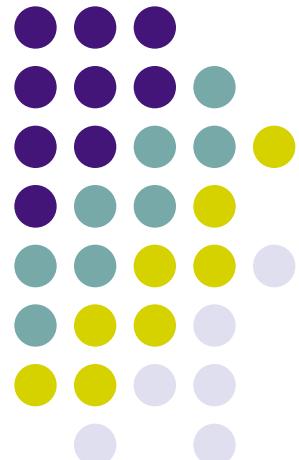


# 数字图像处理

第九讲  
图像复原





# 主要内容

- 图像复原与图像增强
- 重要的噪声函数
- 空间滤波器用于图像复原
- 频域滤波器用于图像复原
- 图像降质模型
- 估计退化函数
- 逆滤波与维纳滤波
- 总结

# 图像复原与图像增强



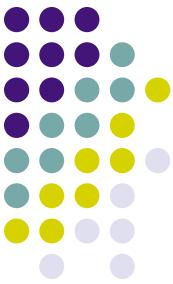
- 图像增强：突出图像中感兴趣的特征，而衰减不需要的特征。**改善后的图像不一定要逼近原图像**。图像增强是个主观的过程。
- 图像复原：根据图像降质原因，设法去补偿降质因素，**从而使改善后的图像尽可能逼近原始图像**。图像复原是个客观的过程。



# 重要的噪声函数

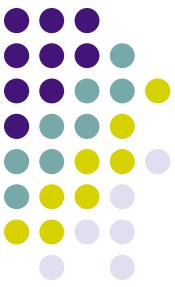
- 假定噪声以某种概率分布函数给出
- 常见的噪声函数
  - 高斯噪声
  - 瑞利噪声
  - 伽马噪声
  - 指数分布噪声
  - 均匀分布噪声
  - 脉冲噪声

# 高斯噪声



- 高斯噪声在空间域和频域上易于处理，常用于实践中

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$$



# 瑞利噪声

## ● 概率分布函数

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

$$\mu = a + \sqrt{\pi b}/4$$

$$\sigma^2 = \frac{b(4 - \pi)}{4}$$



# 伽马噪声

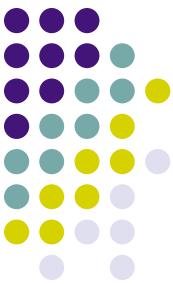
- 概率分布函数

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

$$\mu = \frac{b}{a}$$

$$\sigma^2 = \frac{b}{a^2}$$

# 指数分布噪声



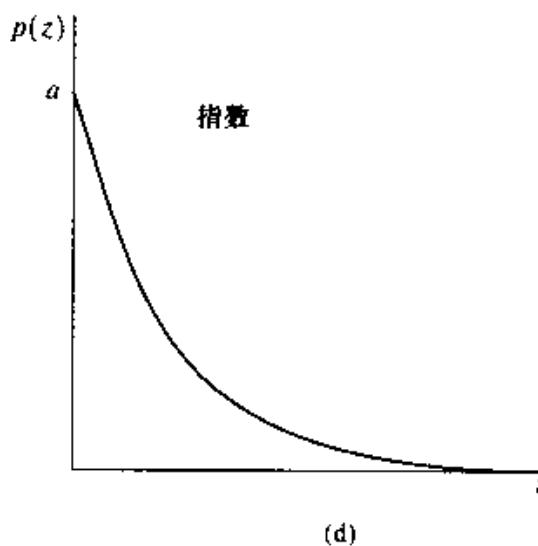
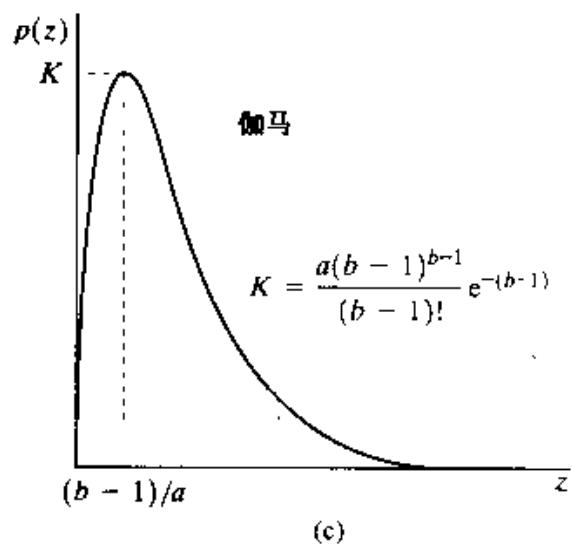
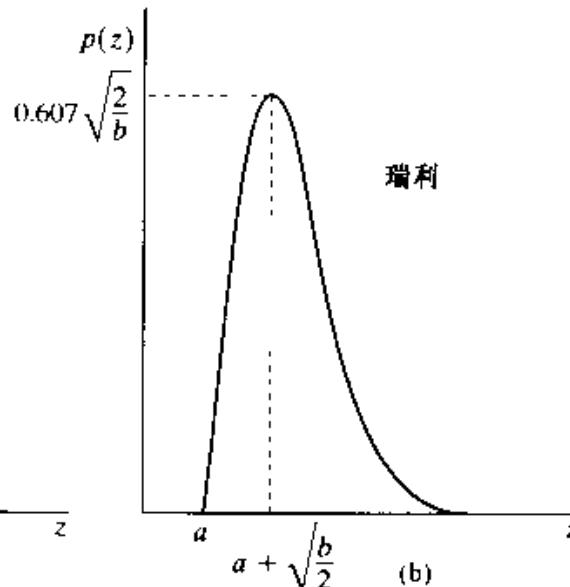
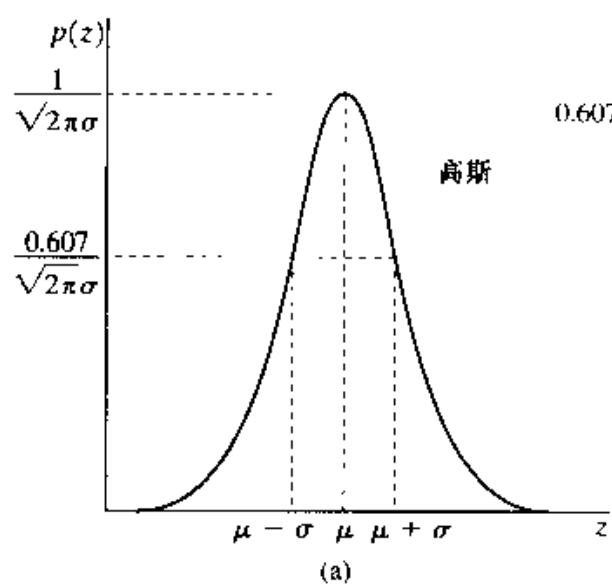
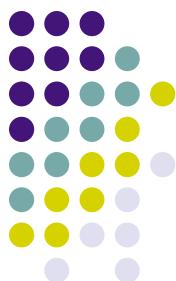
## ● 概率分布函数

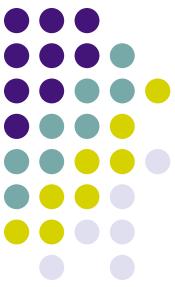
$$p(z) = \begin{cases} a e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

$$\mu = \frac{1}{a}$$

$$\sigma^2 = \frac{1}{a^2}$$

# 分布示意图





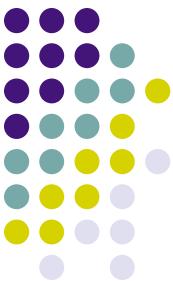
# 均匀分布噪声

- 概率分布函数

$$p(z) = \begin{cases} \frac{1}{b-a} & a \leq z \leq b \\ 0 & \text{其他} \end{cases}$$

$$\mu = \frac{a+b}{2}$$

$$\sigma^2 = \frac{(b-a)^2}{12}$$



# 脉冲噪声 (椒盐噪声)

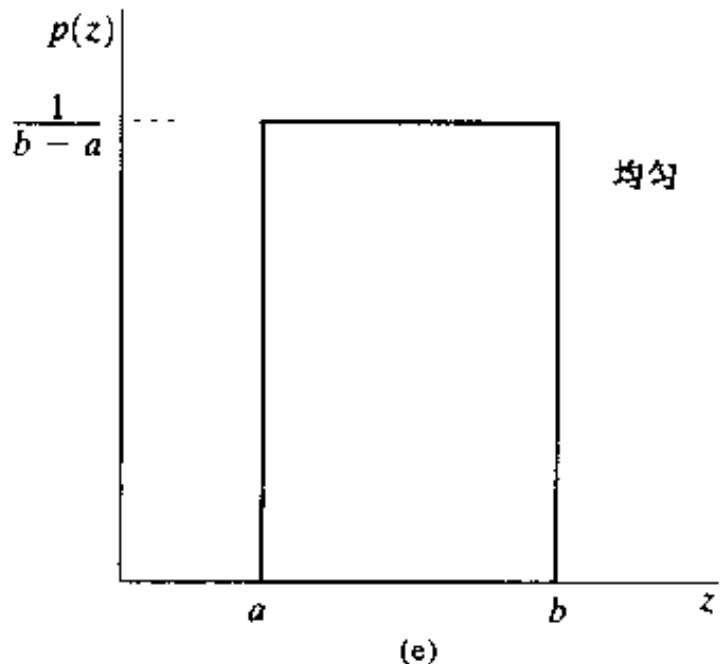
- 椒盐噪声 (Salt and Pepper noise)  
：椒（黑）盐（白）
- 概率分布函数

$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & \text{其他} \end{cases}$$

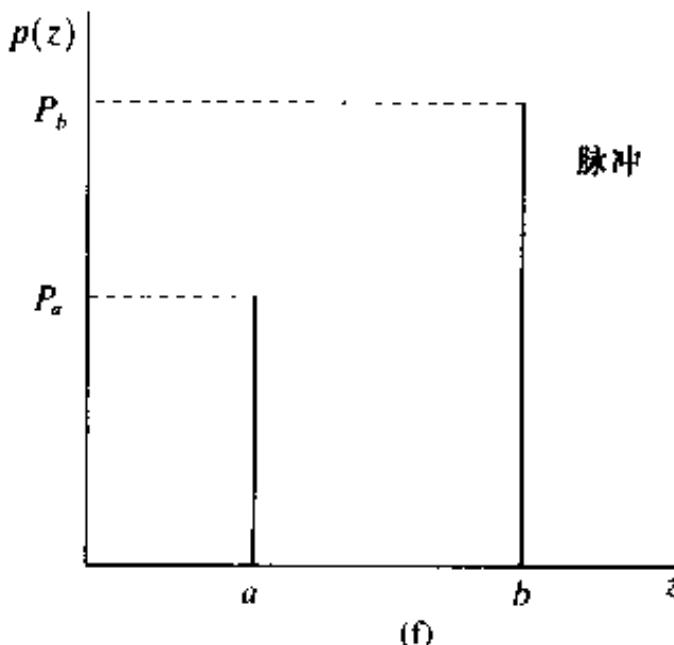
# 分布示意图



(c)



(d)



(e)

(f)

# 不同噪声比较

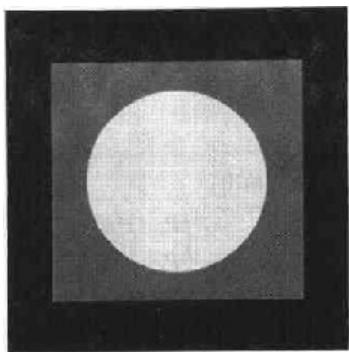


图 5.3 用于说明示于图 5.2 的噪声 PDF 特性的测试图

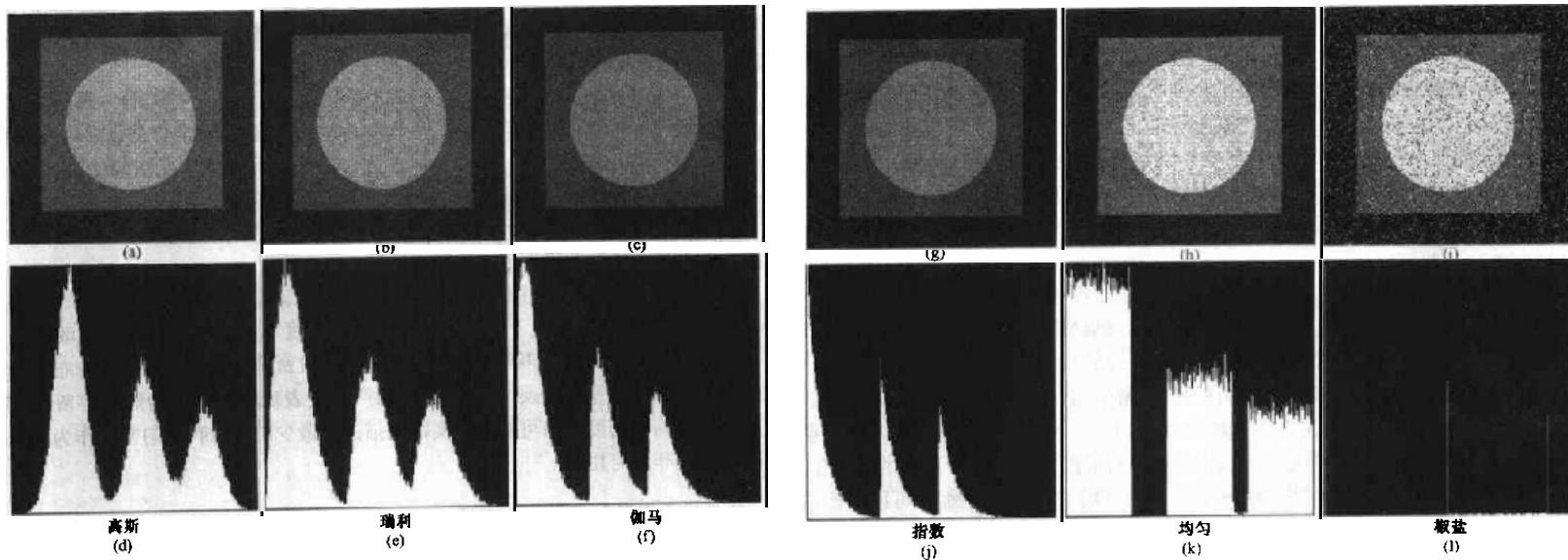
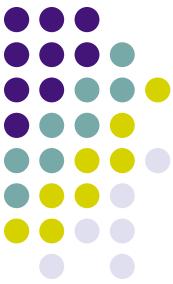


图 5.4 在图 5.3 中附加高斯、瑞利和伽马噪声后的图像和直方图

# 小结



- 熟悉高斯噪声和椒盐噪声
- 了解其它噪声函数



# 周期噪声

- 这类噪声在图像获取中由电力或机电干扰产生，是具有某种频率的信号
- 这类噪声用传统空间域滤波器方法处理不了，需要用频域滤波器来进行图像复原。





# 噪声参数的估计

- 假定知道噪声类型，如何估计噪声参数？

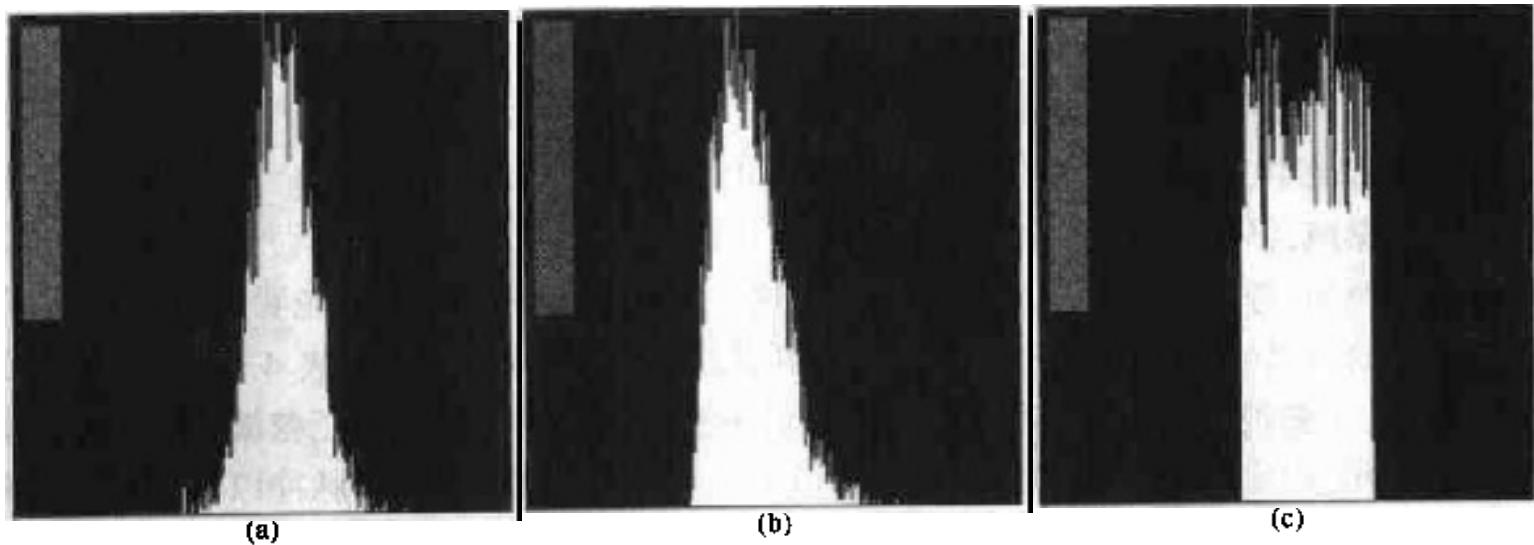


图 5.6 用小带(如图中插入的小图所示)从图 5.4 中的(a)高斯,(b)瑞利,(c)均匀噪声计算的直方图

- 常用做法：根据噪声均值和噪声方差，根据概率论最大似然技术估计参数。

# 主要内容

- 图像复原与图像增强
  - 重要的噪声函数
  - 空间滤波器用于图像复原
  - 频域滤波器用于图像复原
  - 图像降质模型
  - 估计退化函数
  - 逆滤波与维纳滤波
  - 总结
- } 两大类



# 空间域滤波器用于图像复原



- 常用滤波器
  - 均值滤波器
  - 统计排序滤波器
  - 自适应滤波器

# 均值滤波器



- 算术均值滤波器
- 几何均值滤波器
- 谐波均值滤波器
- 逆谐波均值滤波器



# 算术均值滤波器

- 最简单的均值滤波器

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

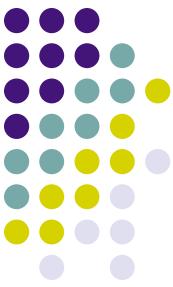
- 平滑图像的局部变化

# 几何均值滤波器



- 几何平均

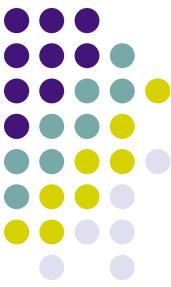
$$\hat{f}(x, y) = \left[ \prod_{(s, t) \in S_{xy}} g(s, t) \right]^{\frac{1}{|S_{xy}|}}$$



# 谐波均值滤波器

- 谐波均值（调和平均）

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



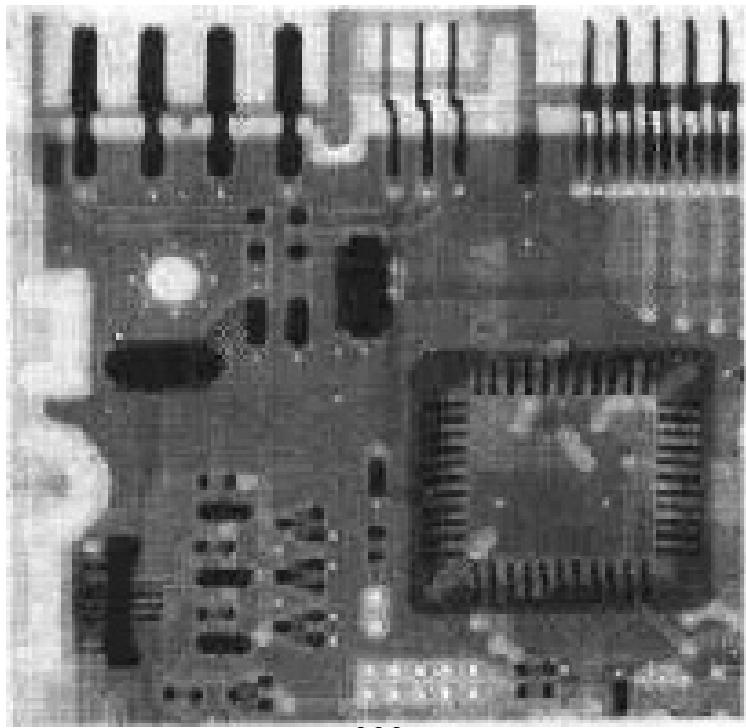
# 逆谐波均值滤波器

- 逆谐波均值

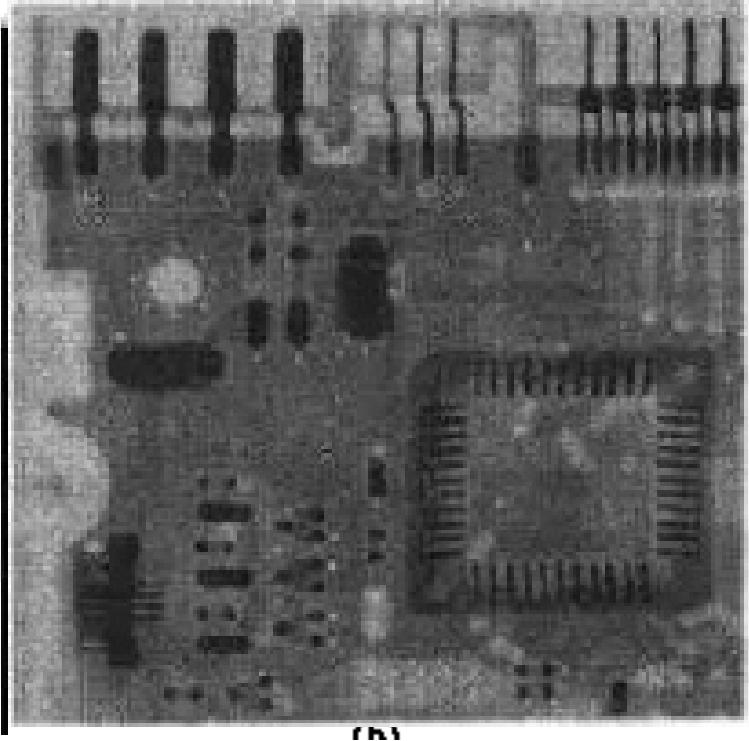
$$\hat{f}(x, y) = \frac{\sum_{(s, t) \in S_{\eta}} g(s, t)^{Q+1}}{\sum_{(s, t) \in S_{\eta}} g(s, t)^Q}$$

- $Q=0$ , 退化为算术均值滤波器
- $Q=-1$ , 退化为谐波均值滤波器

# 效果图

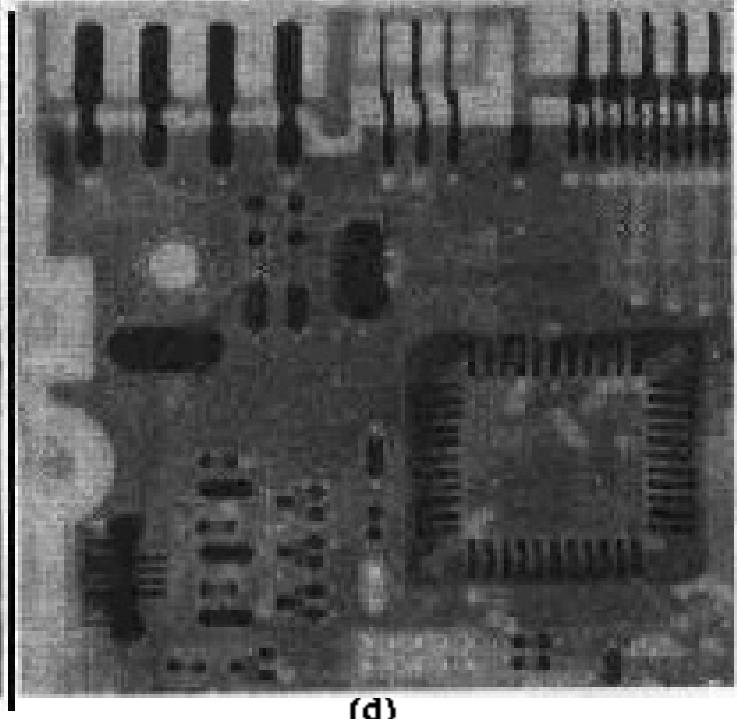
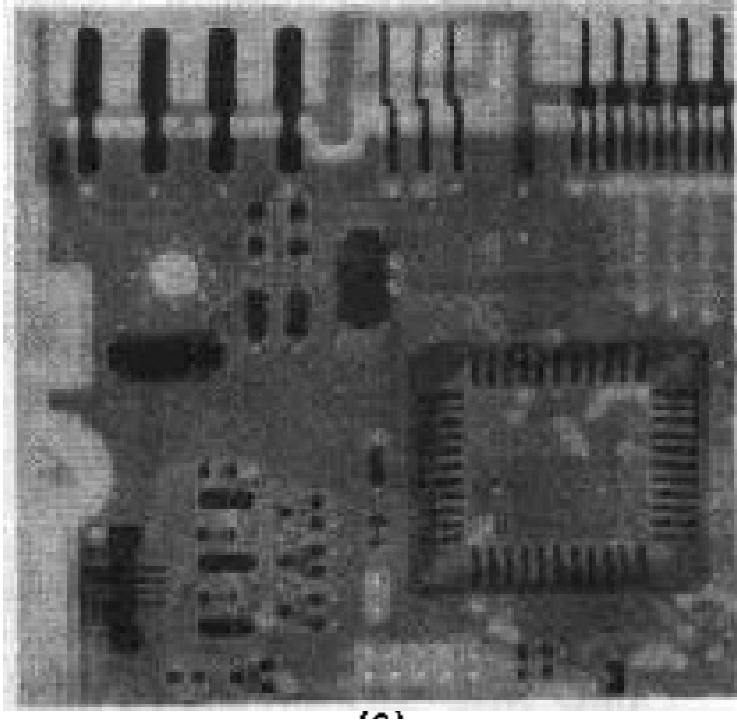
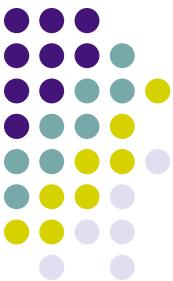


原图



原图+高斯噪声

# 效果图

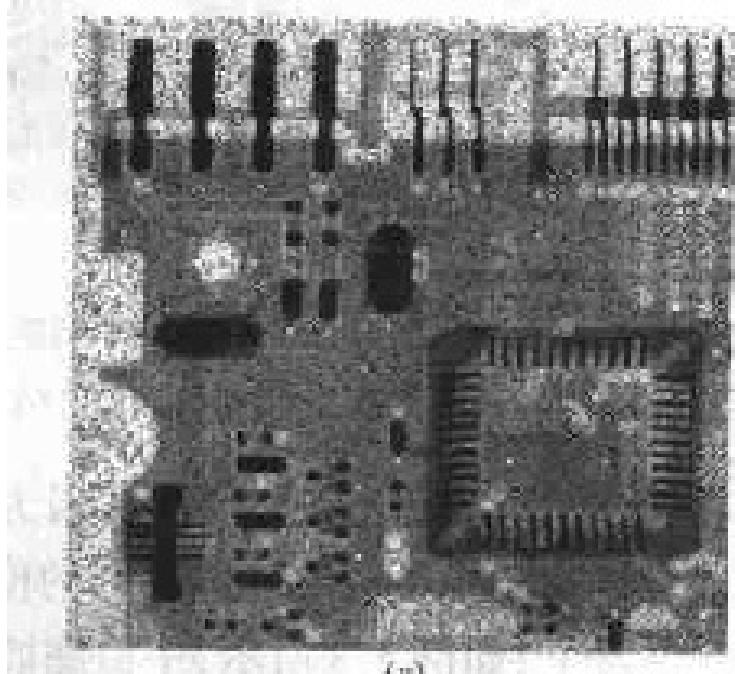


算术均值滤波器

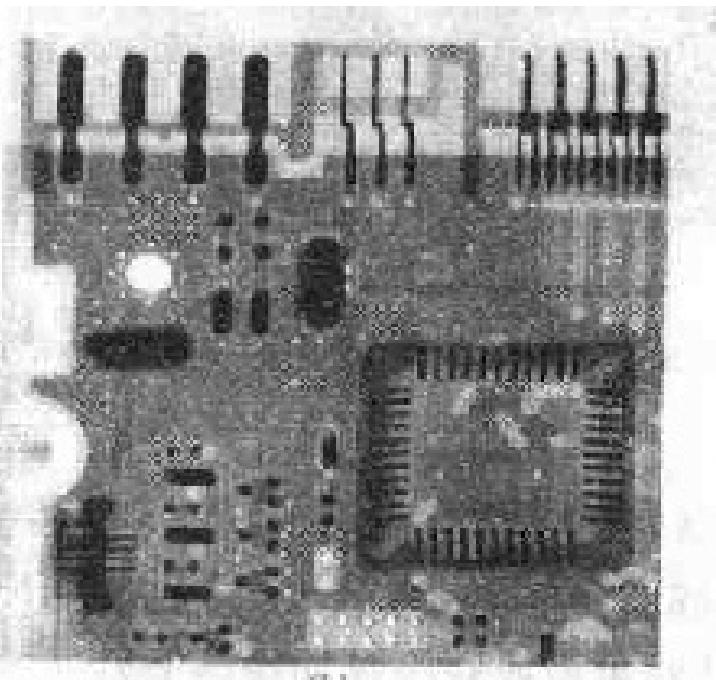
几何均值滤波器

算术均值滤波器会使图像变模糊

# 效果图



(a)

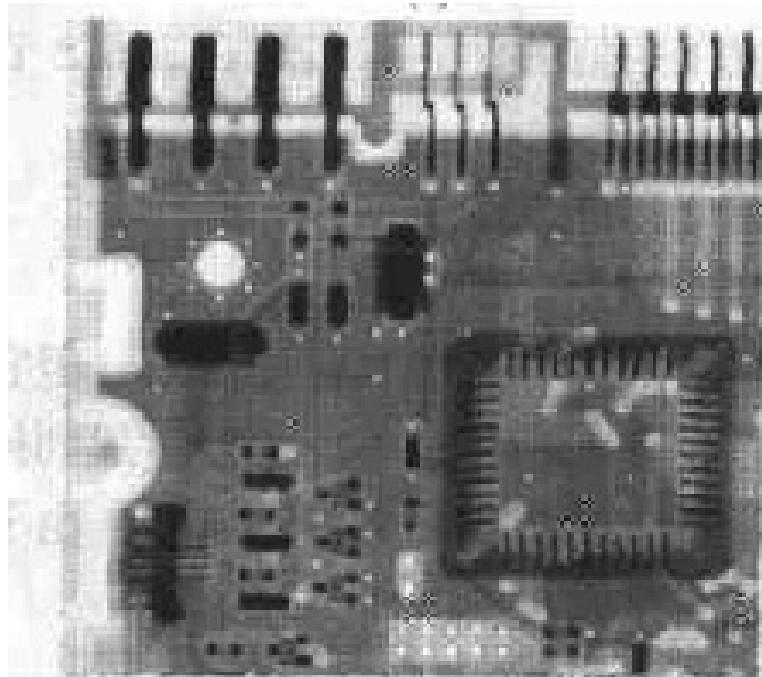
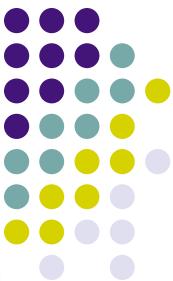


(b)

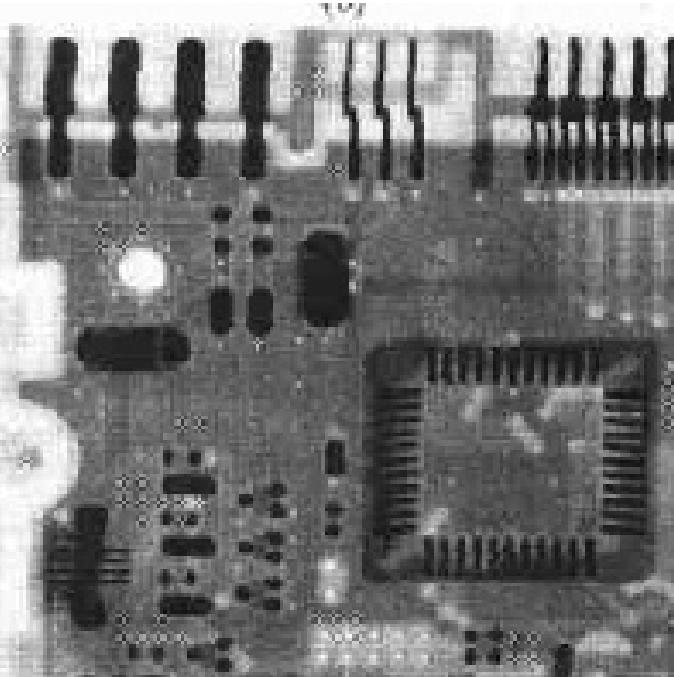
10%的概率被“胡  
椒”噪声污染

10%的概率被  
“盐”噪声污染

# 效果图



(c)

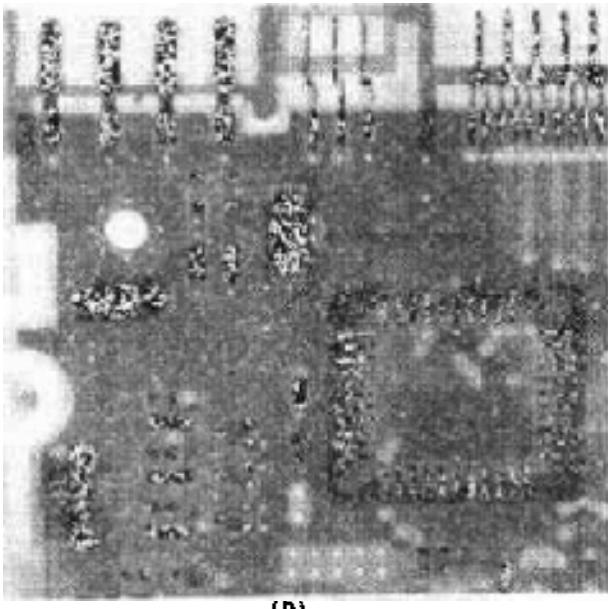
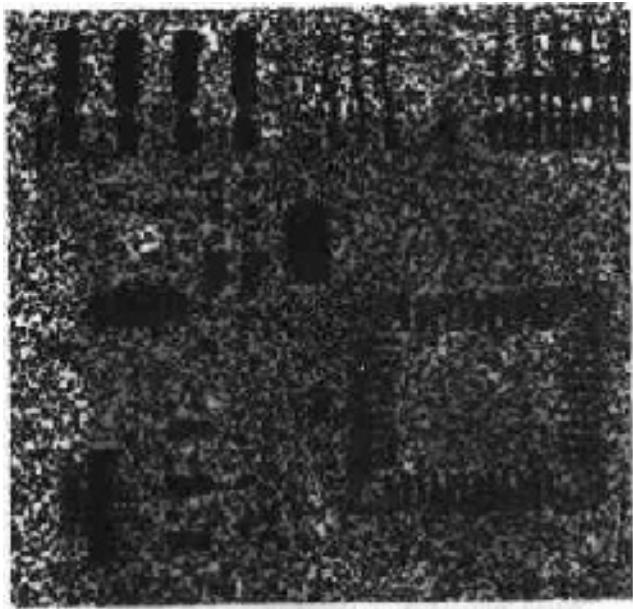


(d)

$Q=1.5$  处理  
“胡椒” 噪声

$Q=-1.5$  处理  
“盐” 噪声

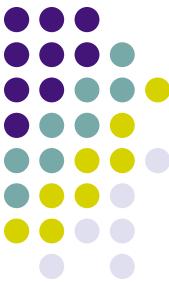
# 效果图



$Q=-1.5$ 处理  
“胡椒”噪声

$Q=1.5$ 处理  
“盐”噪声

# 小结

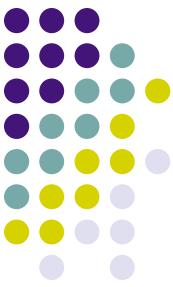


- 除了逆谐波均值之外，不同的均值滤波对应于不同的幂次下的平均值
  - 算术均值，幂为1
  - 几何均值，幂为0
  - 谐波均值，幂为-1
- 逆谐波均值： $Q>0$ 适合处理胡椒噪声； $Q<0$ 适合处理盐噪声；为什么？

# 统计排序滤波器



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

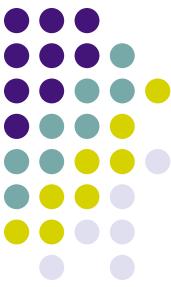


# 中值滤波器

- 取近邻的中值替代该像素的值

$$\hat{f}(x, y) = \underset{(s, t) \in S_x}{\text{median}} \{ g(s, t) \}$$

- 对于单级或双极脉冲噪声非常有效



# 最大最小值滤波器

- 最大值

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

- 适合于处理椒盐噪声

- 最小值

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

- 适合于处理盐噪声



# 中点滤波器

- 取最大最小的中值

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



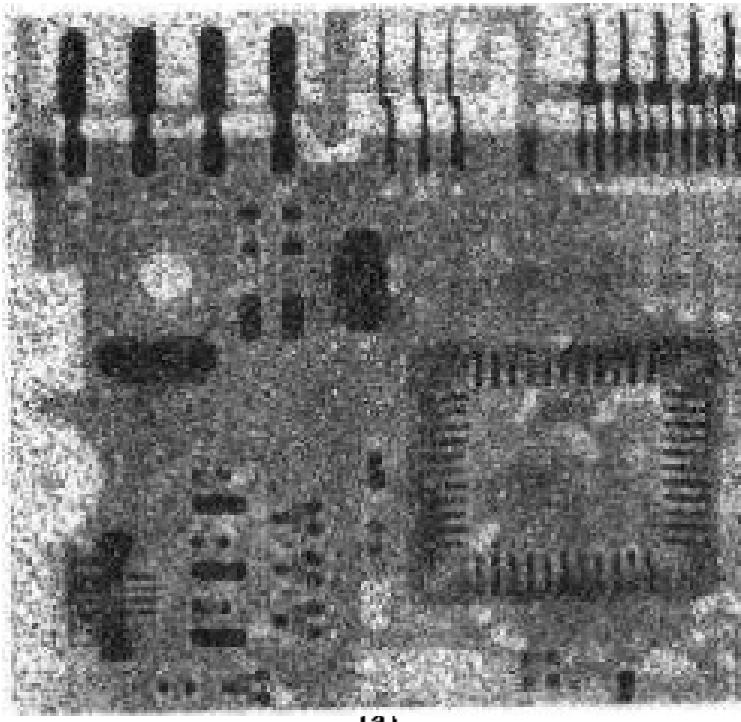
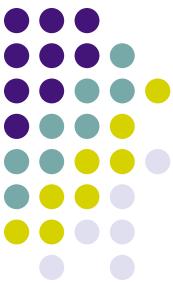
# 修正后的阿尔法均值滤波器

- 去掉少量最高，去掉少量最低，剩下的取平均

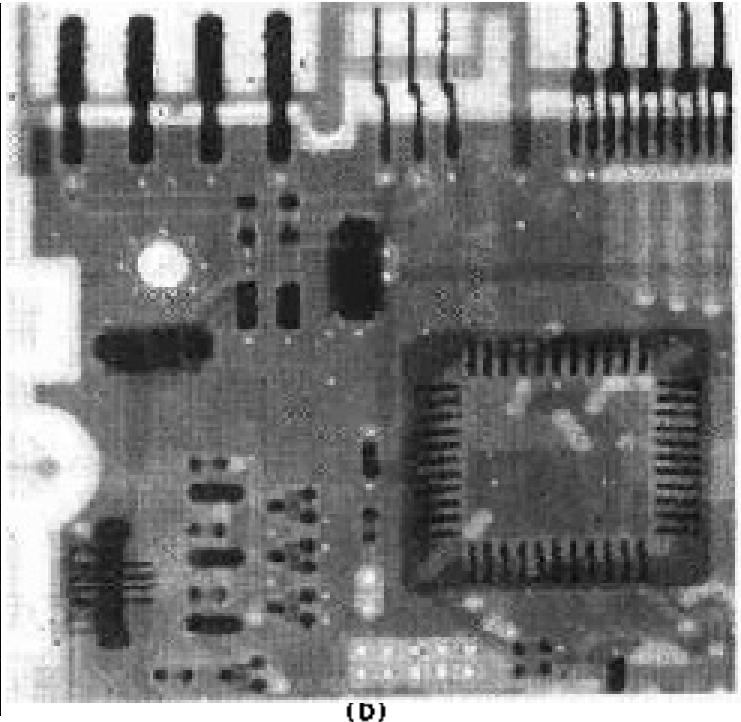
$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s, t) \in S_{xy}} g_r(s, t)$$

- $d=0$ , 退化为均值滤波器
- $d=(mn-1)$ , 退化为中值滤波器

# 效果图



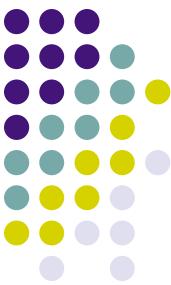
(a)



(b)

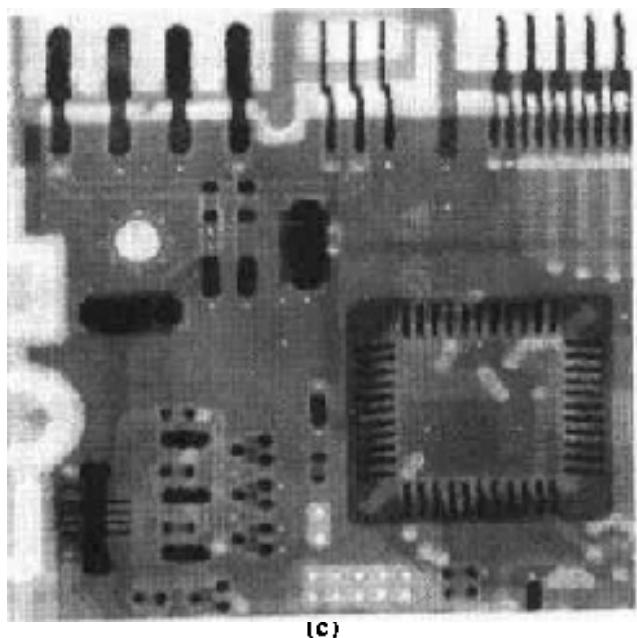
10%的概率被“椒盐”噪声污染

中值滤波器

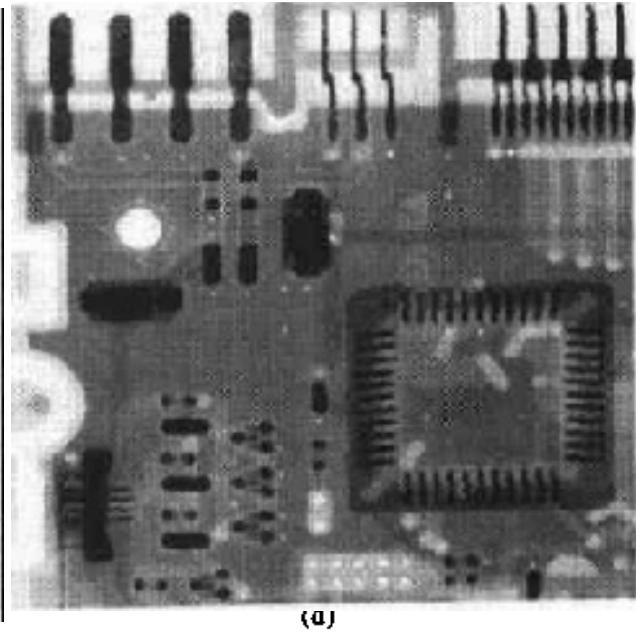


# 效果图

对中值滤波器的结果再进行中值滤  
波器复原的结果



第二次



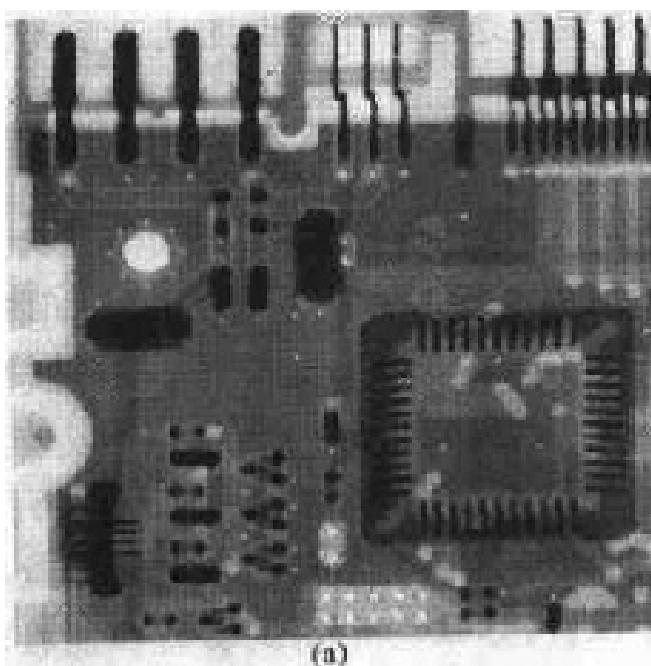
第三次

经过三次中值滤波噪声全部消除。

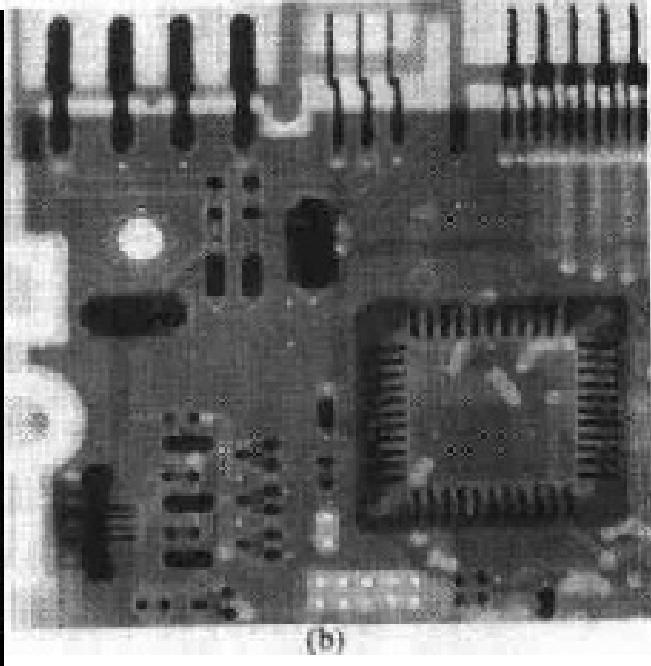


# 效果图

- 对于10%的胡椒噪声采用最大值滤波器的效果图，以及对于10%的盐噪声采用最小值滤波器的效果图



(a)

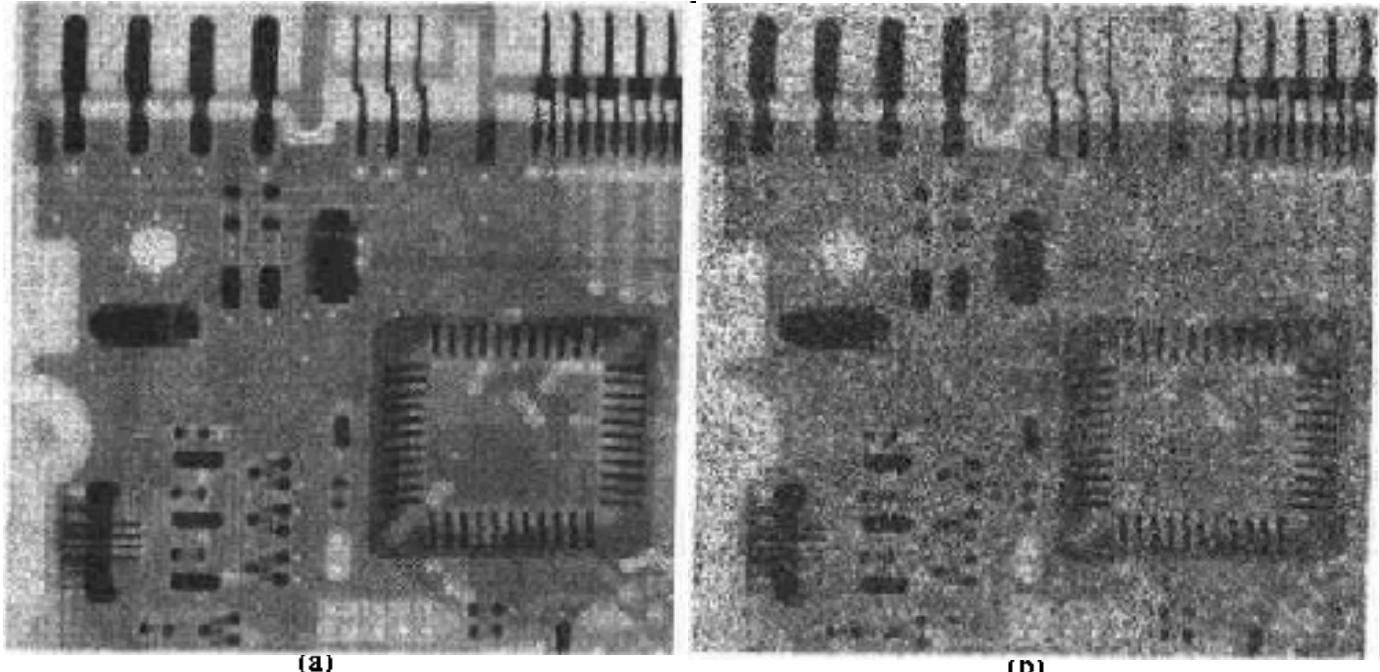
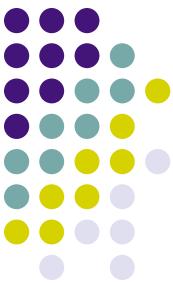


(b)

最大

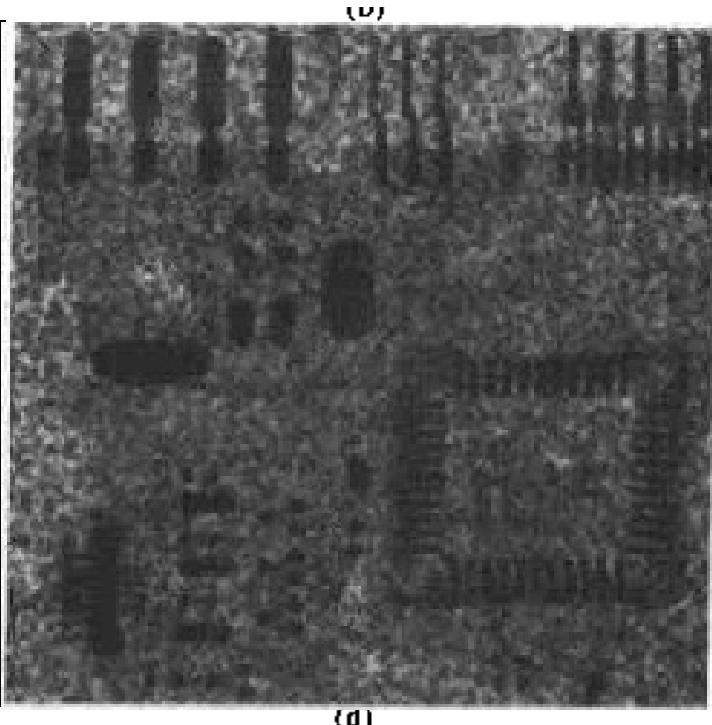
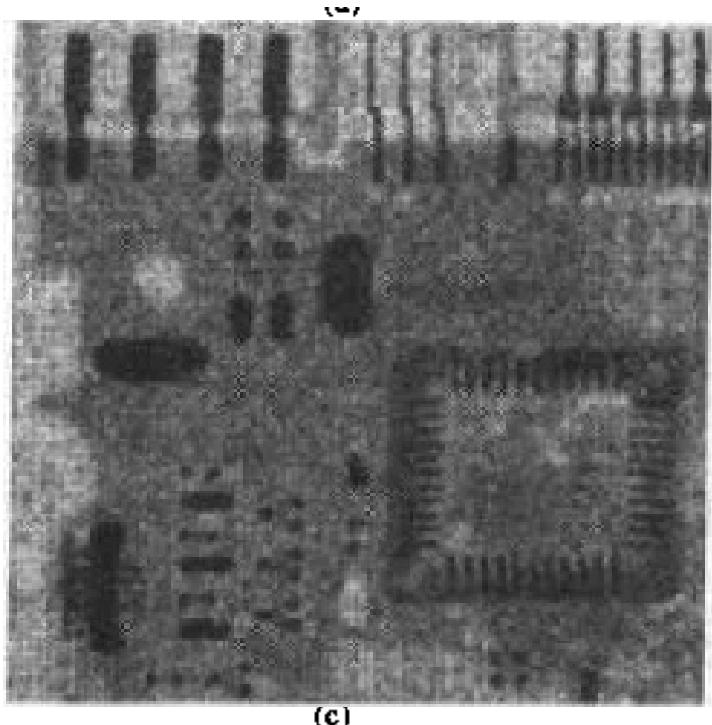
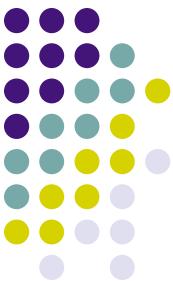
最小

# 效果图



先加入高斯噪声，再加入椒盐噪声后的图像

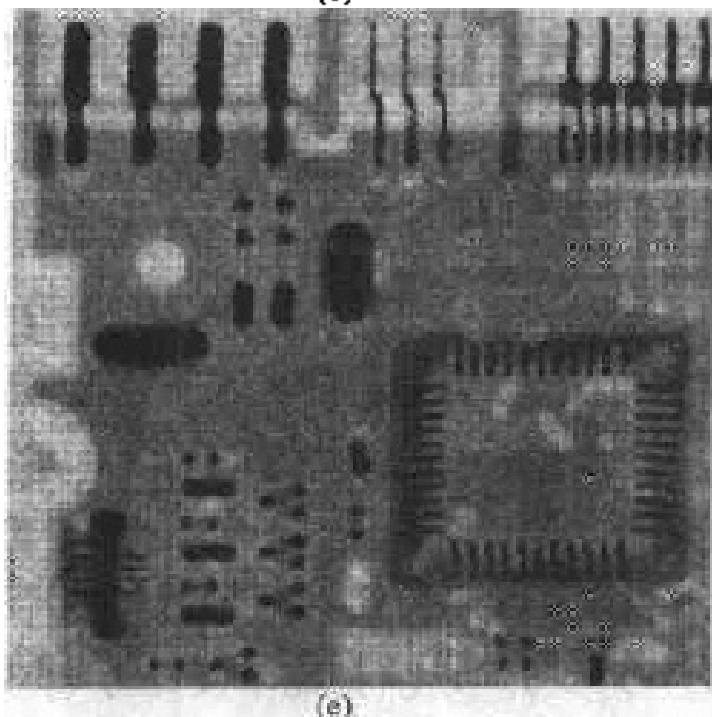
# 效果图



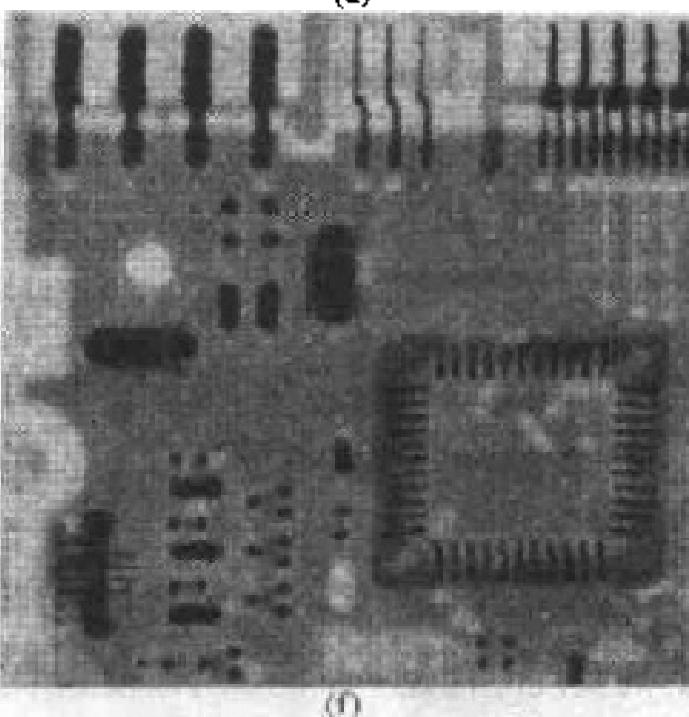
算术均值

几何均值

# 效果图



(e)



(f)

中值濾波

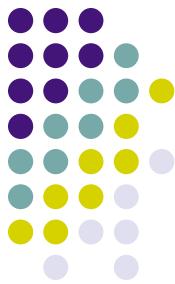
阿尔法濾波



# 小结

- 中值滤波器能很好的处理脉冲型噪声（椒盐噪声）。反复使用中值滤波虽然可以进一步减少噪声，也可能会使图像模糊化。因此一般尽可能减少处理的次数
- 最大值滤波器更适合于“胡椒”噪声；最小值滤波器更适合于“盐”噪声；**为什么？**
- 修正后的阿尔法均值滤波器更适合处理复杂噪声，例如高斯噪声和椒盐噪声混合的情况

# 自适应滤波器



- 问题：之前假定所有像素点采用相同的滤波器。这个假定没有考虑每个像素点的特性。因此，需要发展自适应滤波器
- 两类做法
  - 自适应，局部噪声消除滤波器
  - 自适应中值滤波器



# 自适应，局部噪声消除滤波器

- 四个基本统计量

- (a)  $g(x, y)$  表示噪声图像在点  $(x, y)$  上的值；
- (b)  $\sigma_n^2$ , 干扰  $f(x, y)$  以形成  $g(x, y)$  的噪声方差；
- (c)  $m_L$ , 在  $S_{xy}$  上像素点的局部均值；
- (d)  $\sigma_L^2$ , 在  $S_{xy}$  上像素点的局部方差。

- 基本公式

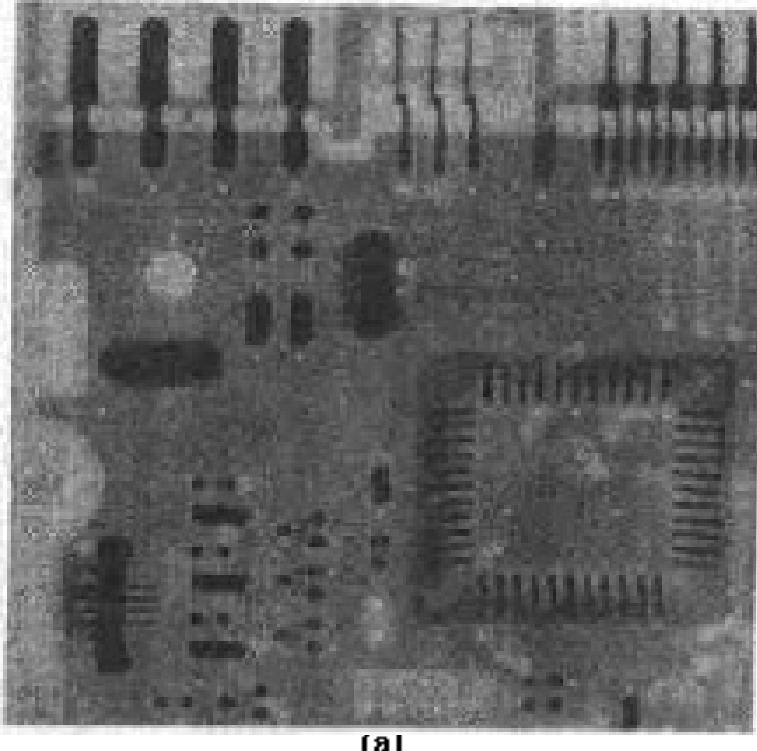
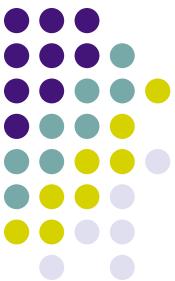
$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_n^2}{\sigma_L^2} [g(x, y) - m_L]$$



## ● 期望结果

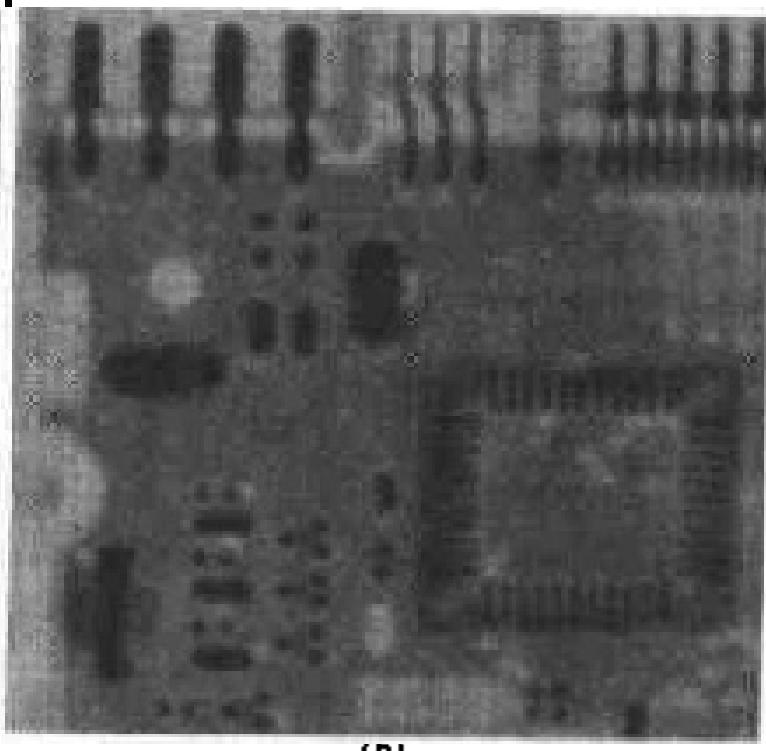
- 如果噪声方差为0， 直接返回像素值
- 如果噪声方差很大， 返回算术均值
- 如果噪声方差在两者之间， 返回中间结果

# 效果图



(a)

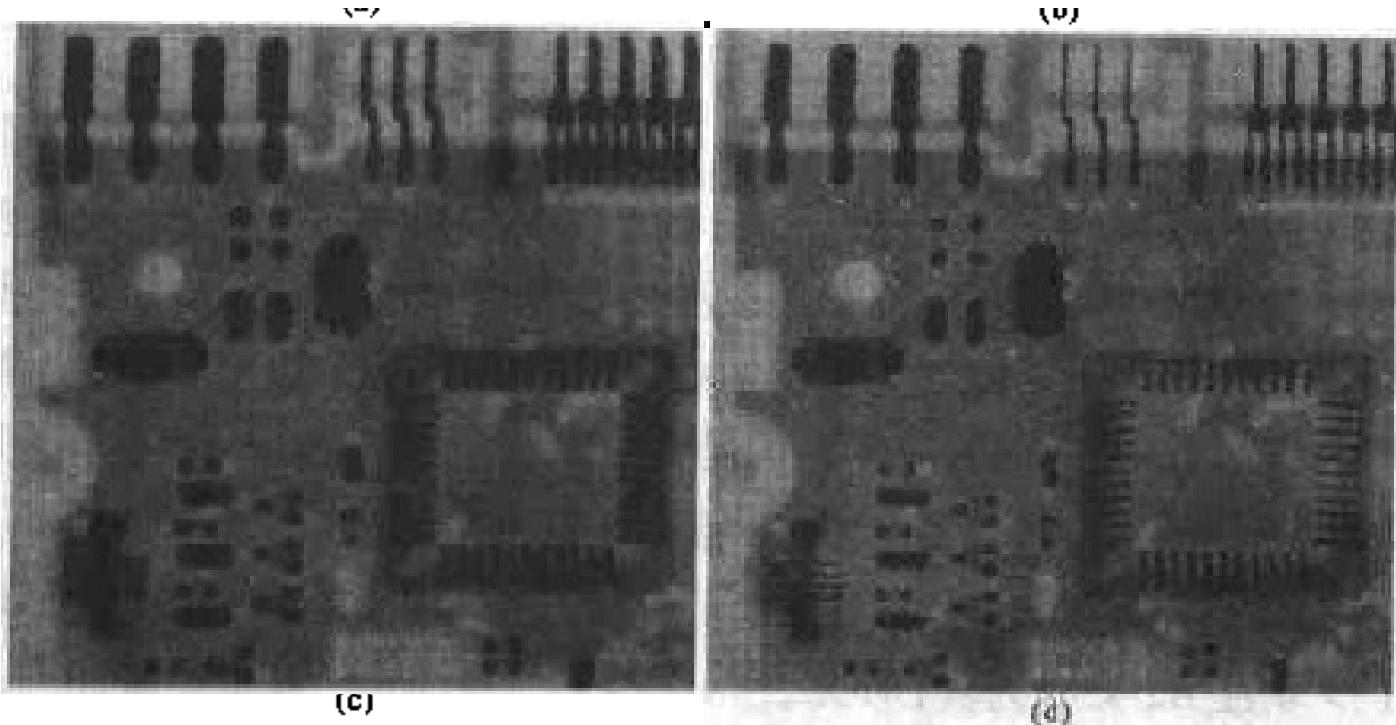
加入方差为1000  
的高斯噪声



(b)

算术均值滤波器

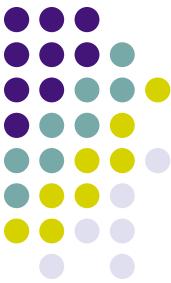
# 效果图



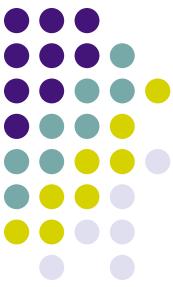
几何均值滤波器

自适应滤波器

# 点评



- 自适应滤波器虽然取得了更好的效果，但基于一个假定：噪声方差已知。
- 很多时候噪声方差未知时，需要估计。估计的错误会影响最后的效果。



# 自适应中值滤波器

- 一些统计量

$z_{\min} = S_{xy}$  中灰度级的最小值

$z_{\max} = S_{xy}$  中灰度级的最大值

$z_{\text{med}} = S_{xy}$  中灰度级的中值

$z_{xy} =$  在坐标  $(x, y)$  上的灰度级

$S_{\max} = S_{xy}$  允许的最大尺寸



# 自适应中值滤波器

## ● 算法

*A 层：*

$$A1 = z_{\text{med}} - z_{\text{min}}$$

$$A2 = z_{\text{med}} - z_{\text{max}}$$

如果  $A1 > 0$  且  $A2 < 0$ , 转到 *B 层*

否则增大窗口尺寸

如果窗口尺寸  $\leq S_{\text{max}}$  重复 *A 层*

否则输出  $z_{xy}$

*B 层：*

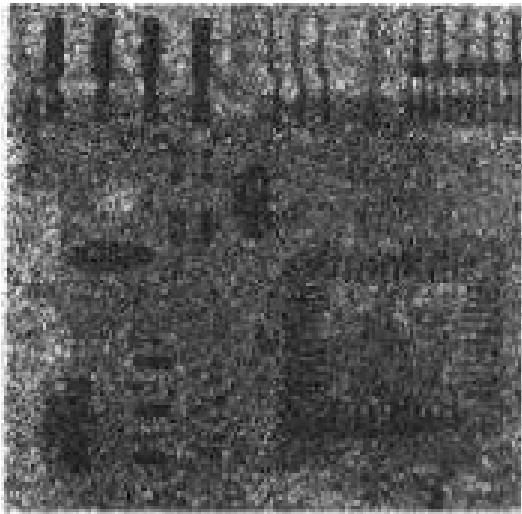
$$B1 = z_{xy} - z_{\text{min}}$$

$$B2 = z_{xy} - z_{\text{max}}$$

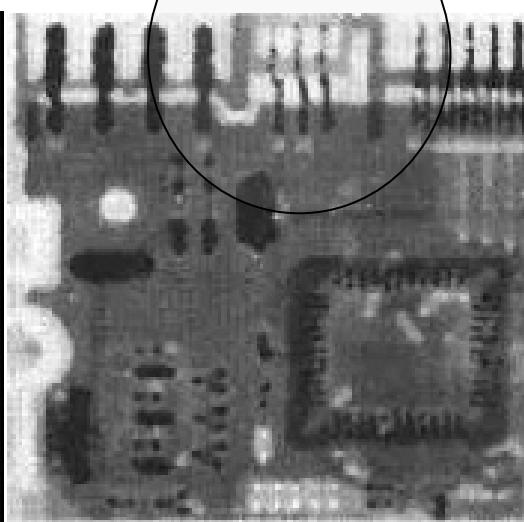
如果  $B1 > 0$  且  $B2 < 0$ , 输出  $z_{xy}$

否则输出  $z_{\text{med}}$

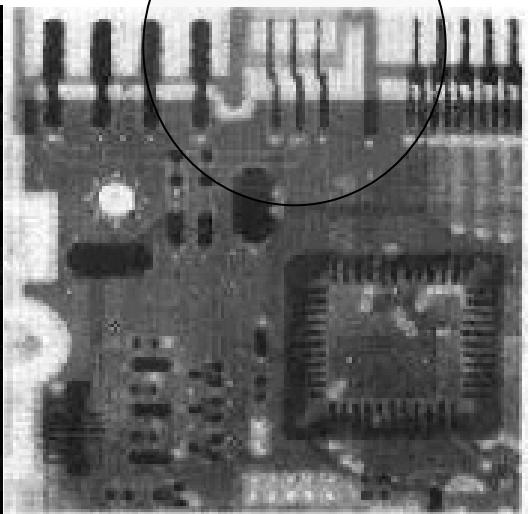
# 效果图



(a)



(b)



(c)

25%的概率被  
椒盐噪声污染

中值滤波器

自适应中值  
滤波器

# 频域滤波器用于图像复原



- 常用的四种频域滤波器用于图像复原
  - 带阻滤波器
  - 带通滤波器
  - 陷波滤波器
  - 最优陷波滤波器



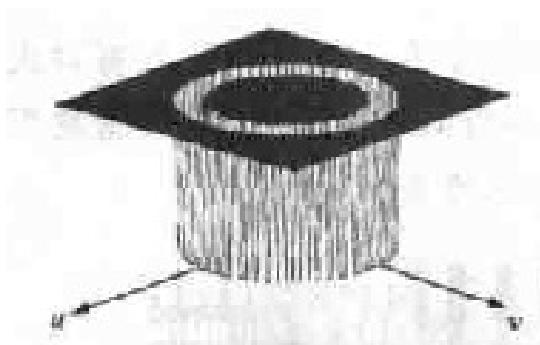
# 带阻滤波器

- 带阻滤波器：消除或衰减傅里叶变换原点处的频段

## 理想带阻滤波器

$$H(u, v) = \begin{cases} 1, & D(u, v) < D_0 - \frac{W}{2} \\ 0, & D_0 - \frac{W}{2} \leq D(u, v) \leq D_0 + \frac{W}{2} \\ 1, & D(u, v) > D_0 + \frac{W}{2} \end{cases}$$

$D_0$ 是频带的中心半径，  
 $W$ 是频带的宽度



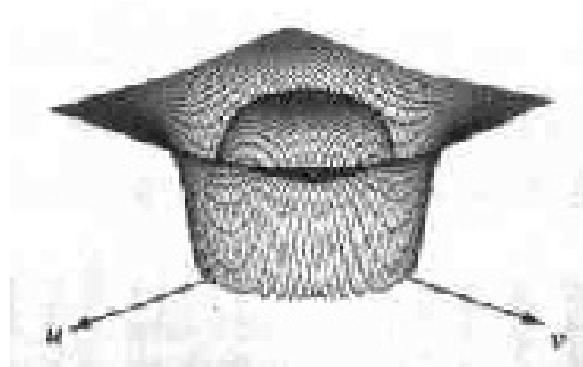


# 带阻滤波器法

- 带阻滤波器：消除或衰减傅里叶变换原点处的频段

n阶的巴特沃思带阻滤波器表达式为

$$H(u, v) = \frac{1}{1 + \left[ \frac{D(u, v)W}{D^2(u, v) - D_0^2} \right]^{2n}}$$



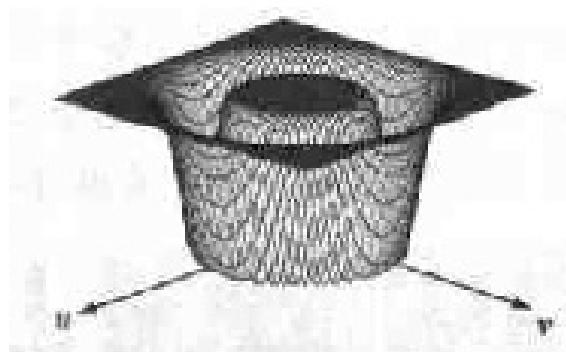


# 带阻滤波器法

- 带阻滤波器：消除或衰减傅里叶变换原点处的频段

高斯带阻滤波器表达式为

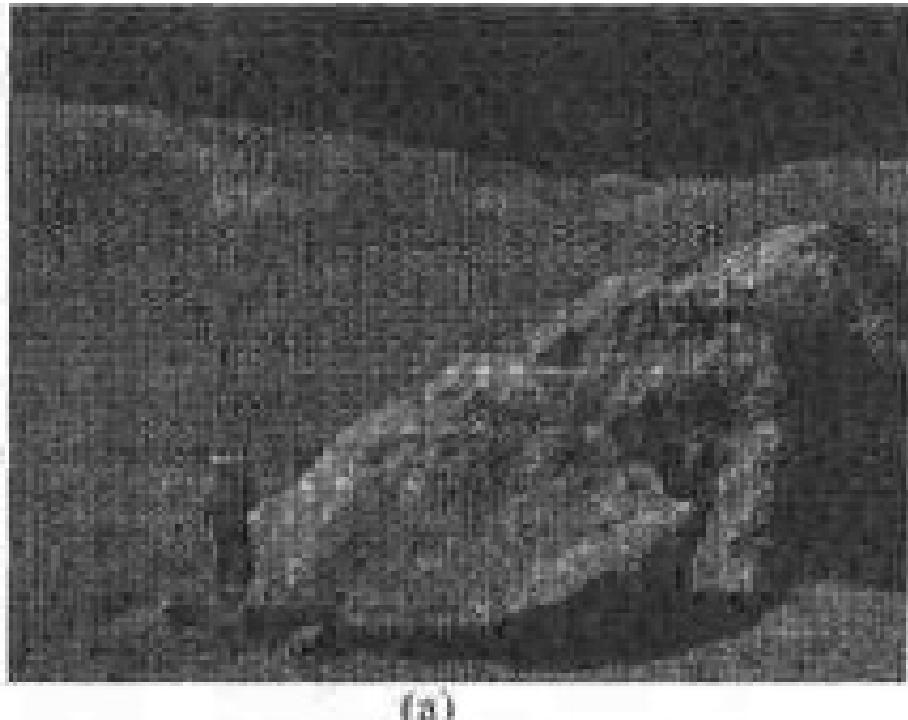
$$H(u, v) = 1 - e^{-\frac{1}{2} \left[ \frac{\rho^2(u, v) - D_0^2}{D(u, v)W} \right]^2}$$





# 带阻滤波器法

- 带阻滤波器：消除或衰减傅里叶变换原点处的频段  
考虑以下这幅图像

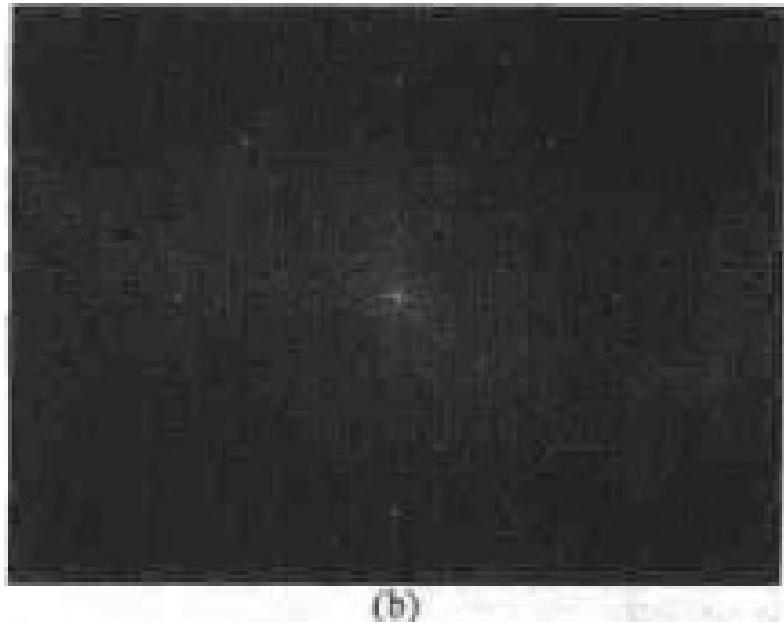


因现实世界的电力或机电干扰，图像被不同频率的正弦噪声干扰



# 带阻滤波器法

- 带阻滤波器：消除或衰减傅里叶变换原点处的频段  
考虑以下这幅图像

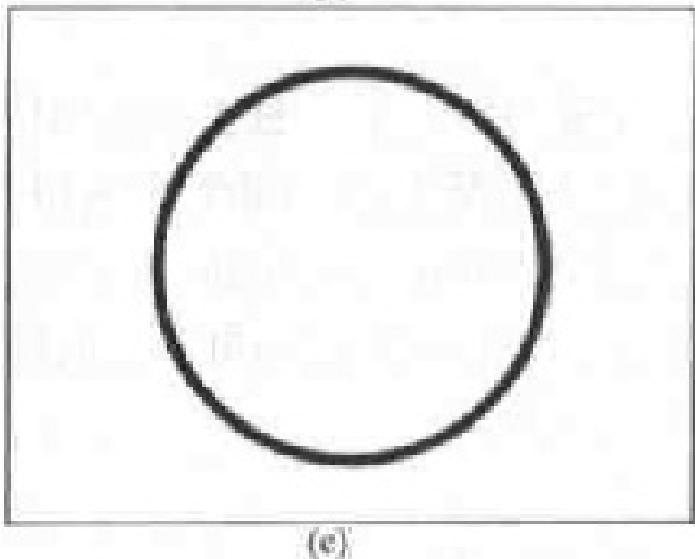


图像的频谱



# 带阻滤波器法

- 带阻滤波器：消除或衰减傅里叶变换原点处的频段  
考虑以下这幅图像



巴特沃思带阻  
滤波器

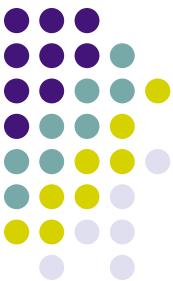


# 带阻滤波器法

- 带阻滤波器：消除或衰减傅里叶变换原点处的频段  
考虑以下这幅图像



滤波后的效果  
图



# 带通滤波器法

- 带通滤波器：执行与带阻滤波器相反的操作

传递函数

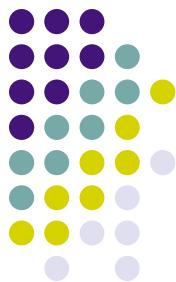
$$H_{\text{bp}}(u, v) = 1 - H_{\text{br}}(u, v)$$



带通滤波器传递函数



带阻滤波器传递函数



# 带通滤波器法

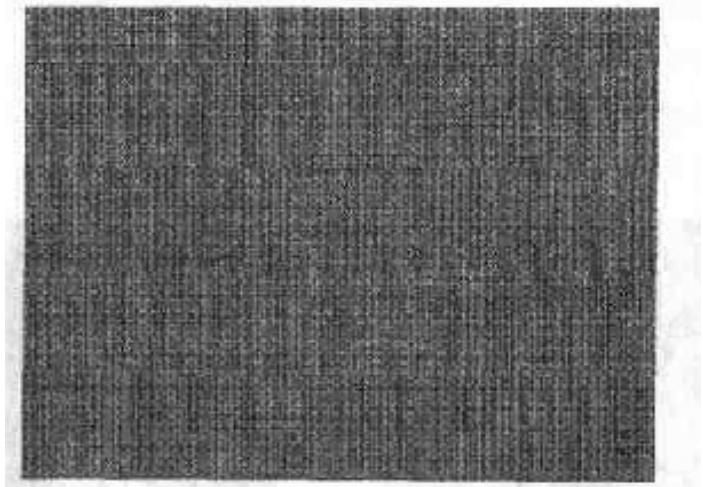
- 带通滤波器

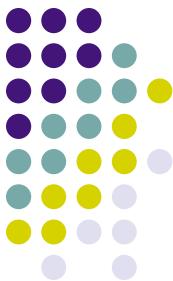
很明显，直接采用带通滤波器会消除掉太多图像细节，不能用于图像增强



(a)

带通滤波器效果图



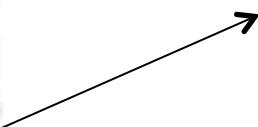
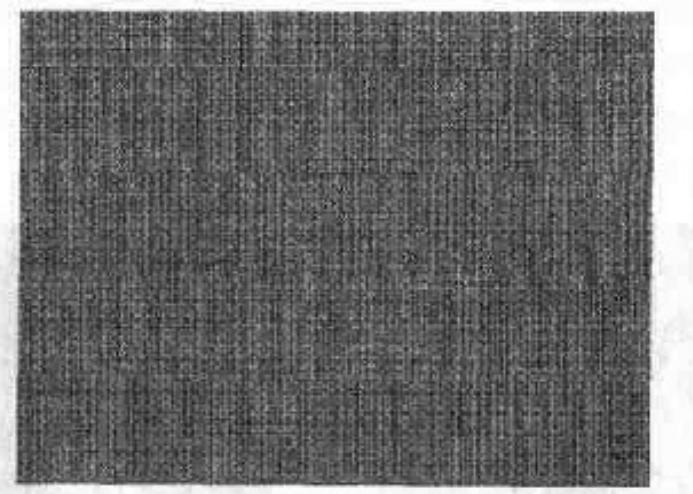


# 带通滤波器法

- 带通滤波器

很明显，直接采用带同滤波器会消除掉太多图像细节，不能用于图像增强

## 带通滤波器效果图



对应于噪声部分

带通可用于分析噪  
声模式



# 陷波滤波器法

- 陷波滤波器：阻止（或通过）事先定义的中心频率领域内的频率

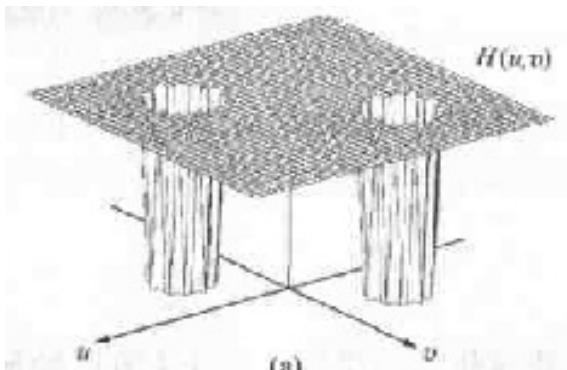
与带阻，带通类似的，由于傅里叶变换是对称的，陷波滤波器也是对称的

与带阻，带通不同的是，陷波滤波器不限定区域的形状，可以是任意的

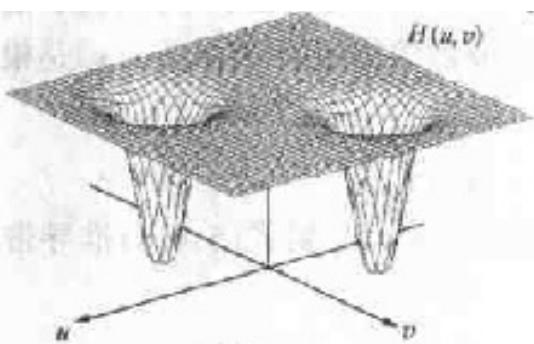


# 陷波滤波器法

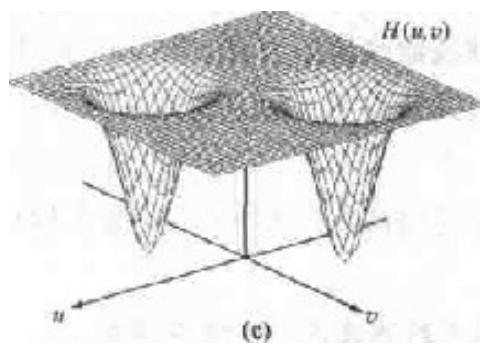
- 陷波滤波器：阻止（或通过）事先定义的中心频率领域内的频率



理想（带阻）  
陷波滤波器



巴特沃思  
(带阻) 陷  
波滤波器



高斯（带阻）  
陷波滤波器



# 陷波滤波器法

- 陷波滤波器的传递函数

$$H(u, v) = \begin{cases} 0 & D_1(u, v) \leq D_0 \text{ 或 } D_2(u, v) \leq D_0 \\ 1 & \text{其他} \end{cases}$$

表示可以有多个形状，  
不限于过原点的圆形



# 陷波滤波器法

- 陷波滤波器的传递函数

例如，可以表示两个不同位置的圆形

$$D_1(u, v) = [(u - M/2 - u_0)^2 + (v - N/2 - v_0)^2]^{1/2}$$

$$D_2(u, v) = [(u - M/2 + u_0)^2 + (v - N/2 + v_0)^2]^{1/2}$$

(M/2-u0,N/2-v0), (M/2+u0,N/2+v0)分别表示圆心



# 陷波滤波器法

- 陷波滤波器的传递函数

阶数为n的巴特沃思陷波带阻滤波器

$$H(u, v) = \frac{1}{1 + \left[ \frac{D_0^2}{D_1(u, v) D_2(u, v)} \right]^n}$$

高斯陷波带阻滤波器

$$H(u, v) = 1 - e^{-\frac{1}{2} \left[ \frac{D_1(u, v) D_2(u, v)}{D_0^2} \right]}$$

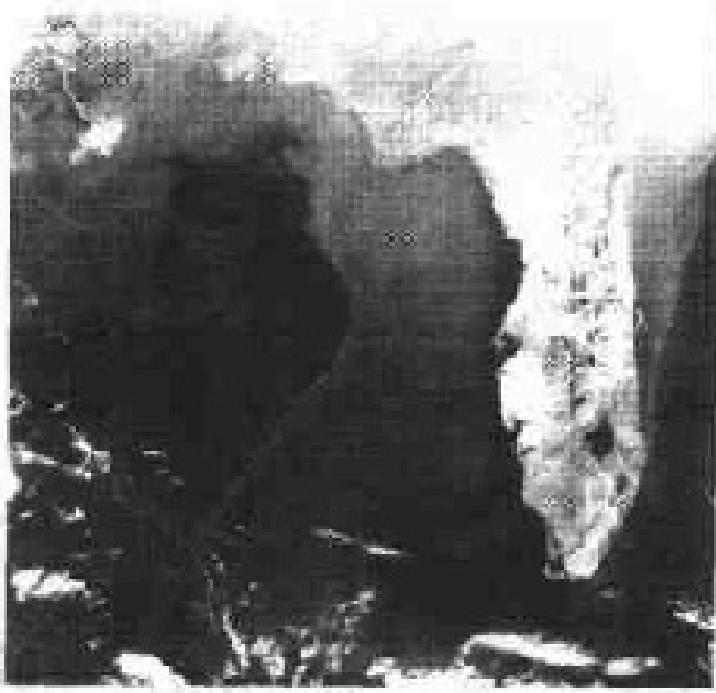
当0,v0为0时，退化为高斯高通滤波器



# 陷波滤波器法

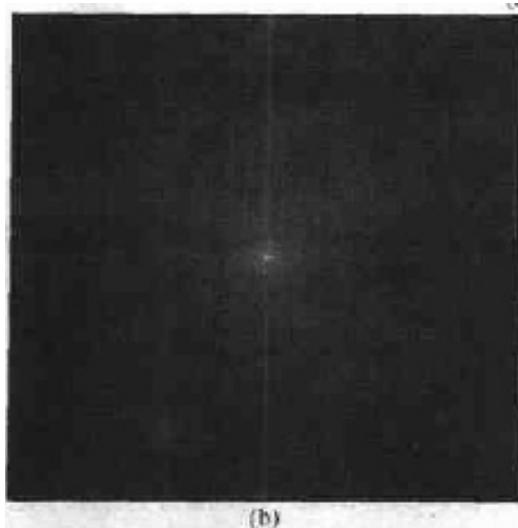
- 陷波滤波器的传递函数

以下图像为扫描线干扰



中心垂直方向有清晰模式的明显噪声

频谱图像



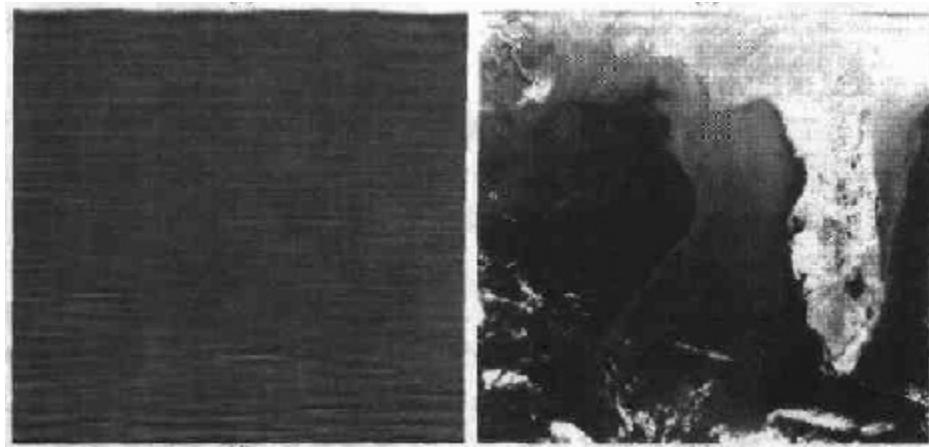
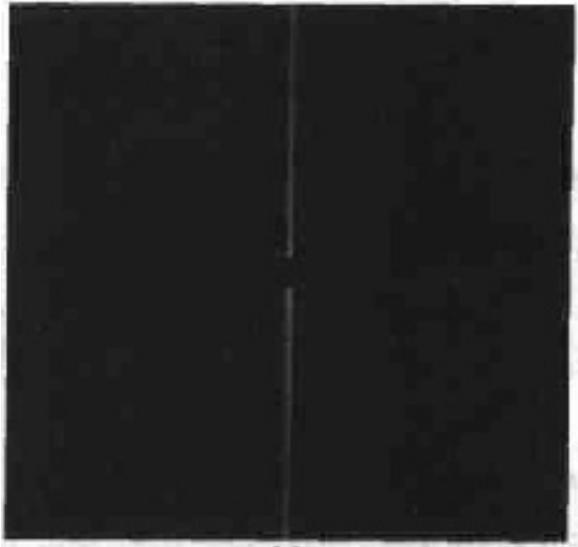


# 陷波滤波器法

- 陷波滤波器的传递函数

设计陷波滤波函数

得到噪声模式与滤波图像





# 最佳陷波滤波器法

- 问题：之前陷波滤波器假设干扰信号的周期模式已知。很多时候干扰信号的周期模式未知。如何处理？
- 做法：先通过一个大致的陷波滤波器进行处理，然后优化一组权值使得总体最小二乘误差最小



# 算法

- 第一步：估计得到干扰模式

$$\eta(x, y) = \mathcal{F}^{-1}\{H(u, v)G(u, v)\}$$

- $H$ 是某一陷波滤波器
- 第二步：假定复原图像表示为

$$\hat{f}(x, y) = g(x, y) - w(x, y)\eta(x, y)$$

- 其中 $w$ 为权值，需要确定

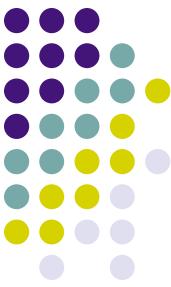


## • 第三步，优化

$$\sigma^2(x, y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^a \sum_{t=-b}^b [\hat{f}(x+s, y+t) - \bar{\hat{f}}(x, y)]^2$$

- $\bar{f}$ 表示领域内均值
- 第四步：求解

$$w(x, y) = \frac{\bar{g}(x, y) \eta(x, y) - g(x, y) \bar{\eta}(x, y)}{\bar{\eta}^2(x, y) - \eta^2(x, y)}$$



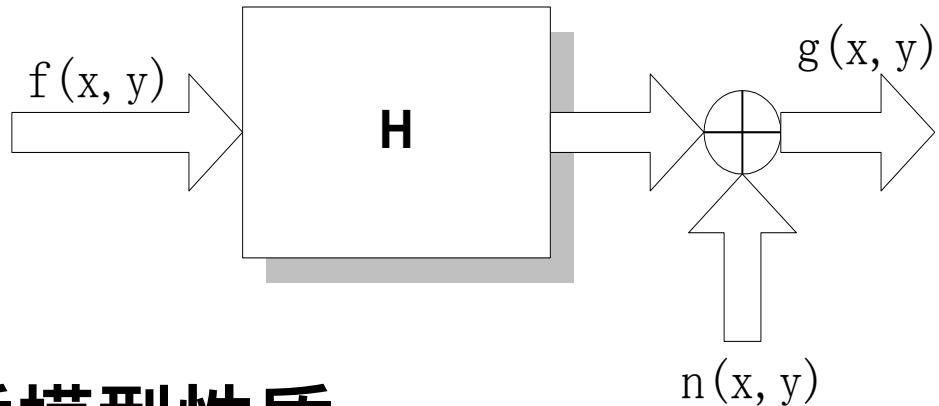
# 频域滤波器

- 熟悉带阻滤波器的原理
- 带通滤波器的作用——分析噪声模式
- 陷波滤波器在带阻滤波器的基础上，放松了干扰模式的形状，允许各种形状
- 最优陷波滤波器考虑在干扰模式不确定的情况下，通过最小化均方误差，学习最优权值来进行图像复原



# 图像降质模型

## • 降质模型



- 降质模型性质
  - $H$ 是线性的；
  - $H$ 是空间移不变的；

$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$

# 问题



$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$

- 如果H未知， 如何估计H？
- 如果H已知， 如何恢复F？



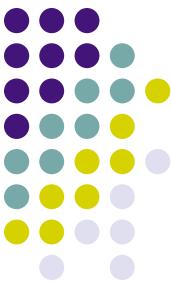
# 退化函数H的估计

- 图像观察法

- 做法

- 选取一些比较可靠的复原频谱，比如信号较强的部分
- 假定没有噪声，根据公式计算H

$$H_s(u, v) = \frac{G_s(u, v)}{\hat{F}_s(u, v)}$$



# 退化函数H的估计

- 实验估计法

- 做法：

- 假定图像退化由某个装置产生
- 输入纯黑或纯白或平滑图像，经该装置得到输出图像的频谱
- 假定噪声为零，根据公式，计算H

$$H(u, v) = \frac{G(u, v)}{A}$$

- 其中A为平滑图像的频谱（已知）



# 逆滤波

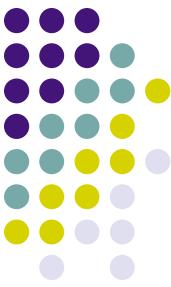
- 假定H已知，如何估计F？

- 对于无噪声情况：

$$\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)}$$

- 对于有噪声情况

$$\hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)}$$



- 问题：会出现**除零**情况
- 解决办法：截断
  - 截断H中非常小的值，仅利用数值较大的值（例如在原点附近的频谱）参与计算

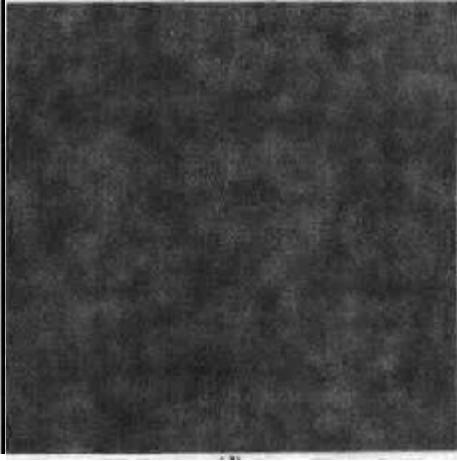
# 效果图



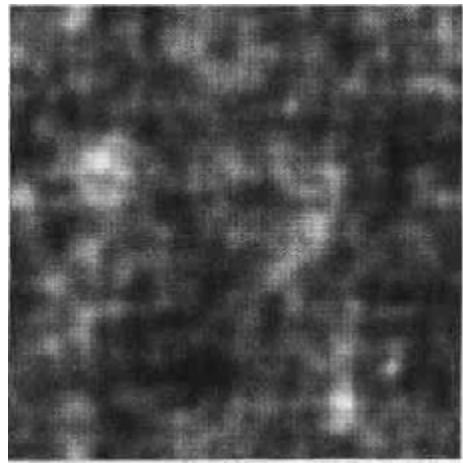
截断半径=40



截断半径=70



截断半径=80



全滤波



# 最小方差滤波（维纳滤波）

## ● 一些统计量

$H(u, v)$  = 退化函数

$H^*(u, v) = H(u, v)$  的复共轭

$|H(u, v)|^2 = H^*(u, v)H(u, v)$

$S_\eta(u, v) = |N(u, v)|^2$  = 噪声的功率谱 [见式(4.2.20)]

$S_f(u, v) = |F(u, v)|^2$  = 未退化图像的功率谱



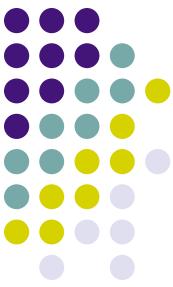
# 最小方差滤波（维纳滤波）

- 误差度量

$$e^2 = E \{ (f - \hat{f})^2 \}$$

$$\begin{aligned}\hat{F}(u, v) &= \left[ \frac{H * (u, v) S_f(u, v)}{S_f(u, v) + |H(u, v)|^2 + S_\eta(u, v)} \right] G(u, v) \\ &= \left[ \frac{H * (u, v)}{|H(u, v)|^2 + S_\eta(u, v) / S_f(u, v)} \right] G(u, v) \\ &= \left[ \frac{1}{|H(u, v)|} \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_\eta(u, v) / S_f(u, v)} \right] G(u, v)\end{aligned}$$

- 如果噪声为0，退化为逆滤波



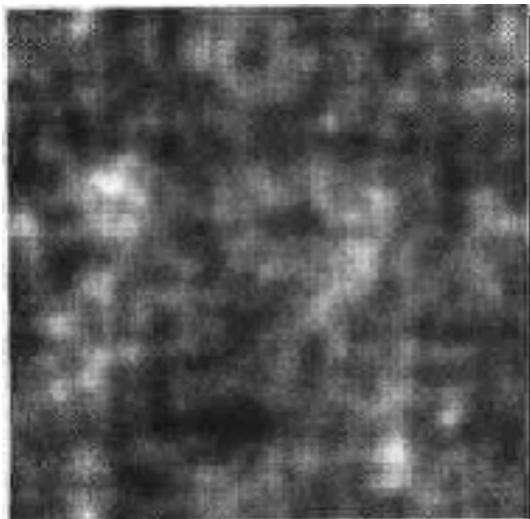
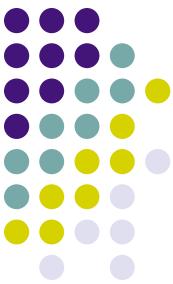
# 最小方差滤波（维纳滤波）

- 特殊情况，处理白噪声 ( $|N(u,v)|^2$  是个常数) 时，维纳滤波可以简化

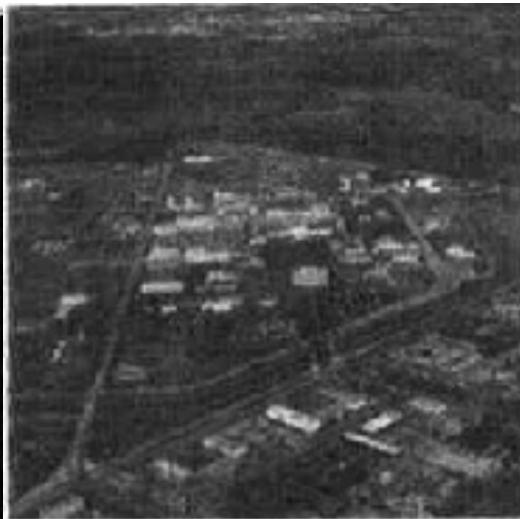
$$\hat{F}(u, v) = \left[ \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v)$$

- 白噪声指噪声的傅里叶频谱为常数

# 效果图



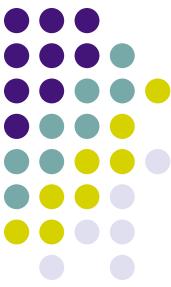
全滤波



半径受限的逆滤波



维纳滤波



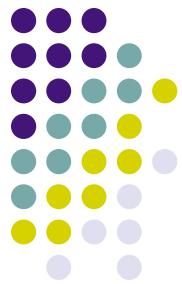
# 要点总结

- 图像复原与图像增强具有什么区别与联系；
- 重要的噪声函数有哪些；
- 均值滤波器，统计排序滤波器为什么可用于复原图像？
- 自适应滤波器的基本思想是什么？
- 频域滤波器都有哪些滤波器可以进行图像复原？各自的基本思想是什么？

# 要点总结



- 了解降质模型
- 降质模型下如何进行图像复原?
  - 退化函数 $H$ 的估计
  - $H$ 已知时, 了解逆滤波的原理及不足, 及解决方案



# 下一讲

