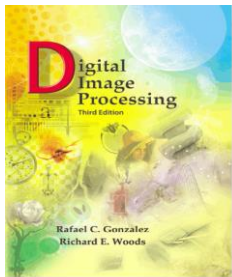
- filter of size  $m*n$  ( $m$  and  $n$  odd)

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$



summation of all coefficient of the mask

一幅 $M*N$ 的图像经过一个 $m*n$ 的加权平均值滤波器滤波的过程



# 模板大小对图像平滑的影响

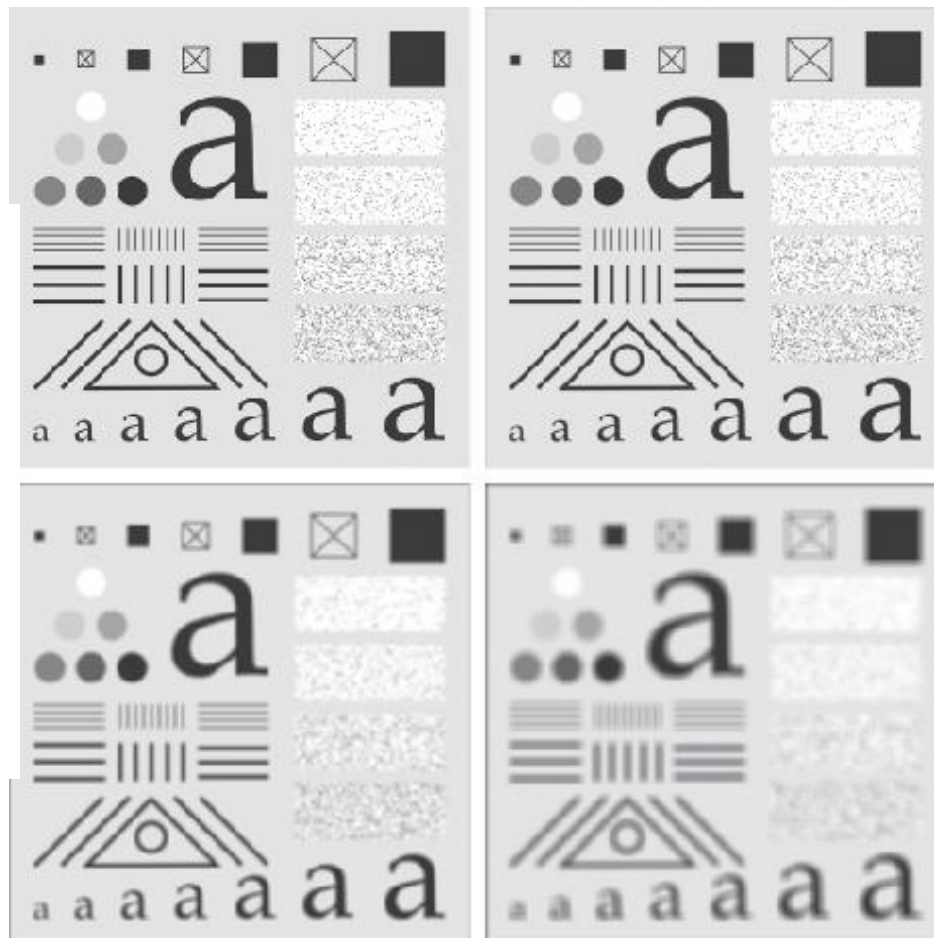


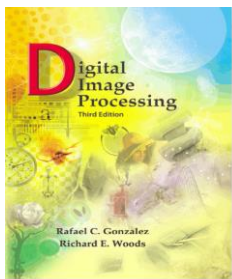
FIGURE 3.33

原图

$n=3, 11, 21$  盒式滤波

模板尺寸的选择  
影响图像平滑的  
效果，尺寸越大  
，图像越模糊





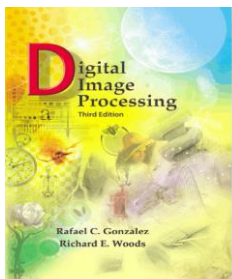
# 模板大小对图像平滑的影响



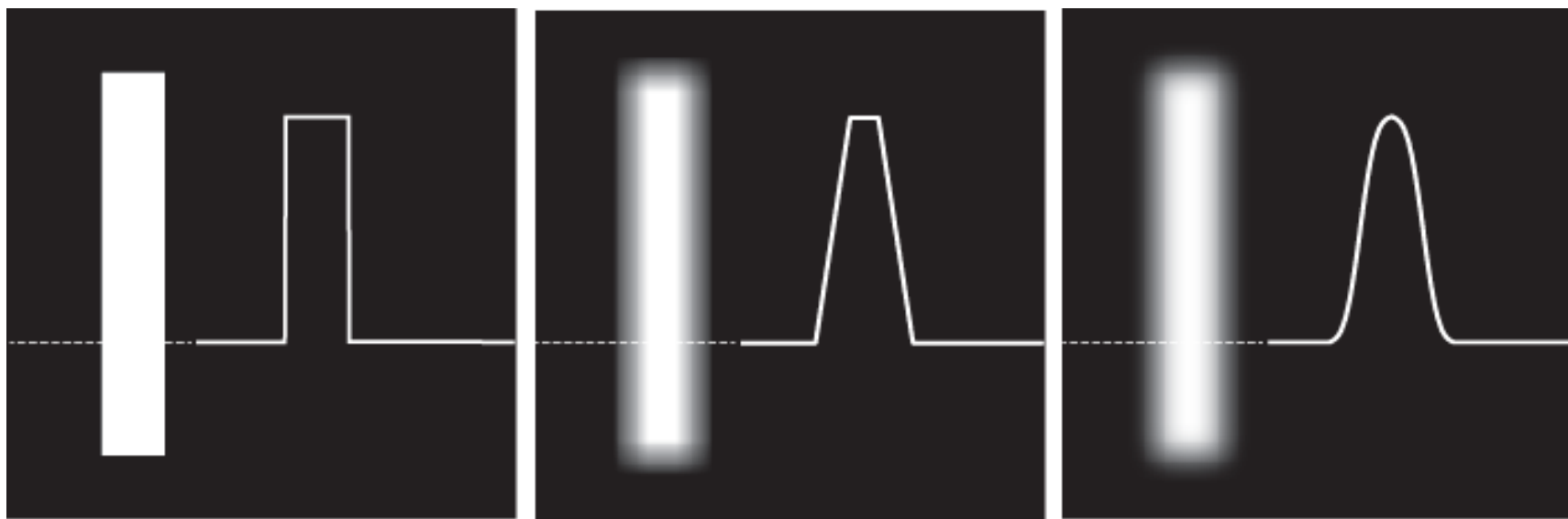
a b c

**FIGURE 3.36** (a) A test pattern of size  $1024 \times 1024$ . (b) Result of lowpass filtering the pattern with a Gaussian kernel of size  $21 \times 21$ , with standard deviations  $\sigma = 3.5$ . (c) Result of using a kernel of size  $43 \times 43$ , with  $\sigma = 7$ . This result is comparable to Fig. 3.33(d). We used  $K = 1$  in all cases.

n=21,43高斯滤波



# 盒式和高斯平滑滤波比较



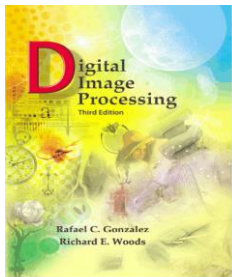
a b c

**FIGURE 3.38** (a) Image of a white rectangle on a black background, and a horizontal intensity profile along the scan line shown dotted. (b) Result of smoothing this image with a box kernel of size  $71 \times 71$ , and corresponding intensity profile. (c) Result of smoothing the image using a Gaussian kernel of size  $151 \times 151$ , with  $K = 1$  and  $\sigma = 25$ . Note the smoothness of the profile in (c) compared to (b). The image and rectangle are of sizes  $1024 \times 1024$  and  $768 \times 128$  pixels, respectively.

原图

盒式滤波

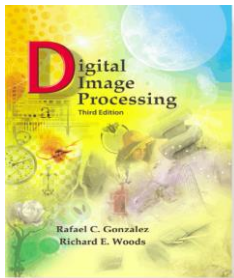
高斯滤波



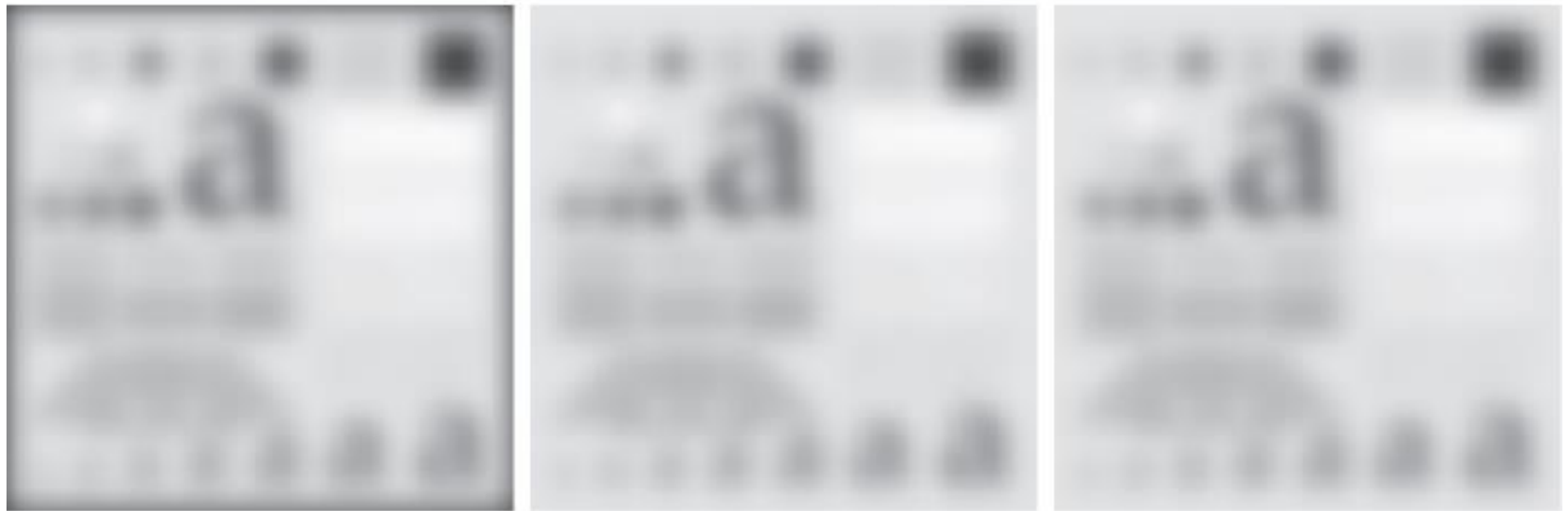
## 邻域处理时的图像边界问题

三种办法（对于 $n*n$ ）方形模板：

- 零填充
- 镜像填充：通过跨边界镜像反射图像的到边界之外的值（边界包含细节）
- 复制填充：边界之外的值设为最接近的图像边界值（边界区域为常数）



# 图像边界处理



a b c

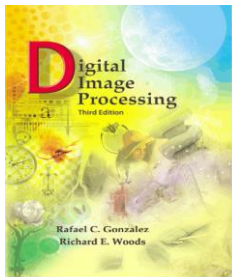
**FIGURE 3.39** Result of filtering the test pattern in Fig. 3.36(a) using (a) zero padding, (b) mirror padding, and (c) replicate padding. A Gaussian kernel of size  $187 \times 187$ , with  $K = 1$  and  $\sigma = 31$  was used in all three cases.

零填充

镜像填充

复制填充

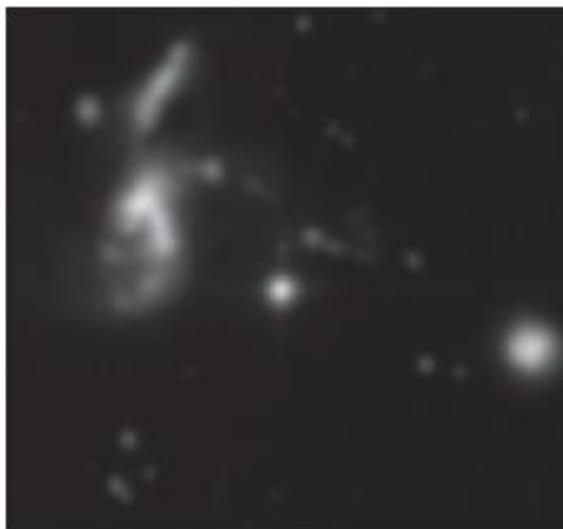




大尺寸模板可以用来消除一些较小物体。为了对感兴趣的物体得到一个粗略描述而模糊一幅图像



原始图像



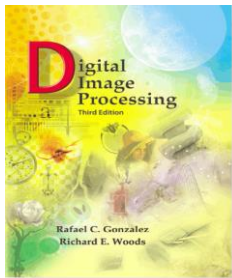
高斯滤波图像



阈值处理结果

**Hubble**望远镜拍摄的星系群图像

平滑掉细节图像，较大物体变的像斑点而易于检测



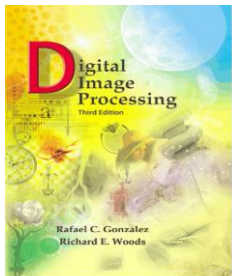
## 3.6.2 统计排序滤波器

统计排序滤波器是一种非线性的空间滤波器，他的响应基于图像区域中像素的**排序**。

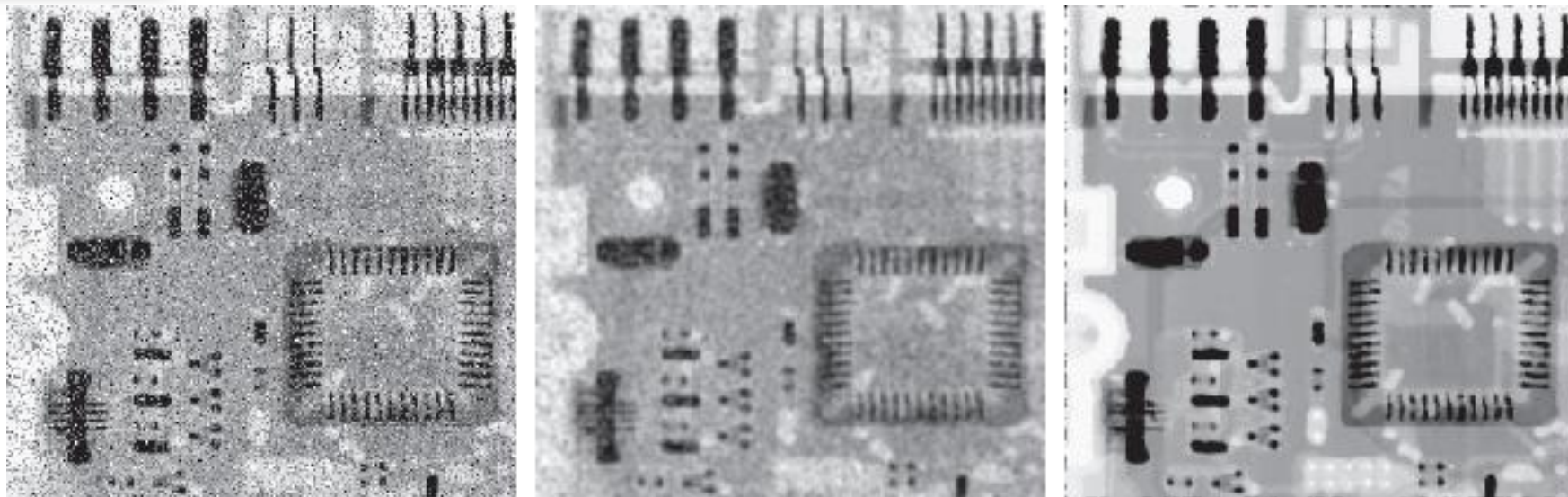
**50%-** 中值滤波：处理脉冲噪声非常有效，还可以保护边缘。

**100%-** 最大值滤波，寻找图像中最大点

**0%-** 最小值滤波



# 中值滤波去除椒盐噪声



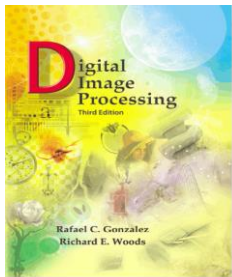
a b c

**FIGURE 3.43** (a) X-ray image of a circuit board, corrupted by salt-and-pepper noise. (b) Noise reduction using a  $19 \times 19$  Gaussian lowpass filter kernel with  $\sigma = 3$ . (c) Noise reduction using a  $7 \times 7$  median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

原图

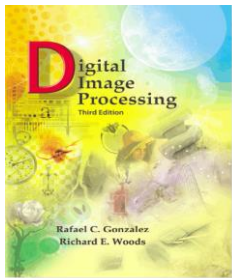
19\*19高斯模板

7\*7中值滤波



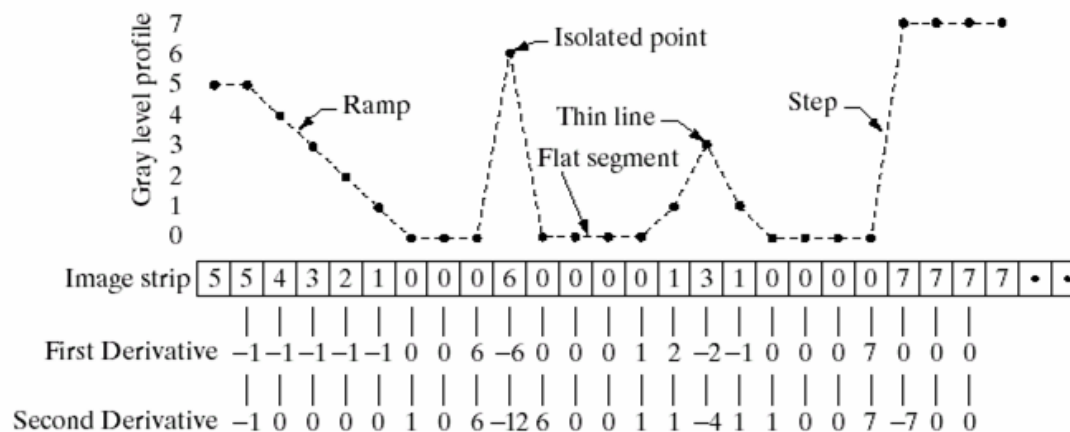
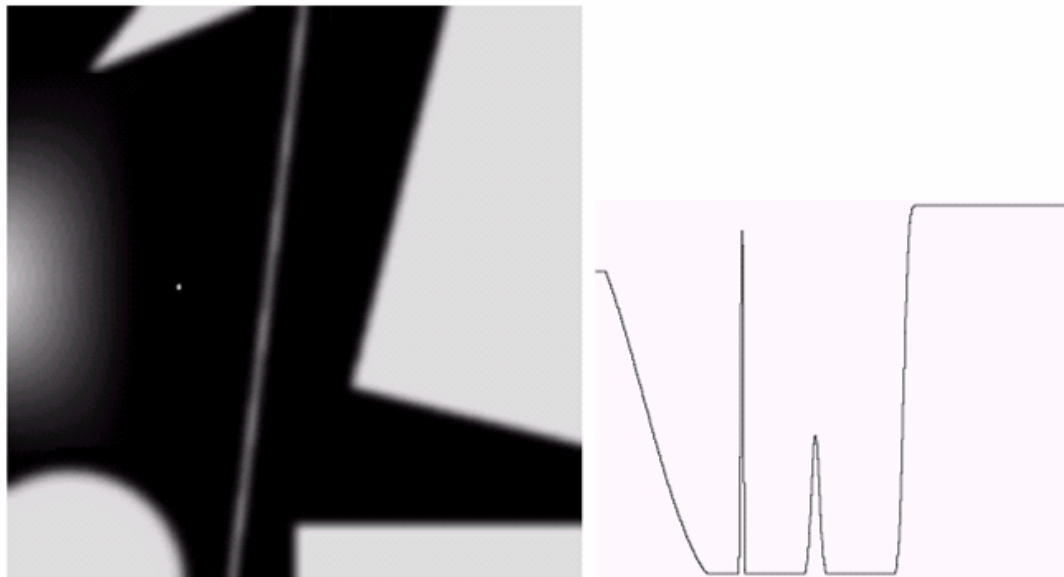
## 3.7锐化空间滤波器

- 锐化处理的主要目的是突出图像中的细节或者增强被模糊了的细节，这种模糊不是由于错误操作，就是特殊图像获取方法的固有影响。
- 前面讲平滑使用的是邻域平均法，均值处理与积分类似，而锐化处理则可用空间微分来完成。微分算子的响应强度与图像在该点的突变程度有关。这样一来，图像微分增强了边缘和其它突变，削弱了灰度变化缓慢的区域。

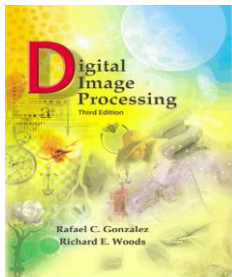


# 3.7 锐化空间滤波器

沿图像中心  
并且包含孤  
立噪声点的  
一维水平灰  
度剖面图



图像带  
一阶微分  
二阶微分

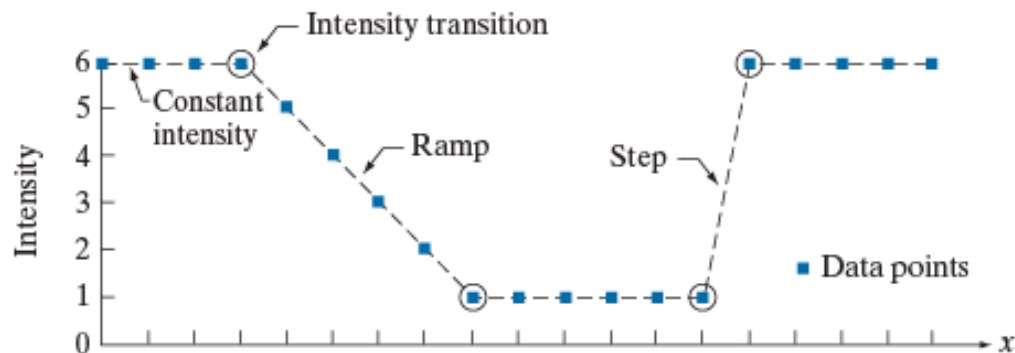


# 3.7 锐化空间滤波器

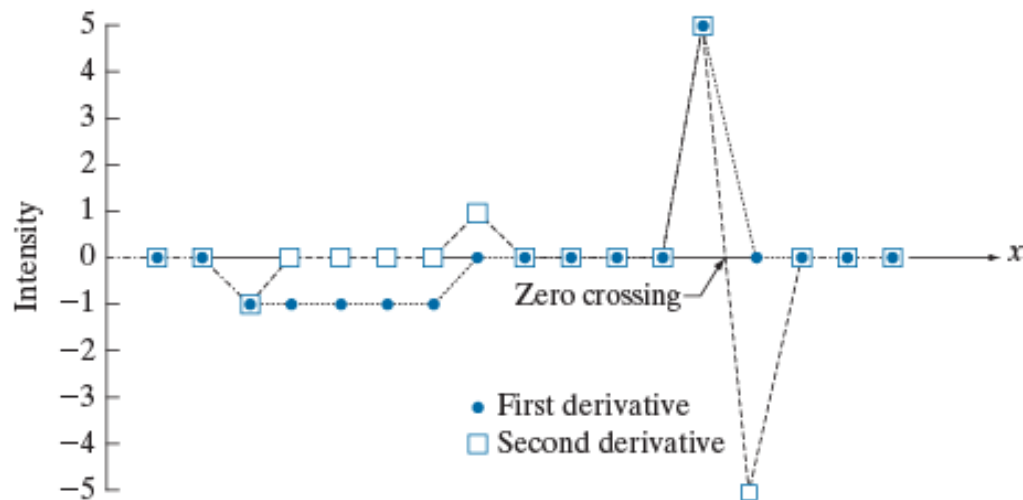
a  
b  
c

FIGURE 3.44

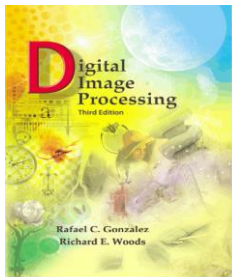
(a) A section of a horizontal scan line from an image, showing ramp and step edges, as well as constant segments.  
(b) Values of the scan line and its derivatives.  
(c) Plot of the derivatives, showing a zero crossing. In (a) and (c) points were joined by dashed lines as a visual aid.



Values of scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6	$\rightarrow x$
1st derivative	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0	5	0	0	0	0	0	
2nd derivative	0	0	-1	0	0	0	0	1	0	0	0	0	0	5	-5	0	0	0	0	







## 3.7 锐化空间滤波器

### 二阶微分的图像增强 —— 拉普拉斯算子

二元图像函数 $f(x,y)$ 的拉普拉斯变换定义为：

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

最简单的各向同性微分算子，  
并且是一个线性操作

离散方式：

**x方向**

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

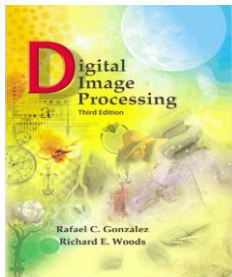
**y方向**

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

各向同性滤波器的响应  
与滤波器作用的图像的  
突变方向无关。也就是  
说各向同性滤波器是旋  
转不变的。

故二维拉普拉斯数字实现由以上两

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



# 拉普拉斯算子

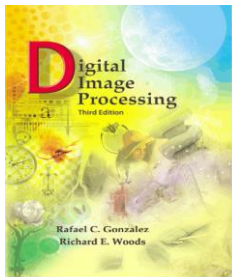
扩展方式:

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

- (a) 执行离散拉普拉斯变换所用的滤波器掩模
- (b) 扩展的拉普拉斯掩模，包括了对角线邻域
- (c) 和(d)其他两种拉普拉斯的实现

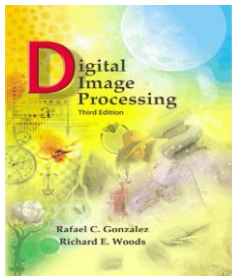




# 拉普拉斯算子

拉普拉斯微分算子**强调图像中灰度的突变,弱化灰度慢变化的区域**。这将产生一幅把浅灰色边线、突变点叠加到暗背景中的图像。





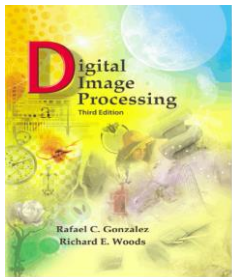
## 3.7 锐化空间滤波器

将原始图像和拉普拉斯图像叠加在一起的简单方法可以获得拉普拉斯**锐化处理**的效果，同时又能复原背景信息。因此拉普拉斯算子用于图像增强的基本方法如下：

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) \\ f(x, y) + \nabla^2 f(x, y) \end{cases}$$

拉普拉斯算子模板  
中心为负值

拉普拉斯算子模板  
中心为正值



## 3.7 锐化空间滤波器

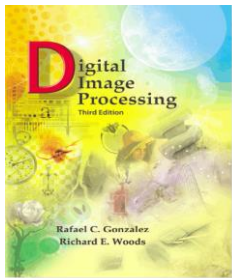
**实际运用时，叠加过程可以简化为：**

$$\begin{aligned} g(x, y) &= f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] + 4f(x, y) \\ &= 5f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] \end{aligned}$$

**可以用下面的掩模  
一次扫描来实现**

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1



# 简化

0	-1	0
-1	5	-1
0	-1	0

=

0	0	0
0	1	0
0	0	0

+

0	-1	0
-1	4	-1
0	-1	0

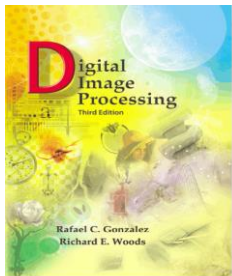
-1	-1	-1
-1	9	-1
-1	-1	-1

=

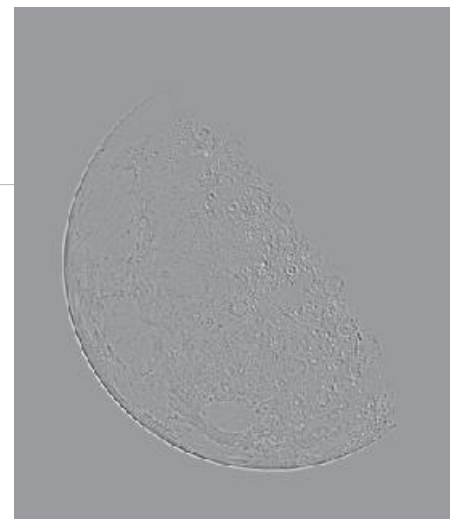
0	0	0
0	1	0
0	0	0

+

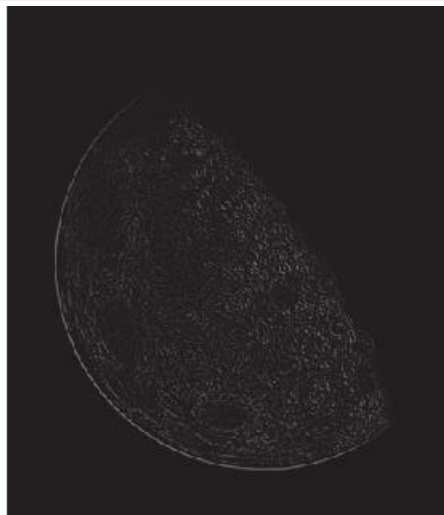
-1	-1	-1
-1	8	-1
-1	-1	-1



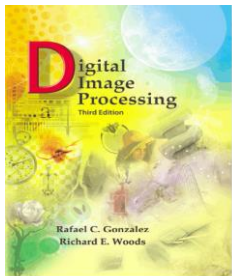
## 3.7 锐化空间滤波器



标定图像



- a. 月球北极的图像
- b. 拉普拉斯滤波后的图像
- c. 4邻域模板锐化后的图像
- d. 8邻域模板锐化后的图像



## 3.7 锐化空间滤波器

### 非锐化掩蔽与高提升滤波处理

长期以来在出版业中使用的图像锐化是将图像模糊形式从原始图像中去除.通过**非锐化掩蔽**来完成. 非锐化掩蔽的基本算法如下:

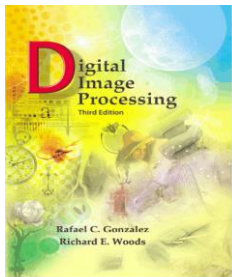
$$f_s(x, y) = f(x, y) - \bar{f}(x, y)$$

式中  $f(x, y)$  是原始图象,  $\bar{f}(x, y)$  是用人为方法模糊的图象.

$$f_{hb}(x, y) = f(x, y) + kf_s(x, y)$$

$$k = 1$$

$f(x, y) - \bar{f}(x, y)$  会使  $f(x, y)$  的低频成分损失很多,而高频成分较完整地被保留下来。将原始图像与非锐化模板叠加, 就得到锐化后的图像。

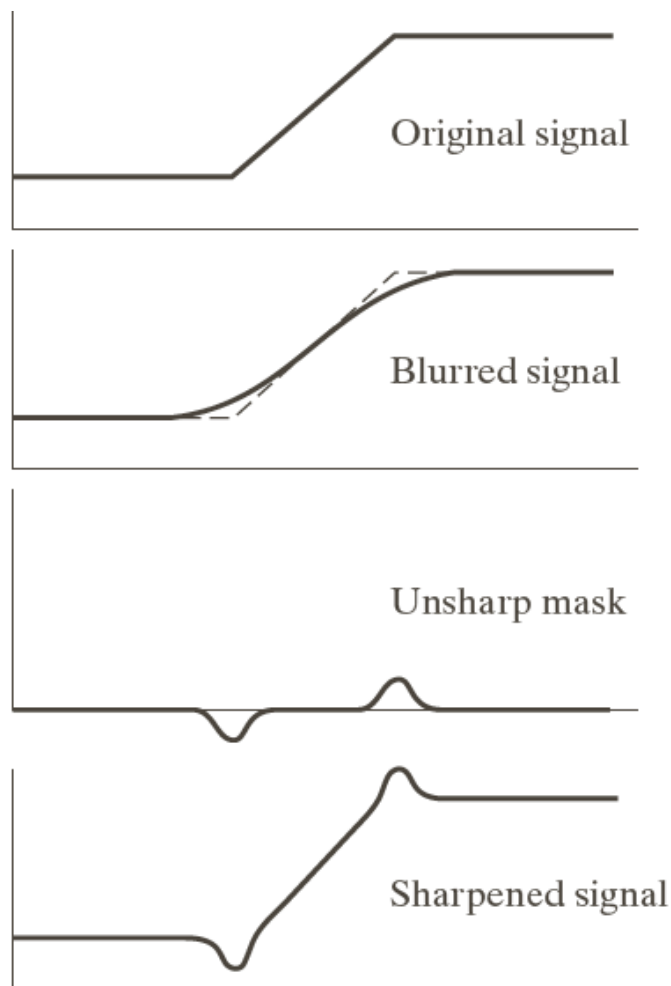


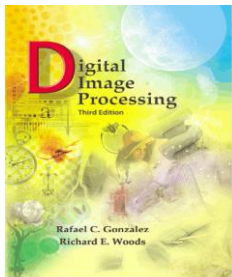
## 3.7 锐化空间滤波器

### 非锐化掩蔽

a  
b  
c  
d

**FIGURE 3.39** 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).





## 3.7 锐化空间滤波器

$$f_{hb}(x, y) = f(x, y) + kf_s(x, y)$$

**k=1时，称为非锐化掩蔽；k>1时，非锐化掩蔽的进一步普遍形式称为高提升滤波**

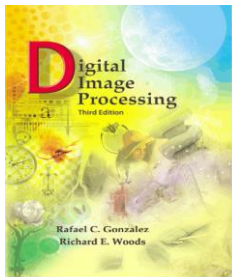


a  
b  
c  
d  
e

**FIGURE 3.40**

(a) Original image.  
(b) Result of blurring with a Gaussian filter.  
(c) Unsharp mask. (d) Result of using unsharp masking.  
(e) Result of using highboost filtering.





## 3.7 锐化空间滤波器

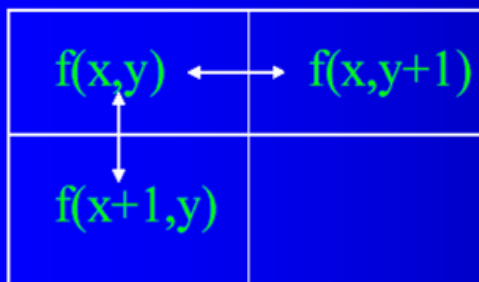
基于一阶微分的

$$G[f(x, y)] \cong |f(x, y) - f(x+1, y)| + |f(x, y) - f(x, y+1)|$$

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

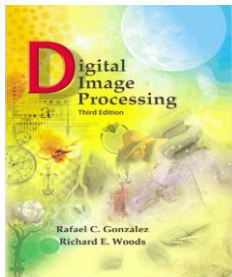
$$\nabla f = \text{mag}(\nabla \mathbf{f}) = [G_x^2 + G_y^2]^{1/2}$$

$$= \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$



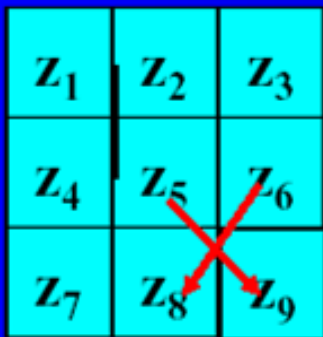
实际运算时用绝对值代替平方根运算:

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



# 基于一阶微分的图像增强——梯度法

## Robert提出的交叉梯度算子



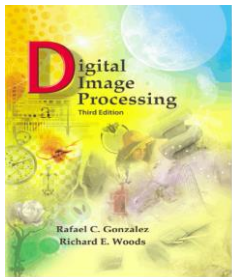
1	0
0	-1

0	1
-1	0

$$G[f(x, y)] \cong |f(x, y) - f(x+1, y+1)| \\ + |f(x+1, y) - f(x, y+1)|$$

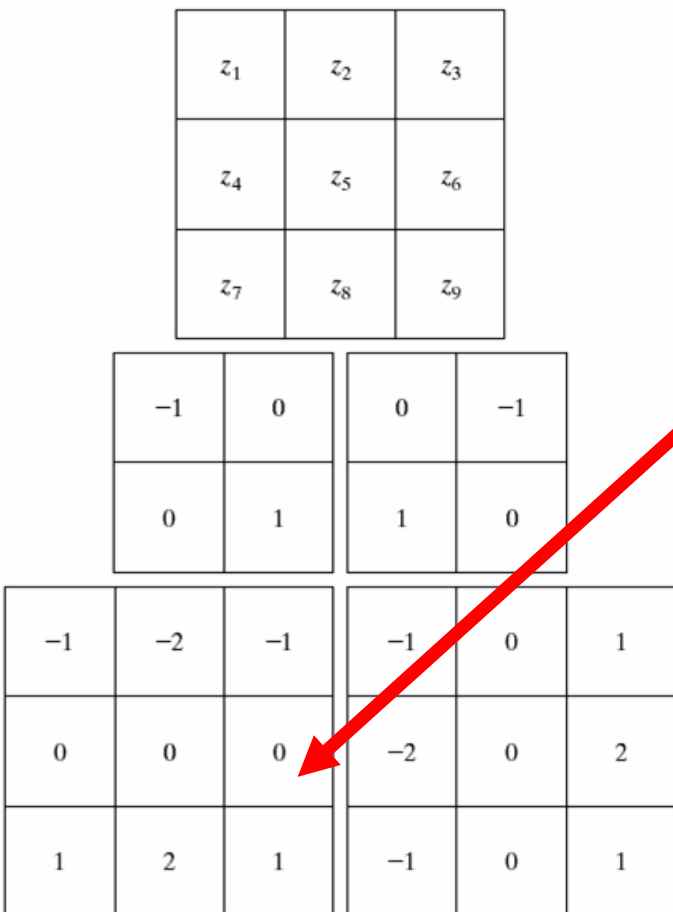
$$G_x = (z_9 - z_5) \quad G_y = (z_8 - z_6)$$

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

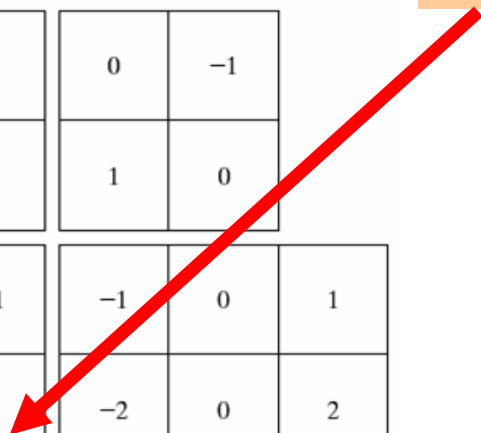


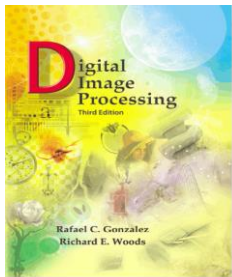
# 基于一阶微分的图像增强——梯度法

## 3X3的掩模: Sobel算子



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

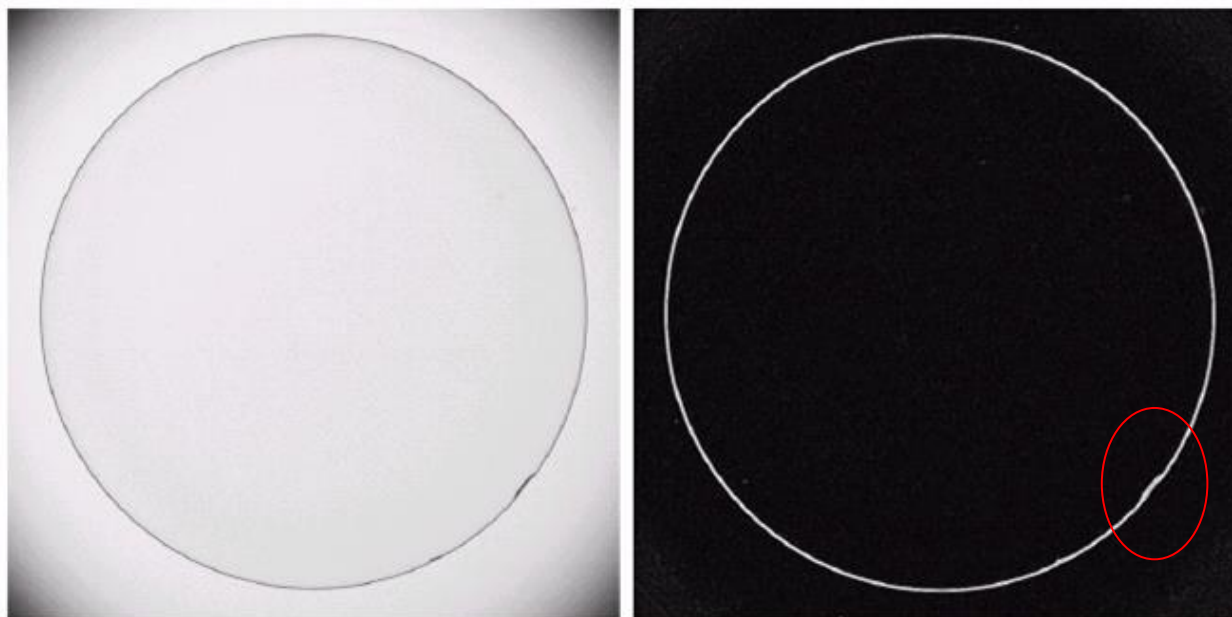




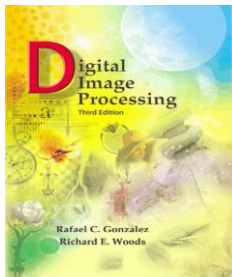
## 用于边缘增强的梯度处理

**梯度处理经常用于工业检测、辅助人工检测缺陷，或者是更为通用的自动检测的预处理。**

**用于  
边缘  
增强  
的梯  
度处  
理**

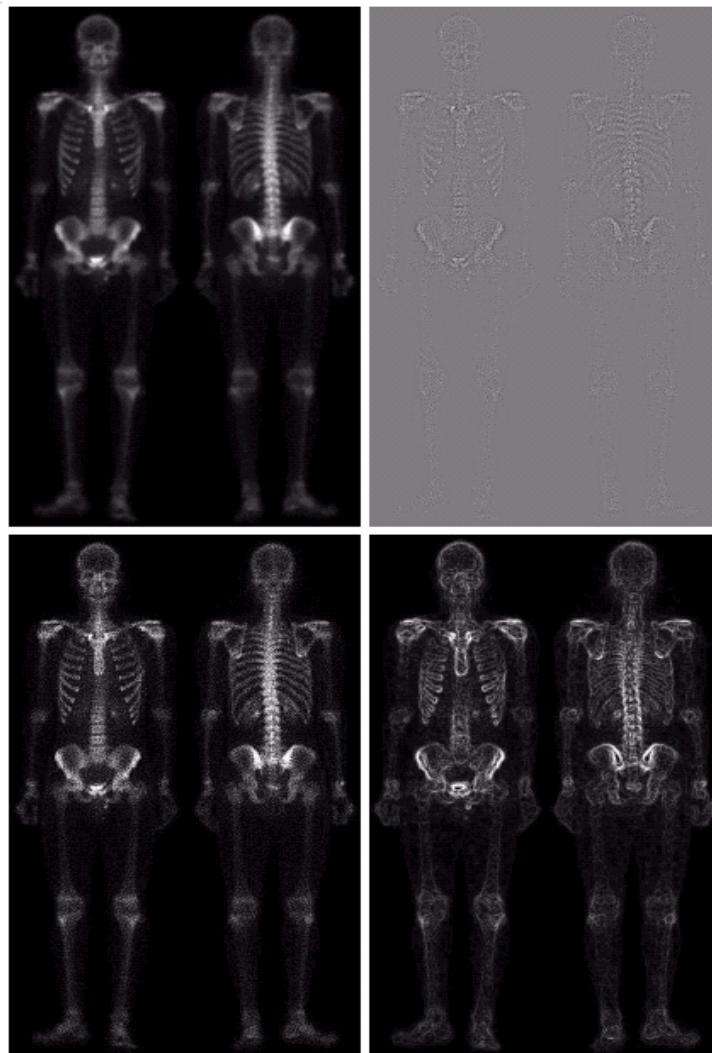


**(a) 隐形眼镜的光学图像**  
**(b) Sobel梯度**



# 混合空间增强

处理步骤  
b、拉普拉斯变换  
c、叠加得到锐化图像  
d、用Sobel算子突出边缘

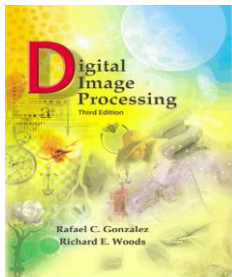


a	b
c	d

**FIGURE 3.46**

(a) Image of whole body bone scan.

(b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).



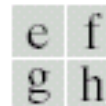
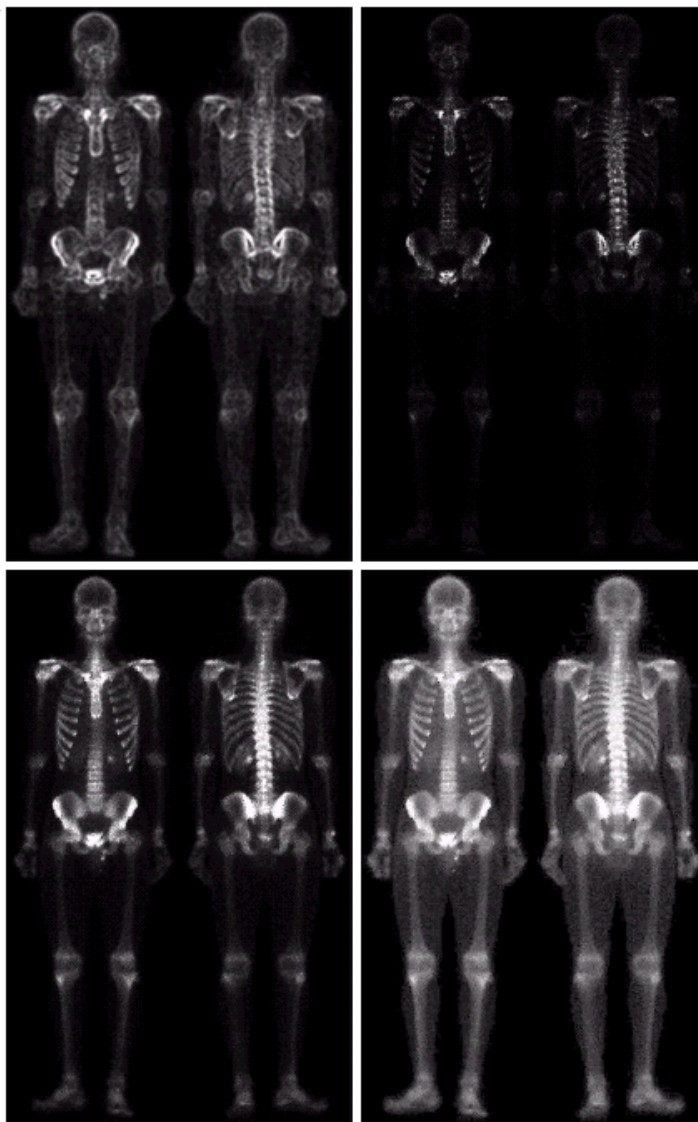
## 混合空间增强

e、对sobel图  
像进行均值  
滤波

f、由c和e相  
乘形成的掩  
蔽图像

g、由a和f求  
和得到的锐  
化图像。

h、对g应用  
幂律变化得  
到结果。



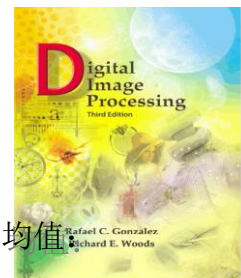
**FIGURE 3.46**

*(Continued)*

(e) Sobel image smoothed with a  $5 \times 5$  averaging filter. (f) Mask image formed by the product of (c) and (e).

(g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)





## 作业题

- 假设一幅图像中，每个坐标点上的噪声不相关且均值为零。在图像平均处理过程中我们对k幅不同的噪声图像取平均形成新的图像来减少噪声。在这一过程中利用了如下的基本原理：

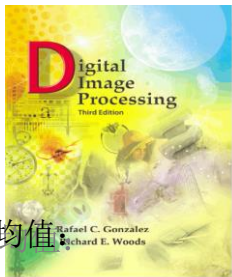
$$g_i(x, y) = f(x, y) + \eta_i(x, y)$$

$$\bar{g}(x, y) = \frac{1}{k} \sum_{i=1}^k g_i(x, y)$$

- 均值：  $E\{\bar{g}(x, y)\} = f(x, y)$

- 方差：  $\sigma_{\bar{g}(x, y)}^2 = \frac{1}{k} \sigma_{\eta(x, y)}^2$

试证明均值和方差公式，并说明图像总数K对处理后图像的影响。



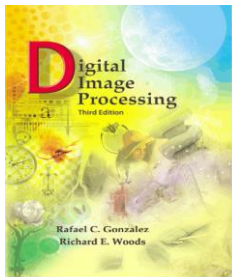
## 答案

• 证明:  $g_i(x, y) = f(x, y) + \eta_i(x, y)$        $\bar{g}(x, y) = \frac{1}{k} \sum_{i=1}^k g_i(x, y)$

• 均值:  $E\{\bar{g}(x, y)\} = \frac{1}{k} \left\{ \sum_{i=1}^k [f(x, y) + \eta_i(x, y)] \right\}$   
 $= \left\{ \frac{1}{k} \sum_{i=1}^k g_i(x, y) - f(x, y) \right\}^2$   
 $= \frac{1}{k} \sum_{i=1}^k f(x, y) + \frac{1}{k} \sum_{i=1}^k \eta_i(x, y) = f(x, y)$

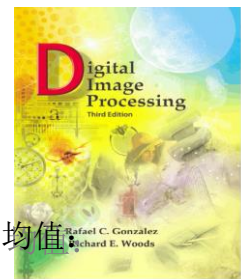
• 方差:  $\sigma_{\bar{g}(x, y)}^2 = E\{(\bar{g}(x, y) - E[\bar{g}(x, y)])^2\}$   
 $= \left\{ \frac{1}{k} \sum_{i=1}^k [f(x, y) + \eta_i(x, y)] - f(x, y) \right\}^2$   
 $= \left\{ \frac{1}{k} \sum_{i=1}^k \eta_i(x, y) \right\}^2 = \frac{1}{k^2} \left[ \sum_{i=1}^k \{\eta_i(x, y)\} \right]^2$   
 $= \frac{1}{k} \cdot \frac{1}{k} \sum_{i=1}^k \{\eta_i(x, y)\}^2 = \frac{1}{k} \sigma_{\eta(x, y)}^2$





## 答案

- $E\{\bar{g}(x, y)\} = f(x, y)$ ，意味着随着图像均值处理中使用的图像的增加， $\bar{g}(x, y)$  将越来越趋近于  $f(x, y)$ 。
- 由方差公式可知，当  $K$  增加时，在各个  $(x, y)$  位置上像素值的噪声变化率将减小。



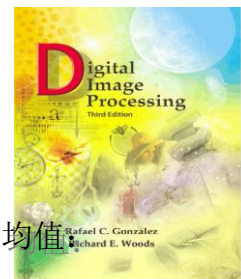
## 作业题

- 有一数字序列为：

(106, 114, 109, 145, 177, 186, 188, 182, 187)

- 1) 利用一维三点平滑模板( $1/3, 1/3, 1/3$ )对数据进行平滑。
- 2) 利用一维拉普拉斯算子 ( $1, -2, 1$ ) 对数据进行锐化。

(边缘处理方式自定义，写出如何定义)



## 答案

答：边缘处理方式为边缘灰度由相邻灰度（处理过的）替代。

- 1) 平滑后的序列为

(110, 110, 123, 144, 169, 184, 185, 185)

- 2) 锐化算子

(-13, -13, 41, -4, -23, -7, -8, 11, 11)

- 锐化后的序列为

(119, 127, 68, 149, 180, 193, 196, 171, 176)