2022春

数字图像处理

图像复原与重建

曹劲舟

助理教授

深圳技术大学

大数据与互联网学院

2022年4月6日

课程概述

周次 Week	周学时 Week Hour	主要教学内容 Course Content	实验实践教学内容 Exercise/Experiment
1	2	图像处理概述及图像处理软件基础	
2	2	图像处理概述及图像处理软件基础	
3	2	灰度变换与空间滤波	
4	2	灰度变换与空间滤波	
5	2+4	频率域滤波	图像增强实验
6	2+4	频率域滤波	图像傅里叶变换实验

课程概述

https://www.caojz.cn/courses/dip/

7	2+4	图像复原与重建	图像的几何变换实验
8	2+4	图像复原与重建	图像复原实验
9	2+4	图像分割	图像分割实验
10	2+4	图像压缩	数字图像压缩与解压
11	2+4	图像压缩	数字图像压缩与解压
12	2+4	彩色图像处理	基于嵌入式设备的图像处理
13	2+4	彩色图像处理	基于嵌入式设备的图像处理
14	2	表示与描述	
15	2	表示与描述	
16	2	基于嵌入式硬件系统的图像处理	
17	2	基于嵌入式硬件系统的图像处理	
18	2	复习与答疑	

概述：图像退化、复原与重建

- **图像复原（image restoration）定义：**利用“图像”退化过程的先验知识，去**恢复已被退化图像**的本来面目。
  - 例如：对图像资料进行大气影响的校正以及对由于设备原因造成的模糊等的改正，将退化图像重建成接近于或完全无退化的原始理想图像的过程。
- **图像重建（image reconstruction）定义：**根据场景的投影数据获取场景中物质分布的信息或重构二、三维场景。
  - 例如：卫星影像立体重建；医学影像重建等。

图像复原和图像增强的区别

图像增强**不考虑图像是如何退化的**，而是试图采用各种技术来增强图像的视觉效果。因此，图像增强可以不顾增强后的图像是否失真，只要看得舒服就行。

而图像复原就完全不同，需知道**图像退化的机制和过程等先验知识**，据此找出一种相应的逆处理方法，从而得到复原的图像。

如果图像已退化，应先作复原处理，再作增强处理。二者的目的都是为了改善图像的质量。

图像退化概念

- **图像退化定义：**图像在**形成、传输和记录**过程中，由于成像系统、传输介质和设备的不完善，使图像的质量变坏，这一过程称为**图像退化**。
- **引起图像退化的原因：**
  - ◆成像系统的散焦
  - ◆成像设备与物体的相对运动
  - ◆成像器材的固有缺陷
  - ◆外部干扰等

图像退化的数学模型

- 成像系统：输入——$f(x,y)$，输出——$g(x,y)$
- 成像系统作用——$H[.]$
- 退化图像为：

$$g(x,y) = H[f(x,y)]$$

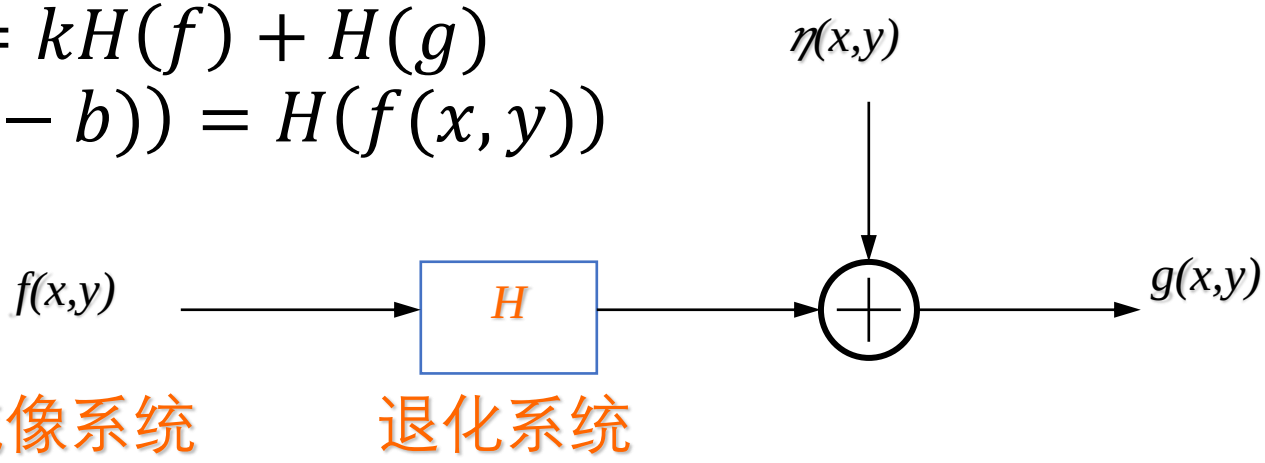
- **线性位移不变成像系统**

退化系统H 满足: $\forall$常数k, a, b 有

(1) 线性: $H(kf + g) = kH(f) + H(g)$

(2) 位移不变性: $H(f(x-a, y-b)) = H(f(x,y))$

计算结果仅取决于输入值，与位置无关



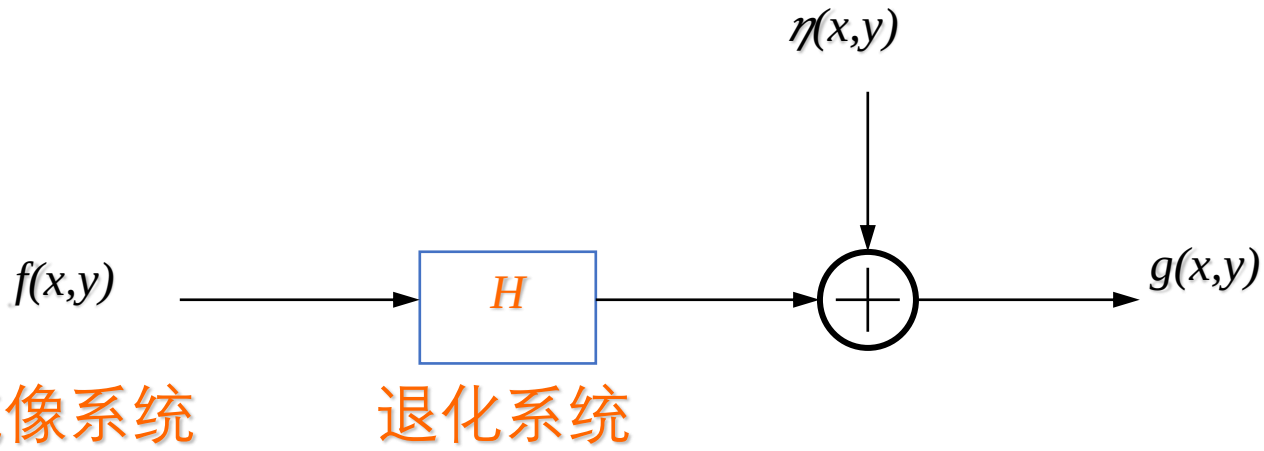
图像退化的数学模型

- **线性位移不变成像系统图像退化模型**

$$g(x,y) = f(x,y) * h(x,y) + \eta(x,y)$$

  - $g(x,y)$ ——退化图像
  - $f(x,y)$ ——理想图像（输入图像）
  - $h(x,y)$ ——点扩散函数（退化函数）
  - $\eta(x,y)$ ——加性噪声

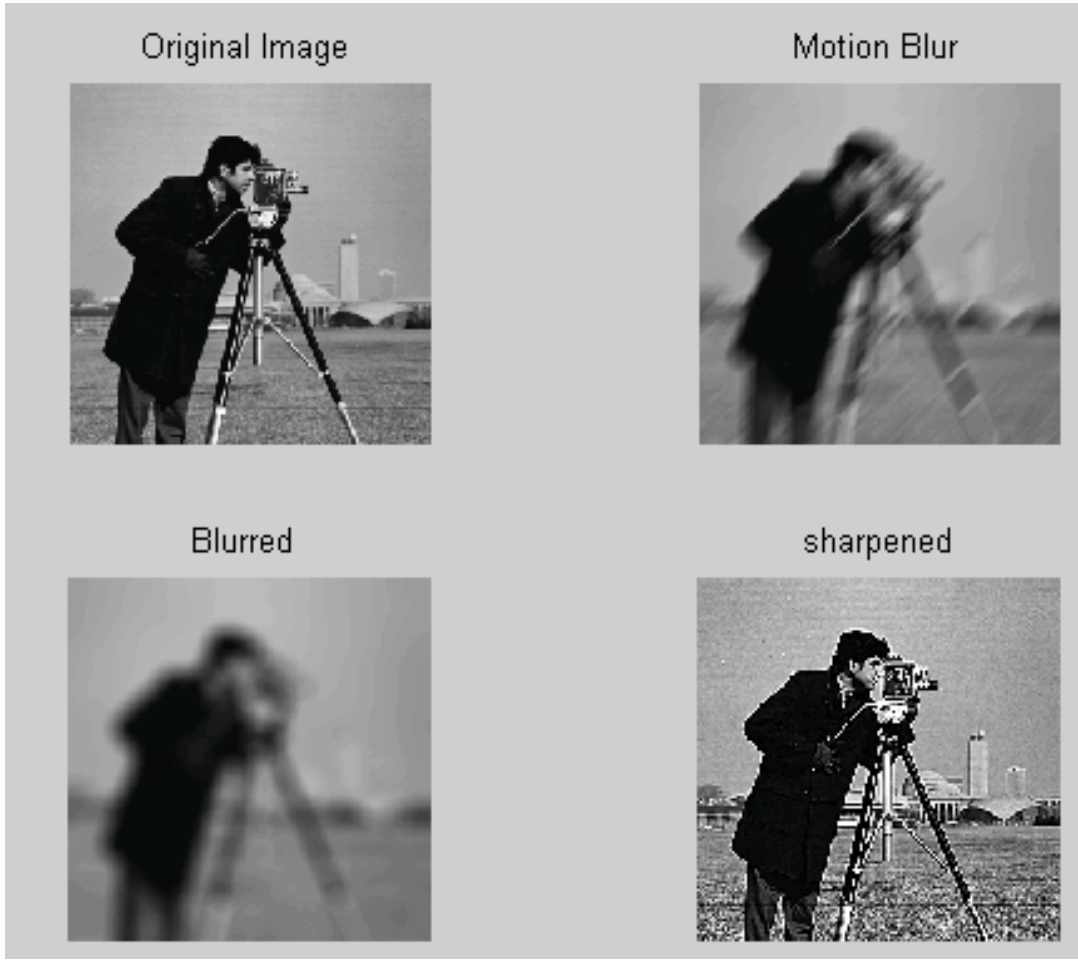
- 空间域 $g(x,y) = h(x,y) \circledast f(x,y) + \eta(x,y)$
- 频率域 $G(x,y) = H(x,y)F(x,y) + N(x,y)$



MATLAB应用实例

模糊图像

```
I=imread('cameraman.tif');
subplot(2,2,1);
imshow(I);title('Original Image');
H=fspecial('motion',20,45);
MotionBlur=imfilter(I,H,'replicate');
subplot(2,2,2);
imshow(MotionBlur);title('Motion Blur');
H=fspecial('disk',10);
blurred=imfilter(I,H,'replicate');
subplot(2,2,3);
imshow(blurred);title('Blurred ');
H=fspecial('unsharp');
sharpened=imfilter(I,H,'replicate');
subplot(2,2,4);
imshow(sharpened);title('sharpened');
```



噪声模型

- **图像噪声**是指造成图像失真、质量下降的图像信号，在图像上表现为引起较强视觉效果的孤立像元点或像元块。
- 噪声来源：
  - **图像获取过程**：成像传感器自身性能和环境条件（热噪声、光照水平等）。
  - **图像传输过程**：信号可能被传输信道中的干扰污染，如通过无线网络传输的图像会受到光或其它大气因素的干扰。

噪声模型

➢空间噪声**不能被预测**，但是可以是由**概率密度函数（PDF）来统计描述**，因此可采用相当简单的概率函数来生成噪声随机数。

➢一些重要的空间噪声类型

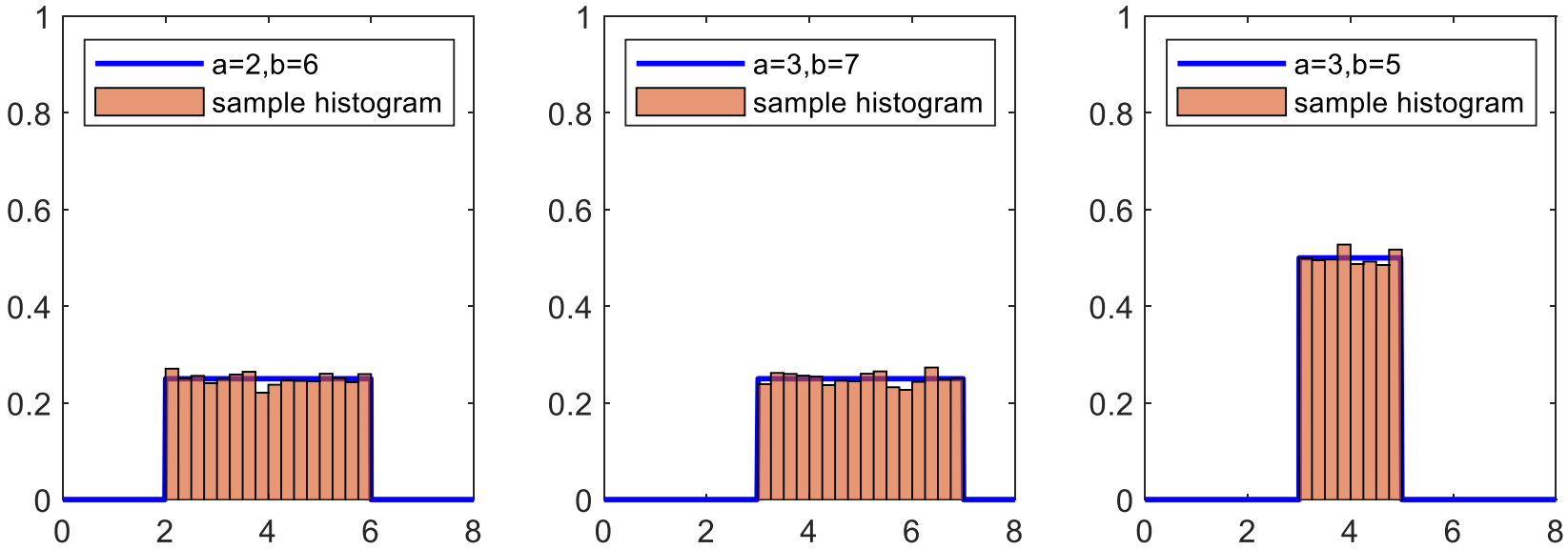
- 高斯噪声
- 瑞利噪声
- 伽马（爱尔兰）噪声
- 指数分布噪声
- 均匀分布噪声
- 脉冲噪声（椒盐噪声）

空间域噪声

1. 均匀噪声

$$p(z)=\begin{cases}\dfrac{1}{\mathrm{b}-\mathrm{a}}, & a\le z\le b\\ 0, & \text{其他}\end{cases}$$

其中，z表示灰度，z的平均值$=(\mathrm{a}+\mathrm{b})/2$，$z$的方差$=\dfrac{(b-a)^2}{12}$。



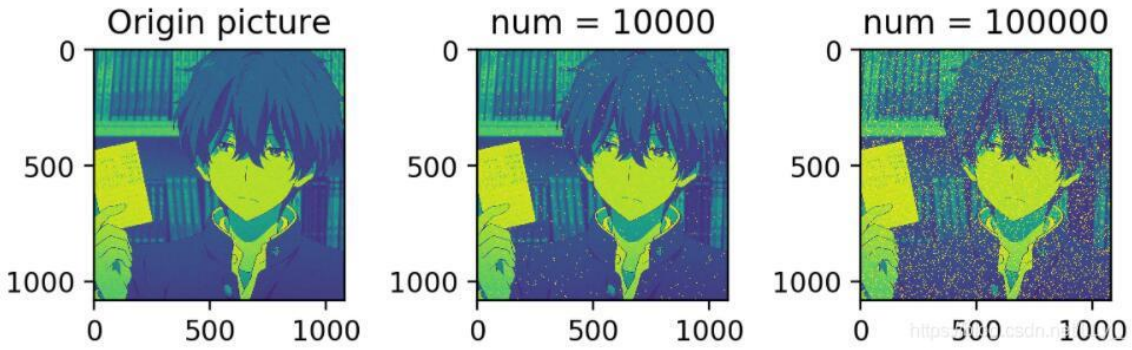
空间域噪声

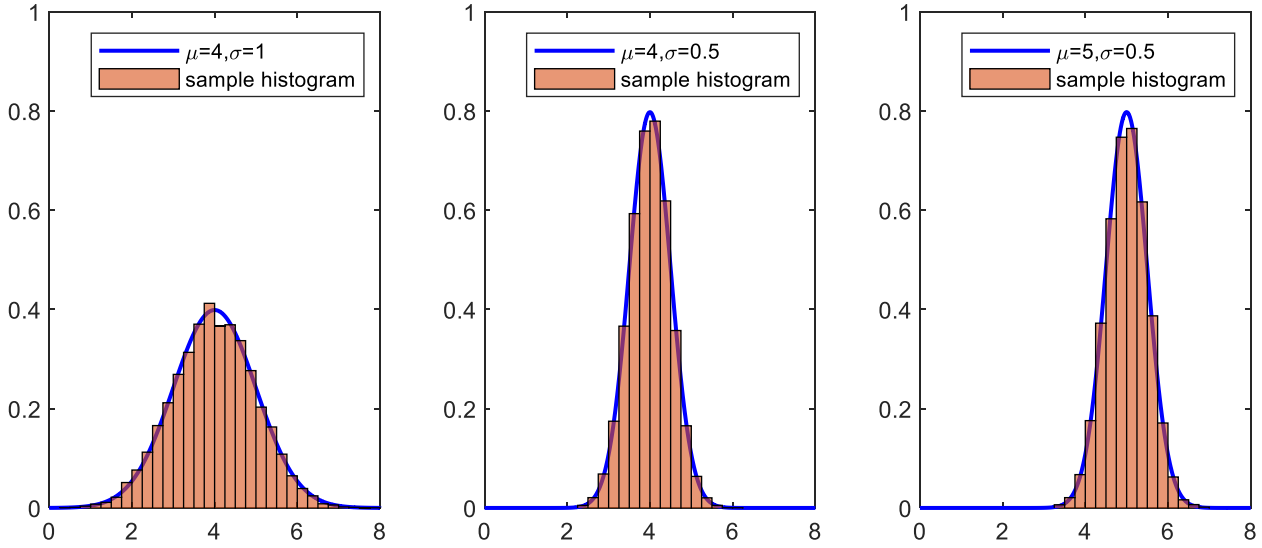
2. 高斯噪声

高斯噪声的概率密度函数(PDF):

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\bar{z})^2}{2\sigma^2}}, \qquad -\infty < z < +\infty$$

其中z表示灰度，$\bar{z}$是灰度z的平均值，σ是灰度z的标准差。





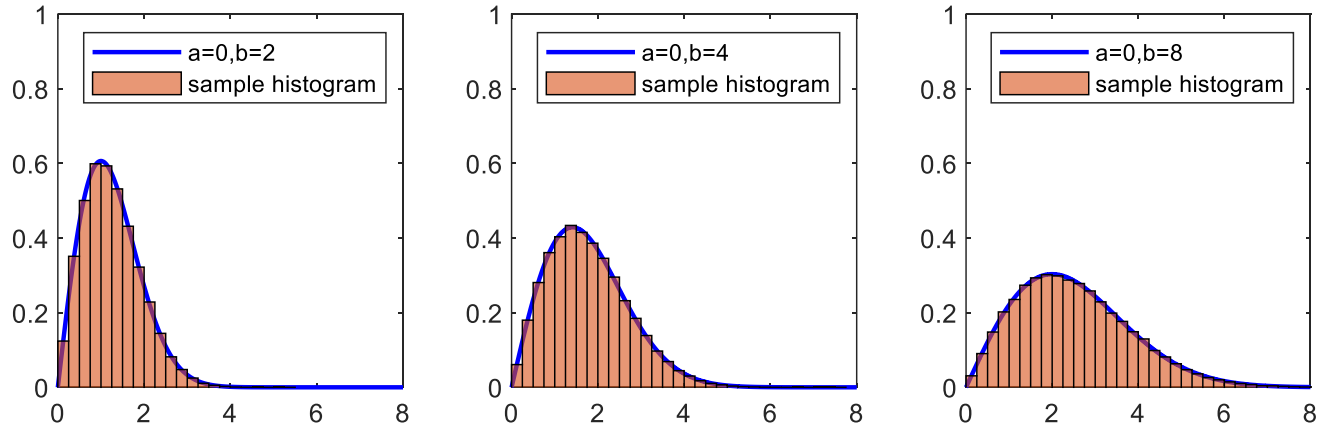
空间域噪声

3. 瑞利噪声

$$p(z)=\begin{cases}\frac{2}{\mathrm{b}}(z-a)e^{-\frac{(z-a)^2}{b}}, & z\geq \mathrm{a}\\ 0, & z<a\end{cases}$$

其中，z表示灰度，z的平均值$=a+\sqrt{\pi b/4}$，z的方差$=b(4-\pi)/4$，a表示距离原点的位移。

当一个随机二维向量的两个分量呈独立的、有着相同的方差的正态分布时，这个向量的模呈瑞利分布。



空间域噪声

4.脉冲噪声（椒盐噪声）

一般是由图像传感器、传输信道及解码处理等产生的黑白相间的亮暗点噪声。

$$p(z)=\begin{cases}P_a & z=a\\ P_b & z=b\\ 1-(P_a+P_b) & \text{其它}\end{cases}$$

- 如果 P_a 或 P_b 为零，则脉冲噪声称为单极脉冲。
- 如果 P_a 或 P_b 均不为零，则脉冲噪声称为双极脉冲噪声或椒盐噪声。

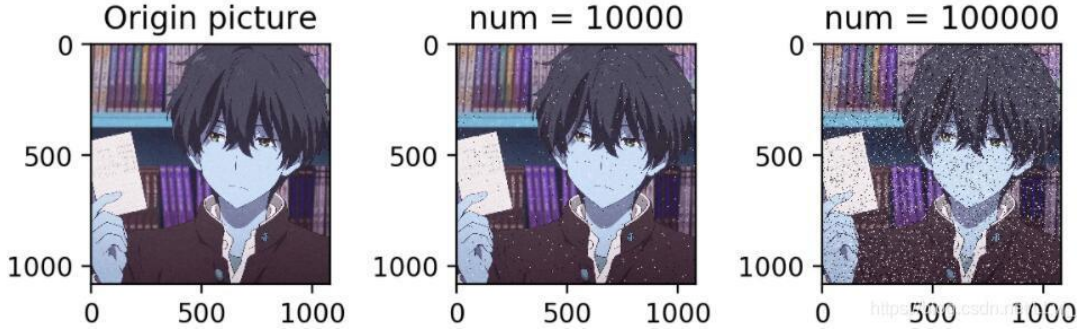
空间域噪声

4. 椒盐噪声

- 椒盐噪声是指两种噪声： 椒噪声 + 盐噪声。盐噪声一般是白色噪声，椒噪声一般是黑色噪声，前者高灰度噪声，后者属于低灰度噪声，一般两种噪声同时出现，呈现在图像上就是黑白杂点。
- 以8比特图像为例，出现0的位置分配0值，出现255的位置分配255，其他值保持不变。

$$p(z)=\begin{cases} P_a & z = 0(胡椒) \\ P_b & z = 2n\text{-}1(盐粒) \\ 1\text{-}(P_a+P_b) & z=k \end{cases}$$

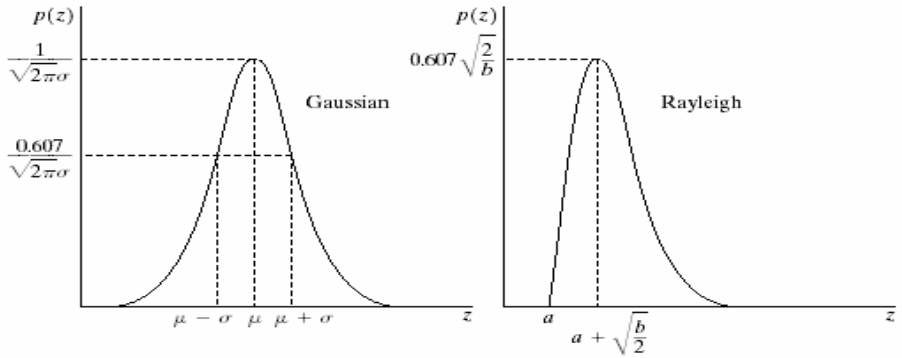
- 脉冲噪声可以为正，也可为负；负脉冲以黑点（胡椒点）出现，正脉冲以白点（盐点）出现。
- 一个像素被椒盐噪声污染的概率为： $P=P_a+P_b$



空间域噪声

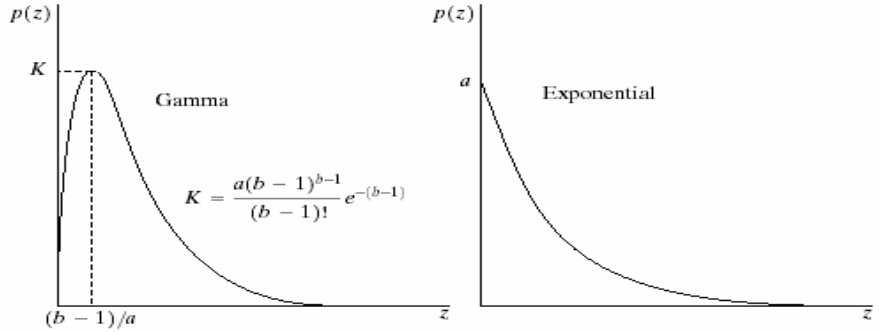
➢一些重要噪声的概率密度函数(PDF)

高斯



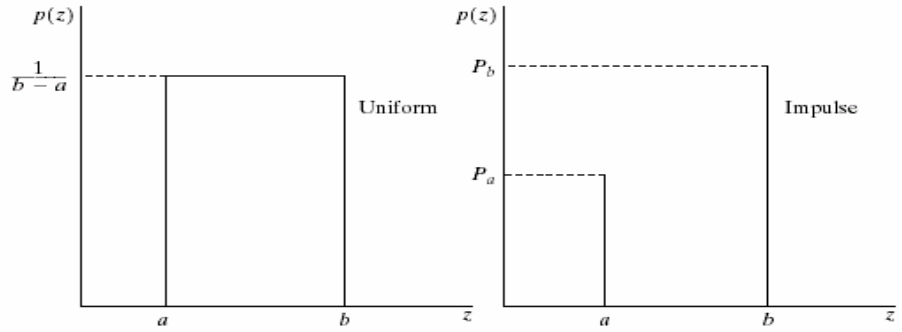
瑞利

伽马



指数

均匀



椒盐

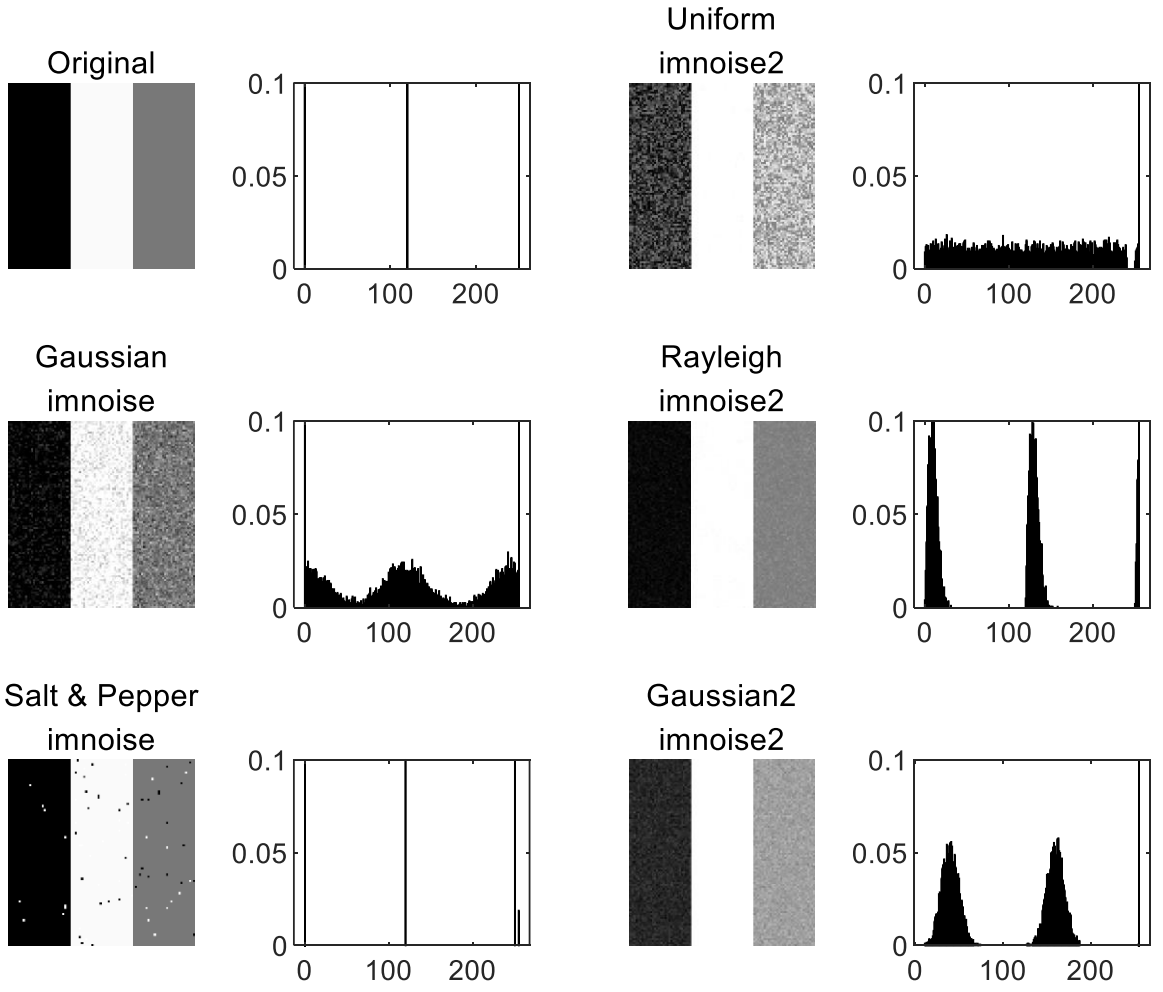
Name	PDF
Uniform	$p(z)=\begin{cases}\frac{1}{b-a} & \text{if } 0\le z\le b\\ 0 & \text{otherwise}\end{cases}$
Gaussian	$p(z)=\frac{1}{\sqrt{2\pi}b}e^{-(z-a)^2/2b^2}$ $-\infty<z<\infty$
Lognormal	$p(z)=\frac{1}{\sqrt{2\pi}bz}e^{-[\ln(z)-a]^2/2b^2}$ $z>0$
Rayleigh	$p(z)=\begin{cases}\frac{2}{b}(z-a)e^{-(z-a)^2/b} & z\ge a\\ 0 & z<a\end{cases}$
Exponential	$p(z)=\begin{cases}ae^{-az} & z\ge 0\\ 0 & z<0\end{cases}$
Erlang	$p(z)=\frac{a^b z^{b-1}}{(b-1)!}e^{-az}$ $z\ge 0$
Salt & Pepper‡	$p(z)=\begin{cases}P_p & \text{for } z=0 \text{ (pepper)}\\ P_s & \text{for } z=2^n-1 \text{ (salt)}\\ 1-(P_p+P_s) & \text{for } z=k\\ & (0<k<2^n-1)\end{cases}$

空间域噪声

$$g(x,y) = f(x,y) + \eta(x,y)$$

- ✓ 用于噪声模型的测试图
- ✓ 由简单、恒定的区域组成
- ✓ 仅仅有3个灰度级的变化

样本噪声图像及其直方图



- **结论**
  - 上述噪声图像的直方图和它们的概率密度函数曲线对应相似
  - 前面5种噪声的图像并没有显著不同，但它们的直方图具有明显的区别

用imnoise函数为图像添加噪声

图像处理工具箱采用***imnoise*** 函数，使噪声污染一幅图像，基本语法

g = imnoise(f, type, parameters)

➢注意：函数 imnoise 在为图像添加噪声之前，将图像转换为范围在[0,1]的double类。

➢将均值为m、方差为var的高斯噪声加到图像上

```
>> f = imread('jimei2.jpg');
>> d = im2double(f); //转换为范围在[0,1]的double类
>> g = imnoise(d, 'gaussian', 0.1, 0.05); //这里均值为0.1，方差为0.05
>> subplot(1,2,1),imshow(d),title('Original Image');
>> subplot(1,2,2),imshow(g,[]),title('Polluted By Gaussian');
```

MATLAB应用实例

加噪声

```
w = imread('x.jpg');
f = im2double(w);    %转换为范围在[0,1]
g = imnoise(f, 'gaussian', 0.1,0.01) ;   %方差为0.01
g = imnoise(f, 'gaussian', 0.1,0.05) ;      %方差为0.05
g = imnoise(f, 'gaussian', 0.1,0.2) ;      %方差为0.2
subplot(2, 2, 1), imshow(w), title('原图像');
subplot(2, 2, 2), imshow(g,[]), title('均值为0.1方差为0.01的高斯噪声加到图像');
subplot(2, 2, 3), imshow(g,[]), title('均值为0.1方差为0.05的高斯噪声加到图像');
subplot(2, 2, 4), imshow(g,[]), title('均值为0.1方差为0.2的高斯噪声加到图像');
```

MATLAB应用实例

加噪声

%图像加噪

```
w = imread('D:\数字图像处理\第四章学习\tree.jpg');
f = im2double(rgb2gray(w));    %把图像变为灰度图像
g = imnoise (f, 'salt & pepper',0.12); %d噪声密度
subplot(1, 3, 1), imshow(w), title('原图像');
subplot(1, 3, 2), imshow(f), title('原图像的灰度图像');
subplot(1, 3, 3), imshow(g), title('椒盐噪声加到图像');
```

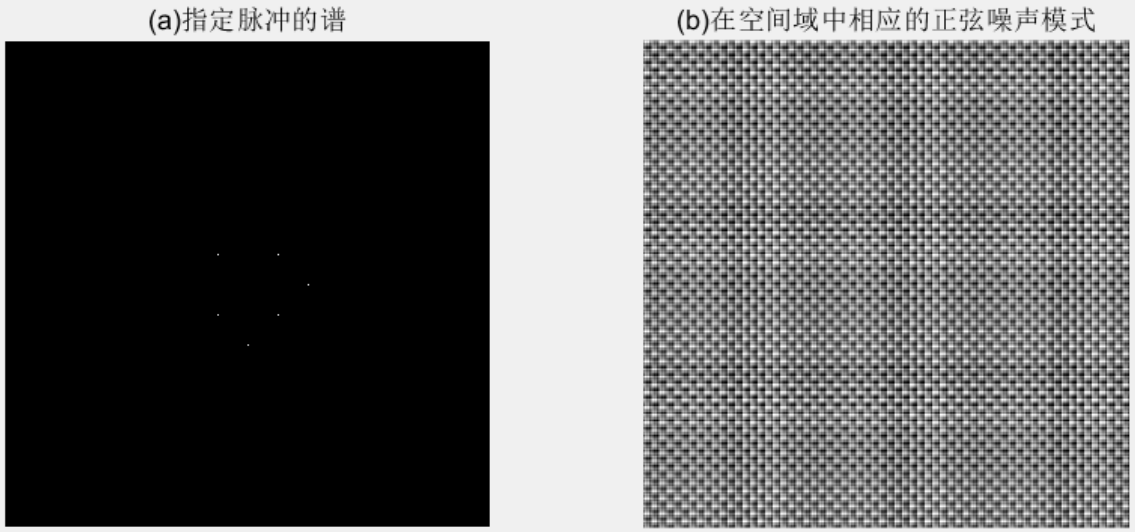
周期噪声

周期噪声：空间上有周期性变化的噪声，是在图像获取中从电力或机电干扰中产生，可以通过频率域滤波显著减少

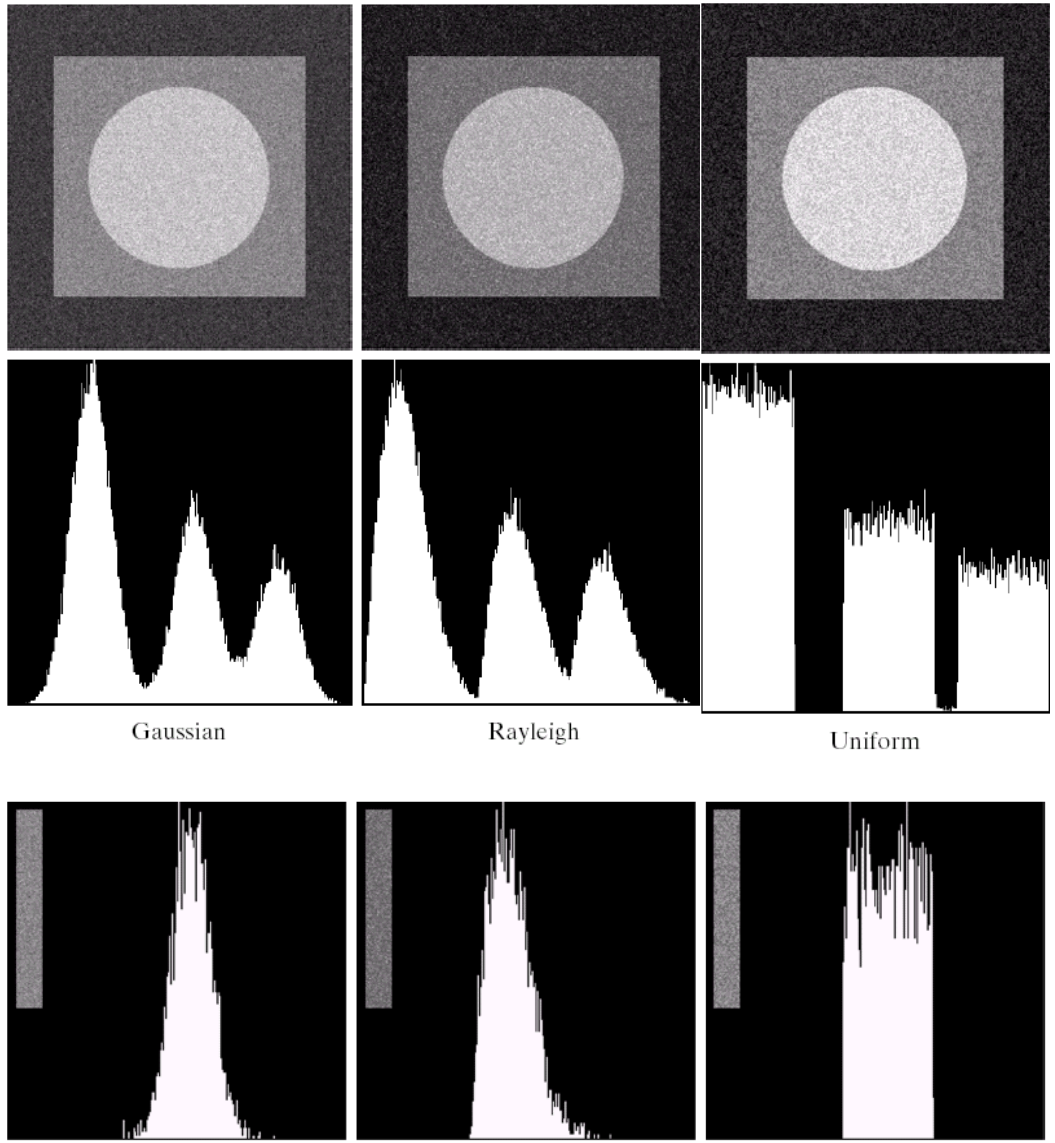
模型：离散的二维正弦波

$$r(x,y) = Asin[\frac{2\pi u_0(x+B_x)}{M} + \frac{2\pi v_0(y+B_y)}{N}]$$

```
>> C = [0 64; 0 128; 32 32; 64 0; 128 0; -32 32];
%K对频域坐标（u，v），表示频域中脉冲的位置
>> [r, R, S] = imnoise3(512,512,C);
% 空间正弦噪声模式r，其傅里叶变换R和频谱S
>> subplot(1,2,1),imshow(S,[]),title('Specified Pulse');
>> subplot(1,2,2),imshow(r,[]),title('Spatial sinusoidal noise');
```



估计噪声参数



无法使用噪声直方图推测噪声PDF

最好选取图像中的一个无特色的背景区域，以便该区域灰度值的变化主要由噪声引起（ROI），来进行推测

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

图像复原

- **空间滤波：仅有噪声的复原**
  - 均值滤波器（算数平均、几何均值、谐波平均、反谐波平均...）
  - 统计排序滤波器（中值滤波、最大最小值、中点、修正阿尔法值...）
  - 自适应滤波器（自适应局部降噪、自适应中值...）

- **频率域滤波：削减周期噪声**
  - 陷波滤波器

空间域滤波复原

- 当唯一的退化是噪声时，

$$g(x,y) = f(x,y) * h(x,y) + \eta(x,y)$$

$$g(x,y) = f(x,y) + \eta(x,y)$$

$g(x,y)$ ——退化图像
$f(x,y)$ ——理想图像（输入图像）
$h(x,y)$ ——点扩散函数
$\eta(x,y)$ ——加性噪声

- 但是，噪声项未知，不能从$g(x,y)$减去噪声
- 可以选择空间滤波方法进行图像复原

空间噪声滤波器

➢图像复原的空间滤波器

- ➢ 均值滤波器

  算术均值滤波器、几何均值滤波器、调和均值滤波器、逆调和均值滤波器

- ➢ 顺序统计滤波器

  中值滤波器、最大值滤波器、最小值滤波器、中点滤波器、修正后的阿尔法均值滤波器

- ➢ 自适应滤波器

  自适应局部噪声消除滤波器、自适应中值滤波器

均值滤波器

- **算数均值滤波器**

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(r,c)\in S_{xy}} g(r,c)$$

其中S_{xy}表示中心为(x,y)、大小为m×n的矩形子图窗口（邻域）的一组坐标。 r,c是邻域S_{xy}中包含的像素的行坐标和列坐标。

- **特点：**
  - 平滑了一幅图像的局部变化，在模糊了结果的同时减少了噪声

均值滤波器

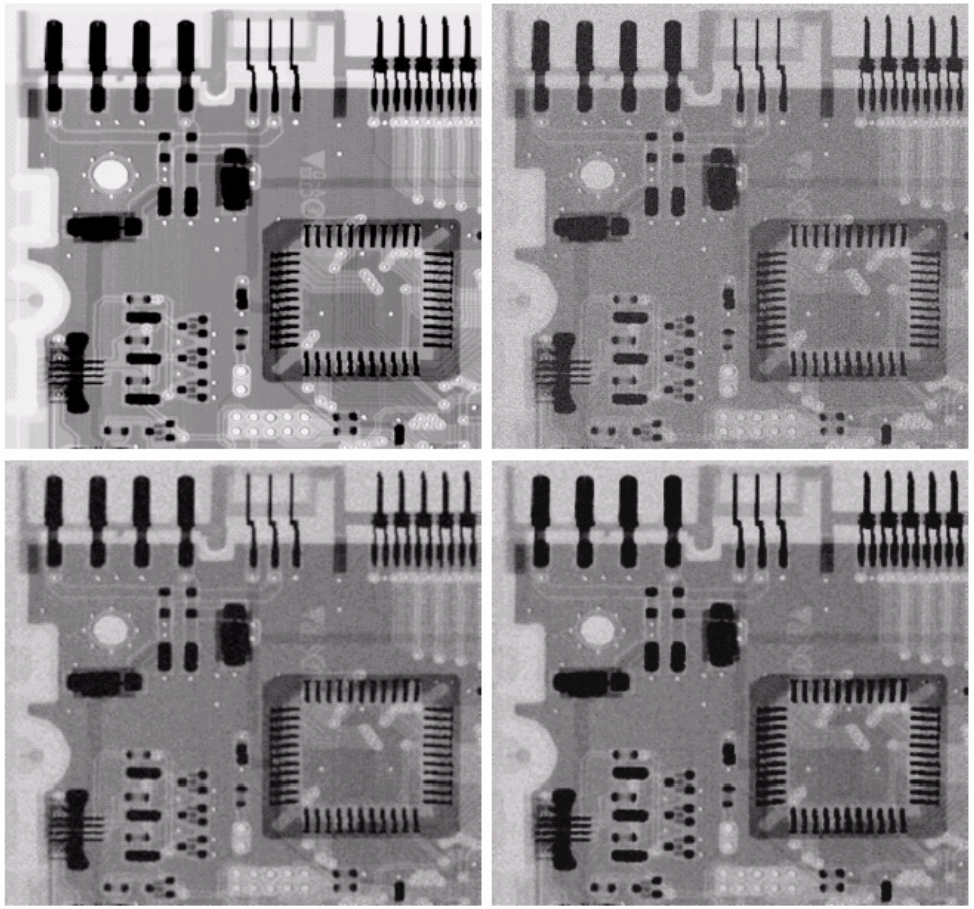
• 几何均值滤波器

$$\hat{f}(x,y) = \left[\prod_{(r,c)\in S_{xy}} g(r,c) \right]^{\frac{1}{mn}}$$

- **特点：**
  - 几何均值滤波器所达到的平滑度可以与算术均值滤波器相比
  - 几何均值滤波器在滤波过程中，与算术均值滤波器相比，会丢失更少的图像细节——相对锐化

均值滤波器

原图　　被均值为0，方差为400的高斯噪声污染



a b
c d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

3×3算术均值滤波器　　3×3几何均值滤波器，图像更清晰

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

均值滤波器

- **调和均值滤波器**

$$\hat{f}(x,y) = \frac{mn}{\sum_{(r,c)\in S_{xy}} \frac{1}{g(r,c)}}$$

- **特点：**
  - 调和均值滤波器对于“盐”噪声效果好，但不适用于“胡椒”噪声。
  - 善于处理高斯噪声。

均值滤波器

- **逆调和平均滤波器**

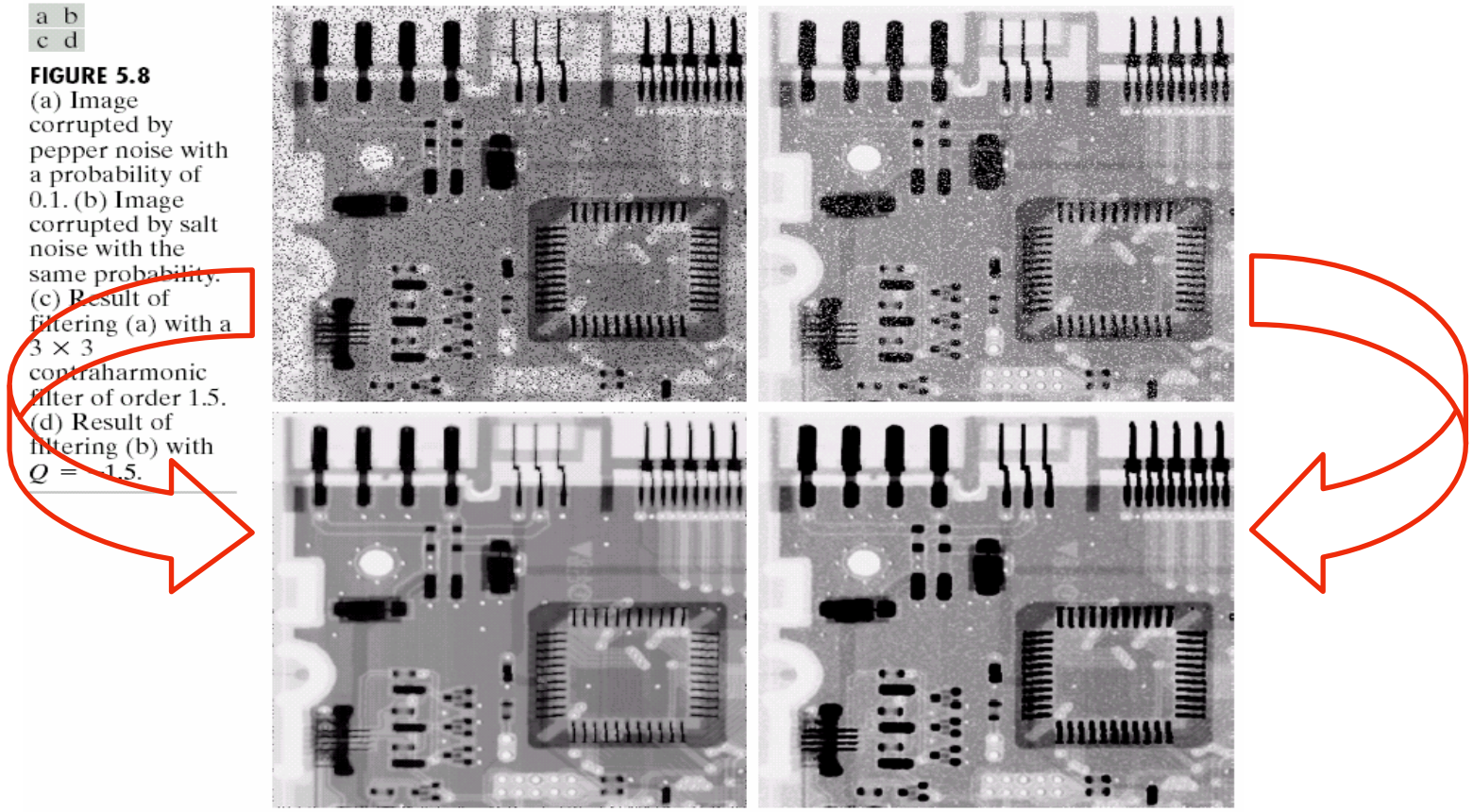
$$\hat{f}(x,y) = \frac{\sum_{(r,c)\in S_{xy}} g(r,c)^{Q+1}}{\sum_{(r,c)\in S_{xy}} g(r,c)^{Q}}$$

- Q称为滤波器的阶数。
- 适用于消除椒盐噪声的影响：当Q为正数时，消除“胡椒”噪声；当Q为负数时，消除“盐”噪声，但不能同时消除“椒盐”噪声。
- 当Q=0，转变为算术均值滤波器。
- 当Q=-1，转变为调和均值滤波器。

均值滤波器

prob. = 0.1

“胡椒”噪声干扰图像　　“盐”噪声干扰图像



a b
c d

FIGURE 5.8
(a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

3×3大小，Q=1.5的逆调和均值滤波器

3×3大小，Q=-1.5的逆调和均值滤波器

DIP4-3-2.m

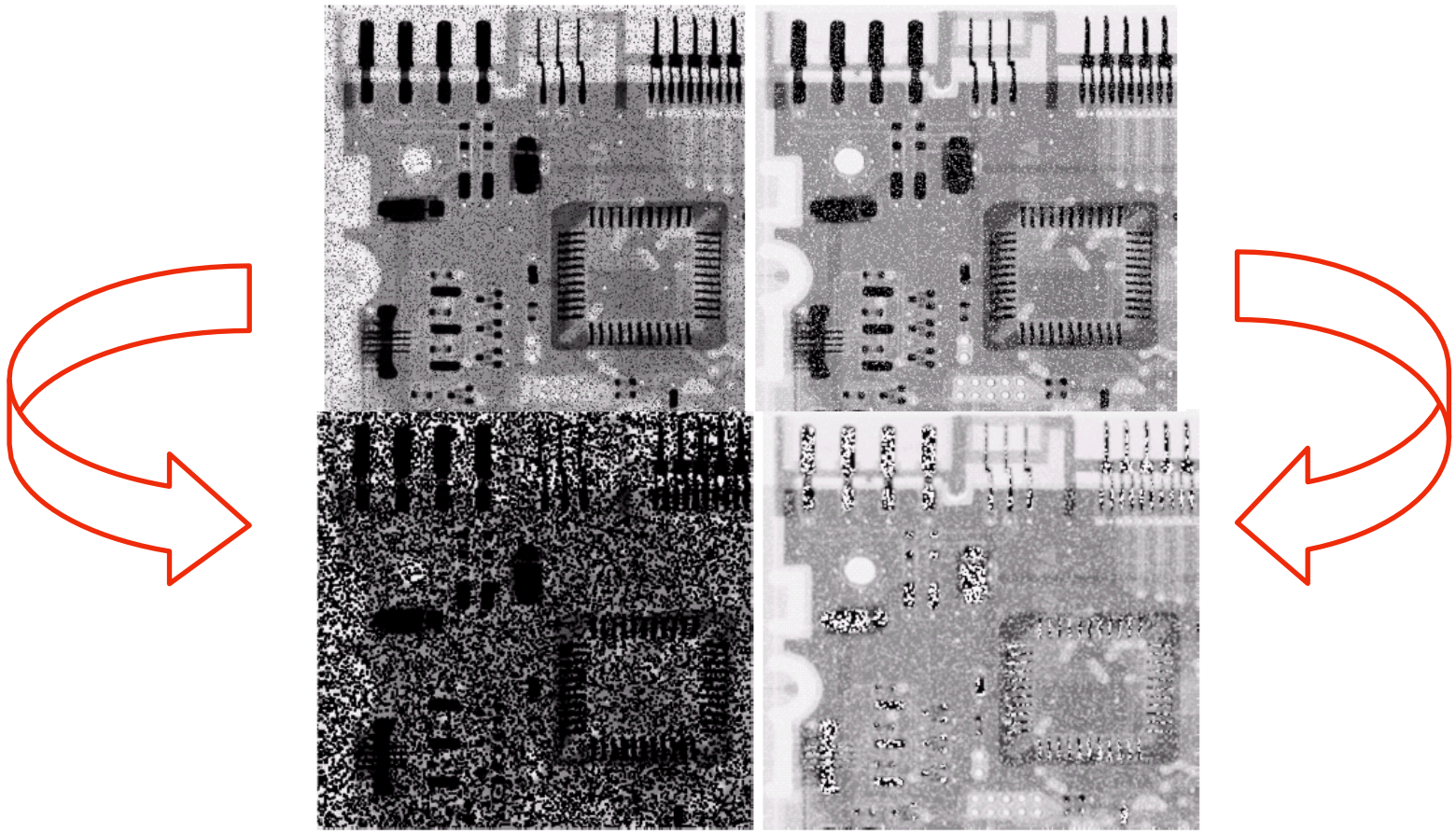
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

均值滤波器

错误案例

prob. = 0.1

“胡椒”噪声干扰图像　　“盐”噪声干扰图像



3×3大小，Q=-1.5的逆调和均值滤波器

3×3大小，Q=1.5的逆调和均值滤波器

DIP4-3-2.m

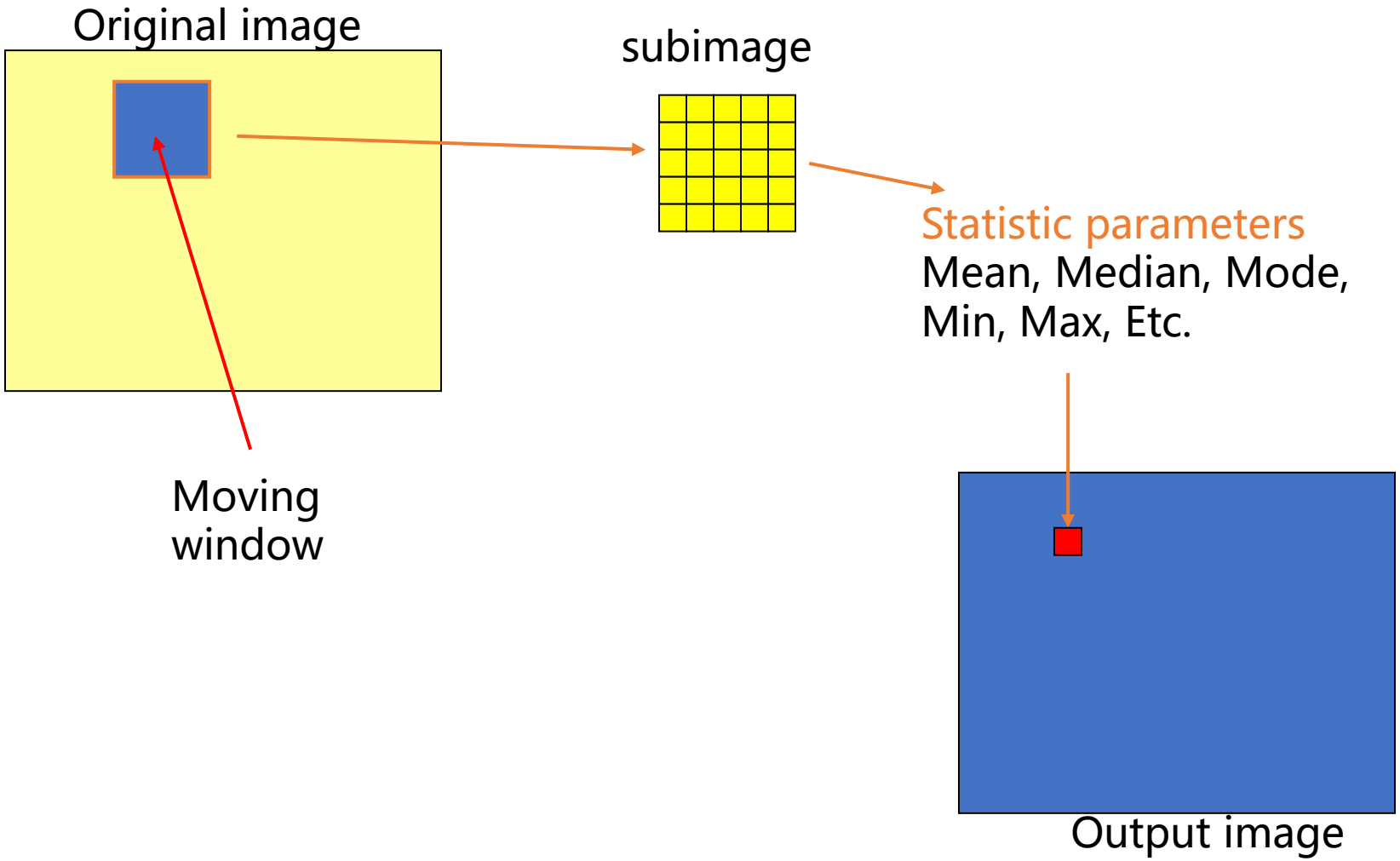
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

均值滤波器

- **总结**
  - **算术均值滤波器**和**几何均值滤波器**适合处理高斯或均匀等随机噪声。
  - **调和均值滤波器**适合处理脉冲噪声（椒盐噪声），但必须事先知道噪声是“盐”还是“胡椒”，以便于选择合适的Q符号。

顺序统计滤波器

- 顺序统计滤波器的响应是基于由滤波器包围的图像区域中像素点的排序，任一点的响应由**排序结果**决定。



顺序统计滤波器

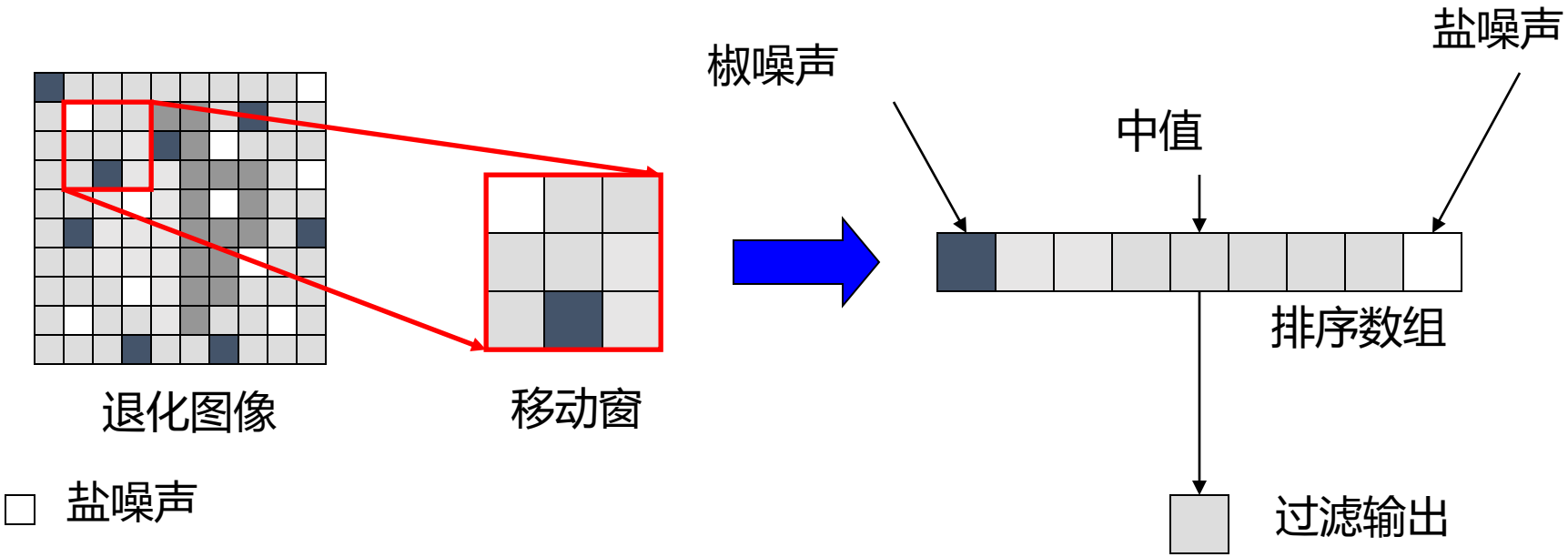
- **中值滤波器**

  使用一个像素邻域中的灰度级的中值来替代该像素的值

$$\hat{f}(x,y) = median_{(r,c)\in S_{xy}}\{g(r,c)\}$$

- **特点：**
  - 在相同尺寸下，比起均值滤波器引起的模糊少。
  - 对椒盐噪声非常有效。

中值滤波器示例

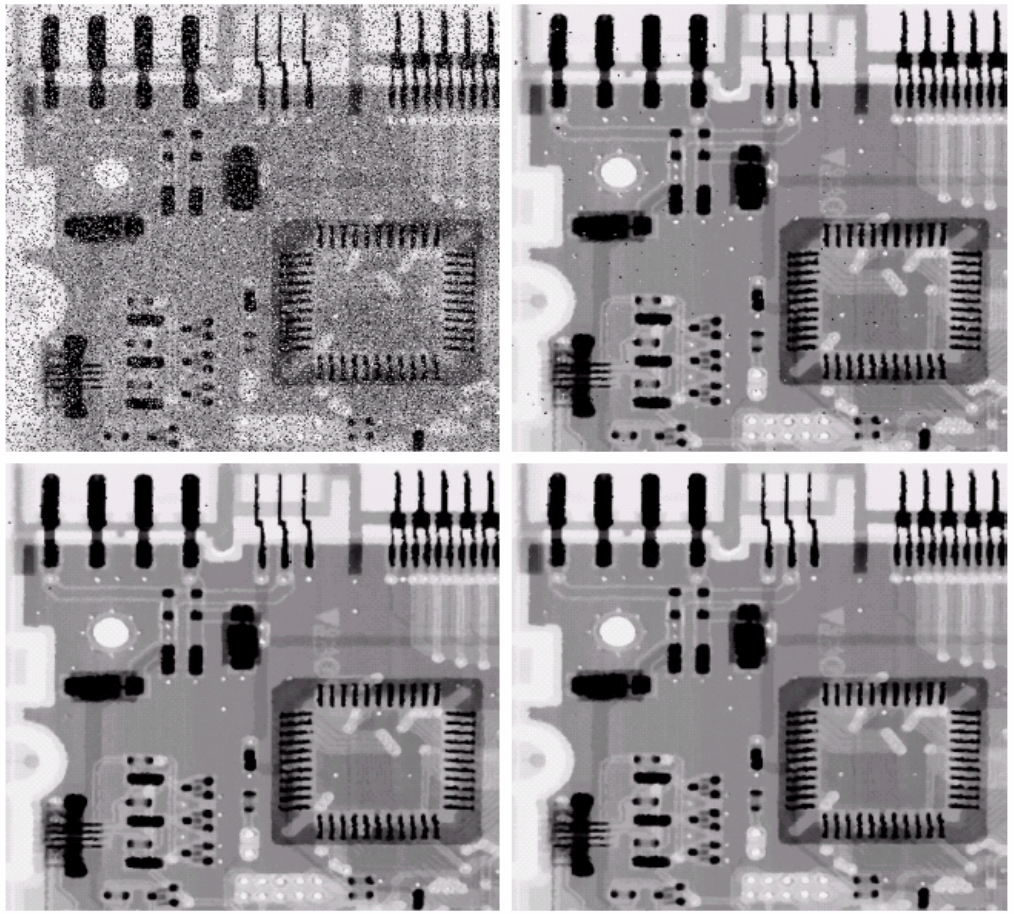


通常, 脉冲噪声是极值。当在移动窗中进行排序时，噪声像元通常位于数组的末端。

噪声像元通常不可能是中值，可有效去除

空间噪声滤波器

P_a=P_b=0.1的脉冲噪声　　3×3的中值滤波器



a b
c d

FIGURE 5.10
(a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$.
(b) Result of one pass with a median filter of size 3 × 3.
(c) Result of processing (b) with this filter.
(d) Result of processing (c) with the same filter.

第二次中值滤波器处理　　第三次中值滤波器处理，全部噪声消除

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

顺序统计滤波器

- **最大/最小值滤波器**

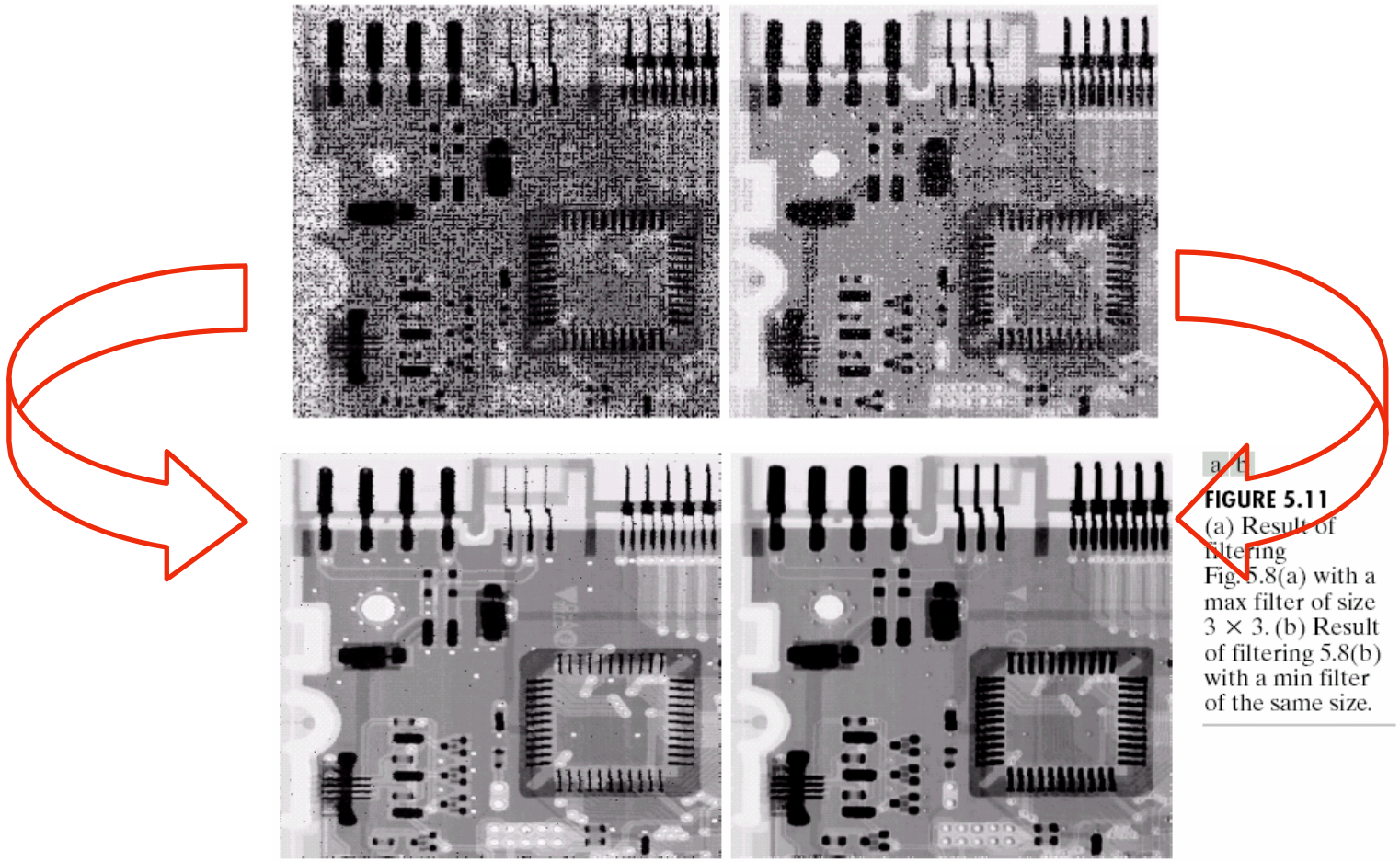
$$\hat{f}(x,y) = max_{(r,c)\in S_{xy}}\{g(r,c)\}$$

$$\hat{f}(x,y) = min_{(r,c)\in S_{xy}}\{g(r,c)\}$$

- **特点：**
  - 最大值滤波器：用于发现图像中的最亮点；可以有效过滤“胡椒”噪声（因为“胡椒”噪声是非常低的值）
  - 最小值滤波器：用于发现图像中的最暗点；可以有效过滤“盐”噪声（因为“盐”噪声是非常高的值）

空间噪声滤波器

“胡椒”噪声干扰图像　　“盐”噪声干扰图像



a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.

最大值滤波器处理　　最小值滤波器处理

顺序统计滤波器

- **中点滤波器**

$$\hat{f}(x,y) = \frac{1}{2}[min_{(r,c)\in S_{xy}}\{g(r,c)\} + max_{(r,c)\in S_{xy}}\{g(r,c)\}]$$

- **特点：**
  - 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效果。

顺序统计滤波器

- **修正阿尔法滤波器**

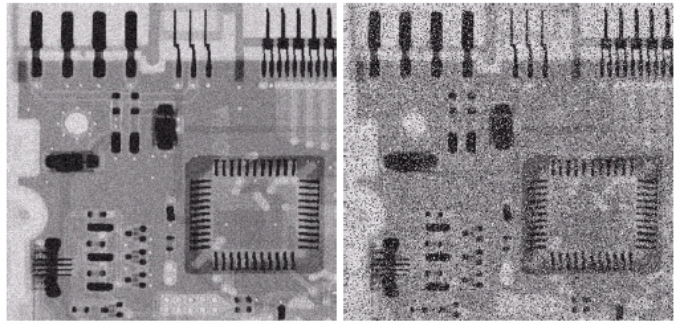
$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(r,c)\in S_{xy}} g_R\,(r,c)$$

- 在S_{xy}邻域内删除$g_R\,(r,c)$ d/2个最低灰度值和d/2个最高灰度值。
- $g_R\,(r,c)$代表剩余的$mn-d$个像素
- 当d=0，退变为算术均值滤波器
- 当d=(mn-1)/2，退变为中值滤波器
- 当d取其它值时，适用于包括**多种噪声**的情况下，例如高斯噪声和椒盐噪声混合的情况

空间噪声滤波器

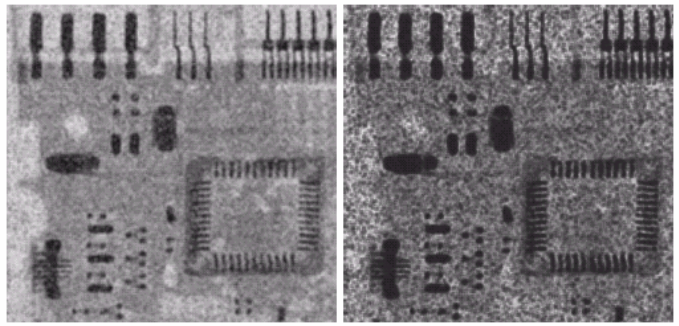
由于脉冲噪声的存在，算术均值和几何均值滤波器没有起到好的作用

均值为0，方差为 800 的噪声干扰的图像



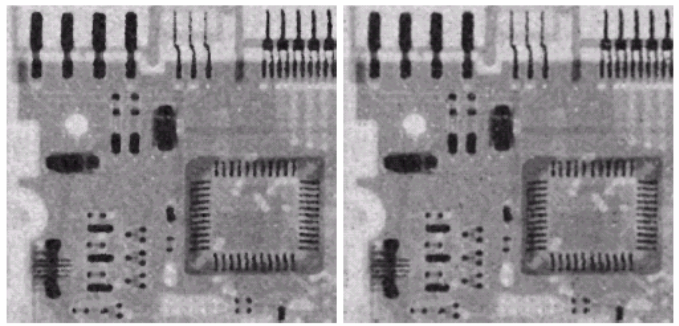
被$P_a=P_b=0.1$的椒盐噪声叠加，进一步恶化

算术均值滤波器 x



几何均值滤波器×

中值滤波器



d=5, 规格为5×5的修正后的阿尔法均值滤波器

√

DIP4-3-2.m

空间噪声滤波器

例 4.5　使用函数 spfilt。

图 4.5(a)所示的图像是一幅被概率只有 0.1 的胡椒噪声污染的 uint8 类图像。这幅图像是使用下面的命令生成的［f 是来自图 2.19(a)的图像］：

```
>> [M, N] = size(f);
>> R = imnoise2('salt & pepper', M, N, 0.1, 0);
>> gp = f;
>> gp(R == 0) = 0;
```

图 4.5(b)中的图像仅被盐粒噪声污染，它是使用如下语句生成的：

```
>> R = imnoise2('salt & pepper', M, N, 0, 0.1);
>> gs = f;
>> gs(R == 1) = 255;
```

过滤胡椒噪声的一种较好办法是，使用 Q 为正值的逆调和滤波器。图 4.5(c)是使用如下语句生成的：

```
>> fp = spfilt(gp, 'chmean', 3, 3, 1.5);
```

同样，盐粒噪声可以使用 Q 为负值的逆调和滤波器过滤：

```
>> fs = spfilt(gs, 'chmean', 3, 3, -1.5);
```

图 4.5(d)显示了结果。使用最大和最小滤波器可以得到类似的结果。例如，图 4.5(e)和图 4.5(f)所示的图像是通过使用如下命令分别由图 4.5(a)和图 4.5(b)生成的：

```
>> fpmax = spfilt(gp, 'max', 3, 3);
>> fsmin = spfilt(gs, 'min', 3, 3);
```

使用 spfilt 的其他解决方法可以类似的方式实现。

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

自适应空间滤波器

- **自适应滤波器：考虑图像中不同点的特征变化**
  - 基于由m×n矩形窗口S_{xy}定义的区域内图像的**统计特性**
  - 与前述滤波器相比，性能更优，但也增加了**算法复杂度**
  - **包括：**
    - **自适应局部降噪滤波器**

      （什么时候应该取均值？什么时候不应该？）
    - **自适应中值滤波器**

      （什么时候应该取中值？什么时候不应该？）

自适应局部降噪滤波器

- **滤波器响应基于以下4个统计特性：**
  - 全局的噪声统计特征：**方差σ^2**
  - 邻域S_{xy}内的统计特征：**局部均值$\bar{z}_{S_{xy}}$**

$$\bar{z}_{S_{xy}} = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

  - 邻域S_{xy}内的统计特征：**局部方差$\sigma^2_{S_{xy}}$**

$$\sigma^2_{S_{xy}} = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} (g(s,t) - m_L)^2$$

自适应局部降噪滤波器

- 目的：保留边缘
- 运行规则：

1. 如果$\sigma^2=0$，->没有噪声

   滤波器不需要操作，直接返回$g(x,y)$。 $g(x,y)$下零噪声的情况等同于$f(x,y)$。

2. 若$\sigma_{S_{xy}}^2 \gg \sigma^2$，->边缘区域（方差大通常是边缘，应保留）

   滤波器返回接近$g(x,y)$的值

3. 若$\sigma_{S_{xy}}^2$接近σ^2，->图像物体内部

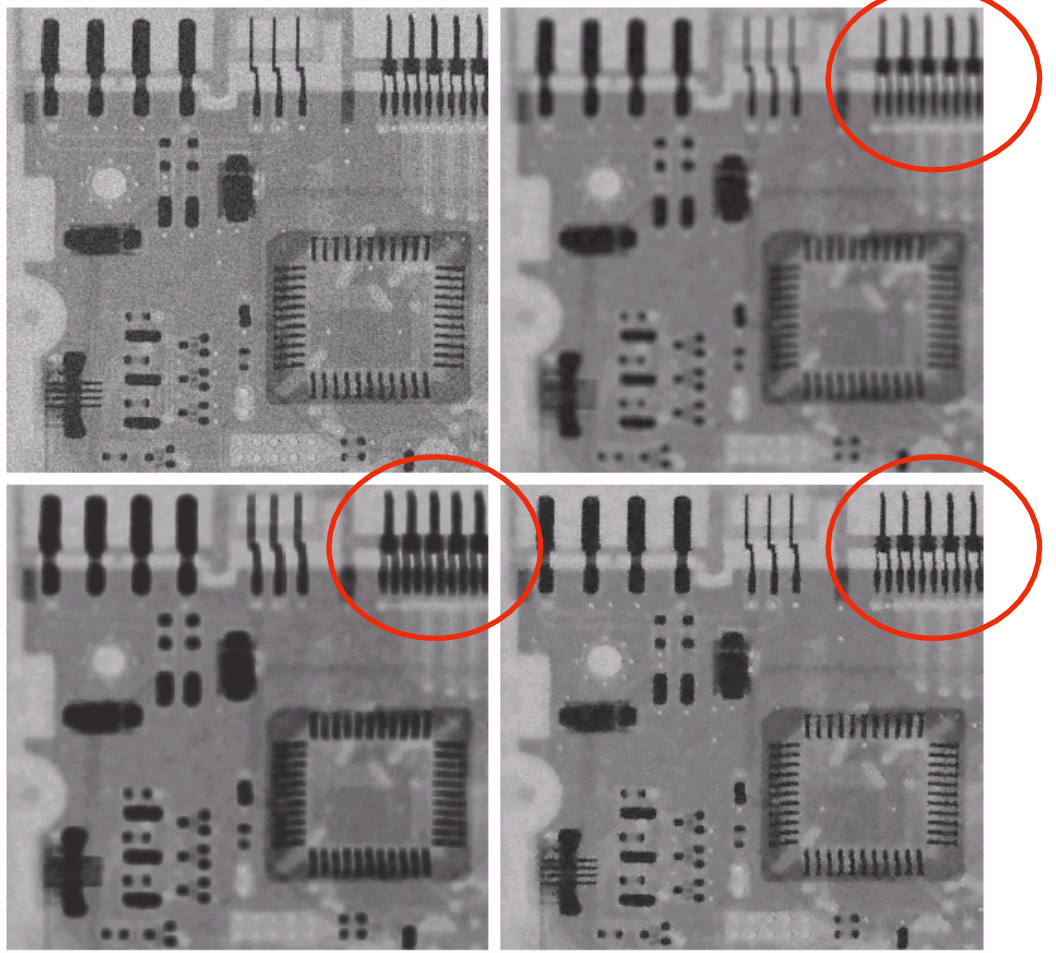
   进行算数平均滤波，返回$\bar{z}_{S_{xy}}$

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma^2}{\sigma_{S_{xy}}^2}[g(x,y) - \bar{z}_{S_{xy}}]$$

自适应局部降噪滤波器

均值为0，方差为1000的高斯噪声

7×7的算术均值滤波器



a b
c d

FIGURE 5.13
(a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7 × 7.

更加尖锐

7×7的几何均值滤波器

7×7的自适应滤波器