计算机视觉

空间域图像处理（3）

2019-3-18

空间域图像处理

- 基本概念
- 基本运算
  - ✓ 点运算
  - ✓ 直方图运算
  - ✓ 代数运算
- 空间滤波器
  - ✓ 平滑空间滤波器
  - ✓ 锐化空间滤波器

加法运算：图像均值去噪

- 图像加法运算：两幅图像f(x,y)和h(x,y)之间的和。

$$g(x,y) = f(x,y) + h(x,y)$$

- 主要应用：图像均值去噪
- 由于噪声的干扰，对于一幅原始图像f(x,y)，都会存在着一个噪声图像集合：

$$\{g_i(x,y)\}, \qquad i = 1,2,\ldots,N$$

其中

$$g_i(x,y) = f(x,y) + h_i(x,y)$$

加法运算：图像均值去噪

- 对所有噪声图像采用均值操作：

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

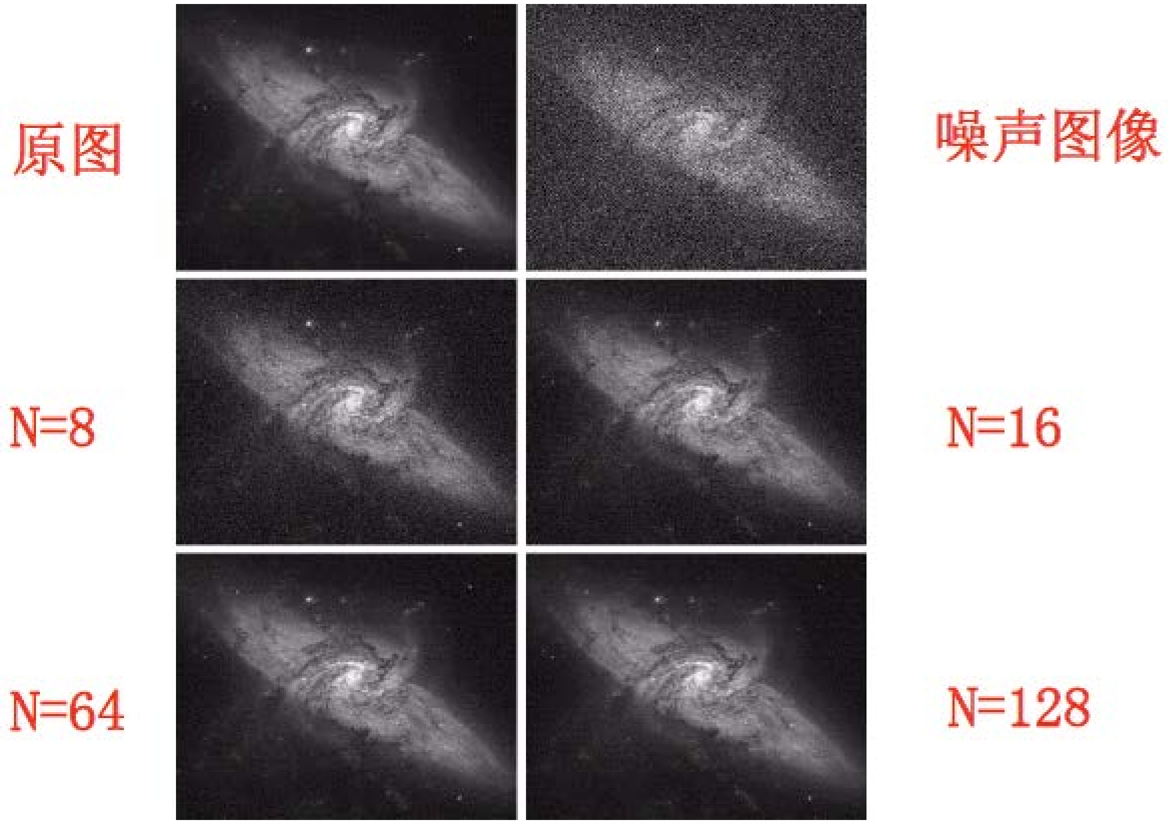
- 假设噪声h(x,y)均值为0，且互不相关，可以证明均值图像的期望值即是原始图像：

$$E[\overline{g}(x,y)]=f(x,y)$$

- 随着N的不断增大，均值图像会不断接近于原图像，即起到了去噪的效果。

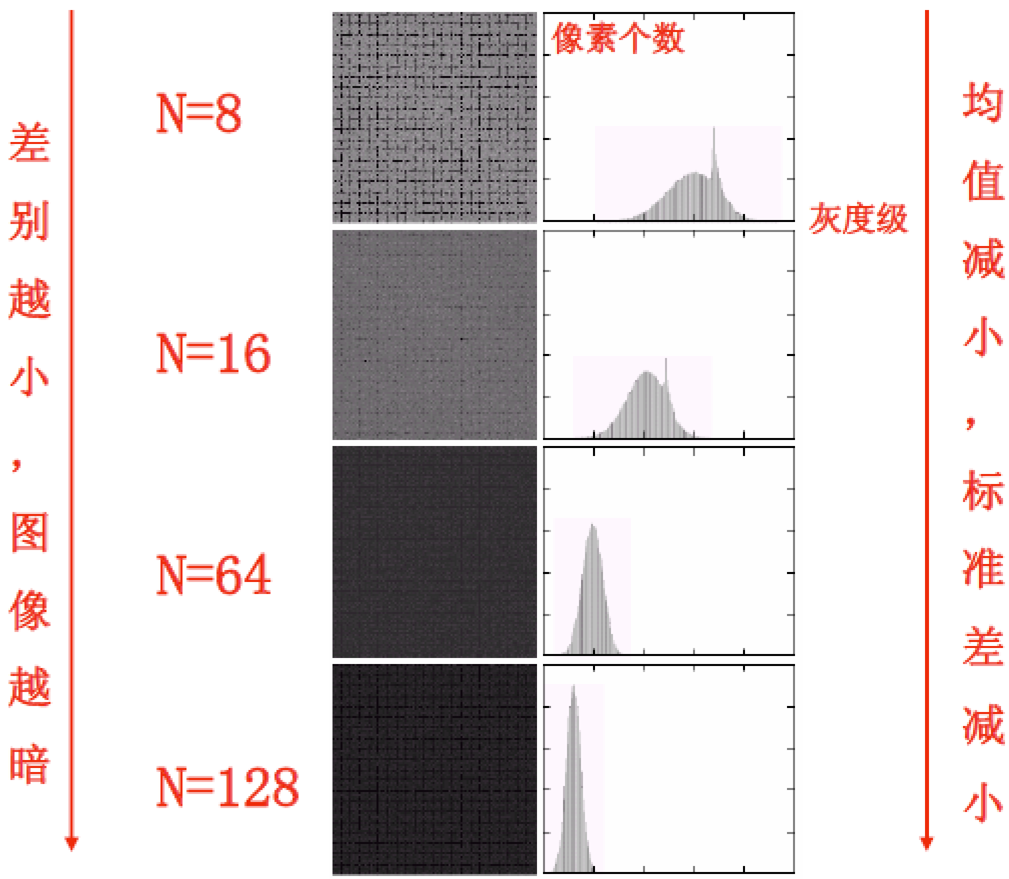
图像均值去噪：示例

- 星系图去噪



图像均值去噪：示例

- 星系图去噪

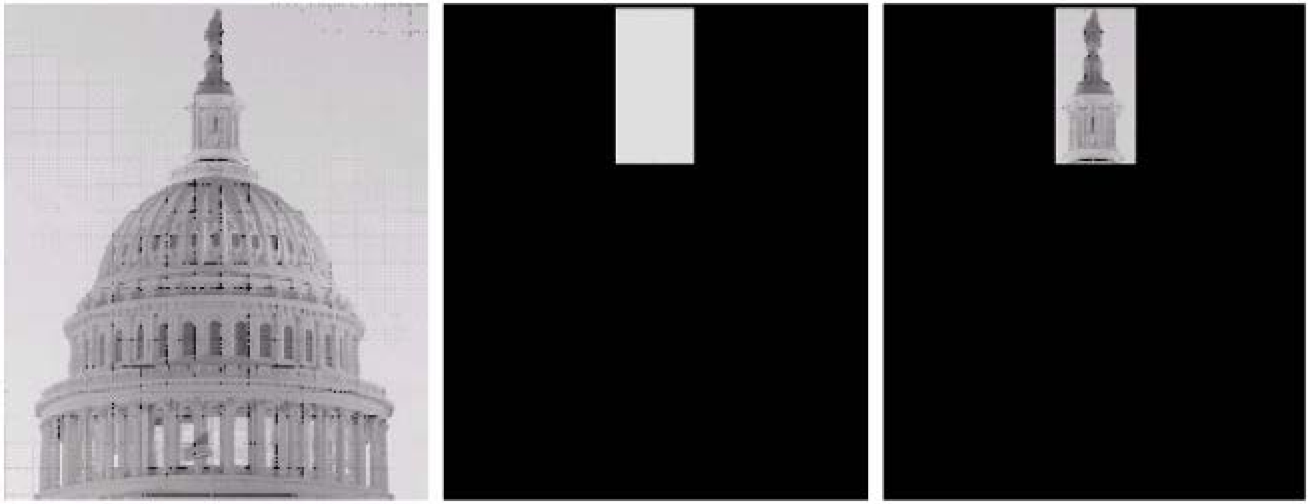


与运算

- 与运算：两幅图像f(x,y)和h(x,y)之间的逻辑与

$$g(x,y) = f(x,y) \wedge h(x,y)$$

- 主要应用：应用于模板运算，来提取感兴趣的子图像

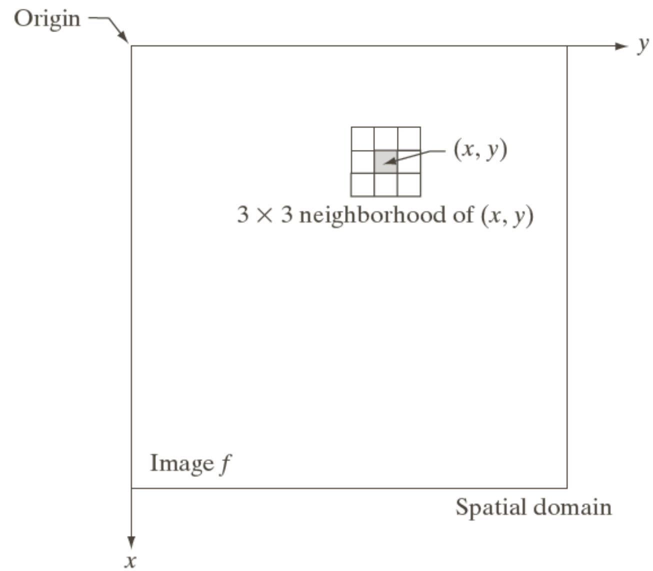


空间域图像处理

- 基本概念
- 基本运算
  - ✓ 点运算
  - ✓ 直方图运算
  - ✓ 代数运算
- 空间滤波器
  - ✓ 平滑空间滤波器
  - ✓ 锐化空间滤波器

空间滤波器的概念

- 图像处理操作：$g(x,y)=T\left[f(x,y)\right]$
- 图像处理算子T是定义在(x,y)的邻域上
- 前面的点运算和代数运算中，(x,y)的邻域都是1×1。
- 如果邻域是一个m×n的掩膜(mask)，且m>1，n>1，该掩膜通常被称为一个滤波器(filter)。



空间滤波的机制

- 空间滤波的数学表达：

滤波器坐标原点在核的中心点。

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

其中，m=2a+1，n=2b+1

w(s,t)是滤波器系数，即滤波器这个子图像中每个像素的值。

空间滤波的机制

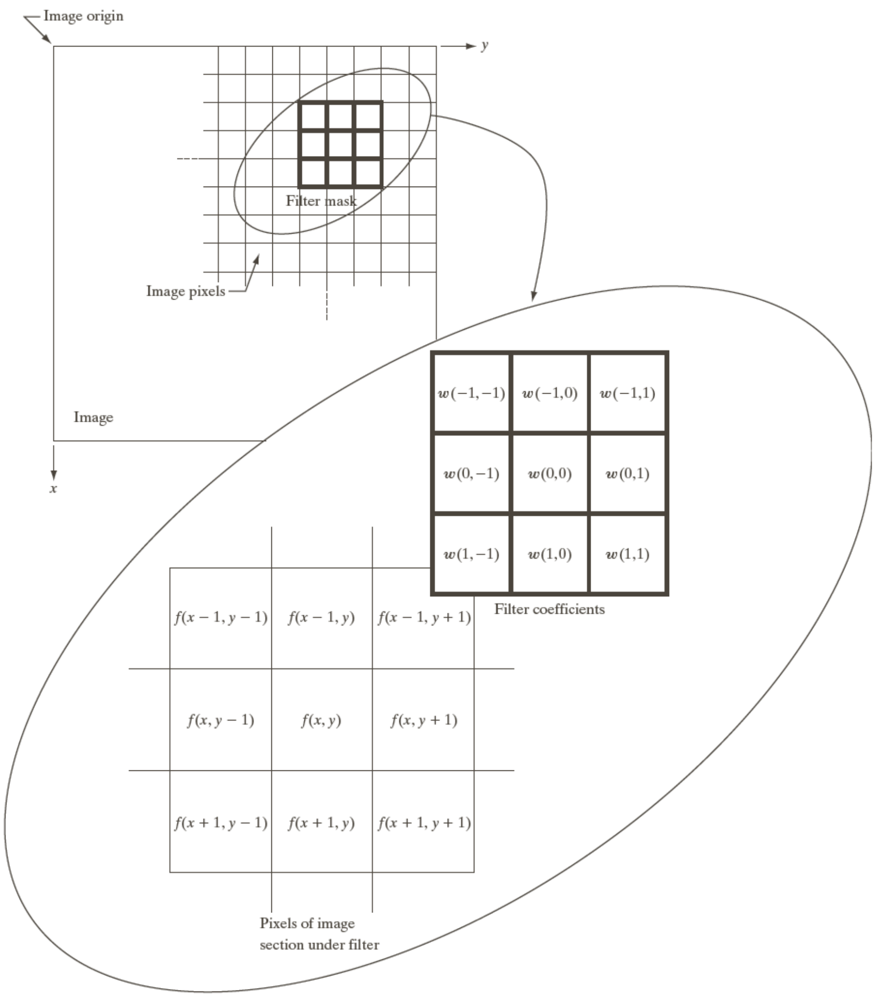


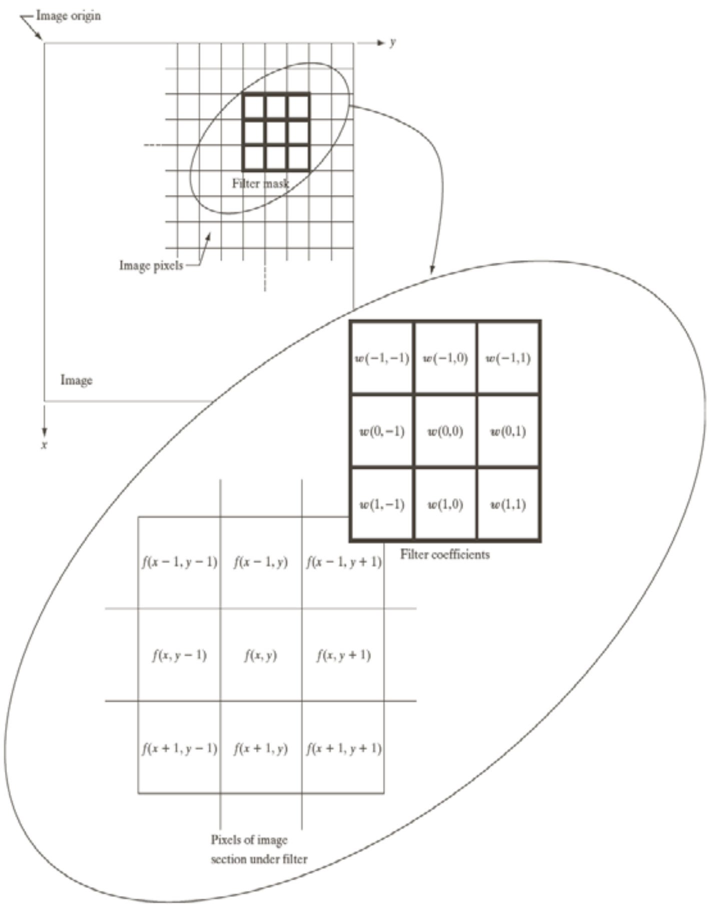
Image origin
y
Filter mask
Image pixels
Image
x
w(−1,−1)
w(−1,0)
w(−1,1)
w(0,−1)
w(0,0)
w(0,1)
w(1,−1)
w(1,0)
w(1,1)
Filter coefficients
f(x − 1, y − 1)
f(x − 1, y)
f(x − 1, y + 1)
f(x, y − 1)
f(x, y)
f(x, y + 1)
f(x + 1, y − 1)
f(x + 1, y)
f(x + 1, y + 1)
Pixels of image
section under filter

空间滤波计算步骤

- 空间滤波器：

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

其中，m=2a+1，n=2b+1，
w(s,t)是滤波器,其值是滤波器系数。



- 空间滤波的计算步骤：
- 1. 以图像中的一个像素为中心，将该像素m×n邻域内的灰度值与对应的滤波器系数相乘，得到的结果作为该中心像素滤波后的值。
- 2. 将滤波器从一个像素移动到它的下一个相邻像素，重复步骤1.

空间域图像处理

- 基本概念
- 基本运算
  - ✓ 点运算
  - ✓ 直方图运算
  - ✓ 代数运算
- 空间滤波器
  - ✓ 平滑空间滤波器
  - ✓ 锐化空间滤波器

平滑空间滤波器

- 平滑空间滤波器(Smoothing Spatial Filters)
  - 作用：
    - 模糊处理：去除图像中一些不重要的细节
    - 减少噪声
  - 分类：
    - 均值（线性）滤波器
    - 统计排序（非线性）滤波器
    - 自适应滤波器

线性（均值）滤波器

- 线性滤波器(Linear Filters)
  - 原理：计算包含在滤波器内所有相邻像素的平均值。因此，也称之为均值滤波器。
  - 作用：减少图像灰度的“尖锐”变化，减少噪声。即滤掉了高频信号，因此也被称之为低通滤波器（Low-pass Filters）。

算术均值滤波器

a

$$\frac{1}{9}\times\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

b

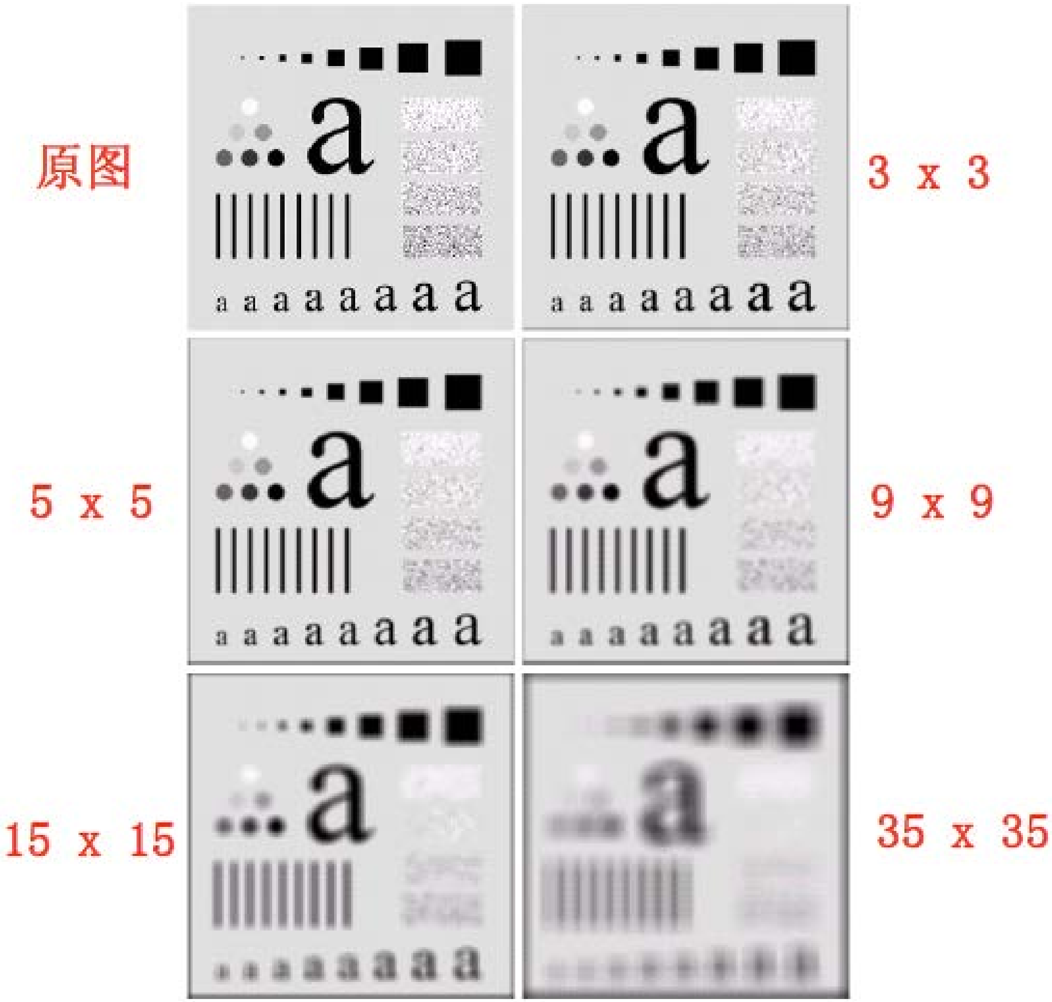
$$\frac{1}{16}\times\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

- 图a：标准的像素算术平均值
- 图b：像素的加权平均，表明一些像素更为重要

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

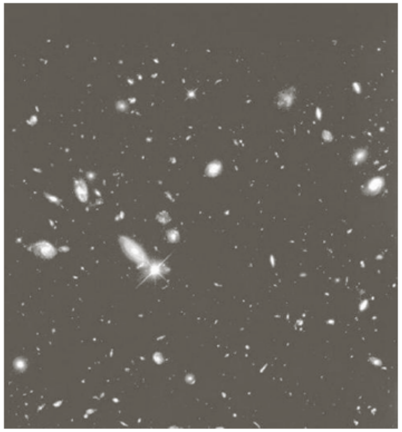
算术均值滤波器：示例1

- 滤波器尺寸大小对于图像平滑的影响。

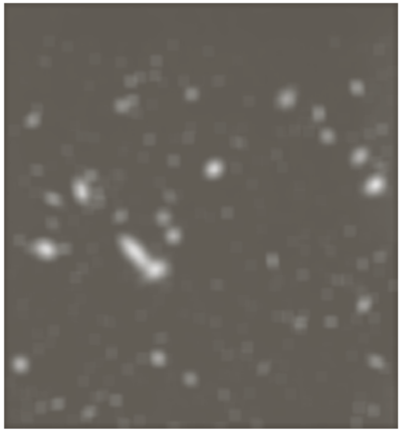


算术均值滤波器：示例2

- 提取感兴趣的物体而模糊图像（哈勃空间望远镜图像）
- 1. 根据感兴趣物体的尺寸大小平滑图像，使得不感兴趣的部分融入到背景中。
- 2. 在平滑后的图像上，应用相应的阈值，达到提取感兴趣目标的目的。



原图



15×15

应用阈值后

几何均值滤波器

- 几何均值滤波器(Geometric Mean Filter)

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

- 其中， S_{xy}代表以(x,y)为中心的m×n的矩形区域
- 特点：
- 1.几何均值滤波器的平滑效果与算术均值滤波器效果相当。适合于处理高斯等随机噪声。
- 2.几何均值滤波器比算术均值滤波器丢失的图像细节信息要少，即几何均值滤波器得到的图像更加锐化一些。

谐波均值滤波器

- 谐波均值滤波器(Harmonic Mean Filter)

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

- 其中， S_{xy}代表以(x,y)为中心的m×n的矩形区域
- 特点：
- 1. 对于“盐”(salt)噪声效果好，但不适合去除“胡椒”(pepper)噪声。
- 2. 善于处理高斯噪声。

逆谐波均值滤波器

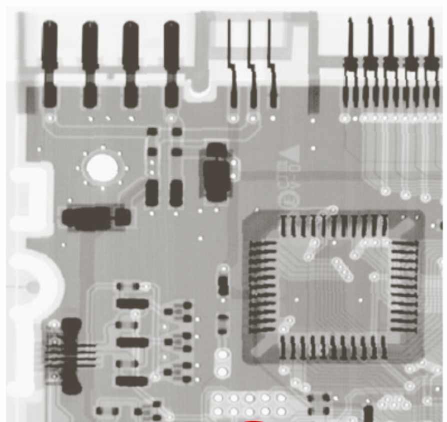
- 逆谐波均值滤波器(Contraharmonic Mean Filter)

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

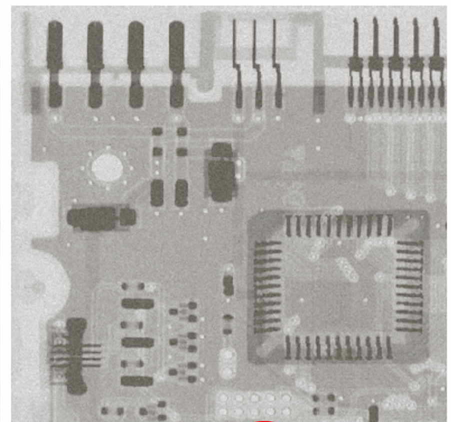
- 其中，Q是滤波器阶数。
- 特点：
- 1. 当Q为负数时，可以消除“盐”(salt)噪声
- 2. 当Q为正数时，可以消除“胡椒”(pepper)噪声
- 因此，需要事先知道噪声是暗噪声还是亮噪声
- 3. 当Q＝－1时，成为谐波均值滤波器
- 4. 当Q＝0时，成为算术均值滤波器

均值滤波器：示例—高斯噪声

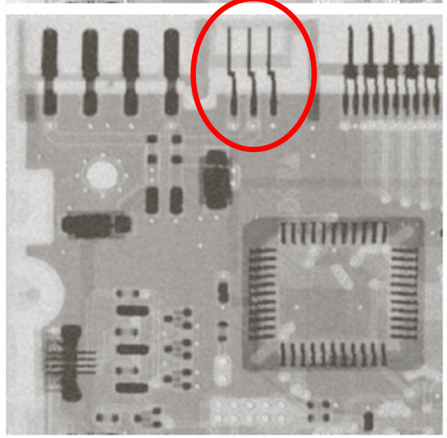
原图



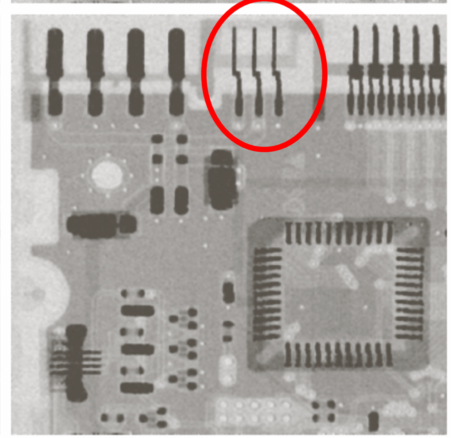
加入高斯噪声的图像：噪声均值为0，方差为400



3×3算术均值滤波器的结果



3×3几何均值滤波器的结果

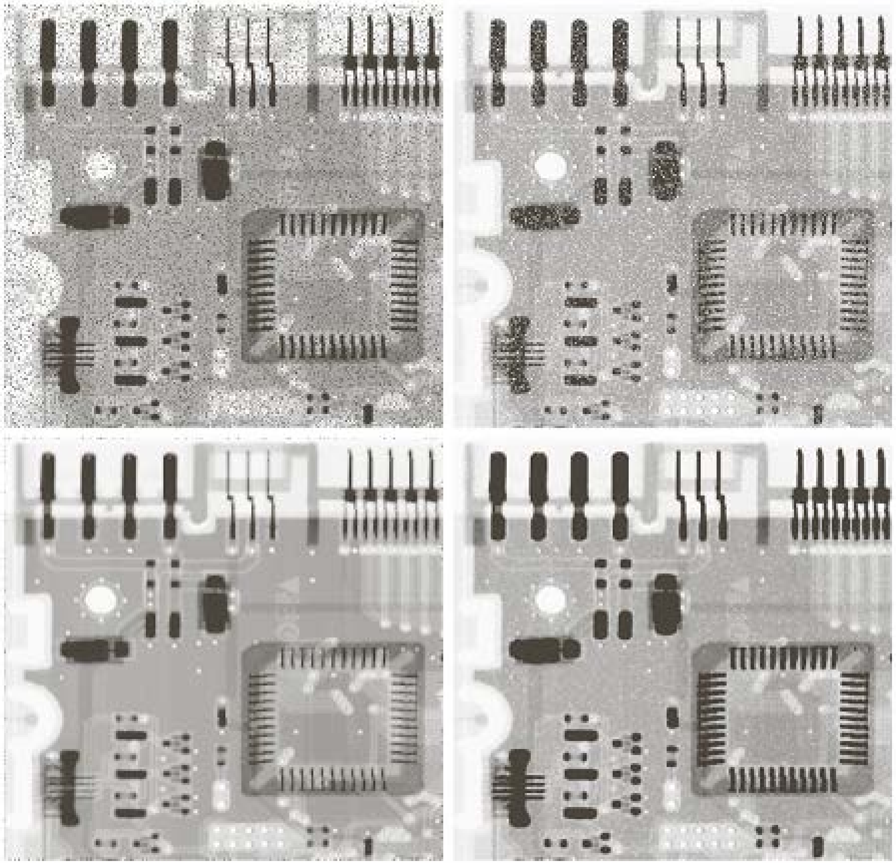


图像更加清晰

均值滤波器：盐噪声和胡椒噪声

加入胡椒噪声的图像：概率为0.1

加入盐噪声的图像：概率为0.1



3×3逆谐波均值滤波器的结果，Q＝1.5

3×3逆谐波均值滤波器的结果，Q＝－1.5

非线性滤波器：统计排序滤波器

- 统计排序滤波器(Order-Statistics Filters)
- 它是一种非线性滤波器
- 它是基于滤波器所在图像区域内像素的排序，用排序结果决定的值来代替中心像素的值。

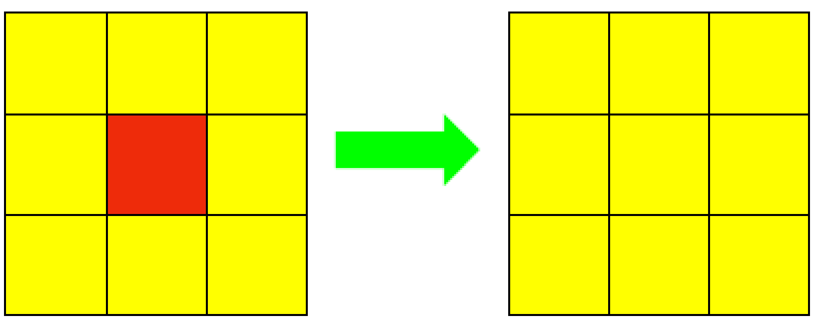
- 分类
- 1. 中值滤波器
- 2. 最大值滤波器
- 3. 最小值滤波器
- 4. 中点滤波器
- 5. 修正的阿尔法均值滤波器

中值滤波器

- 中值滤波器(Median Filters)
- 原理：用滤波器区域内所有像素的中间值，作为结果值来代替中心像素值。

$$R = med\{z_k \mid k = 1,2,\ldots,n\}$$

- 强迫滤波器区域内突出的亮点（暗点）更像它周围的值，以消除孤立的亮点（暗点）。
- 作用：去除噪声。

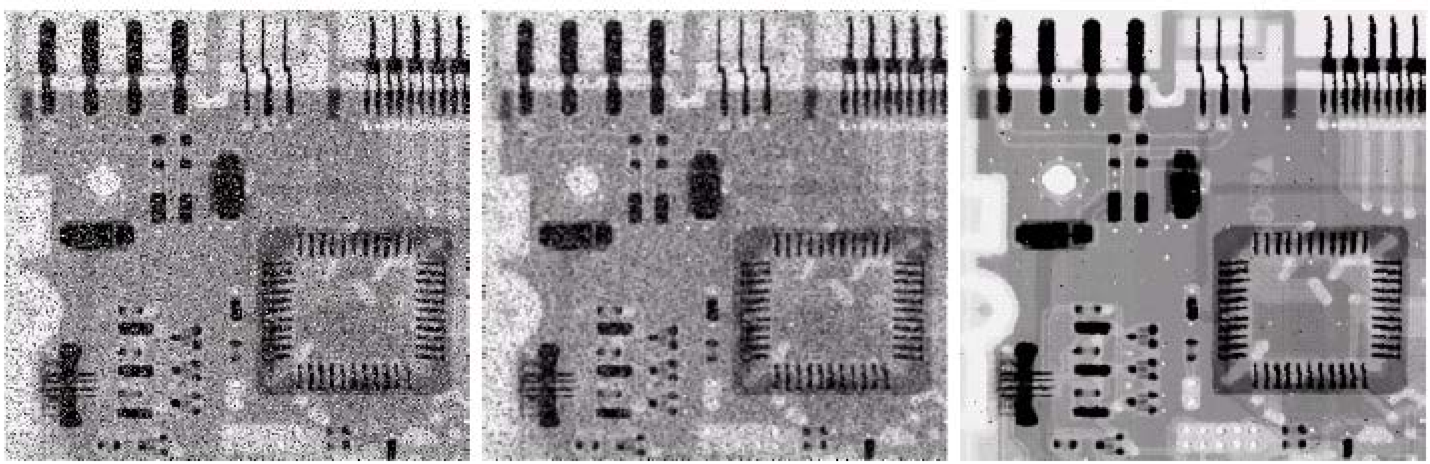


中值滤波器

- 中值滤波器的特点：
- 1. 对于去除椒盐噪声(salt-and-pepper noise)特别有效。
  椒盐噪声：以黑白点叠加在图像上。
- 2. 在去除噪声的同时，可以比较好的保留边缘的锐度和图像的细节。

中值滤波器：示例

原图　　3x3均值滤波　　3x3中值滤波



最大值和最小值滤波器

- 最大值和最小值滤波器 (Max and Min Filter)

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

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

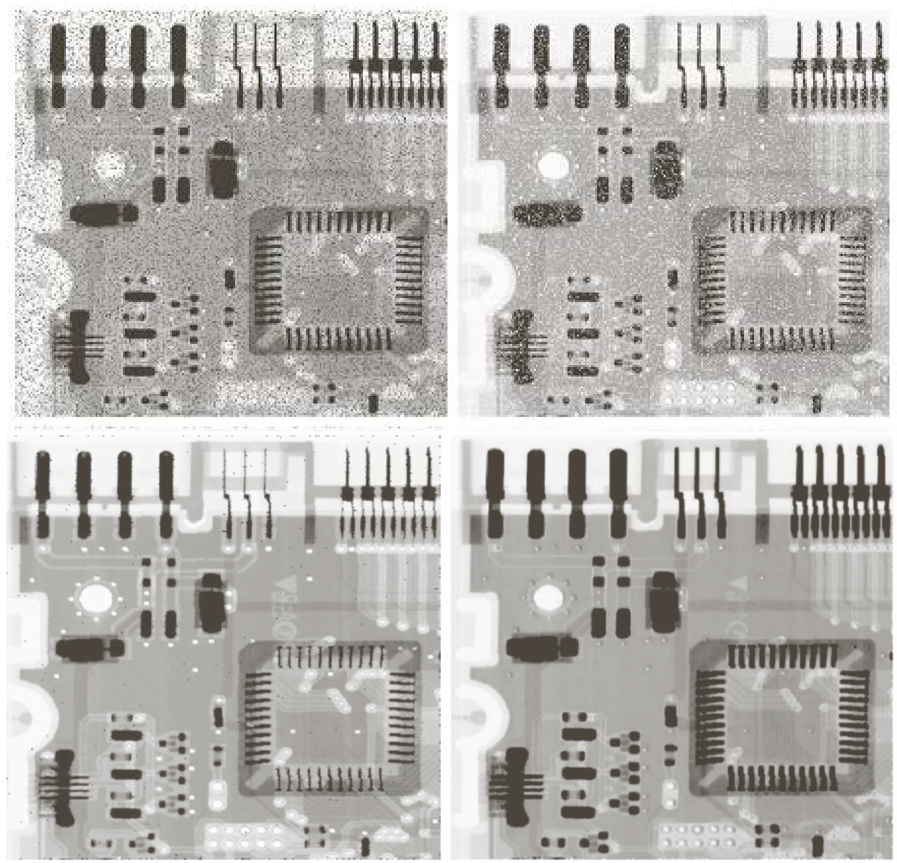
- 其中， S_{xy}代表以(x,y)为中心的m×n矩形区域。

- 特点：
- 1. 最大值滤波器用于发现图像中最亮的点，因此可以用来过滤“胡椒”噪声。
- 2. 最小值滤波器用于发现图像中最暗的点，因此可以用来过滤“盐”噪声。

最大最小值滤波器示例：胡椒/盐噪声

加入胡椒噪声的图像：概率为0.1

加入盐噪声的图像：概率为0.1



3×3最大值滤波器的结果

3×3最小值滤波器的结果

中点滤波器

- 中点滤波器 (Midpoint Filter)

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

- 其中， S_{xy}代表以(x,y)为中心的m×n矩形区域。

- 特点：
- 1. 中点滤波器综合了统计滤波器和均值滤波器的特点。
- 2. 它对高斯、均匀随机分布噪声有很好的滤波效果。

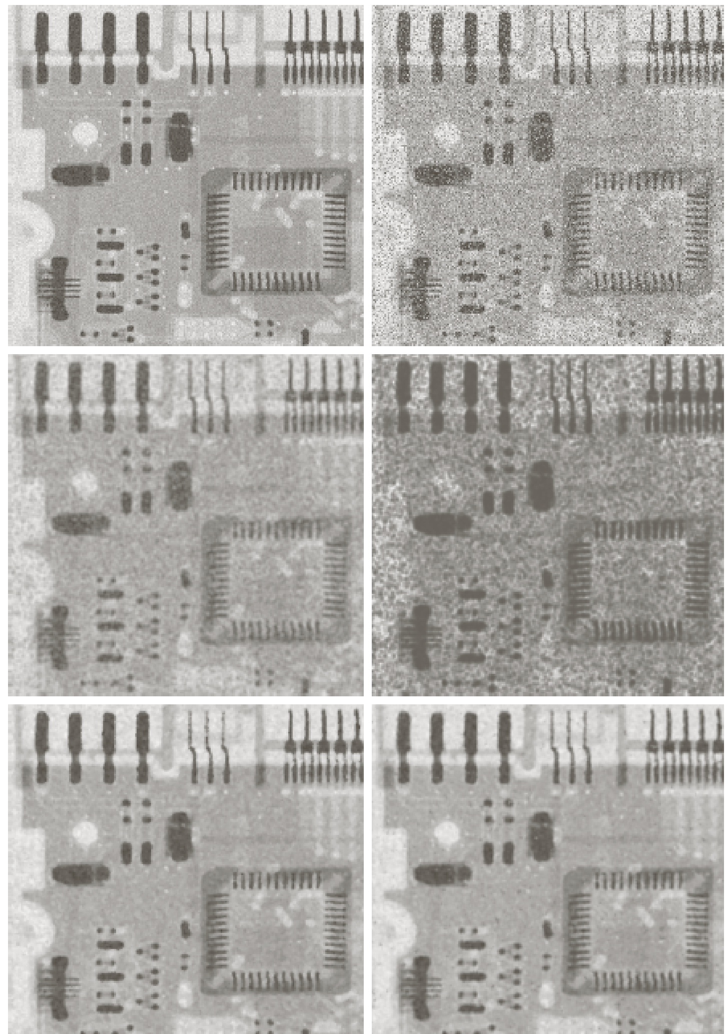
修正的阿尔法均值滤波器

- 修正的阿尔法均值滤波器 (Alpha-trimmed Mean Filter)

$$\hat{f}(x,y)=\frac{1}{mn-d}\sum_{(s,t)\in S_{xy}} g_r(s,t)$$

- 其中， g_r代表以(x,y)为中心的m×n矩形区域中去掉d/2个最大值像素和d/2个最小值像素后剩下的像素。
- 特点：
- 1. 当0<d<(mn-1)/2时，适合于有多种类型噪声混合在一起情况下的除噪，例如高斯和椒盐噪声混合在一起。
- 2. 当d=0时，该滤波器成为算术均值滤波器。
- 3. 当d=(mn-1)/2时，该滤波器成为中值滤波器。

统计滤波器示例：均匀噪声+椒盐噪声



加入均匀分布噪声的图像：均值：0，方差：800

又加入椒盐噪声的图像，Pa= Pb =0.1

5×5算术均值滤波器的结果

5×5几何均值滤波器的结果

5×5中值滤波器的结果

5×5修正的阿尔法均匀滤波器的结果

自适应的局部噪声消除滤波器

- 自适应的局部噪声消除滤波器 (Adaptive, local noise reduction filter)

$$\hat{f}(x,y)=g(x,y)-\frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x,y)-m_L\right]$$

- σ_η^2 : 噪声方差；M_L ，σ_L^2 ： 区域S_{xy}内像素灰度值的均值和方差。
- 设计思想：
- 1. 当 $\sigma_\eta^2=0$ 时，意味着这是个零噪声情形，即g(x,y)=f(x,y)，所以滤波器输出为g(x,y)。
- 2. 当 $\sigma_\eta^2\le\sigma_L^2$时，意味着该区域的灰度值方差大于噪声方差，即该区域是图像的边缘区域。所以该区域应该予以保留，即滤波器输出为g(x,y)。
- 3. 当 $\sigma_\eta^2=\sigma_L^2$ 时，意味着该区域灰度值分布和噪声分布混合程度较深，应该予以滤除，这里使用均值滤波，即滤波器输出为该区域均值。

自适应的局部噪声消除滤波器

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x,y) - m_L\right]$$

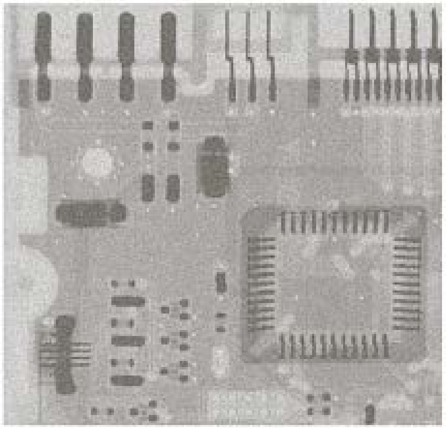
- 问题1：当 $\sigma_\eta^2 > \sigma_L^2$ 时，怎么办？
- 在此情况下，该滤波器仍然使用均值滤波器来消除噪声，即强行设置

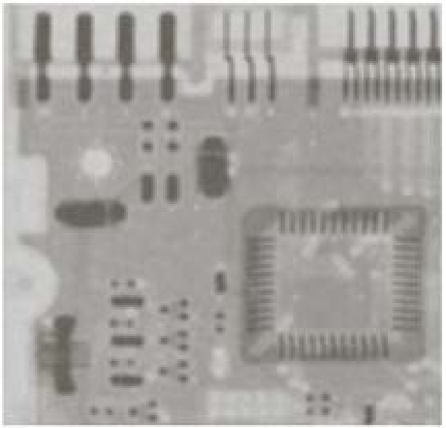
  $\sigma_\eta^2 = \sigma_L^2$

- 问题2：如何估计 σ_η^2 ？
- 无法估计！

自适应的局部噪声消除滤波器示例

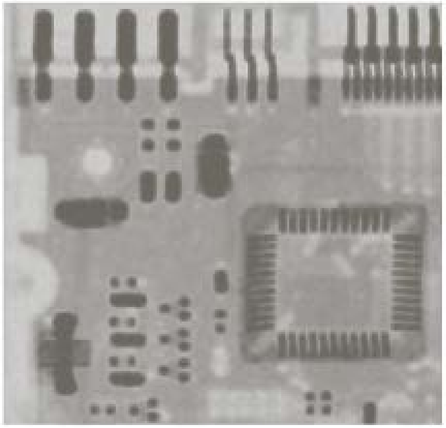
加入高斯噪声的图像：均值：0，方差：1000

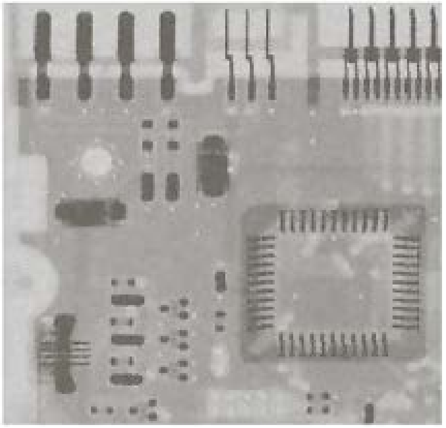




7×7算术均值滤波器的结果

7×7几何均值滤波器的结果





7×7自适应噪声消除滤波器的结果

自适应中值滤波器

- 自适应中值滤波器 (Adaptive Median Filter)
- 目的：
- 1. 消除椒盐噪声
- 2. 降低滤波产生的失真现象

- 定义如下符号：
- Z_{min}: S_{xy}区域中灰度值最小值
- Z_{max}: S_{xy}区域中灰度值最大值
- Z_{med}: S_{xy}区域中灰度值中值
- Z_{xy}: 点(x,y)的灰度值
- S_{max}: 允许的最大S_{xy}区域

自适应中值滤波器

- 自适应中值滤波器 (Adaptive Median Filter)
- Level 1： $A1 = Z_{med} - Z_{min}$
- $A2 = Z_{med} - Z_{max}$
- If A1>0 and A2<0，去 level 2
- Else 增加邻域窗口大小
- If 邻域窗口大小<S_{max}，重复Level 1
- Else 输出 Z_{xy}
- Level 2: $B1 = Z_{xy} - Z_{min}$
- $B2 = Z_{xy} - Z_{max}$
- If B1>0 and B2 <0，输出 Z_{xy}
- Else 输出 Z_{med}

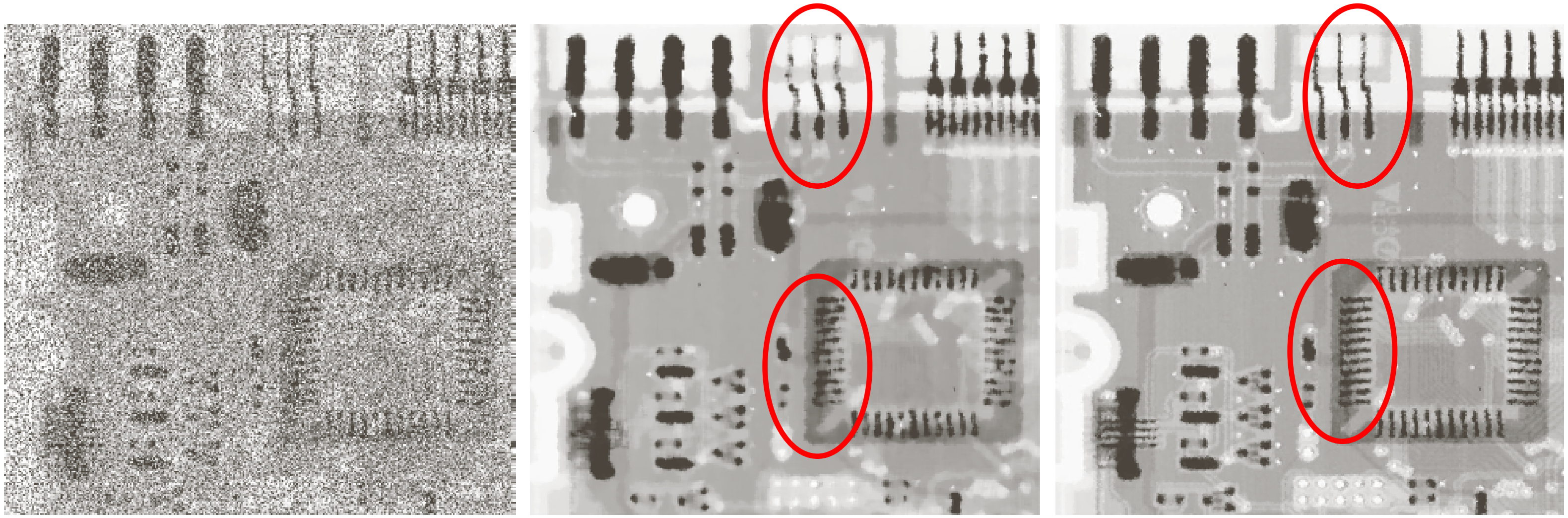
说明中值Z_{med}不是脉冲，下一步检测Z_{xy}是不是脉冲

说明中值Z_{med}是脉冲，扩大窗口来找到非脉冲中值

说明Z_{xy}也不是脉冲，所以输出图像原值，从而不会产生图像失真。

说明Z_{xy}是脉冲，所以输出中值，从而产生中值滤波的效果。

自适应中值滤波器：示例



a b c

FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{\max} = 7$.

左图：加入椒盐噪声的图像，Pa=Pb=0.25

中图：7×7中值滤波器的结果

右图：S_{max}为7的自适应中值滤波器的结果

空间域图像处理

- 基本概念
- 基本运算
  - ✓ 点运算
  - ✓ 直方图运算
  - ✓ 代数运算
- 空间滤波器
  - ✓ 平滑空间滤波器
  - ✓ 锐化空间滤波器

锐化空间滤波器

- 锐化空间滤波器(Sharpening Spatial Filters)
- 核心思想：突出图像中的细节，增强被模糊了的细节

- 应用领域：
- 1. 印刷中的细微层次强调。弥补扫描对图像的钝化。
- 2. 超声探测成像，分辨率低，边缘模糊，通过锐化来改善
- 3. 图像识别中，做图像分割前，进行边缘提取，从而促进目标识别和定位。
- 4. 恢复过度钝化、曝光不足的图像

锐化滤波器的设计原理

- 均值滤波产生钝化的效果，而均值与积分具有相似的性质。
- 那么，微分能不能产生相反的效果，即锐化的效果呢？
- 结论是肯定的。因此，可以用微分运算来设计锐化滤波器。
- 在图像处理中，数字函数的微分运算通常是用差值来定义。例如，数字函数f(x)关于x轴的一阶导数可以表达为：

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

锐化滤波器的分类

- 锐化滤波器的分类：
- 1. 二阶微分滤波器：拉普拉斯算子
- 2. 一阶微分滤波器：梯度算子

拉普拉斯算子

- 一个图像函数f(x,y)的拉普拉斯变换可以定义为：

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

- 沿x，y轴的二阶偏导可以定义为：

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

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

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

- 相加可得：

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

拉普拉斯算子：实施

标准实施

中心系数为负

偏导沿着x,y轴正向

0	f(x, y+1) 1	0
f(x−1, y) 1	f(x, y) −4	f(x+1, y) 1
0	1 f(x, y−1)	0

f(x−1, y+1)　　f(x+1, y+1)

1	1	1
1	−8	1
1	1	1

f(x−1, y−1)　　f(x+1, y−1)

扩展实施

（考虑了对角邻域）

中心系数为正

偏导沿着x,y轴反向

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1