

# **Ch5 Image Restoration**

**Wang Jian**

**SEIE-TJU**

**Autumn 2017**



# Index

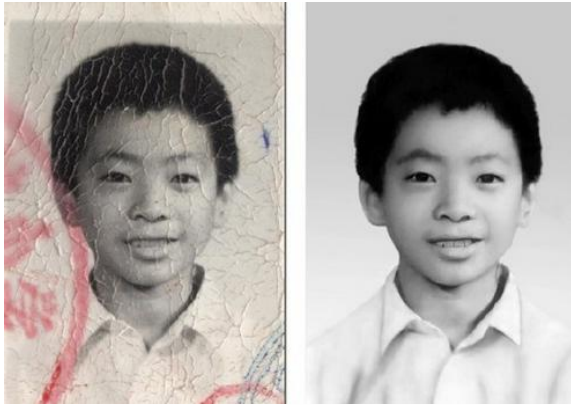
---

- **Introduction**
- Noise models
- Spatial filtering methods: noise only case
- Frequency domain filtering methods
- Linear position-invariant degradation
- Degradation function estimation
- Inverse filtering
- Wiener filtering
- Other methods
- Image reconstruction

# Image restoration v.s enhancement

- 与图像增强一样，图像复原技术的**最终目的**是改善给定的图像。
- **图像增强**是一个**主观过程**。**图像复原**是一个**客观过程**。
- 图像增强是一个探索性过程，根据人类视觉特性设计一种改善图像的方法。
- 图像复原则利用**退化现象**的某种先验知识来重建或恢复被退化的图像。

去模糊

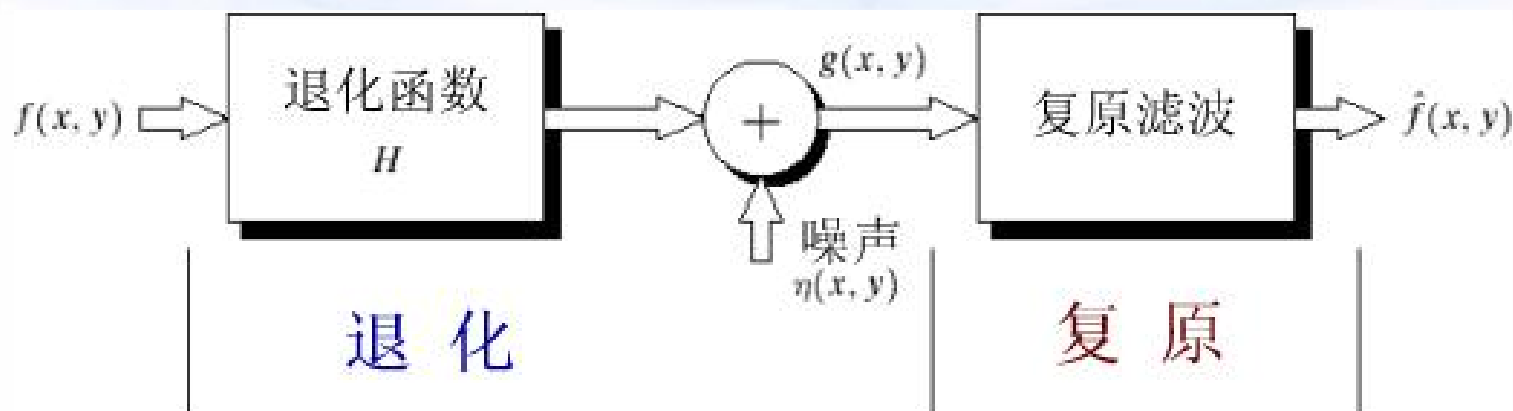


照片修复



去雾

# Image degradation /restoration model



- 退化图像  $g(x, y) \longleftrightarrow G(u, v)$ 
  - 退化过程模型化为一个**退化函数**和一个**加性噪声**项；
  - 图像复原的目的是获得关于原始图像的近似估计；
  - 知道 传输函数  $H$  和噪声  $\eta$  的信息越多，近似估计越接近原图像；

- 空域退化模型**

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

- 频域退化模型**

$$\updownarrow DFT$$

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



# Index

---

- **Image degradation /restoration model**
- **Noise models**
- **Spatial filtering methods: noise only case**
- **Frequency domain filtering methods**
- **Linear position-invariant degradation**
- **Degradation function estimation**
- **Inverse filtering**
- **Wiener filtering**
- **Other methods**
- **Image reconstruction**





# Noise models

- 数字图像的噪声主要来源于：
  - 图像的获取（数字化过程）
    - 图像传感器的工作情况受到各种因素的影响，如图像获取中的环境条件和传感器元件的自身质量。例如 **CCD** 摄像机，产生噪声的主要因素是光照强度和传感器温度。
  - 传输过程
    - 图像在传输过程中主要由于所用的传输信道的干扰受到噪声污染。例如无线传输网络，多径、光或其它大气因素的干扰而产生噪声。
- **噪声模型假设：**
  - 噪声独立于空间坐标；
  - 与图像不相关：像素值和噪声分量之间没有相关性。
- 噪声空间描述采用噪声分量灰度值的统计特性，用概率密度函数 (pdf) 表示。



# Noise models

---

- 噪声类型

- 高斯噪声

- 瑞利噪声

- 指数分布噪声

- 伽马噪声

- 均匀噪声

- 脉冲噪声

- 椒噪声

- 盐噪声

- 椒盐噪声

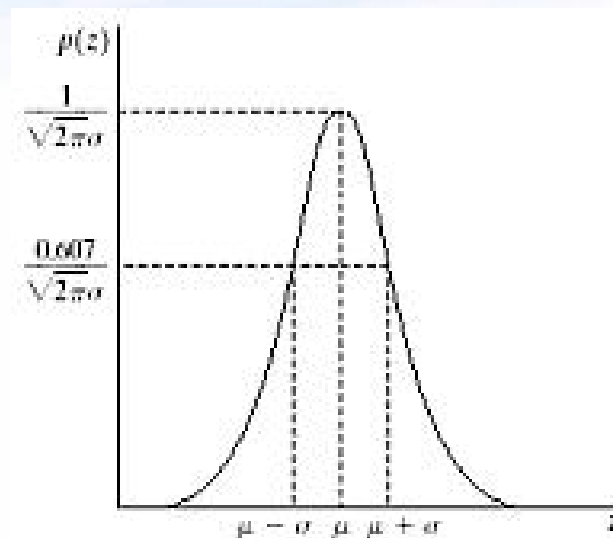
# Noise models



- 高斯噪声 **Gaussian** (正态噪声)

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

- **z** 表示灰度值，高斯随机变量。
- $\mu$  表示  $z$  的均值（数学期望）
- $\sigma$  表示  $z$  的标准差，标准差的平方  $\sigma^2$  称为  $z$  的方差。
- 当  $z$  服从高斯分布的时候，70% 落在  $[(\mu-\sigma), (\mu+\sigma)]$  范围内，有 99.5% 落在  $[(\mu-3\sigma), (\mu+3\sigma)]$  范围内。
- 高斯噪声在空间和频域中数学上易于处理，在实践中常用。
- 高斯噪声源于电子电路噪声和由低照明度和高温带来的传感器噪声；





# Noise models

- 瑞利噪声 **Rayleigh**

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

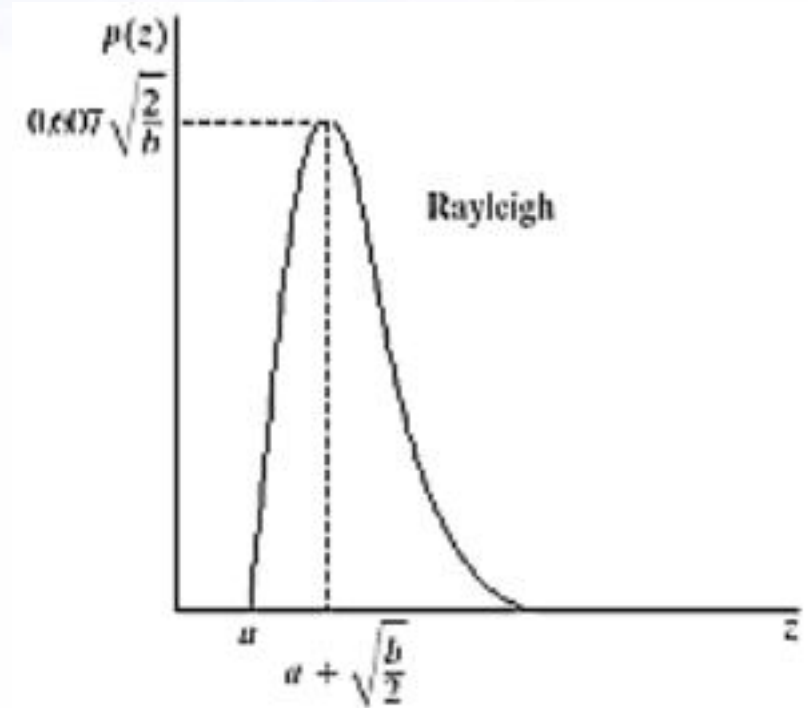
– 均值

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

– 方差

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

- 距原点的位移和其密度图形的基本形状向右变形，瑞利密度对于近似偏移的直方图十分适用。
- 深度成像中的噪声，如自然图像中的大气衰减（瑞利散射定律）

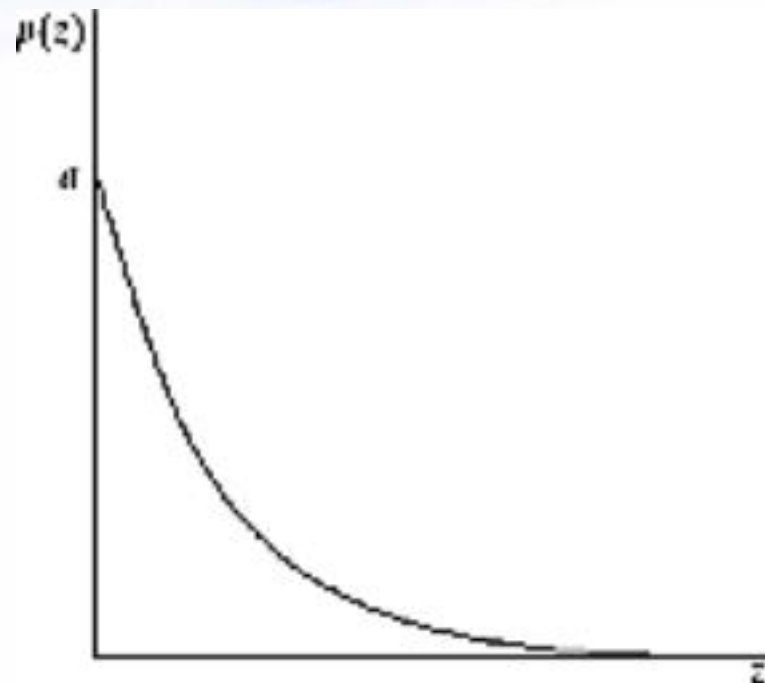


- 指数分布噪声 **Exponential**

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

其中  $a > 0$

- 均值  $\mu = 1/a$
- 方差  $\sigma^2 = 1/a^2$
- 指数分布的概率密度函数可视为当  $b=1$  时爱尔兰概率分布的特殊情况。
- 指数密度分布和伽马密度分布常出现在激光成像中;



# Noise models



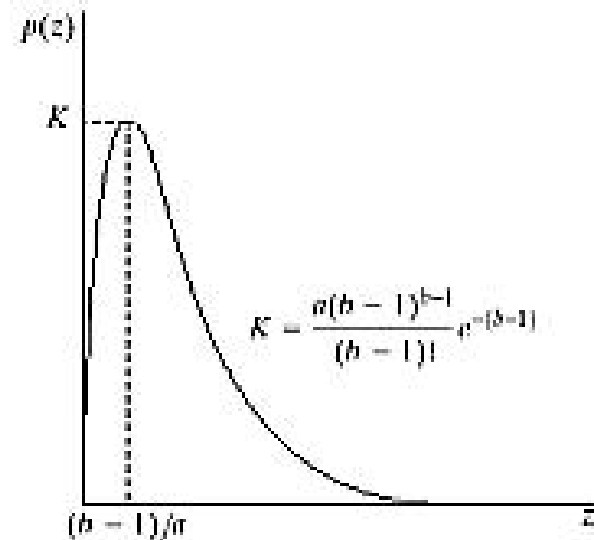
- 伽马 (Gamma) 噪声(爱尔兰 Erlang)

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

- $a > 0$ ,  $b$  为正整数,  $!$  表示阶乘。

- 均值  $\mu = b/a$

- 方差  $\sigma^2 = b/a^2$



- 伽马噪声的 **pdf** 表达式常被用来表示伽马密度, 严格的说, 只有当分母为伽马函数  $\Gamma(b)$  时才正确。分母如上式所示时, 该密度近似为爱尔兰密度。

# Noise models

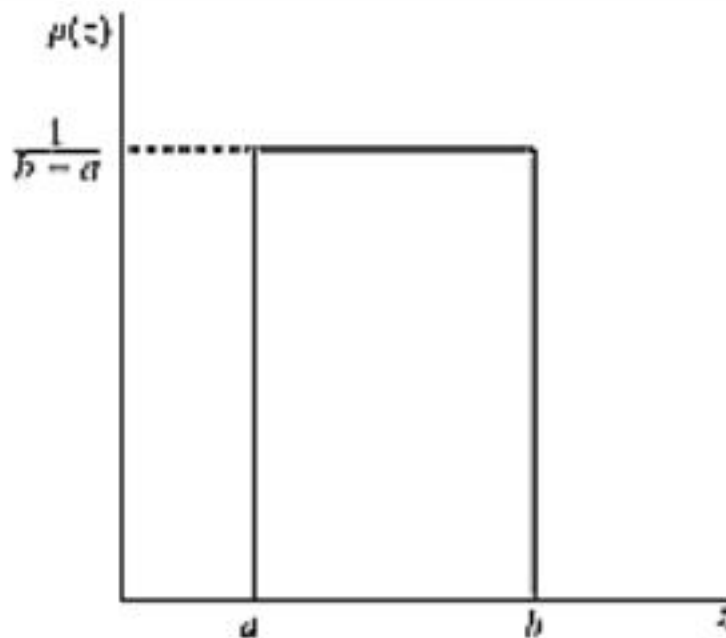


- 均匀分布噪声 **Uniform**

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

- 均值  $\mu = \frac{a+b}{2}$

- 方差  $\sigma^2 = \frac{(b-a)^2}{12}$



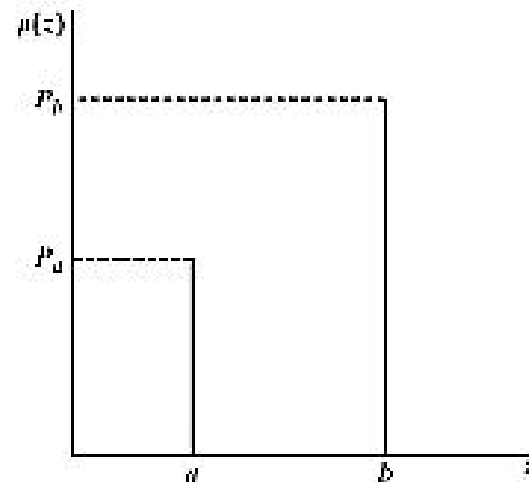
- 很少噪声服从均匀分布，常用于产生随机数

# Noise models



- (双极)脉冲噪声 **Impulse**  
(椒盐噪声)

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

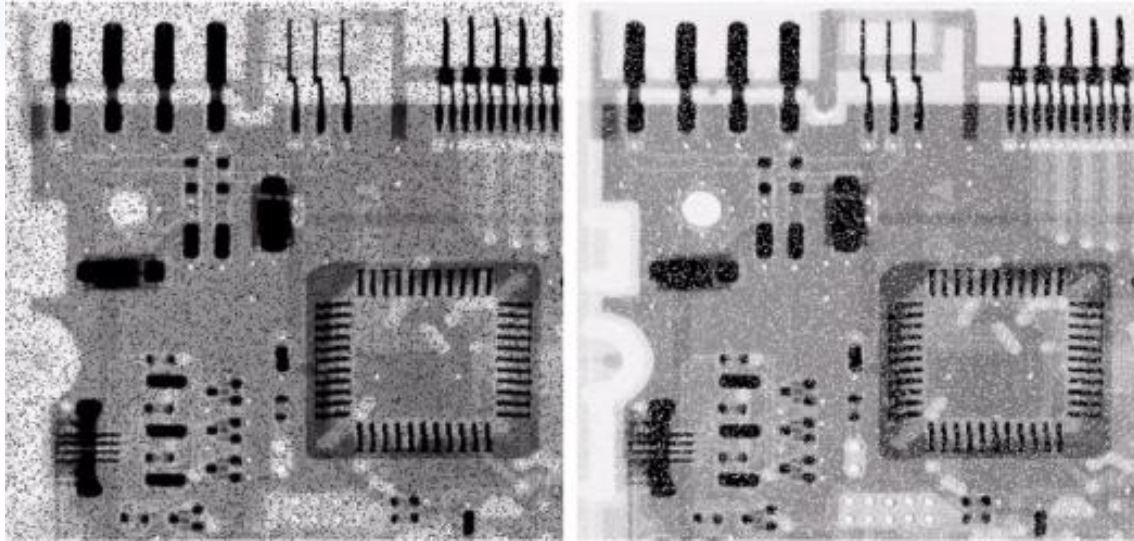


- 如果  $b > a$ ，灰度值  $b$  在图像中将显示为一个亮点(盐)；相反， $a$  的值将显示为一个暗点(胡椒)。
- 若  $p_a$  或  $p_b$  为零，则脉冲噪声称为单极性噪声。
- 若  $p_a$  或  $p_b$  均不为零，尤其是它们近似相等时( $p_a \approx p_b$ )，则称为双极性脉冲噪声，脉冲噪声将类似于随机分布在图像上的胡椒盐粉微粒，又称椒盐噪声。

# Noise models

- 脉冲噪声

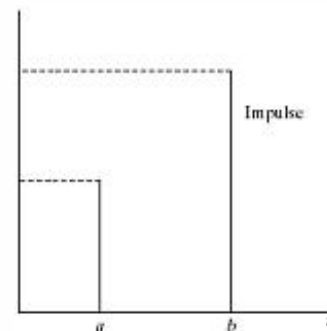
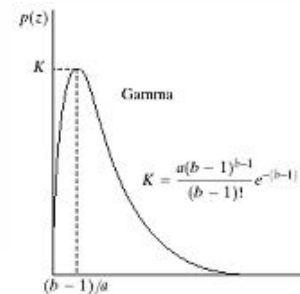
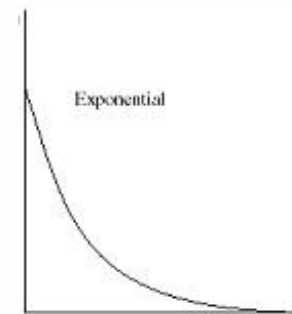
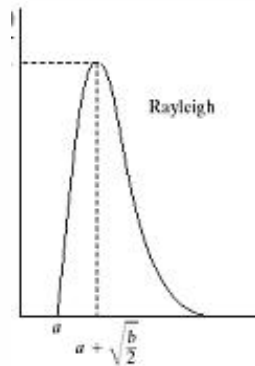
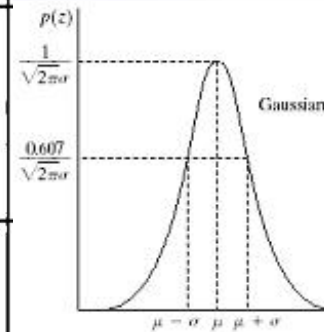
- 噪声脉冲可正可负
- 脉冲干扰通常与图像信号的强度相比较强，在图像数字化过程中常常标定为最大值（纯白或纯黑）
  - 假设 **a** 和 **b** 是饱和值，在数字化图像中，它们等于所允许的最大值和最小值。
- 负脉冲以一个黑点（胡椒点）出现在图像中；正脉冲以一个白点（盐点）出现在图像中。
- 对于一个 **8** 位的图像，意味着 **a=0** (黑), **b=255** (白)。





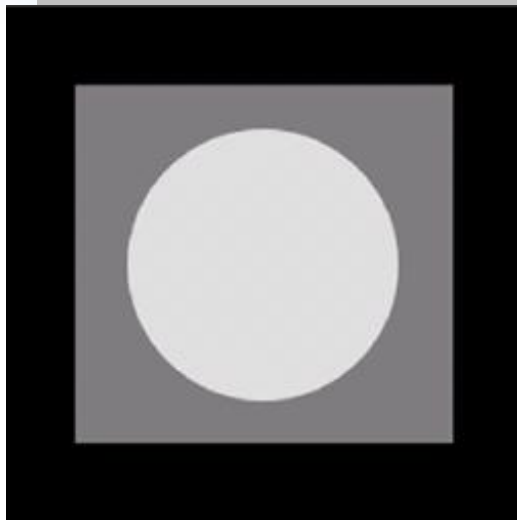
# Noise models

名 称	PDF	均值和方差
均匀	$p(z) = \begin{cases} \frac{1}{b-a} & 0 \leq z \leq b \\ 0 & \text{其他} \end{cases}$	$m = \frac{a+b}{2}, \sigma^2 = \frac{(b-a)^2}{12}$
高斯	$p(z) = \frac{1}{\sqrt{2\pi}b} e^{-(z-a)^2/2b^2}$ $-\infty < z < \infty$	$m = a, \sigma^2 = b^2$
瑞利	$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & z \geq a \\ 0 & z < a \end{cases}$	$m = a + \sqrt{\pi b/4}, \sigma^2 = \frac{b(4-\pi)}{4}$
指数	$p(z) = \begin{cases} ae^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$	$m = \frac{1}{a}, \sigma^2 = \frac{1}{a^2}$
厄兰	$p(z) = \frac{a^b z^{b-1}}{(b-1)!} e^{-az}$ $z \geq 0$	$m = \frac{b}{a}, \sigma^2 = \frac{b}{a^2}$
椒盐	$p(z) = \begin{cases} P_p & z = 0 \text{ (pepper)} \\ P_s & z = 2^n - 1 \text{ (salt)} \\ 1 - (P_p + P_s) & z = k \\ & (0 < k < 2^n - 1) \end{cases}$	$m = (0)P_p + k(1 - P_p - P_s) + (2^n - 1)P_s$ $\sigma^2 = (0-m)^2 P_p + (k-m)^2 (1 - P_p - P_s) + (2^n - 1 - m)^2 P_s$



# Noise models

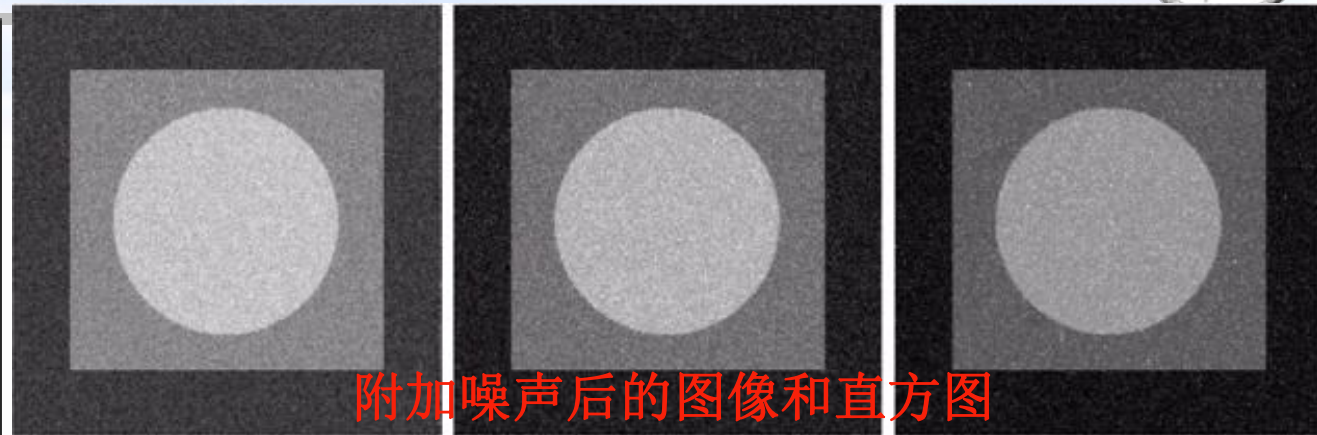
matlab代码



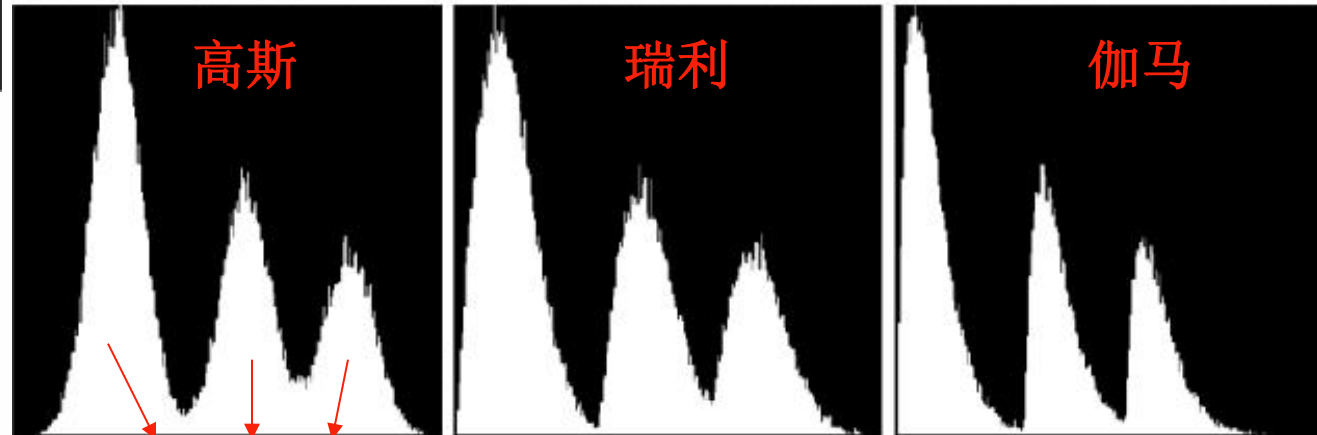
原始图像

由简单、恒定的区域构成，从黑到灰、最后到白仅三个灰度级跨度，非常适合于噪声模型的测量。

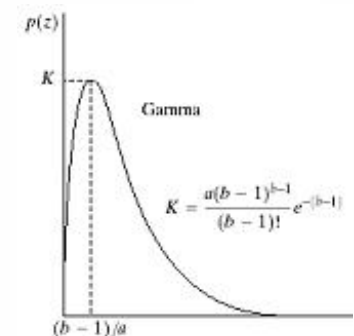
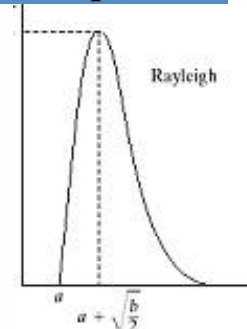
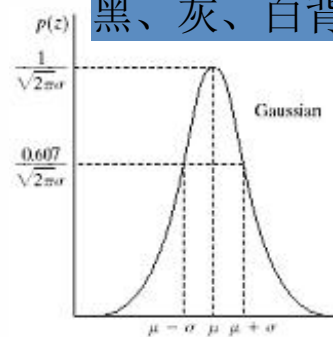
比较噪声图像的直方图和噪声的 pdf，可以看到两者之间的相似性。



附加噪声后的图像和直方图

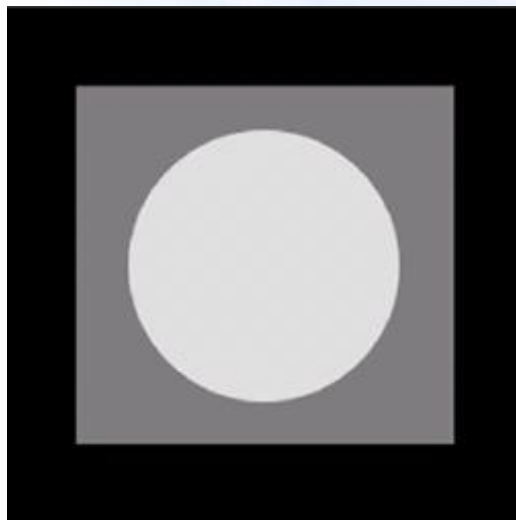


黑、灰、白背景下的噪声 pdf

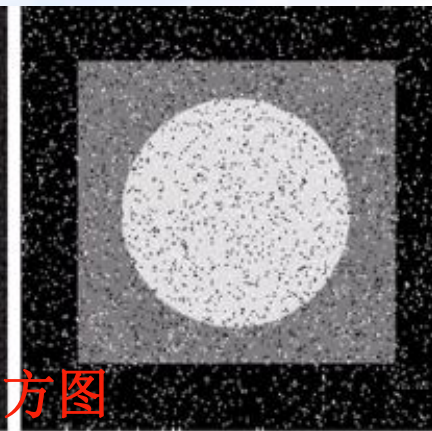
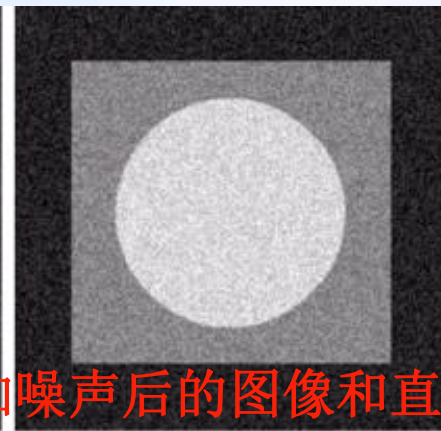
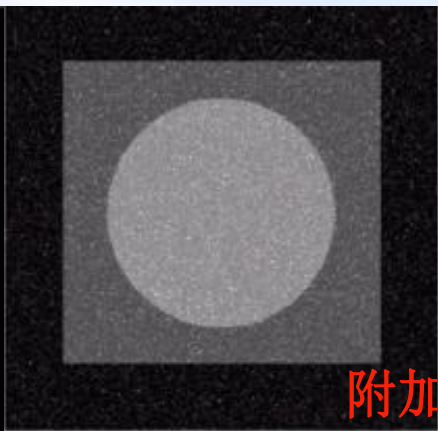


# Noise models

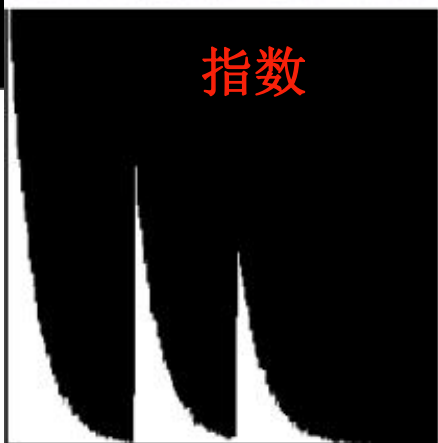
matlab代码



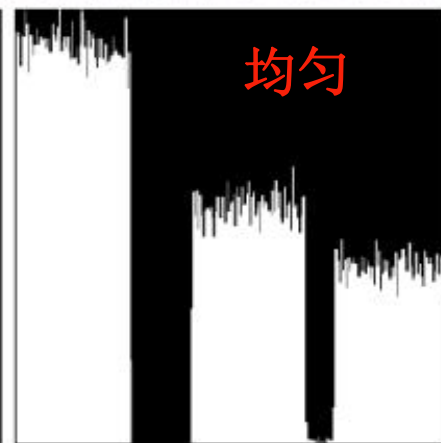
原始图像



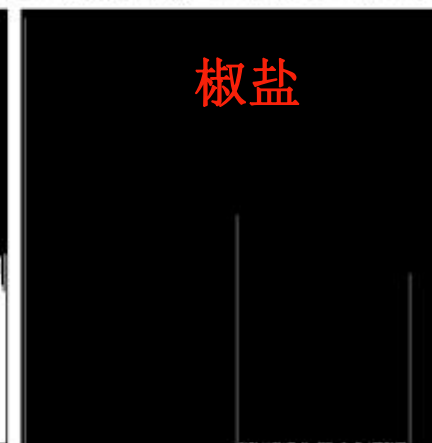
附加噪声后的图像和直方图



指数

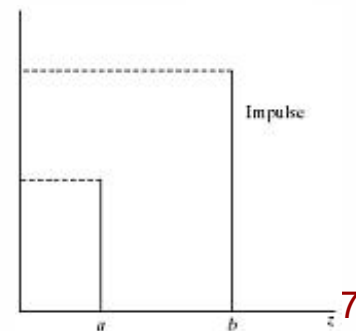
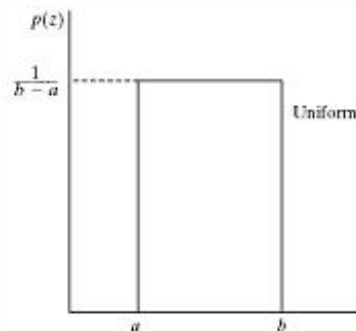
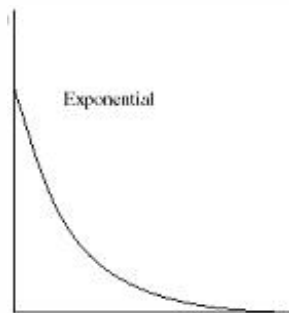


均匀



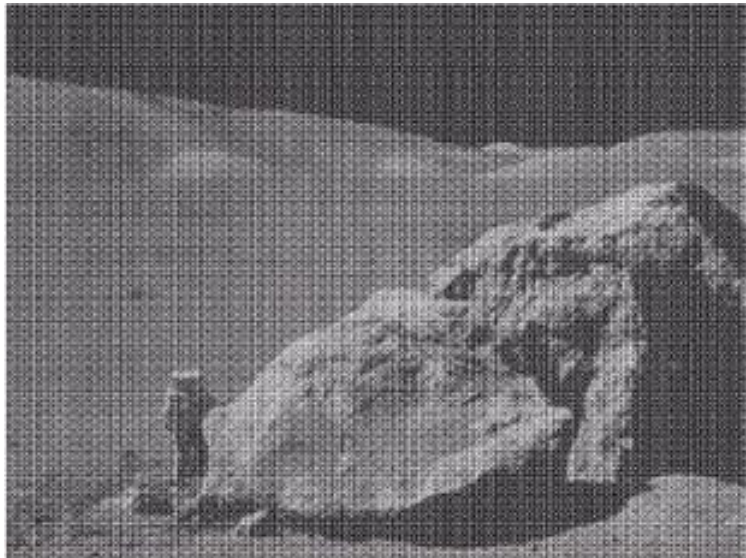
椒盐

虽然前五种图像直方图明显不同，但是图像除了亮度少许变化外，没有显著不同。椒盐噪声是唯一一种能引起明显可视退化的噪声类型。



# Noise models

- 周期噪声
  - 噪声源：在图像获取过程中电力或机电干扰产生的。
  - 一种空间依赖型噪声。
  - 周期噪声可以通过频域滤波显著的减少。
- 举例：被不同频率的正弦噪声污染的图像
  - 纯正弦的 **FT** 是位于正弦波共扼频率处的一对共扼脉冲。
  - 在此特殊例中，不同频率的正弦波 **DFT** 后的脉冲对以近似于圆的形状出现。



被不同频率的正弦噪声污染的图像

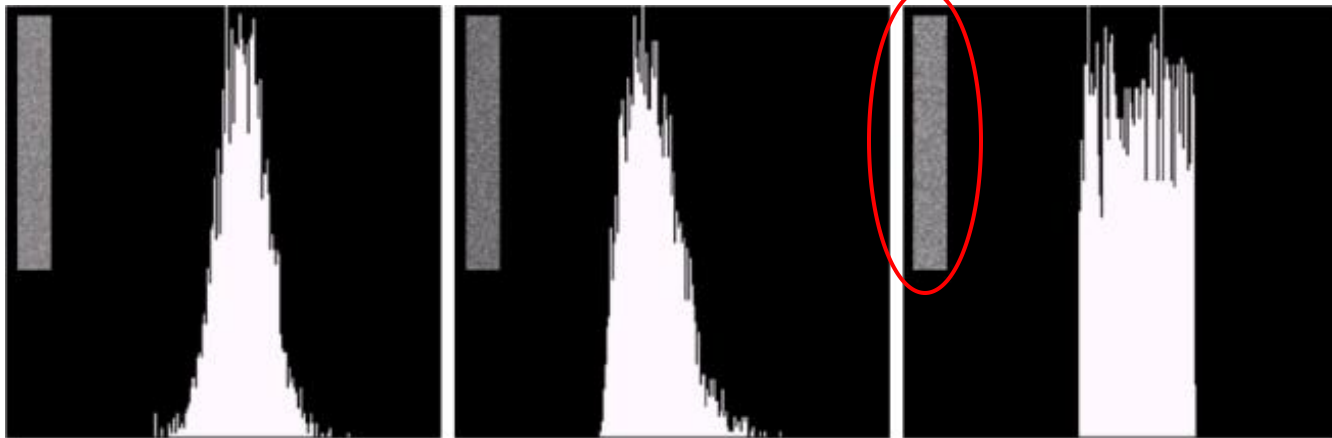


频谱（一个正弦波对应一对共轭脉冲）



# Noise models

- 典型的周期噪声参数是通过检测图像的**傅里叶频谱**来进行估计的。周期噪声趋向于产生频率尖峰，易于判断和检测；对于简单的情况，尽可能直接从图像推断噪声的周期性。
- 从成像系统出发：系统的噪声特性通过截取一组恒定亮度下的“平坦”背景图像来研究。结果图像是一个良好的、典型的系统噪声指示器。
- 从图像本身出发：从相对**恒定**灰度值的一小部分估计噪声pdf的参数。



用子图像计算的高斯、瑞利、均匀噪声直方图

# Noise models

- 一旦 pdf 模型确定了，估计模型参数(均值 $\mu$ 、方差 $\sigma^2$ ) 或  $(a,b)$ 。考虑由  $S$  定义的一个子图像，利用图像带中的数据，估计灰度值的均值和方差，计算参数  $a$  和  $b$ :

$$\mu = \sum_{z_i \in S} z_i p(z_i)$$
$$\sigma^2 = \sum_{z_i \in S} (z_i - \mu)^2 p(z_i)$$

- $z_i$  是  $S$  中像素的灰度值
- $P(z_i)$  表示相应的归一化直方图

- 直方图的形状指出最接近的噪声 pdf 的匹配。
  - 如果其形状近似于高斯，那么由均值和方差就可确定高斯噪声的 pdf。
  - 如果近似于其它噪声，用均值和方差可解出噪声 pdf 中参数  $a$  和  $b$ 。
  - 对于脉冲噪声，采用不同的处理方法。在一个相对恒定的中等灰度区域内估计黑白像素发生的实际概率，即黑白像素尖峰高度。





# Index

---

- Image degradation /restoration model
- Noise models
- **Spatial filtering methods: noise only case**
- Frequency domain filtering methods
- Linear position-invariant degradation
- Degradation function estimation
- Inverse filtering
- Wiener filtering
- Other methods
- Image reconstruction



# Spatial filtering method

- 当噪声是图像中唯一存在的退化时，退化图像的模型如下：

$$\begin{aligned} g(x, y) &= f(x, y) + \eta(x, y) \\ &\quad \updownarrow DFT \\ G(u, v) &= F(u, v) + N(u, v) \end{aligned}$$

- 噪声项  $\eta(x, y)$  或  $N(u, v)$  是未知的，从  $g(x, y)$  或  $G(u, v)$  去除它们比较困难。
  - 对于周期噪声
    - 通常从  $G(u, v)$  的频谱估计  $N(u, v)$
    - 从  $G(u, v)$  中减去  $N(u, v)$  得到原始图像的估计
  - 当仅有加性噪声存在时，可以选择空间滤波方法



# Spatial filtering method

- 下面讨论两类基于空间域处理的噪声减少滤波器：
  - **均值滤波器 Mean Filters**
    - 算术均值滤波器 **Arithmetic Mean Filter**
    - 几何均值滤波器 **Geometric Mean Filter**
    - 谐波均值滤波器 **Harmonic Mean Filter**
    - 逆谐波均值滤波 **Contraharmonic Mean Filter**
  - **统计排序滤波器 Order-statistics Filters**
    - 中值滤波器 **Median Filter**
    - 最大和最小滤波器 **Max and Min Filter**
    - 中点滤波器 **Midpoint Filter**
    - 修正的阿尔发均值滤波器 **Alpha-trimmed Mean Filter**

# Arithmetic Mean Filter

- 算术均值滤波器

- 令  $S_{xy}$  表示中心在  $(x,y)$  的点，尺寸为  $m \times n$  的矩形子图像窗口的坐标集。
- 算术均值滤波的过程就是计算由  $S_{xy}$  定义的区域中退化图像  $g(x,y)$  的均值：

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

- 均值简单的平滑了一幅图像的局部变化，在模糊了图像的同时，减少了噪声。
- 算术均值滤波器可以用系数为  $1/mn$  的卷积模板实现。

# Arithmetic Mean Filter

- 几何均值滤波器

- 每个被复原像素由子图像窗口中像素点的乘积结果求  $1/mn$  次幂给出

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

- 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。
- 几何均值滤波过程中丢失更少的图像细节。

# Harmonic Mean Filter

- 谐波均值滤波器

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

- 对“盐”噪声效果更好，不适用于“胡椒”噪声。
- 也善于处理类似于高斯噪声那样的图像噪声。



# Contrahamonic Mean Filter

- 逆谐波均值滤波器

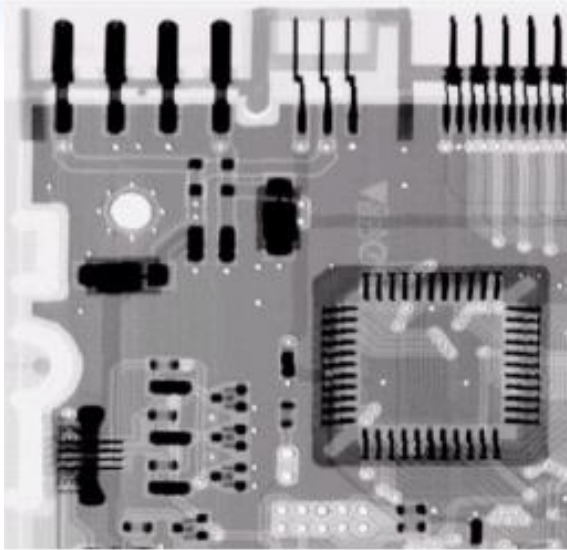
$$\hat{f}(x, y) = \frac{\sum_{(s, t) \in S_{xy}} g(s, t)^{Q+1}}{\sum_{(s, t) \in S_{xy}} g(s, t)^Q}$$

- $Q$  为滤波器的阶数
- 逆谐波均值滤波器适合减少或消除椒盐噪声影响
  - 当  $Q > 0$ ，用于消除“胡椒”噪声
  - 当  $Q < 0$ ，用于消除“盐”噪声
  - 但不能同时消除这两种噪声
- 当  $Q = 0$ ，逆谐波均值滤波器变化为算术均值滤波器
- 当  $Q = -1$ ，逆谐波均值滤波器变化为谐波均值滤波器

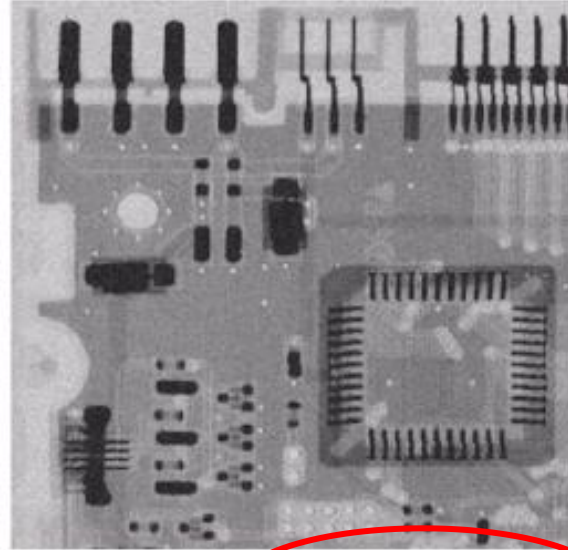
# Mean Filters: applications

matlab代码

电路板的X  
射线图像

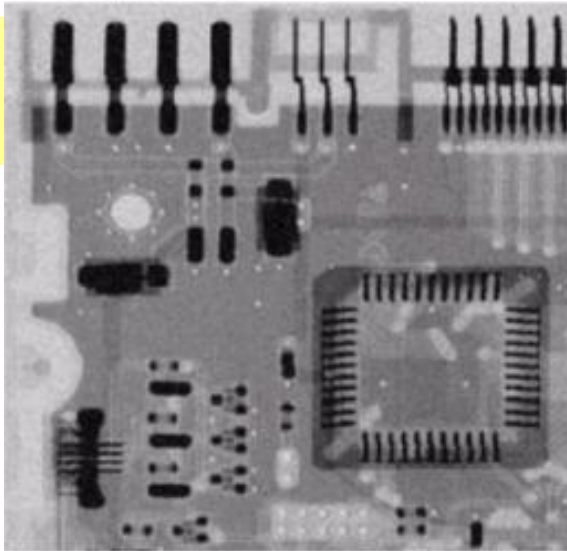


附加高斯噪声  
污染的图像

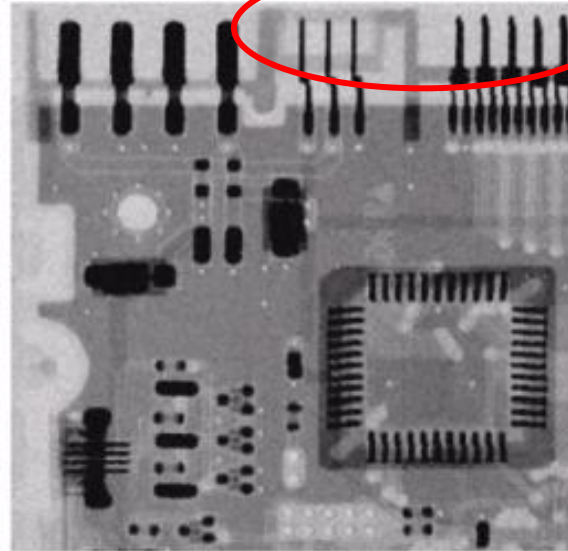


( $\mu=0, \sigma=20$ )

3×3算术均值  
滤波后的结果



3×3几何均值  
滤波后的结果

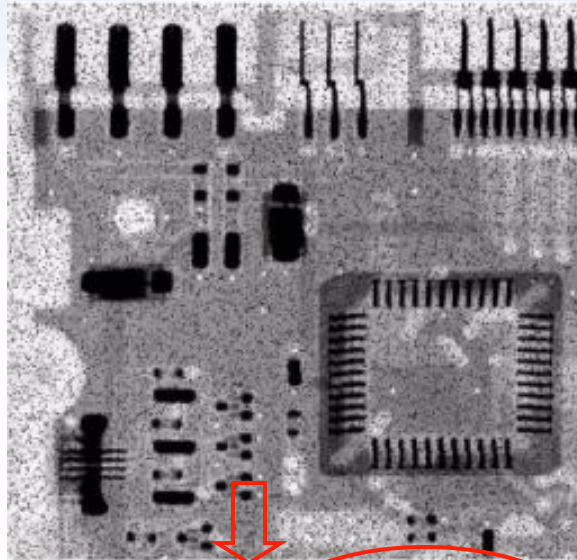


保持了更多的  
图像细节，模  
糊较轻，例  
如图像顶端连  
接片更清晰。

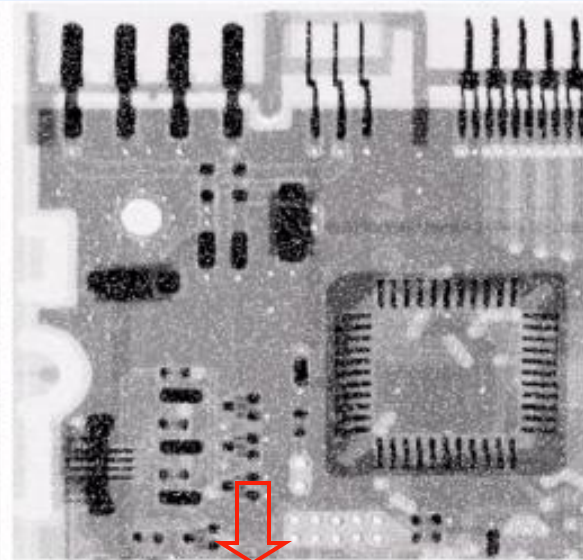
# Mean Filters: applications

matlab代码

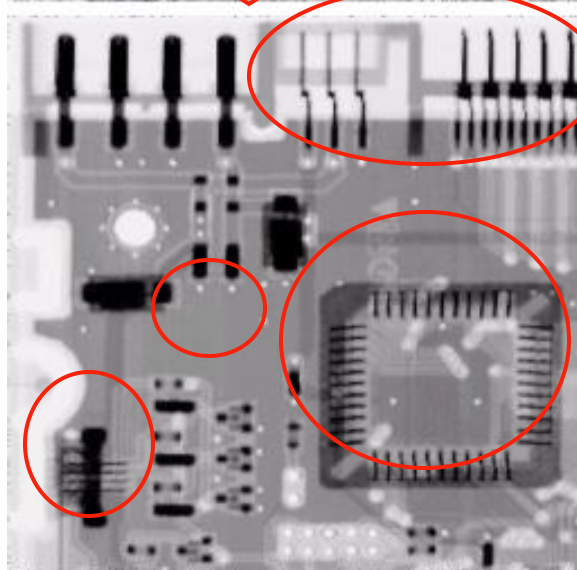
以0.1的概率被“胡椒”噪声污染了的图像



以0.1的概率被“盐”噪声污染了的图像

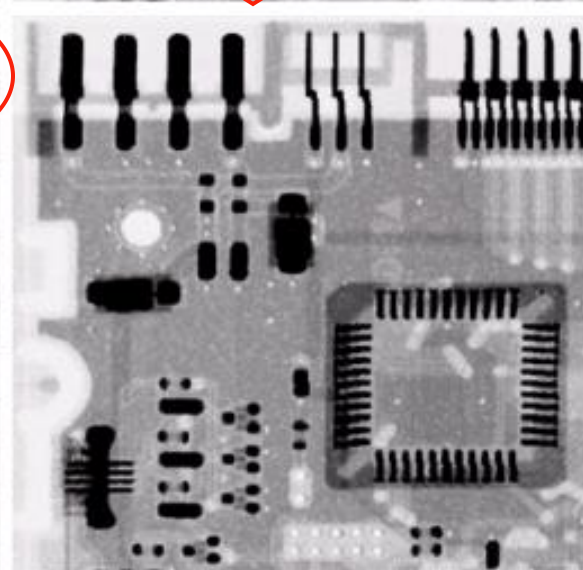


$3 \times 3$ 大小， $Q=1.5$ 的逆谐波滤波器滤波结果



正阶滤波器使暗区变得模糊，亮背景较为清晰。

$3 \times 3$ 大小， $Q=-1.5$ 对上图的滤波结果



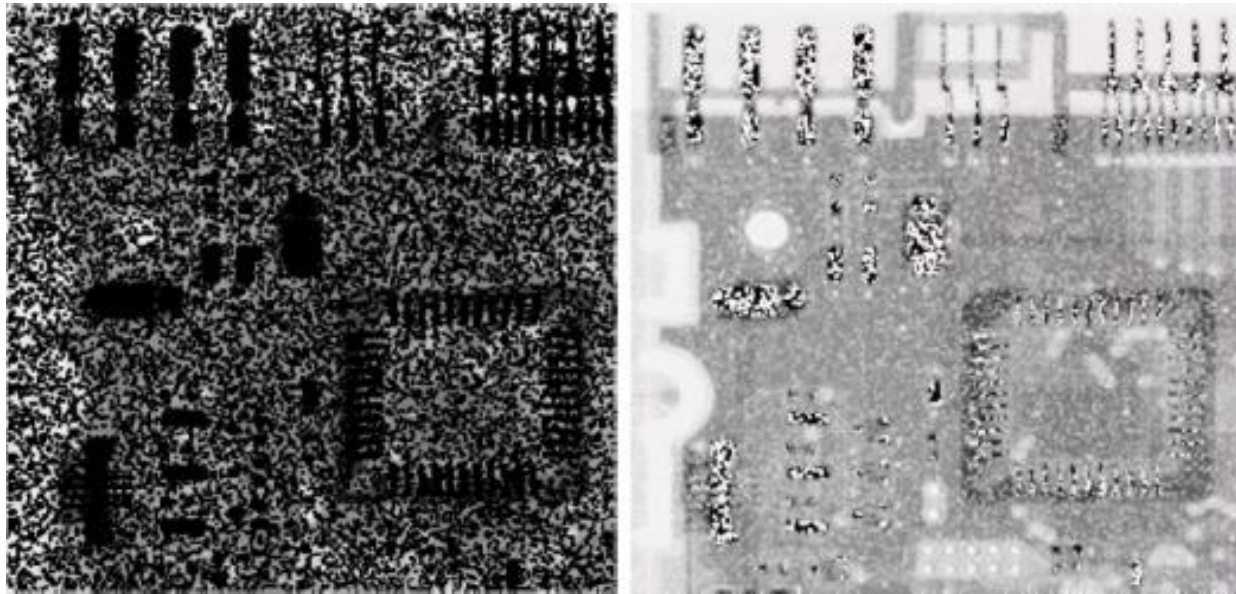
负阶滤波器情况相反，暗区清晰，亮背景模糊。

(两种滤波器都有很好的去除噪声效果)



# Mean Filters: applications

- 算术和几何均值滤波器 (尤其后者) 更适合处理高斯或均匀等随机噪声。
- 谐波均值滤波器更适合处理脉冲噪声, 但是要注意: 它处理噪声之前要知道噪声是暗噪声(胡椒)还是亮噪声(盐), 以决定  $Q$  的符号, 否则会引起灾难性后果。



在逆谐波均值滤波器中错误选择符号的结果

$3 \times 3$ 大小,  $Q = -1.5$ 的逆谐波滤波器滤波处理“胡椒”噪声结果

$3 \times 3$ 大小,  $Q = 1.5$ 的逆谐波滤波器滤波处理“盐”噪声结果



# Order-Statistic filters

- 统计排序滤波器

- 统计排序滤波器是空间域滤波器，其响应基于由滤波器包围的图像区域中像素点的**排序(最大、最小、中间或其它)**。
- 滤波器任何点的响应由排序结果决定。

- 中值滤波器

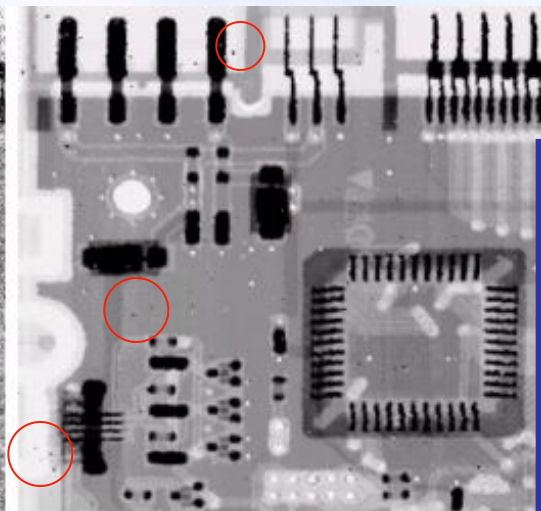
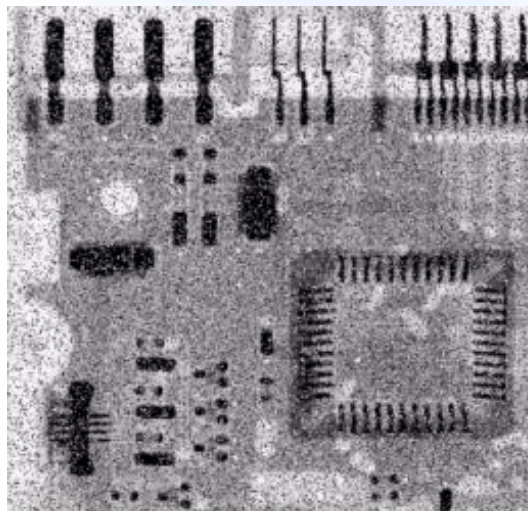
- 用相邻区域像素灰度的中值代替该点的像素值

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

- 对于多种随机噪声去噪能力好，在相同尺寸下比线性平滑滤波器引起的模糊少。
- 对单极性/双极性脉冲噪声尤其有效。

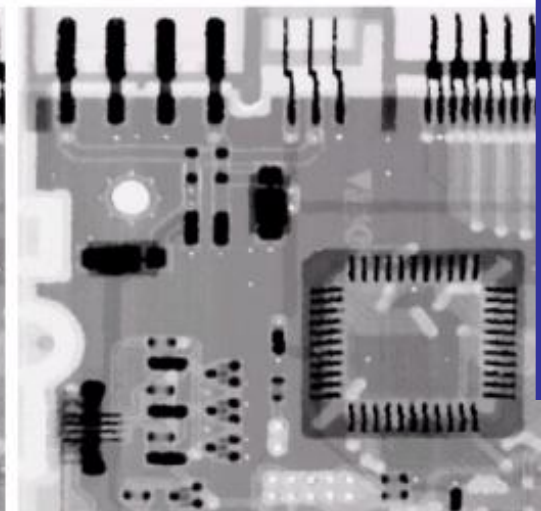
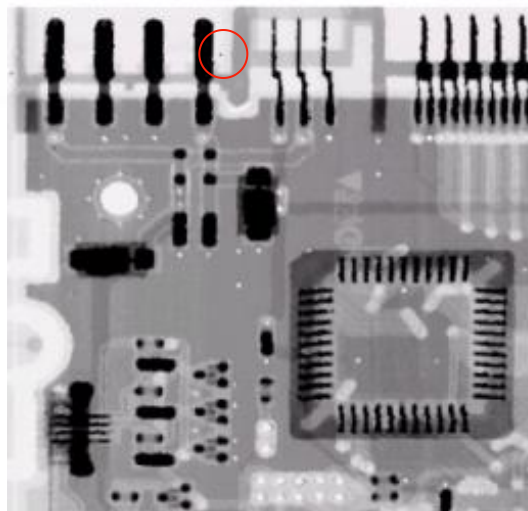
# Order-Statistic filters

matlab代码



左上图：被概率为 $P_a=P_b=0.1$ 的脉冲噪声干扰的电路图像。

右上图：用规格 $3 \times 3$ 的中值滤波器处理后的结果，图像效果明显改善，但是仍然存在一些噪声点。



左下图：二次中值滤波后的结果，仅剩余非常少数的噪声点

右下图：三次中值滤波的结果，噪声点全部消除。

**注意：**多次重复使用中值滤波可能会使图像模糊，应尽可能保持少的处理次数。



# Max and Min filters

- 最大值和最小值滤波

- 最大值滤波器：排序序列中的最后(最大、最亮)一个灰度数值。

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

- 在寻找图像中最亮点时非常有用。
  - 最大值滤波适合消除“胡椒”噪声，“胡椒”噪声是暗像素，可以通过选择图像邻域的最大值代替当前点像素值来消除。
  - 最小值滤波器：排序序列中的起始(最暗)位置的数值。

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

- 在寻找图像中最暗点时非常有用。
  - 相应的，最小值滤波适合处理“盐”噪声

# Midpoint filters

- 中点滤波器

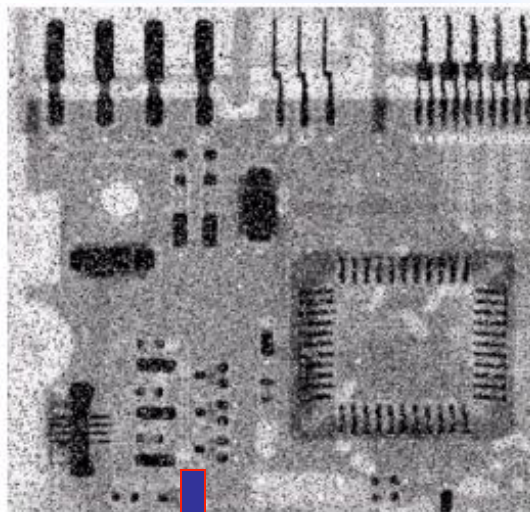
- 在滤波器模板内计算灰度级最大值和最小值的中点

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

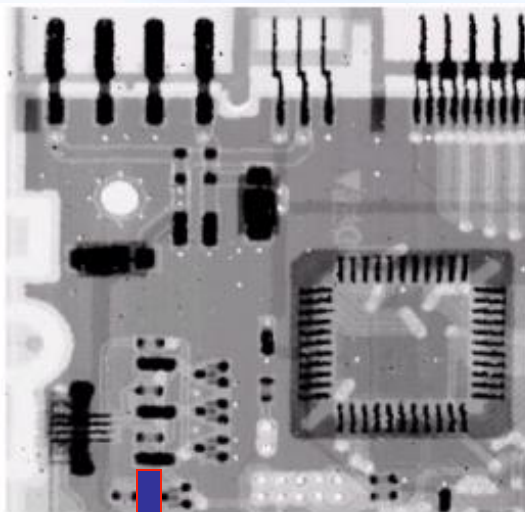
- 结合了顺序统计和求平均，对于高斯和均匀随机分布这类的噪声有很好的效果。

# Order-Statistic filters

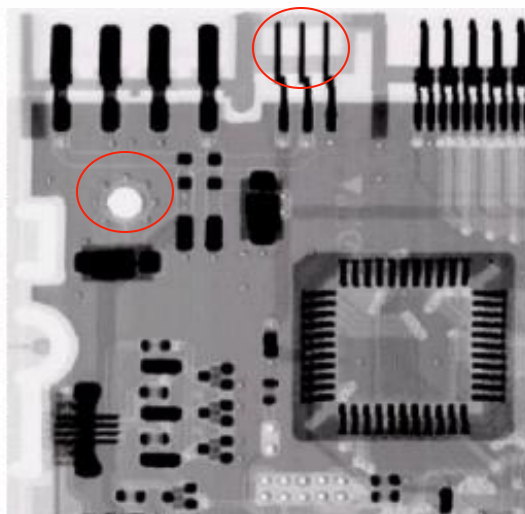
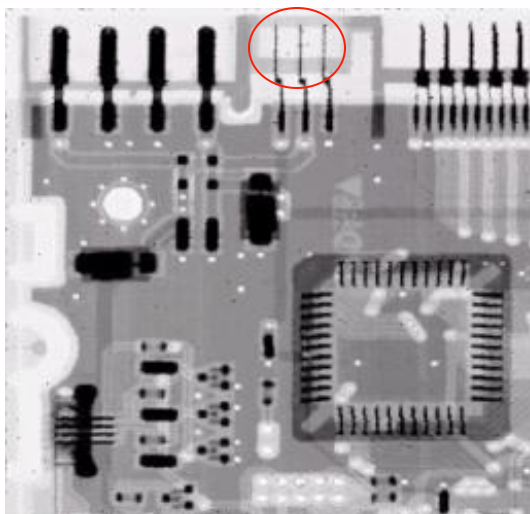
matlab代码



最大值滤波



最小值滤波



左上图：被概率为 $P_a=P_b=0.1$ 的脉冲噪声干扰的电路图像

右上图：用规格 $3 \times 3$ 的中值滤波器处理后的结果，图像效果明显改善，但是仍然存在一些噪声点。

左下图：最大值滤波适合处理“胡椒”噪声，同时也从黑色物体边缘移走了一些黑色像素。

右下图：在这种特殊情况下，最小值滤波优于最大值滤波，但同时也从亮物体边缘移走了一些白色像素，亮色物体变小，暗色物体变大。

# Alpha-trimmed mean filter

- 修正的阿尔法均值滤波器

- 假设在  $S_{xy}$  的邻域内去掉最高灰度值的  $d/2$  个像素和最低灰度值的  $d/2$  个像素，用  $g_r(s,t)$  代表剩余的  $(mn - d)$  个像素。由剩余像素点的均值形成修正后的阿尔法均值滤波器：

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

— 其中

- $0 \leq d \leq (mn-1)$  任意值
- $d=0 \rightarrow$  算术均值滤波器
- $d=mn-1 \rightarrow$  中值滤波器
- $d$  取其它值时，修正后阿尔法均值滤波器适用于包含多种噪声情况下的图像复原，例如高斯噪声和椒盐噪声混合的情况。



# Order statistic filters: applications

由加性均匀噪声  
声污染的图像

$$\mu = 0, \sigma^2 = 800$$

5×5算术均值  
滤波处理后的  
结果

由于存在脉冲  
噪声，滤波效  
果不好

5×5中值滤波  
处理后的结果

中值和修正的阿尔法均值  
滤波器滤波效果好，在消  
除噪声方面，阿尔法均值  
滤波器滤波更好些。

被“椒盐”噪声进  
一步叠加后的图像

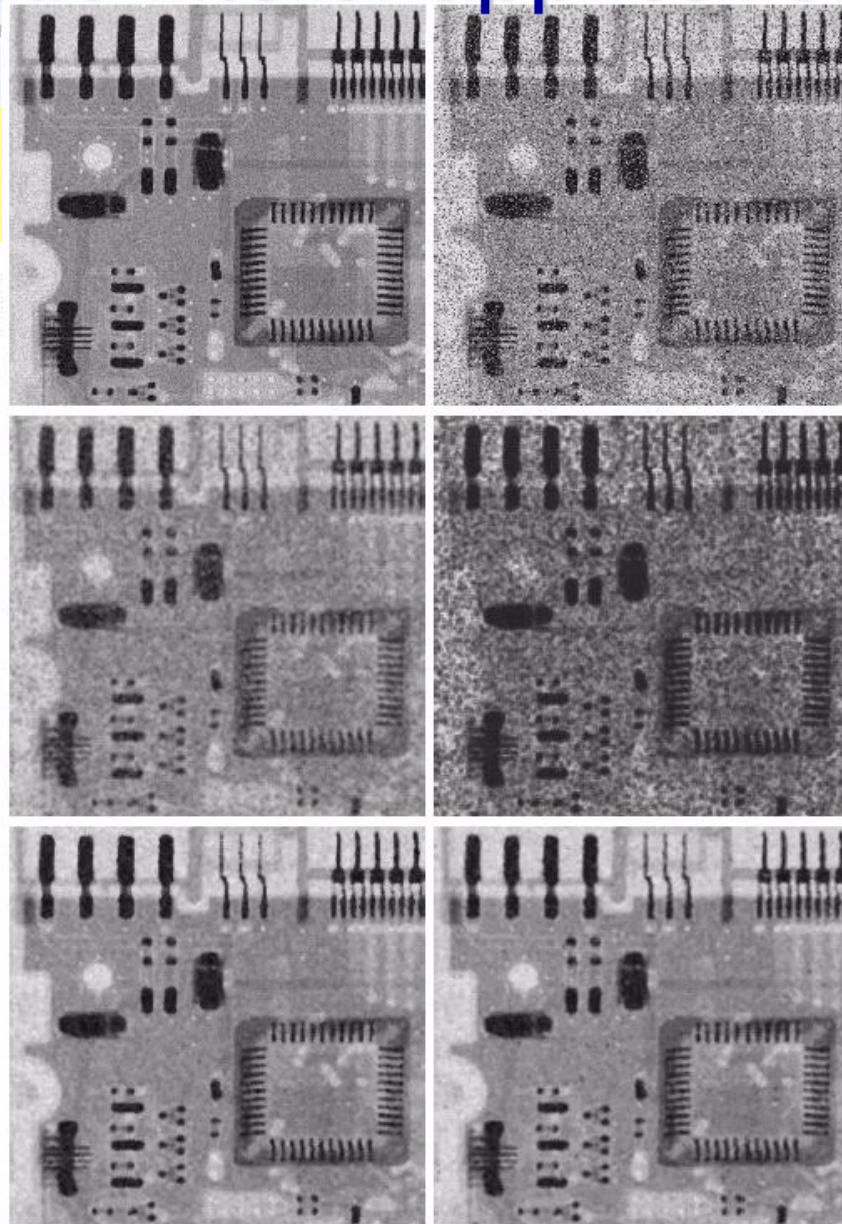
$$(P_a = P_b = 0.1)$$

matlab代码

5×5几何均值  
滤波处理后的  
结果

由于存在脉冲  
噪声，滤波效  
果不好

5×5， $d=5$ 的  
修正后的阿尔  
法均值滤波处  
理后的结果





# Adaptive filters (self study)

- 前面讨论的滤波器是固定的，一旦确定就不随图像内容而变化。自适应滤波器的行为变化基于图像的统计特性。
- 考虑基于由  $m \times n$  矩形窗口  $S_{xy}$  定义的区域图像的统计特性
  - 均值：区域中的平均灰度值的量度；
  - 方差：区域平均对比度的量度；
- 提高滤波性能的代价是滤波器的复杂度
- 书上介绍的自适应滤波器包括：
  - 自适应局部降低噪声滤波器
  - 自适应中值滤波器



# Index

---

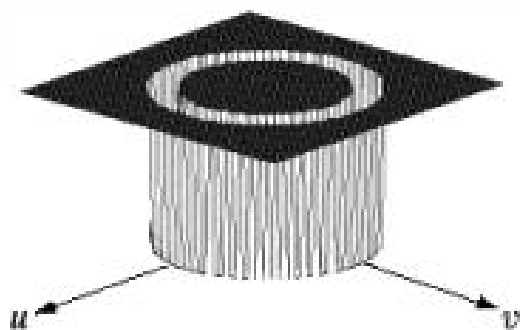
- Image degradation /restoration model
- Noise models
- Spatial filtering methods: noise only case
- **Frequency domain filtering methods**
- Linear position-invariant degradation
- Degradation function estimation
- Inverse filtering
- Wiener filtering
- Other methods
- Image reconstruction
- Image reconstruction

# Frequency domain filtering methods(自学)

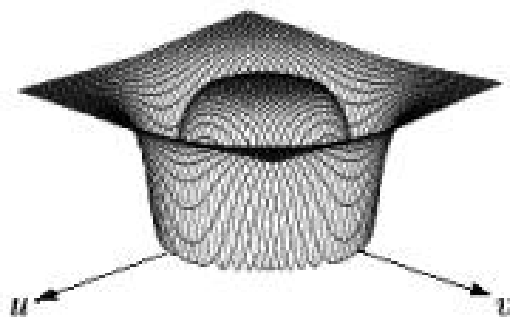


- 带阻、带通和陷波滤波器，它们用于削减和消除周期性噪声。

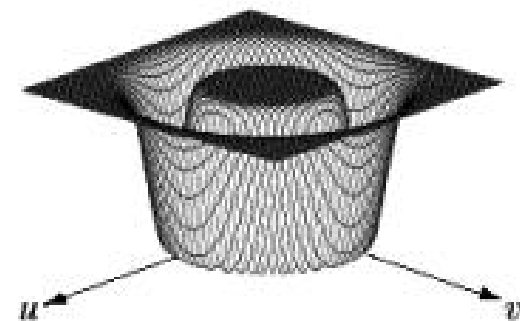
## 三种带阻滤波器透视图



理想带阻滤波器



1阶巴特沃思带阻滤波器



高斯带阻滤波器



# Index

---

- Image degradation /restoration model
- Noise models
- Spatial filtering methods: noise only case
- Frequency domain filtering methods
- **Linear position-invariant degradation**
- Degradation function estimation
- Inverse filtering
- Wiener filtering
- Other methods
- Image reconstruction



# Linear system

- 线性系统  $H$  满足:

$$H[af_1(x, y) + bf_2(x, y)] = aH[f_1(x, y)] + bH[f_2(x, y)]$$

- $a$ 、 $b$  为比例常数,  $f_1(x, y)$  和  $f_2(x, y)$  是任意两幅输入图像。
- **叠加性:**  $a=b=1$ , 若  $H$  为线性算子, 两个输入之和的响应等于两个响应之和。

$$H[f_1(x, y) + f_2(x, y)] = H[f_1(x, y)] + H[f_2(x, y)]$$

- **齐次性:**  $f_2(x, y) = 0$ , 与常数相乘的输入的响应等于该输入响应乘以相同的常数。

$$H[af_1(x, y)] = aH[f_1(x, y)]$$



# Linear position invariant degradation

- 位移不变系统（或空间不变系统）
  - 对于任意 图像 $f(x,y)$ ，位置 $\alpha$ 和 $\beta$ ，有

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



$$H[f(x-\alpha, y-\beta)] = g(x-\alpha, y-\beta)$$

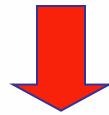
- 图像中任一点的响应只取决于该点的输入值，而与该点的位置无关。

# Linear position invariant degradation

- 用连续单位脉冲函数定义在  $(x,y)$  位置的  $f(x,y)$ :

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) \delta(x - \alpha, y - \beta) d\alpha d\beta$$

$H$ 是线性算子



$$\begin{aligned} g(x, y) &= H[f(x, y)] = H\left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) \delta(x - \alpha, y - \beta) d\alpha d\beta\right] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H[f(\alpha, \beta) \delta(x - \alpha, y - \beta)] d\alpha d\beta \end{aligned}$$

$f(\alpha, \beta)$  独立于  $(x, y)$ , 使用均匀性



$$\begin{aligned} g(x, y) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) H[\delta(x - \alpha, y - \beta)] d\alpha d\beta \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta \end{aligned}$$

又称为点  
扩散函数  
(PSF)

对于线性系统, 任意输入的响应是脉冲响应和输入函数的简单卷积





# Linear position invariant degradation

- 考虑加性噪声  $\eta(x,y)$  的线性位置不变退化模型

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta + \eta(x, y)$$



卷积形式

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



频域形式

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

- 具有加性噪声的线性空间不变退化系统，可以在空间域被模型化为退化函数(点扩散函数)与图像的卷积，并加上噪声。完成这类线性图像恢复功能的滤波器又被称为“图像去卷积滤波器”
- 在频域内表示为退化函数与图像的乘积，加上噪声。



# Index

---

- **Image degradation /restoration model**
- **Noise models**
- **Spatial filtering methods: noise only case**
- **Frequency domain filtering methods**
- **Linear position-invariant degradation**
- **Degradation function estimation**
- Inverse filtering
- Wiener filtering
- Other methods
- Image reconstruction



# Estimation by observation

- 估计退化函数  $h(x,y)$  或  $H(u,v)$  的方法：
  - 观察法
  - 实验法
  - 数学建模法
- 图像观察估计法
  - 没有  $H$  的知识，用收集图像自身的信息进行估计；
  - 利用位置不变性质，用**强信号内容区 (受噪声影响小，可以忽略噪声影响)**的子图像估计整幅图像的退化函数；
  - $g_s(x,y)$  为观察**子图像**， $\hat{f}_s(x,y)$  为构建的估计**子图像**，**假设噪声可以忽略**，并且位置不变，则：

$$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)} \Rightarrow \text{完全函数 } H(u,v)$$

# Estimation by experimentation

## • 试验估计法

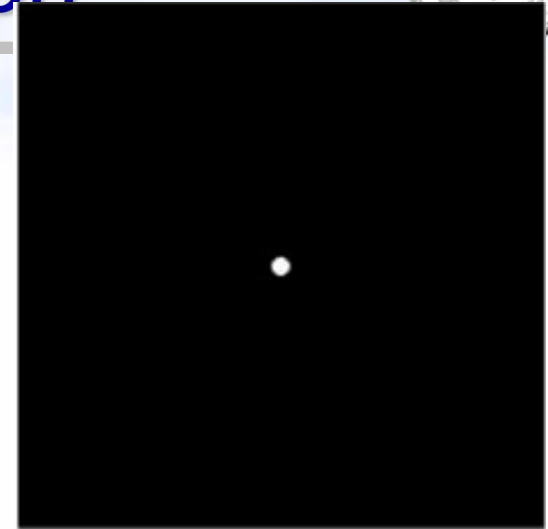
- 使用与获取退化函数的设备相似的装置，将与退化图像相类似的图像退化到尽可能接近需要复原的图像。
- 利用相同的系统设置，由一个脉冲成像(小亮点)得到退化系统的脉冲响应。线性位置不变系统完全可由它的脉冲响应来描述。
- 一个脉冲可由明亮的点来模拟，并尽可能亮以便可以忽略噪声干扰。
  - 输入：  $f(x,y)=\delta(x,y)$ ,  $DFT\{\delta(x,y)\}=1$ ;
  - 输出：  $G(u,v)=H(u,v)F(u,v)=H(u,v)$  或包括一个常数  $A$ 。

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

观察图像的 FT

常量，表示脉冲强度

很多情况下难以得到与原始成像系统完全一致的条件



一个亮脉冲(放大显示)



退化的脉冲

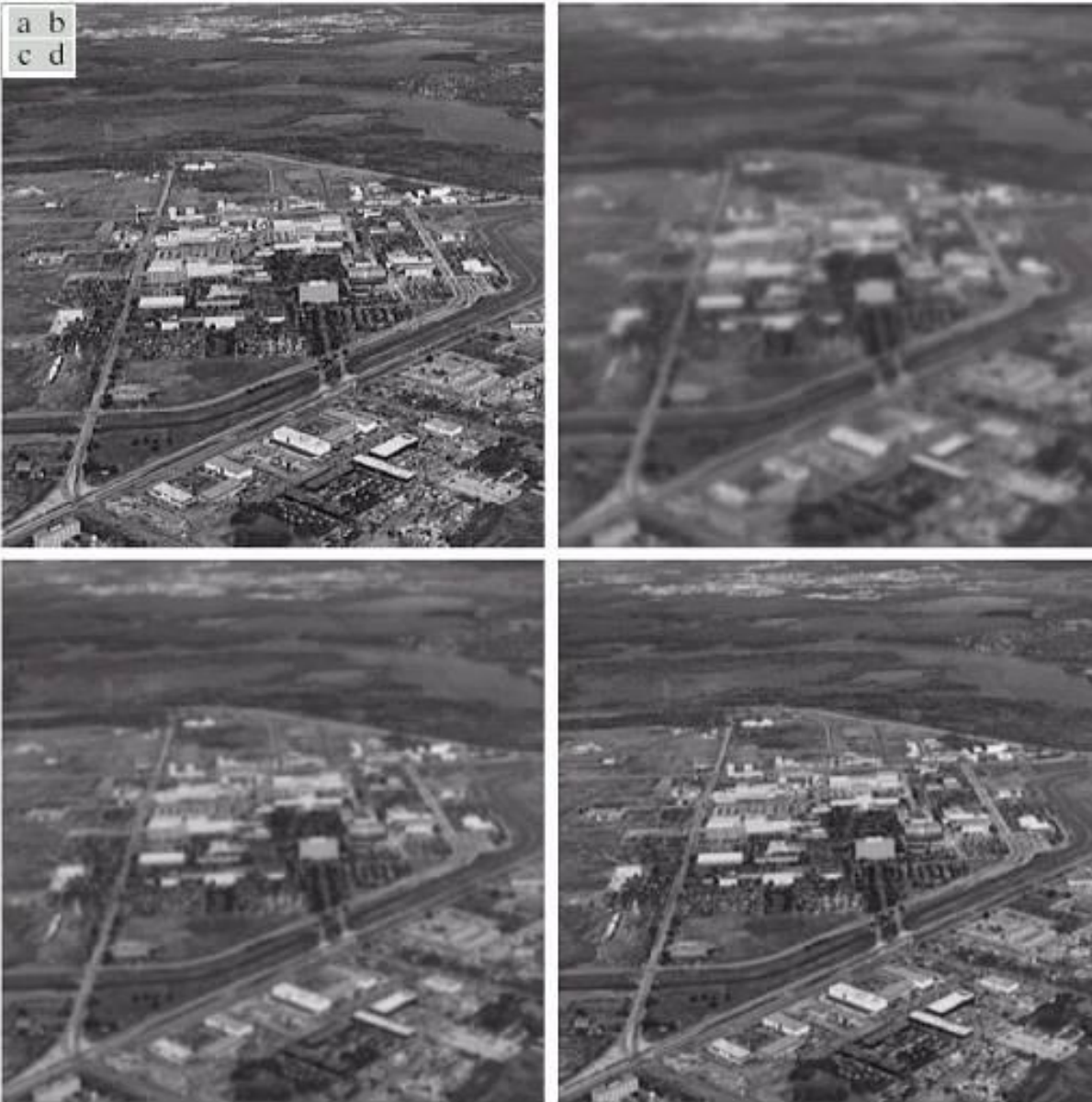


# Estimation by modeling

- 模型估计法：
  - 方法一：把引起退化的环境因素考虑在内
  - 例如，Hufnagel 和 Stanley 提出的退化模型是基于大气湍流的物理特性：
$$H(u, v) = e^{-k(u^2 + v^2)^{5/6}}$$
  - k 为常数，它与湍流的特性有关，大的 k 值表示剧烈湍流。
  - 此模型类似于GLPF（除了 5/6 指数外），GLPF 可用来淡化模型，均匀模糊。



# Estimation by modeling



$$H(u, v) = e^{-k(u^2 + v^2)^{5/6}}$$

大气湍流模型的解释

(a) 可以忽略的湍流；

(b) 剧烈湍流  $k=0.0025$ ；

(c) 中等湍流  $k=0.001$ ；

(d) 轻微湍流  $k=0.00025$ ；





# Estimation by modeling

- 模型估计法：
  - 方法二：从基本原理开始推导出一个数学模型
  - 用实例来说明：图像获取时被图像和传感器之间的均匀线性运动模糊。
  - 图像  $f(x,y)$  进行平面运动， $x_0(t)$  和  $y_0(t)$  分别是  $x$  和  $y$  方向上相应的随时间变化的运动参数。
  - 记录介质任意点的曝光总数是通过对时间间隔  $T$  内瞬时曝光数的积分得到的。
    - 若  $T$  足够短，那么光学成像过程不受图像运动干扰的影响，图像非常完美。
    - 若  $T$  较长，那么模糊图像为：

$$g(x, y) = \int_0^T f(x - x_0(t), y - y_0(t)) dt$$

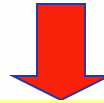
# Estimation by modeling

傅里叶变换

$$G(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{-j2\pi(ux+vy)} dx dy$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ \int_0^T f[x - x_0(t), y - y_0(t)] dt \right] e^{-j2\pi(ux+vy)} dx dy$$

改变积分顺序



$$G(u, v) = \int_0^T \left[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f[x - x_0(t), y - y_0(t)] e^{-j2\pi(ux+vy)} dx dy \right] dt$$

变量置换

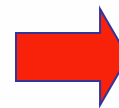


$F(u, v)$  与时间无关

$$G(u, v) = \int_0^T F(u, v) e^{-j2\pi[ux_0(t)+vy_0(t)]} dt = F(u, v) \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]} dt$$



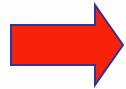
$$H(u, v) = \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]} dt$$



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

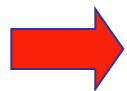
# Estimation by modeling

- 假设图像只在  $x$  方向以给定速率做匀速直线运动,  $x_0(t) = at/T$ ,  $t=T$  时图像运动总距离为  $a$ , 令  $y_0(t)=0$ 。则得到退化函数  $H(u,v)$ :



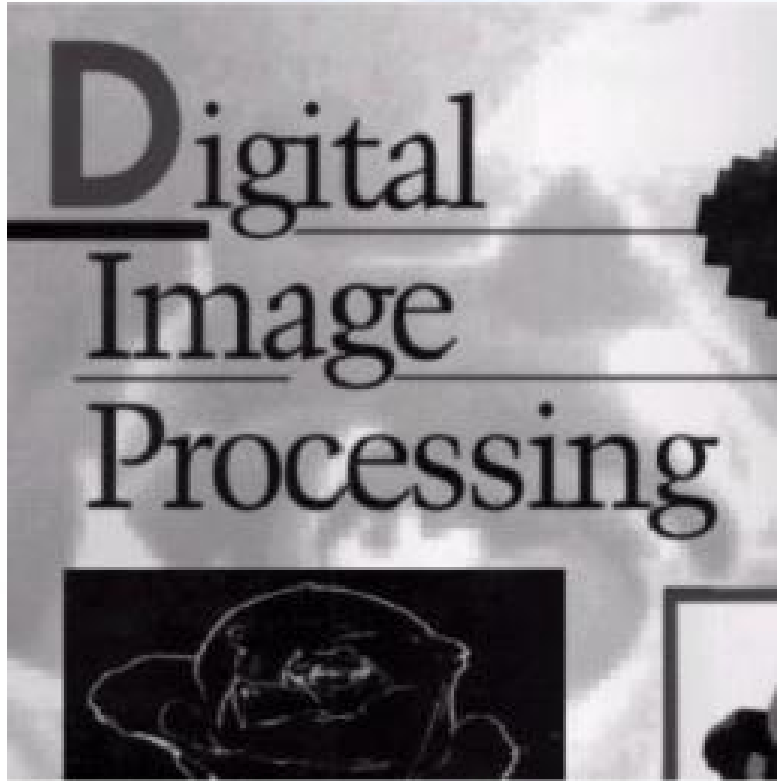
$$\begin{aligned} H(u, v) &= \int_0^T e^{-j2\pi ux_0(t)} dt = \int_0^T e^{-j2\pi uat/T} dt \\ &= \frac{T}{\pi ua} \sin(\pi ua) e^{-j\pi ua} \end{aligned}$$

- 假设图像在  $y$  方向也以给定速率做匀速直线运动  $y_0(t) = bt/T$ , 则退化函数  $H(u,v)$  变为:



$$H(u, v) = \frac{T}{\pi(ua + vb)} \sin[\pi(ua + vb)] e^{-j\pi(ua + vb)}$$

# Estimation by modeling



matlab代码

- 由于运动而引起 的图像模糊
  - 右图是左图经过均匀线性运动模糊的图像，它用原图像的傅里叶变换乘以上述的  $H(u,v)$ ，再求反变换获得。
  - 从模糊图像获得复原图像比较困难，尤其是当退化图像存在噪声的时候。



# Index

---

- **Image degradation /restoration model**
- **Noise models**
- **Spatial filtering methods: noise only case**
- **Frequency domain filtering methods**
- **Linear position-invariant degradation**
- **Degradation function estimation**
- **Inverse filtering**
- **Wiener filtering**
- **Other methods**
- **Image reconstruction**



# Inverse filtering

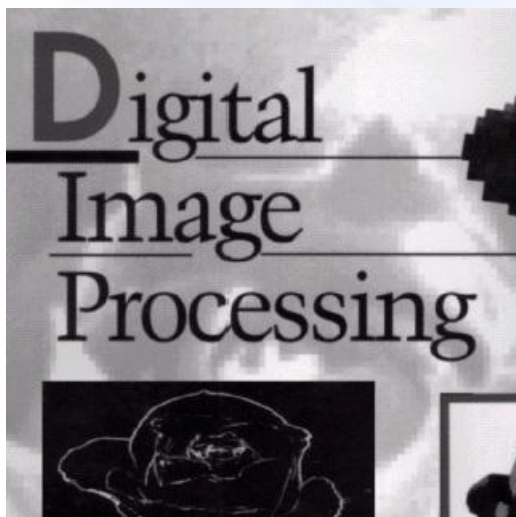
- 逆滤波：用退化函数除退化图像的傅里叶变换，得到原始图像的傅里叶变换的估计  $\hat{F}(u, v)$ ：

$$\begin{aligned}\hat{F}(u, v) &= \frac{G(u, v)}{H(u, v)} = \frac{F(u, v)H(u, v) + N(u, v)}{H(u, v)} \\ &= F(u, v) + \frac{N(u, v)}{H(u, v)}\end{aligned}$$

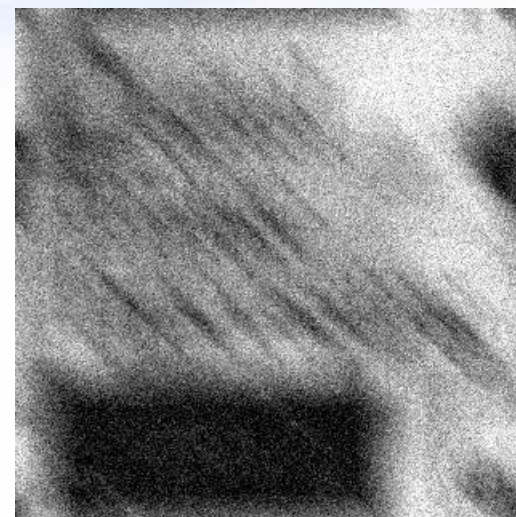
- 从此式可知，即使知道退化函数  $H$ ，也不能准确的复原原始图像  $F(u, v)$ ，因为：
  - $N(u, v)$  是一个随机函数，傅里叶变换未知，所以不能准确的恢复原始图像。
  - 更糟的是若  $H(u, v)$  接近 0 值或者非常小时， $N(u, v)/H(u, v)$  会影响  $\hat{F}(u, v)$  的估计值。
  - 因此，直接使用逆滤波时，去噪性能较差

# Inverse filtering

matlab代码

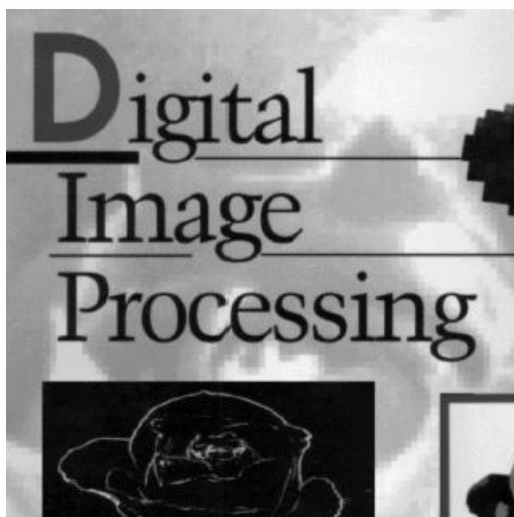


运动模糊图像

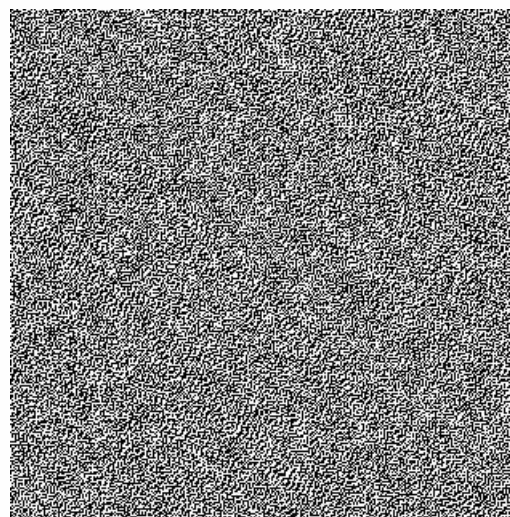


运动模糊+噪声图像

逆滤波结果1:  
没有噪声



逆滤波结果2:  
含有噪声





# Inverse filtering

- 解决退化  $H(u,v)$  为 0 或者很小值的途径：限制滤波频率使其尽量接近原点值。
  - $H(0,0)$  等于  $h(x,y)$  的均值，而且常常是  $H(u,v)$  在频域的最大值。频率越接近原点， $H(u,v)$  为 0 和很小值的几率就越小。
  - 举例说明限制频率对滤波结果的影响
    - 使用精确退化函数

$$H(u, v) = e^{-k \left[ (u - M/2)^2 + (v - N/2)^2 \right]^{5/6}}$$

- $k=0.0025$ ，常数  $M/2$  和  $N/2$  是偏移量，可以中心化函数，与傅立叶变换中心化相对应。
- $M=N=480$ ，比较全滤波，半径  $R=40$ 、 $70$ 、 $85$  时(用一个 10 阶 BLPF 实现)的滤波结果。



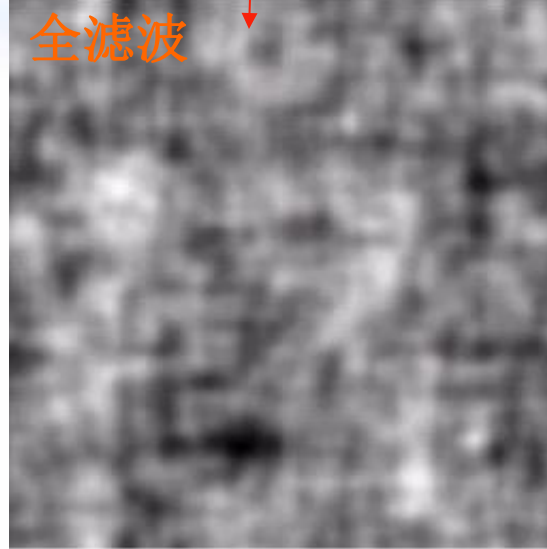
# Inverse filtering

原始图像



噪声淹没了图像，全滤波变得毫无用处。

全滤波



滤波恢复的图像

R=40



变得模糊

使用BLPF进行截断

退化图像

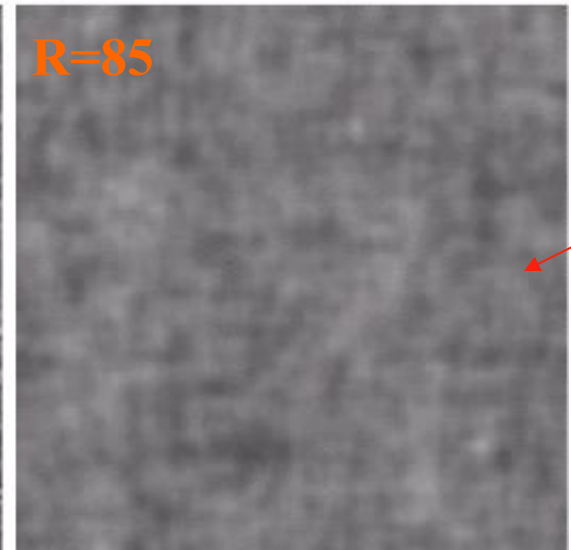


R=70



视觉效果最好

R=85



开始退化，噪声影响越来越大。



# Index

---

- **Image degradation /restoration model**
- **Noise models**
- **Spatial filtering methods: noise only case**
- **Frequency domain filtering methods**
- **Linear position-invariant degradation**
- **Degradation function estimation**
- **Inverse filtering**
- **Wiener filtering**
- **Other methods**
- **Image reconstruction**



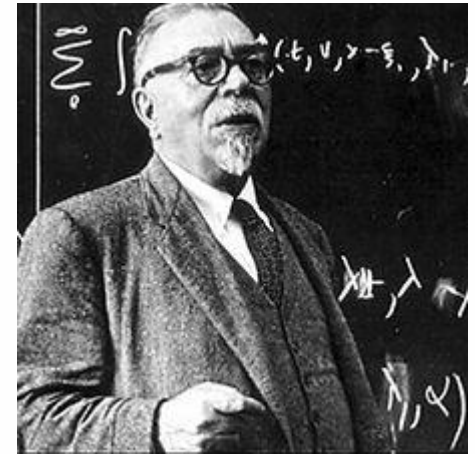
# Wiener filtering

- 综合考虑退化函数  $H$  和噪声统计特征 (逆滤波中未详细讨论)。
- 方法: 建立在图像和噪声是随机过程的基础上
- 目标: 找到一个未污染图像  $f$  的估计  $\hat{f}$  , 使其均方误差最小。
- 误差度量:

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

假定

- 噪声和图像不相关
- 其中一个有零均值 (通常是指噪声)
- 估计的灰度级是退化图像灰度级的线性函数



Norbert Wiener(1894-1964)

# Wiener filtering

- 1942 年 N.Wiener 提出**维纳滤波器** (详细推导过程见《随机信号处理》教材):

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

- **H(u,v)** 为退化函数;  $H^*(u,v)$  为  $H(u,v)$  的复共扼;
- $|H(u,v)|^2 = H^*(u,v) H(u,v)$  复数和其共扼的乘积等于复数模值的平方;
- $S_\eta(u,v) = |N(u,v)|^2 =$  **噪声的功率谱**  $= R^2(u,v) + I^2(u,v)$
- $S_f(u,v) = |F(u,v)|^2 =$  **未退化**图像的功率谱

# Wiener filtering

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



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

- 维纳滤波又称为最小均方误差滤波器，或最小二乘方误差滤波器；
- 除非对于相同的  $u$ 、 $v$  值， $H(u, v)$  和  $S_\eta(u, v)$  都是 0，否则维纳滤波没有逆滤波中退化函数为 0 的问题；
- 如果噪声是 0，噪声功率谱消失，维纳滤波退化为逆滤波。
- 当处理白噪声的时候， $|N(u, v)|^2$  为常数，简化了处理，但未退化图像的功率谱很少已知。当这些值未知或不能估计时，用下式近似：

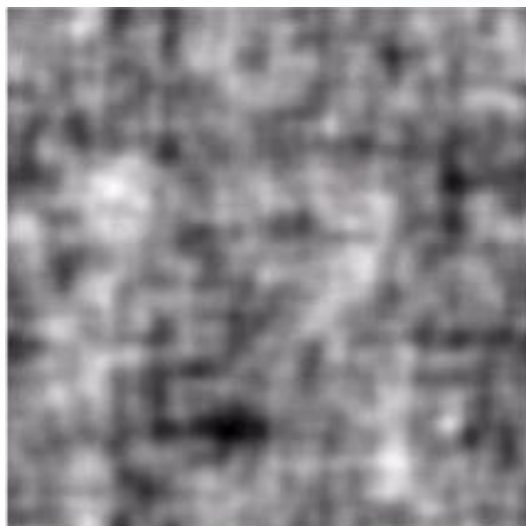
$$\hat{F}(u, v) = \left[ \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v)$$

- $K$  是一个特殊常数；通过交互式选择。

# Wiener filtering



全逆滤波



半径受限  $R=70$  逆滤波



维纳滤波



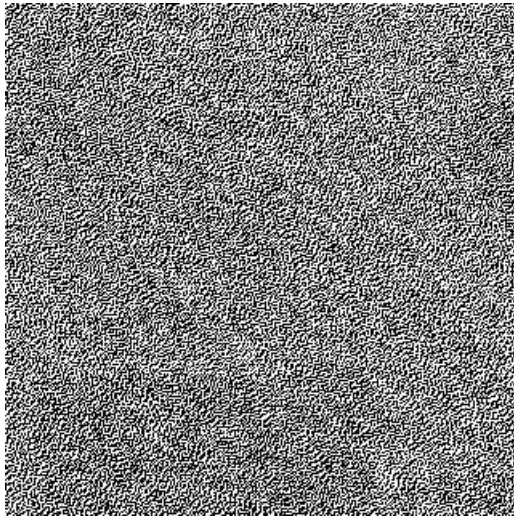
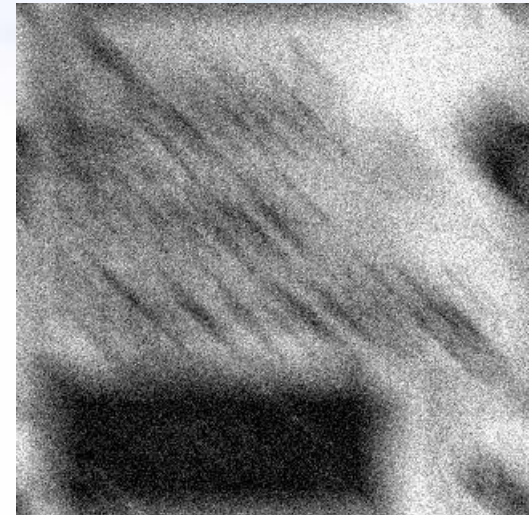
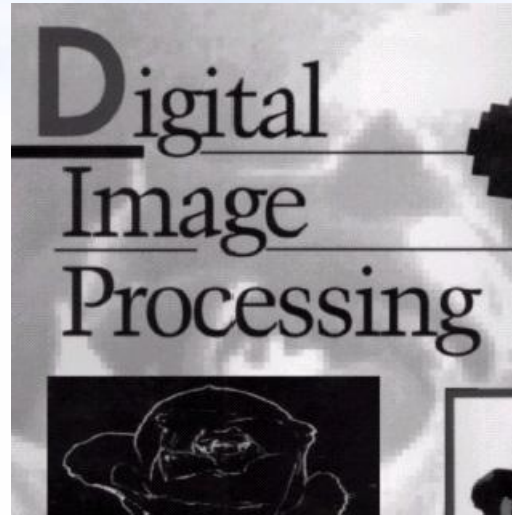
逆滤波和维纳滤波的结果比较

原始图像



# Wiener filtering

逆滤波和维纳滤波的结果比较



全逆滤波



维纳滤波（信噪比为常数）



维纳滤波（根据自相关系数）



# Inverse filtering V.S. Wiener filtering

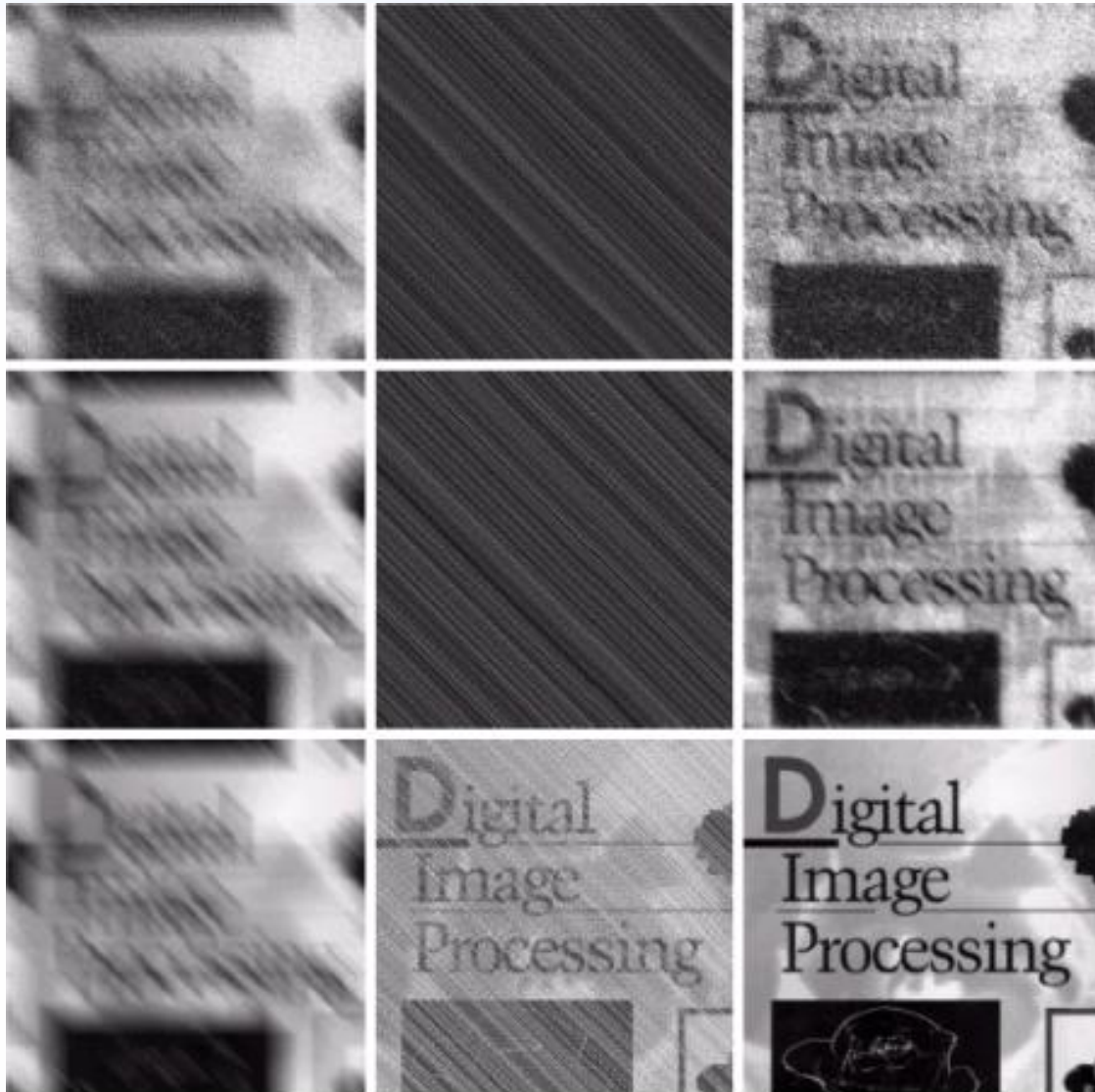
模糊污染图像      半径受限逆滤波      维纳滤波

$$\mu = 0, \sigma^2 = 650$$

由运动模糊和  
加性噪声污染  
的图像

噪声方差减少  
1个数量级

噪声方差减少  
5个数量级



逆滤波图像不太适用，噪声非常强烈；维纳滤波效果不完美，但有图像内容的线索。

噪声减少对逆滤波没有影响，但维纳滤波效果明显改进，文字容易辨认多了。

原始图像中实际已没有可见的噪声，逆滤波消除了图像模糊，但噪声仍很明显，支配着结果，字符仍有一点模糊



# Index

---

- **Image degradation /restoration model**
- **Noise models**
- **Spatial filtering methods: noise only case**
- **Frequency domain filtering methods**
- **Linear position-invariant degradation**
- **Degradation function estimation**
- **Inverse filtering**
- **Wiener filtering**
- **Other methods**
- **Image reconstruction**

# Other methods

1. 有约束的最小二乘方滤波
2. 几何均值滤波器
3. 盲卷积

$$\hat{F}(u, v) = \left[ \frac{H^*(u, v)}{|H(u, v)|^2 + \gamma |P(u, v)|^2} \right] G(u, v) \quad C = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\nabla^2 f(x, y)]^2$$

$$\hat{F}(u, v) = \left[ \frac{H^*(u, v)}{|H(u, v)|^2} \right]^\alpha \left[ \frac{H^*(u, v)}{|H(u, v)|^2 + \beta \left[ \frac{S_\eta(u, v)}{S_f(u, v)} \right]} \right]^{1-\alpha} G(u, v)$$

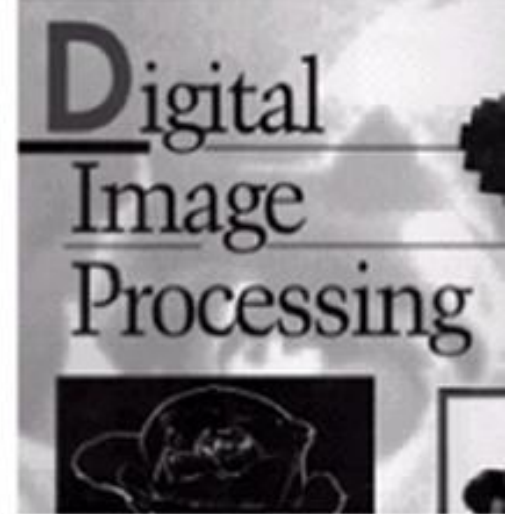
# 维纳滤波和约束最小二乘方滤波比较

$\mu = 0, \sigma^2 = 650$

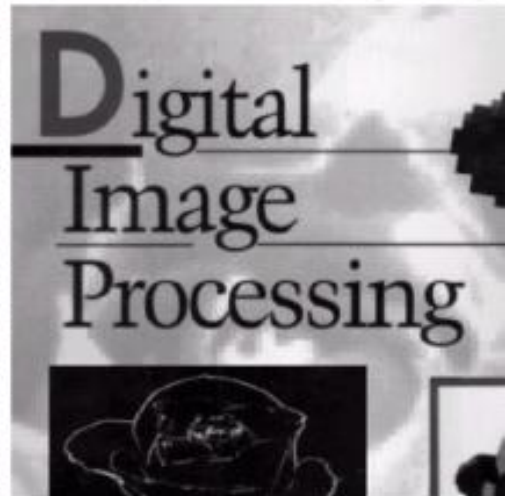
噪声方差减少 1  
个数量级

噪声方差减少 5  
个数量级

维纳滤波



约束最小二乘方滤波



高、中噪声下，约束最小二乘方滤波略优。

在低噪声下两者差不多



# Homework



1. 简述图像复原和图像增强两类技术的异同点；
2. 试结合框图说明图像退化-复原的模型，并写出对应的空域和频域表达式
3. 简述各类噪声的性质和产生原因
4. 简述估计图像噪声pdf的基本过程
5. 比较说明四类均值滤波器的去噪性能
6. 比较说明四类统计排序滤波器的性能
7. 在频域去除图像噪声时，要求尽量减少对图像内容本身的影响，此时对噪声类型有何限制？试以带阻滤波器为例说明频域去噪的过程。



# Homework



8. 写出含有加性噪声的线性空间不变退化系统模型的数学表达式，并简述“图像去卷积滤波器”的含义
9. 简述估计退化函数中的图像观察估计法和实验估计法的基本原理。
10. 以运动模糊为例，简述模型估计法的实现思路
11. 简述逆滤波的实现思路，以及影响逆滤波性能的原因
12. 简述维纳滤波的实现思路，试解释维纳滤波优于逆滤波的原因



# Appendix : imnoise

- **J=imnoise(I, type, parameters)**

- 给图像添加噪声:

J = imnoise(I,'gaussian',M,V)

J = imnoise(I,'localvar',V)

J = imnoise(I,'localvar',image\_intensity,var)

J = imnoise(I,'poisson')

J = imnoise(I,'salt & pepper',d)

J = imnoise(I,'speckle',v)

```
RGB = imread('saturn.png');  
I = rgb2gray(RGB);  
J = imnoise(I,'gaussian',0,0.025);  
K = wiener2(J,[5 5]);  
figure, imshow([I J K])
```

```
I = imread('eight.tif');  
J = imnoise(I,'salt & pepper',0.02);  
figure, imshow(I)  
figure, imshow(J)
```

Value	Description
'gaussian'	Gaussian white noise with constant mean and variance
'localvar'	Zero-mean Gaussian white noise with an intensity-dependent variance
'poisson'	Poisson noise
'salt & pepper'	On and off pixels
'speckle'	Multiplicative noise



# Appendix : fspecial

- **h = fspecial('motion', len, theta)**
  - 运动补偿滤波:

**h = fspecial('motion', len, theta)** returns a filter to approximate, once convolved with an image, the linear motion of a camera by len pixels, with an angle of theta degrees in a counterclockwise direction. The filter becomes a vector for horizontal and vertical motions. The default len is 9 and the default theta is 0, which corresponds to a horizontal motion of nine pixels.

To compute the filter coefficients, h, for 'motion':

1. Construct an ideal line segment with the desired length and angle, centered at the center coefficient of h.
2. For each coefficient location (i,j), compute the nearest distance between that location and the ideal line segment.
3.  $h = \max(1 - \text{nearest\_distance}, 0)$ ;
4. Normalize h:  $h = h / (\text{sum}(h(:)))$

```
I = imread('cameraman.tif');  
H = fspecial('motion',20,45);  
MotionBlur = imfilter(I,H,'replicate');  
imshow([I MotionBlur]);
```



# Appendix : wiener2

- **J = wiener2(I,[m n],noise)**
  - 使用维纳滤波滤除噪声：

```
J = wiener2(I,[m n],noise)
[J,noise] = wiener2(I,[m n])
```

wiener2 lowpass-filters a grayscale image that has been degraded by constant power additive noise. wiener2 uses a pixelwise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel.

`J = wiener2(I,[m n],noise)` filters the image I using pixelwise adaptive Wiener filtering, using neighborhoods of size m-by-n to estimate the local image mean and standard deviation. If you omit the [m n] argument, m and n default to 3. The additive noise (Gaussian white noise) power is assumed to be noise.

`[J,noise] = wiener2(I,[m n])` also estimates the additive noise power before doing the filtering. wiener2 returns this estimate in noise.