

空间滤波和空间滤波器的定义

- 在 $M \times N$ 的图像 f 上，使用 $m \times n$ 的滤波器：

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

其中， $m=2a+1, n=2b+1$,

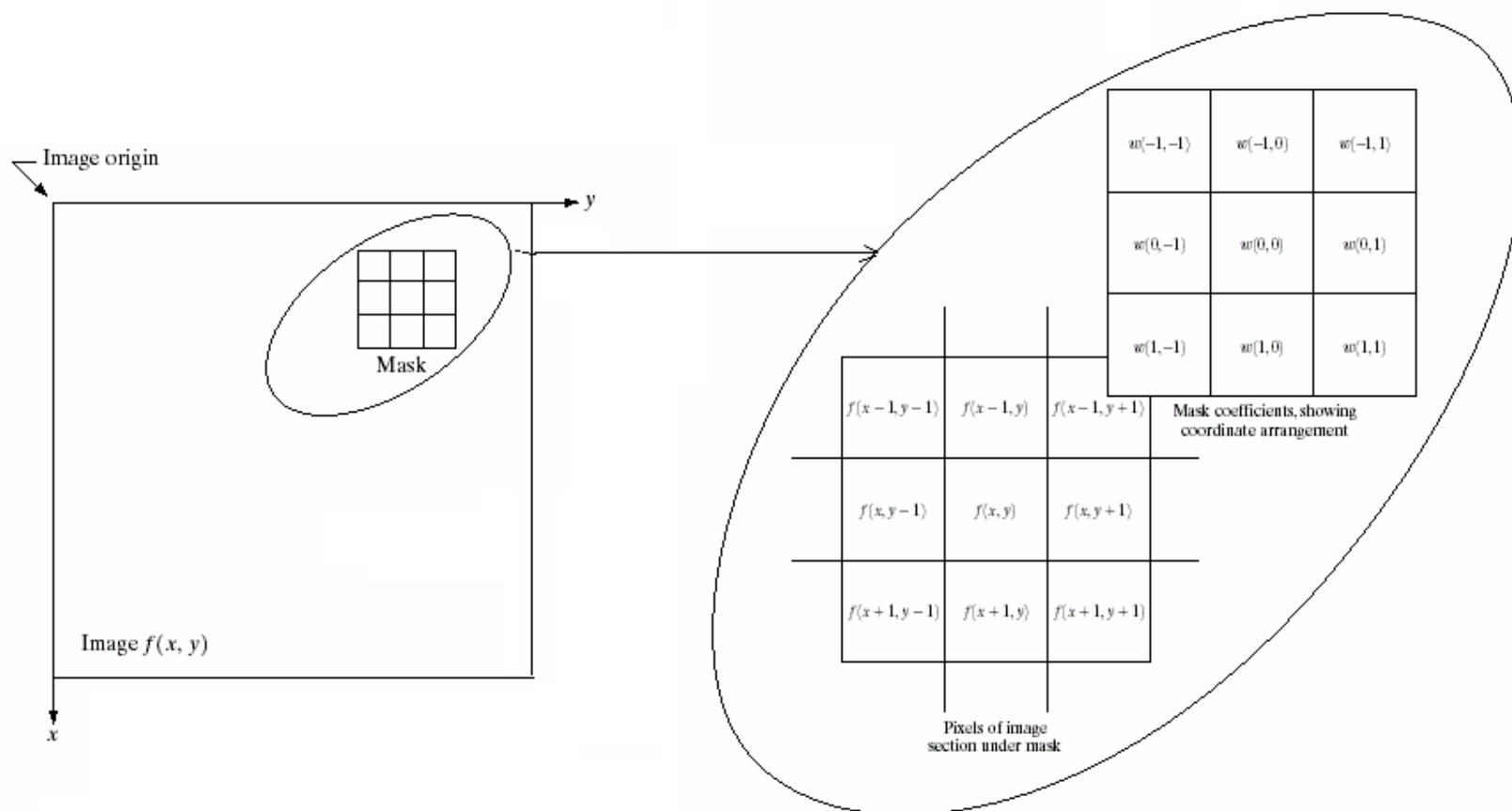
$w(s, t)$ 是滤波器系数， $f(x, y)$ 是图像值

- 空间滤波的简化形式：

$$R = w_1 z_1 + w_2 z_2 + \dots + w_{mn} z_{mn} = \sum_{i=1}^{mn} w_i z_i$$

其中， w 是滤波器系数， z 是与该系数对应的图像灰度值， mn 为滤波器中包含的像素点总数

三、线性空间滤波



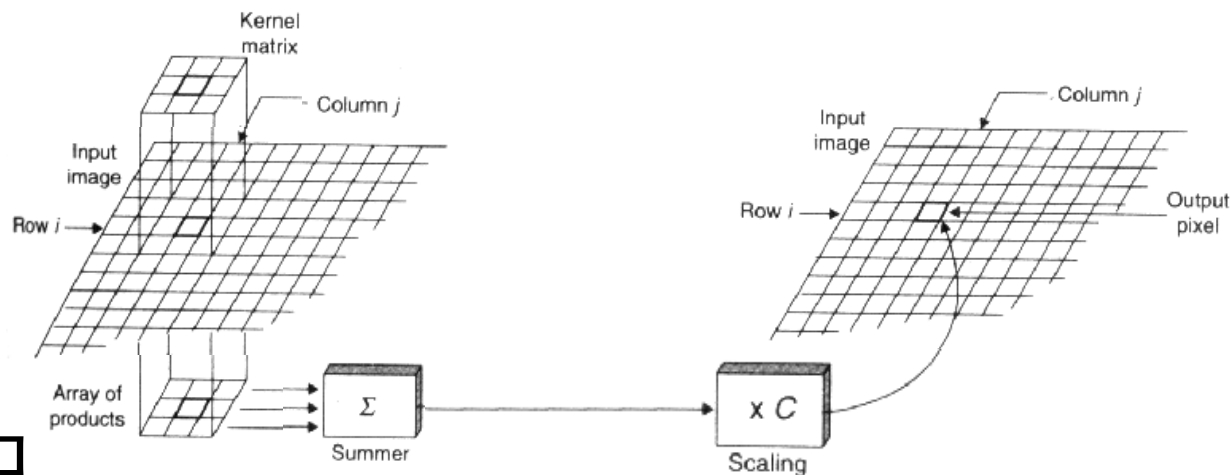
空间滤波基础

三、线性空间滤波

- 线性空间滤波有时也称为图像与掩模的卷积
- 滤波器也称卷积模板
- 对于3x3掩模，可简化写成：

$$R = \sum_{i=1}^9 w_i z_i$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9



四、模板卷积

- 模板卷积编程极为方便
- 只需要一个固定的卷积程序
- 改变不同模板（即3x3或5x5数组中的值），即可适应不同的滤波要求
- 模板卷积在图像预处理中得到非常广泛的应用

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0								

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10							

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20						

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30					

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30				

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30	30			

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30	30	20		

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

均值滤波器



Filtered by



$$h = \frac{1}{25}$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1



均值滤波器

- 包含在滤波器领域内像素的平均值，也称为均值滤波器
- 作用
 - 减小图像灰度的“尖锐”变化，减小噪声
 - 由于图像边缘是由图像灰度尖锐变化引起的，所以也存在边缘模糊的问题

均值滤波器

$$R = \frac{1}{9} \sum_{i=1}^9 z_i$$

a

1	1	1
1	1	1
1	1	1

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

b

1	2	1
2	4	2
1	2	1

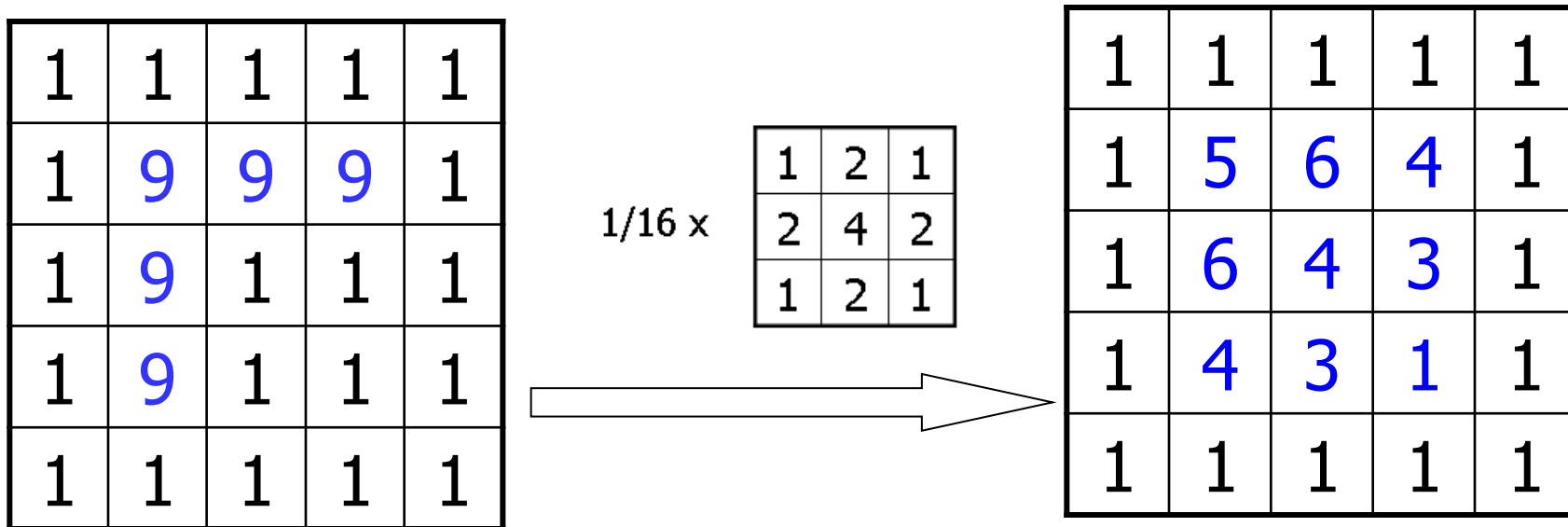
$\frac{1}{16} \times$

图**a**是标准的像素平均值

图**b**是像素的加权平均，表明一些像素更为重要

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

一、平滑线性空间滤波器[计算举例]



均值滤波器——例1



- 模板尺寸对过滤器效果的影响
模板尺寸越大，图像越模糊，图像细节丢失越多

均值滤波器——例3

图像的邻域平均法



(a) 原始图像



(b) 邻域平均后的结果

均值滤波器——例2

— 平滑空域滤波的缺点和问题

如果图像处理的目的是去除噪音，那么，平滑滤波在去除噪音的同时也钝化了图像的边和尖锐的细节



3×3 模板

9×9 模板



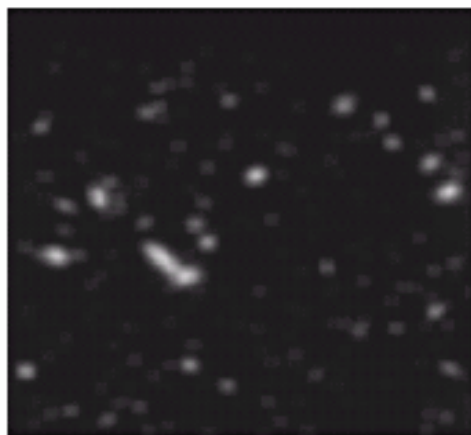
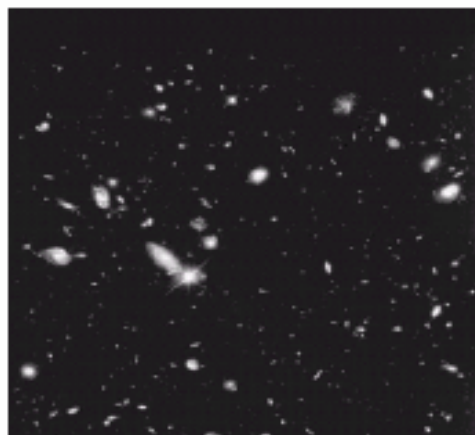
线性滤波器——例4

提取感兴趣物体而模糊图像

原图

15 x 15

阈值=25% x b图像的最高亮度



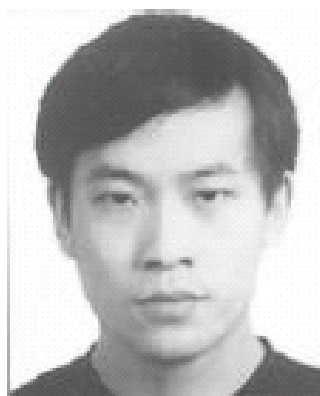
a b c

FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15×15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)



0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

?



0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0



No change



0	0	0	0	0
0	0	0	0	0
0	0	0	0	1
0	0	0	0	0
0	0	0	0	0

?



0	0	0	0	0
0	0	0	0	0
0	0	0	0	1
0	0	0	0	0
0	0	0	0	0



Shift right by 2 pixel

- 平滑空间滤波器的作用
 - 模糊处理:去除图像中一些不重要的细节
 - 减小噪声
- 平滑空间滤波器的分类
 - 线性滤波器: 均值滤波器
 - 非线性滤波器
 - 最大值滤波器
 - 中值滤波器
 - 最小值滤波器

统计排序滤波器

- 什么是统计排序滤波器？
 - 是一种非线性滤波器
 - 基于滤波器所在图像区域中像素的排序，由排序结果决定的值代替中心像素的值
- 分类
 - 中值滤波器：用像素领域内的中间值代替该像素
 - 最大值滤波器：用像素领域内的最大值代替该像素
 - 最小值滤波器：用像素领域内的最小值代替该像素

统计排序滤波器

- 中值滤波器

- ✓ 主要用途：去除噪声

- ✓ 计算公式： $R = \text{mid} \{z_k \mid k = 1, 2, \dots, n\}$

- 最大值滤波器

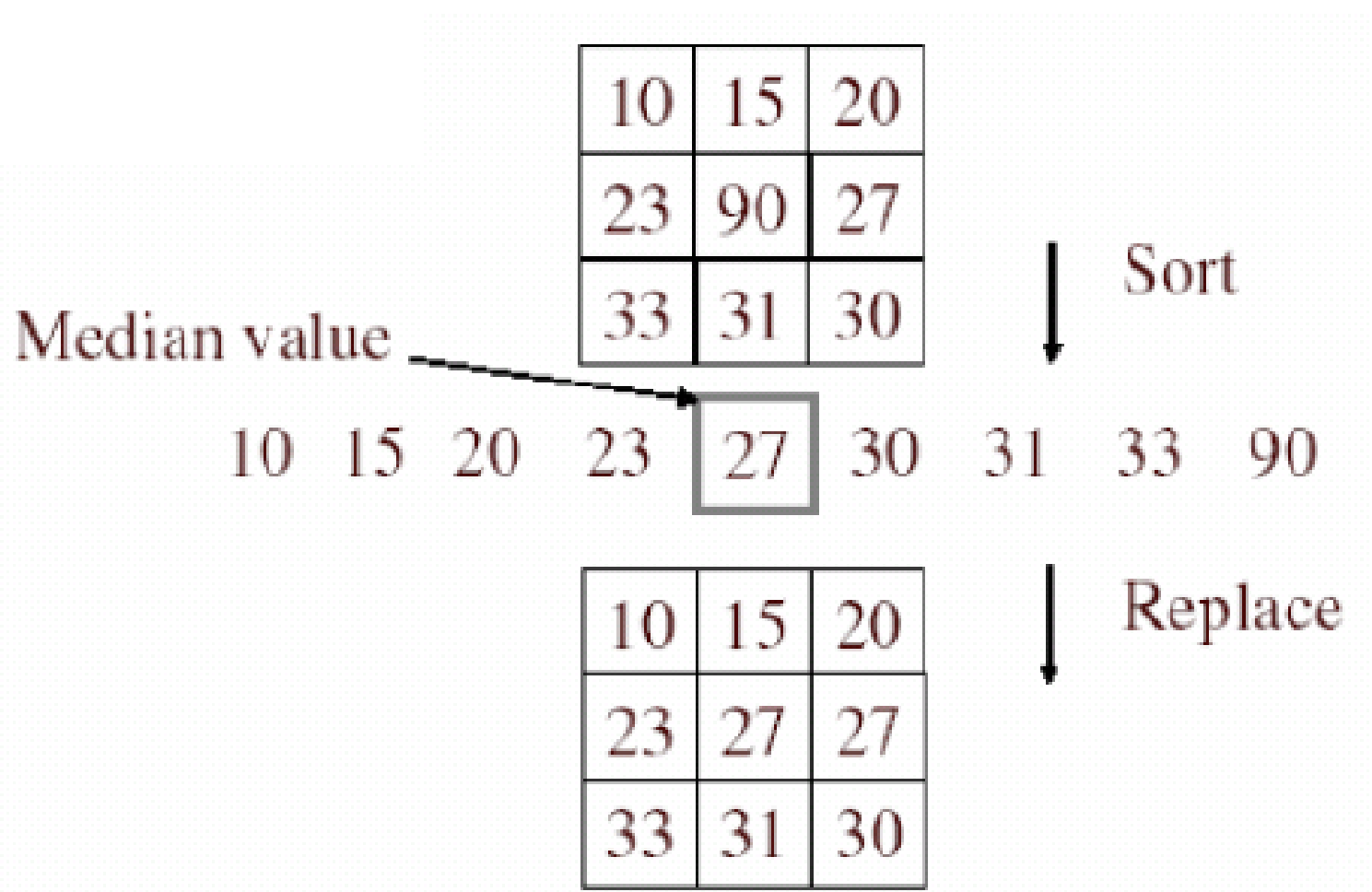
- ✓ 主要用途：寻找最亮点

- ✓ 计算公式： $R = \max \{z_k \mid k = 1, 2, \dots, n\}$

- 最小值滤波器

- ✓ 主要用途：寻找最暗点

- ✓ 计算公式： $R = \min \{z_k \mid k = 1, 2, \dots, n\}$



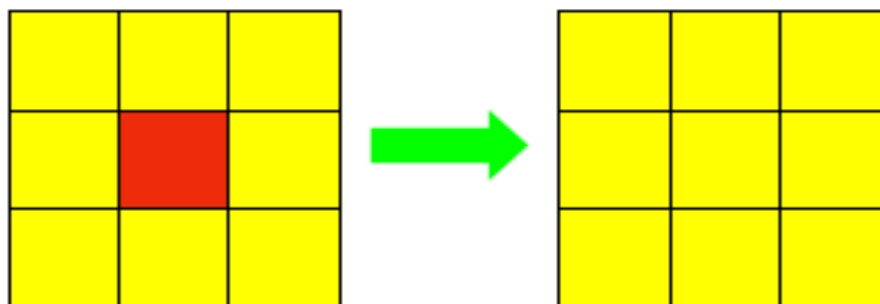
中值滤波器

- 中值滤波的原理

- ✓ 用模板区域内像素的中间值，作为结果值

$$R = \text{mid} \{z_k \mid k = 1, 2, \dots, n\}$$

- ✓ 强迫突出的亮点（暗点）更象它周围的值，以消除孤立的亮点（暗点）



中值滤波器

- 中值滤波算法的实现

- ✓ 将模板区域内的像素排序，求出中间值

- 例如：3x3的模板，第5大的是中值，

- 5x5的模板，第13大的是中值，

- 7x7的模板，第25大的是中值，

- 9x9的模板，第41大的是中值。

- ✓ 对于同值像素，连续排列。

- 如（10, 15, 20, 20, 20, 20, 20, 25, 100）

中值滤波器

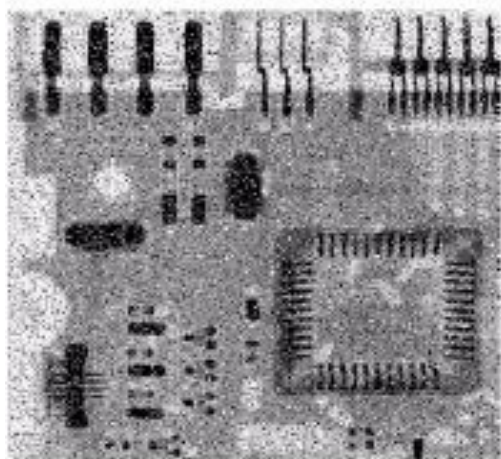
二维中值滤波的窗口形状和尺寸对滤波效果影响较大，不同的图像内容和不同的应用要求，往往采用不同的窗口形状和尺寸。常用的二维中值滤波窗口有线状、方形、圆形、十字形以及圆环形等。窗口尺寸一般先用 3×3 ，再取 5×5 逐渐增大，直到滤波效果满意为止。就一般经验来讲，对于有缓变的较长轮廓线物体的图像，采用方形或圆形窗口为宜。对于包含有尖顶物体的图像，用十字形窗口，而窗口大小则以不超过图像中最小有效物体的尺寸为宜。如果图像中点、线、尖角细节较多，则不宜采用中值滤波。

中值滤波器

- 中值滤波算法的特点
 - 在去除噪音的同时，可以比较好地保留边的锐度和图像的细节（优于均值滤波器）
 - 能够有效去除脉冲噪声：以黑白点叠加在图像上

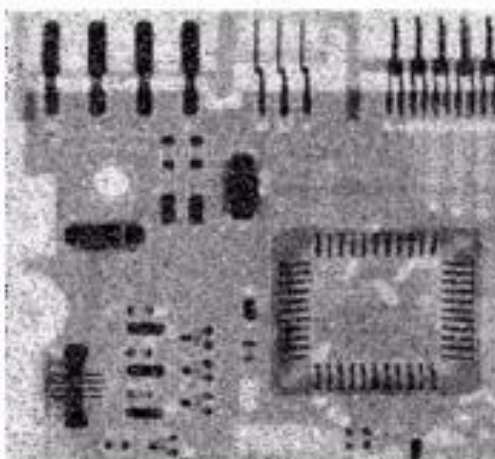
中值滤波器

原图



a b c

3x3均值滤波



3x3中值滤波

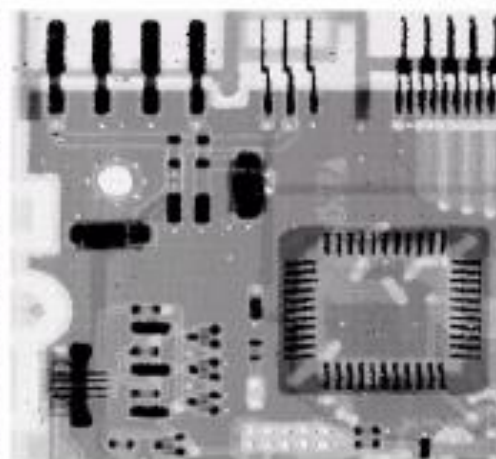
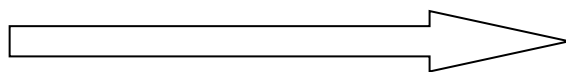


FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

1	1	1	1	1
1	3	4	9	1
1	0	3	4	1
1	4	3	3	1
1	1	1	1	1

中值滤波



1	1	1	1	1
1	1	3	1	1
1	3	3	3	1
1	1	3	1	1
1	1	1	1	1

第4章 图像增强

第六节 图像平滑化处理

二、统计排序滤波器[最大值滤波]

- 最大值滤波器取统计排序结果的最大值作为新图像的像素值

- 例如：3x3掩模覆盖的图像区域值为：

2	3	4
3	8	9
2	4	3

- 排序后： 2, 2, 3, 3, 3, 4, 4, 8, 9

- 取最大值9代替原来的像素值8

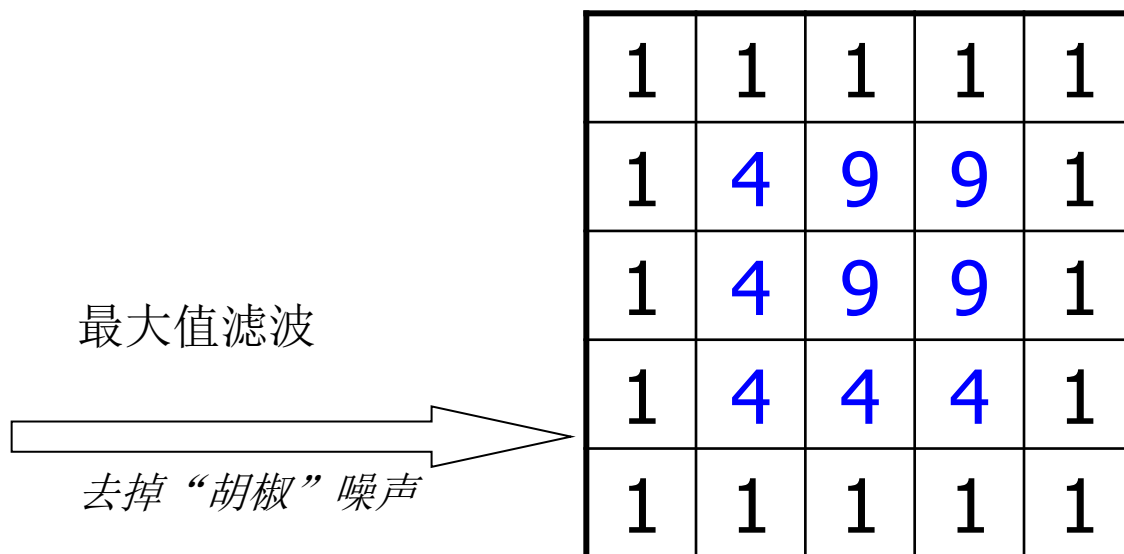
第4章 图像增强

第六节 平滑空间滤波器

二、统计排序滤波器[最大值滤波--计算举例]

- 左边图像经最大值滤波器处理后, 得到右边图像

1	1	1	1	1
1	3	4	9	1
1	0	3	4	1
1	4	3	3	1
1	1	1	1	1

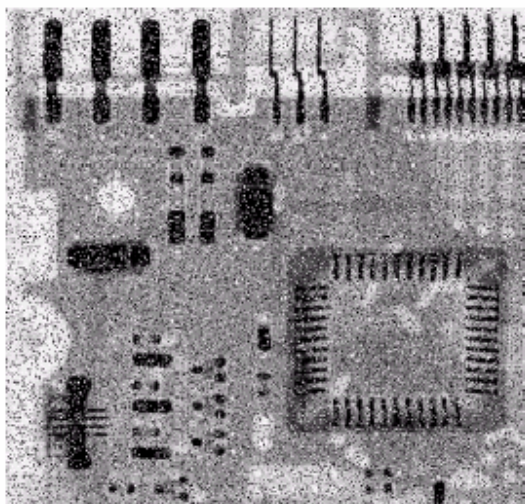


第4章 图像增强

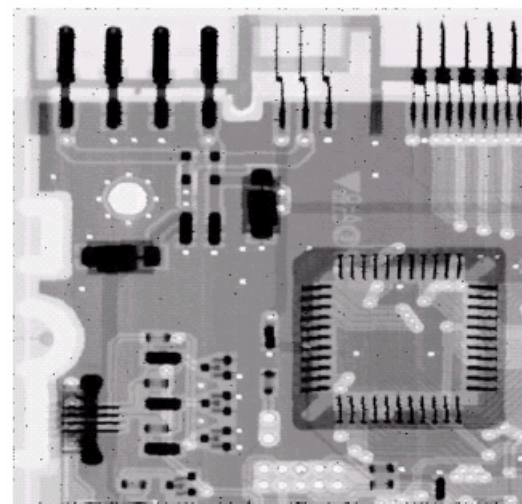
第六节 图像平滑化处理

二、统计排序滤波器[最大值滤波—效果举例]

- 最大值滤波器对处理暗脉冲(胡椒)噪声最为有效



盐椒噪声污染图像



3x3最大值滤波

最大值滤波器



第4章 图像增强

第六节 图像平滑化处理

二、统计排序滤波器[最小值滤波]

- 最小值滤波器取统计排序结果的最小值作为新图像的像素值

- 例如：3x3掩模覆盖的图像区域值为：

2	3	4
3	8	9
2	4	3

- 排序后： 2, 2, 3, 3, 3, 4, 4, 8, 9

- 取最小值2代替原来的像素值8

第4章 图像增强

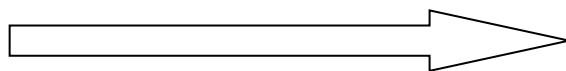
第六节 平滑空间滤波器

二、统计排序滤波器[最小值滤波—计算举例]

- 左边图像经最小值滤波器处理后, 得到右边图像

1	1	1	1	1
1	3	4	9	1
1	0	3	4	1
1	4	3	3	1
1	1	1	1	1

最大值滤波



去掉“盐”噪声

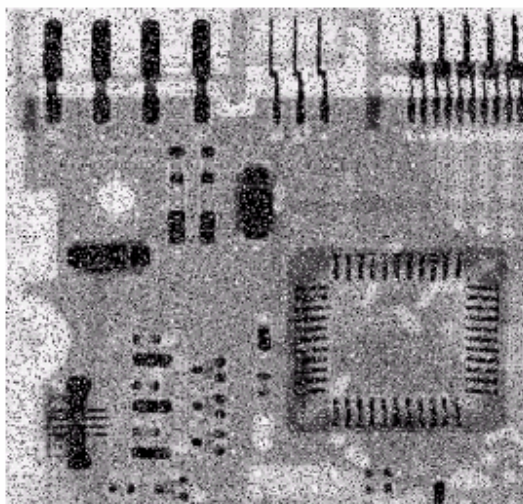
1	1	1	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	1	1	1	1

第4章 图像增强

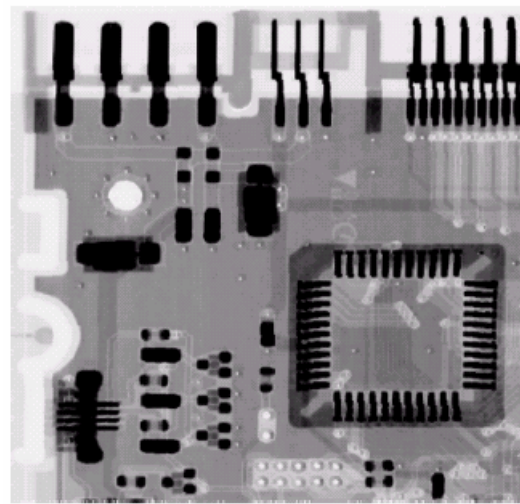
第六节 图像平滑化处理

二、统计排序滤波器[最小值滤波—效果举例]

- 最小值滤波器对处理亮脉冲(盐)噪声最为有效



盐椒噪声污染图像



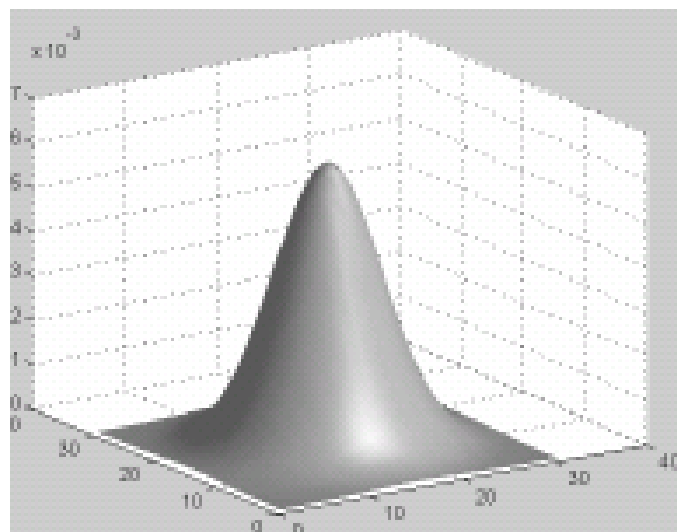
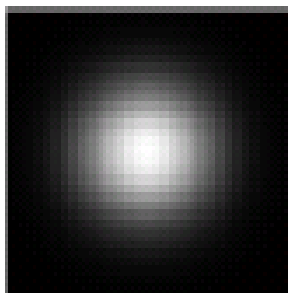
3x3最小值滤波

最小值滤波器



Gaussian平滑滤波器

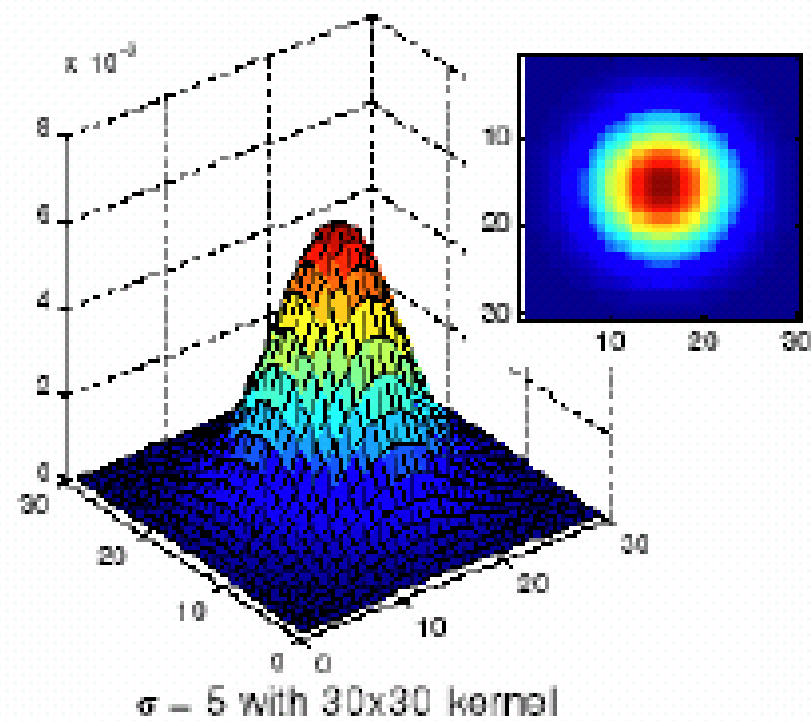
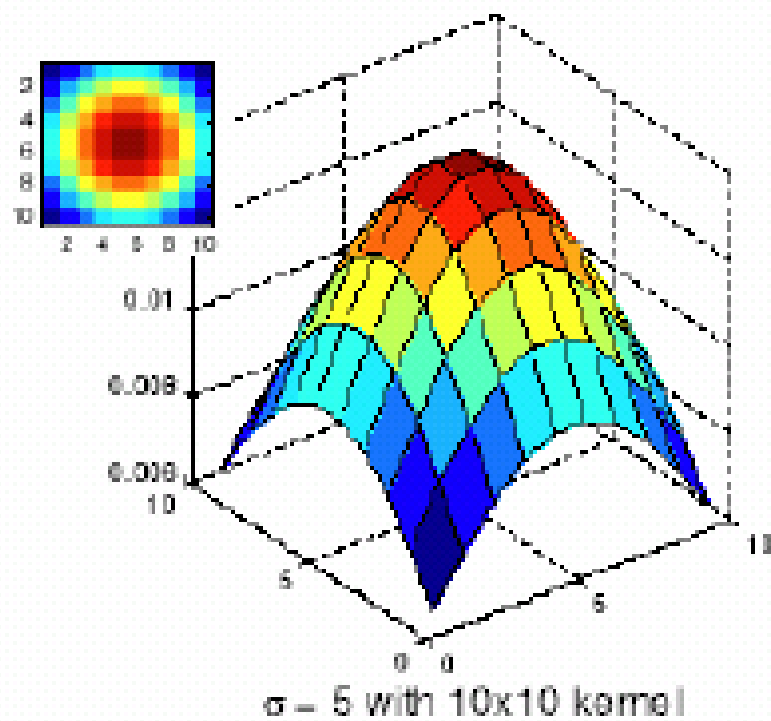
$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (\text{Recall 1D Gaussian})$$

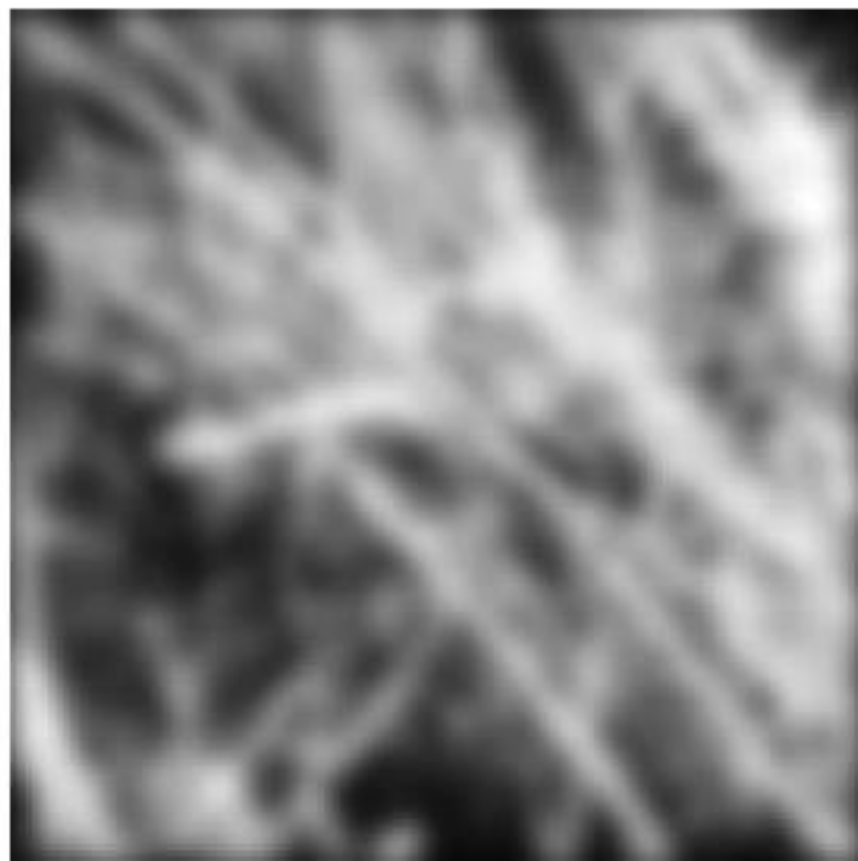


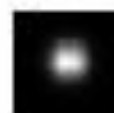
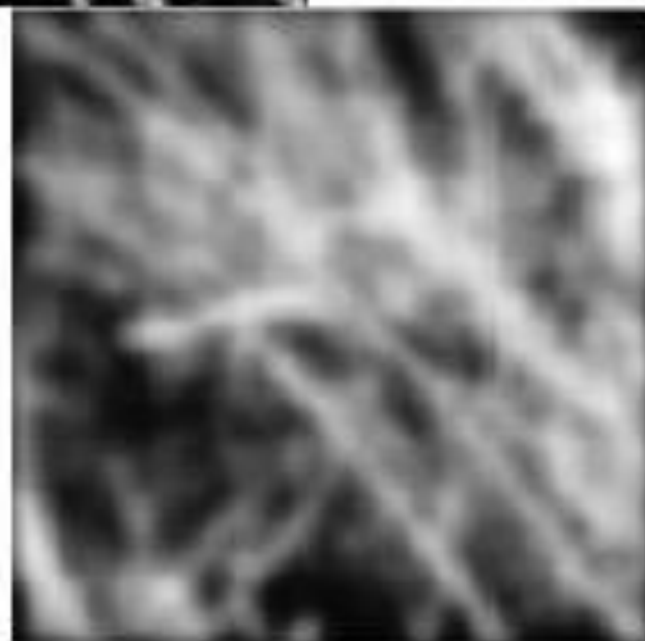
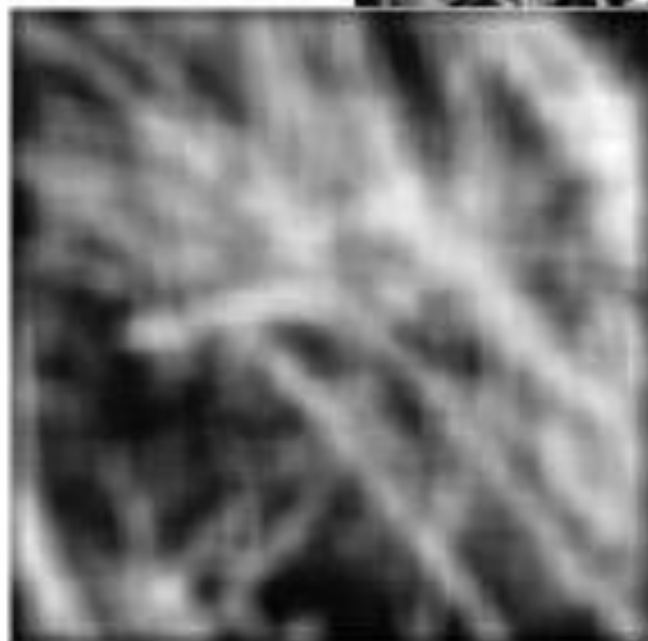
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

$$H_2 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

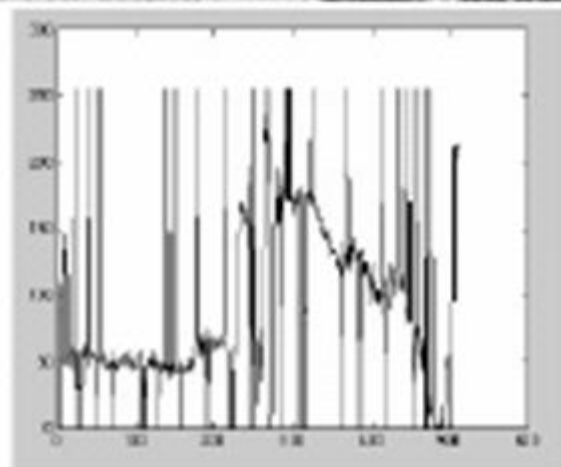




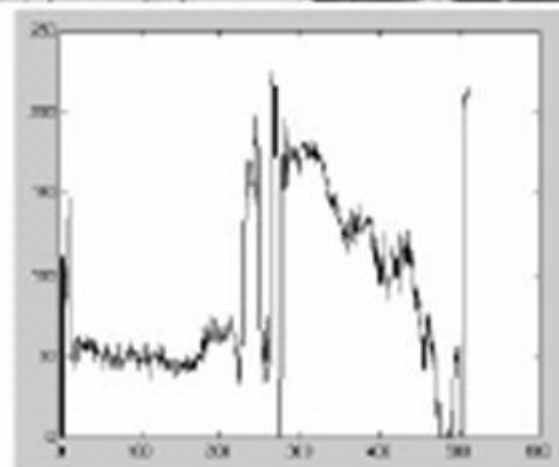


- 椒盐噪声 (Salt and Pepper Noise)
 - 出现位置随机，噪声幅值基本相同
 - 亮噪声
 - 暗噪声
- 高斯噪声 (Gaussian Noise)
 - 每个像素
 - 噪声的幅值随机，服从高斯正态分布

Salt-and-pepper noise

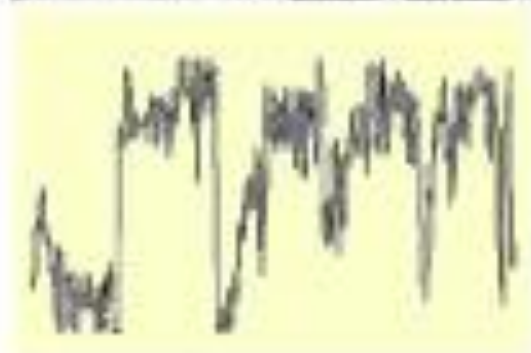


Median filtered



MATLAB: `medfilt2(image, [h w])`

Image Noise



$$f(x, y) = \underbrace{\hat{f}(x, y)}_{\text{Real Image}} + \underbrace{\eta(x, y)}_{\text{Noise process}}$$

Gaussian i.i.d. ('white') noise:
 $\eta(x, y) \sim \mathcal{N}(\mu, \sigma)$



Original



Salt and pepper noise



Impulse noise



Gaussian noise

- **Salt and pepper noise:** contains random occurrences of black and white pixels
- **Impulse noise:** contains random occurrences of white pixels
- **Gaussian noise:** variations in intensity drawn from a Gaussian normal distribution

3x3



5x5



7x7



What's wrong with the results?



高斯滤波, $\sigma^2=1$



高斯滤波, $\sigma^2=3$



Original image



Add pepper noise



3×3 Gaussian filter, $\sigma = 0.5$



3×3 Median filter

