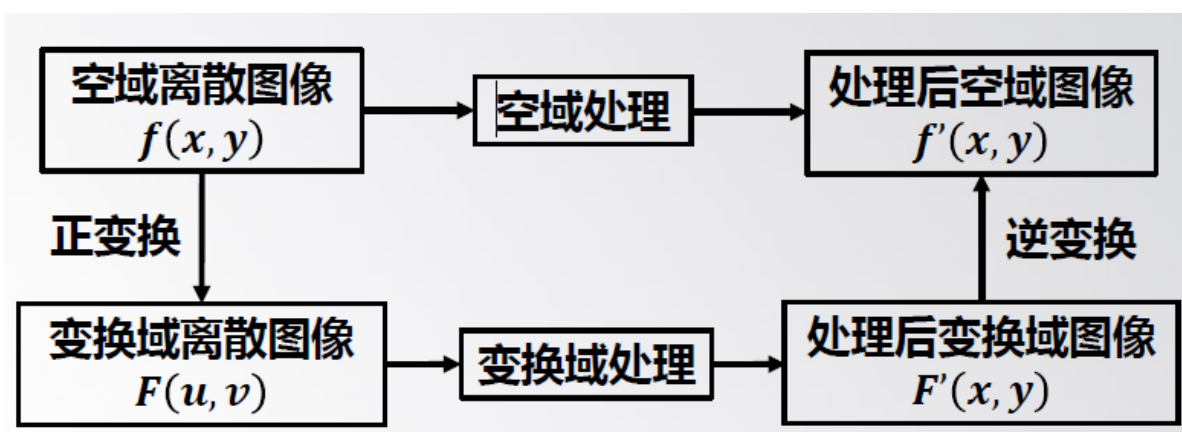


数字图像处理复习

图像变换

概述

- 本章所谓的图像变换特指数字图像经过某种数学工具的处理，把原先**二维空间域**中的数据，变换到另外一个“**变换域**”形式描述的过程。例如，傅里叶变换将时域或空域信号变换成频域的能量分布描述。
- 任何图像信号处理都不同程度改变图像信号的**频率成份分布**。



Fourier 变换

一维傅里叶变换的定义: $F(u) = \int_{-\infty}^{+\infty} f(x) \exp(-j2\pi ux) dx$

一维傅里叶逆变换定义: $f(x) = \int_{-\infty}^{+\infty} F(u) \exp(j2\pi ux) du$

根据欧拉公式： $\exp(-j2\pi ux) = \cos 2\pi ux - j\sin 2\pi ux$

傅里叶变换系数可以写成如下式的复数和极坐标形式：

$$F(u) = R(u) + jI(u) = |F(u)|\exp[j\phi(u)]$$

其中：

傅里叶谱（幅值函数）为： $|F(u)| = [R^2(u) + I^2(u)]^{1/2}$

相位角为： $\phi(u) = \arctan[I(u)/R(u)]$

功率谱为： $P(u) = |F(u)|^2$

- 实际问题的时间或空间函数的区间是**有限的**，或者是频谱有**截止频率**
- 将 $f(x)$ 和 $F(u)$ 的有效宽度同样等分为 N 个小间隔，对连续傅里叶变换进行近似的数值计算，得到 **离散傅里叶变换** 定义。

一维离散傅立叶变换与逆变换：

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp(-j2\pi ux/N), u = 0, 1, \dots, N-1$$

一维离散傅立叶逆变换：

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp(j2\pi ux/N), x = 0, 1, \dots, N-1$$

二维离散傅立叶变换：对于 $N \times N$ 图像

$$F(u, v) = \frac{1}{NN} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp \left(-j2\pi \left(\frac{ux}{N} + \frac{vy}{N} \right) \right)$$

二维离散傅立叶逆变换：对于 $N \times N$ 图像

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \exp \left(j2\pi \left(\frac{ux}{N} + \frac{vy}{N} \right) \right)$$

二维傅里叶变换性质

1. **可分离性**：可以先计算行的傅里叶变换，再计算列的变换，复杂度由 $O(N^4)$ 减为 $O(N^2)$

$$F(x, v) = \frac{1}{N} \sum_{y=0}^{N-1} f(x, y) \exp[-j2\pi vy / N]$$

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} F(x, v) \exp[-j2\pi ux / N]$$

2. **平移性**：

空域平移：

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v) \exp[-j2\pi(ux_0 + vy_0)/N]$$

频域平移：

$$f(x, y) \exp[j2\pi(u_0x + v_0y)/N] \Leftrightarrow F(u - u_0, v - v_0)$$

3. 利用平移性，当二维图像傅里叶变换原点在顶点时，将其移到方阵中心。

当 $u_0 = v_0 = N/2$ 时有:

$$\exp\left[j2\pi(u_0x + v_0y)/N\right] = e^{j\pi(x+y)} = (-1)^{x+y}$$

$$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - N/2, v - N/2)$$

可以简单地用 $(-1)^{x+y}$ 乘以 $f(x, y)$, 将其的傅里叶变换的原点移到相应 $N \times N$ 频率方阵的中心。

4. 周期性

- 如果 $f(x, y)$ 是实函数, 则它的傅里叶变换具有共轭对称性

$$F(u, v) = F^*(-u, -v)$$

- 离散的傅里叶变换和它的反变换具有周期为 N 的周期性

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

5. 旋转性质

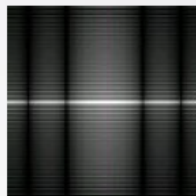
在极坐标系下有: $f(x, y) = f(r, \theta) \Leftrightarrow F(\omega, \varphi)$

如果 $f(x, y)$ 被旋转 θ_0 , 则 $F(u, v)$ 被旋转同一角度:

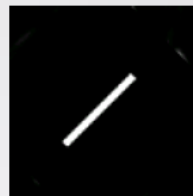
$$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$$



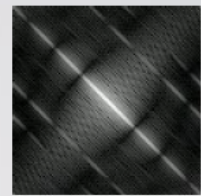
简单图像



傅里叶谱



旋转后图像



旋转后图像的傅里叶谱

6. 分配率

$$af_1(x, y) + bf_2(x, y) \Leftrightarrow aF_1(u, v) + bF_2(u, v)$$

7. 尺度缩放

$$af(x, y) \Leftrightarrow aF(u, v)$$

$$f(ax, by) \Leftrightarrow \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right)$$

8. $F(0, 0)$ 与图像均值的关系

二维图像灰度均值定义： $\bar{f}(x, y) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)$

同时有： $F(0, 0) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)$

因此有： $\bar{f}(x, y) = F(0, 0)$

即 $F(0, 0)$ 等于图像灰度均值。

9. 卷积性质

$$f(x) * g(x) \Leftrightarrow F(u)G(u)$$

$$f(x)g(x) \Leftrightarrow F(u) * G(u)$$

二维变换：

$$f(x, y) * g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(p, q)g(x - p, y - q)dpdq$$

$$f(x, y) * g(x, y) \Leftrightarrow F(u, v)G(u, v)$$

$$f(x, y)g(x, y) \Leftrightarrow F(u, v) * G(u, v)$$

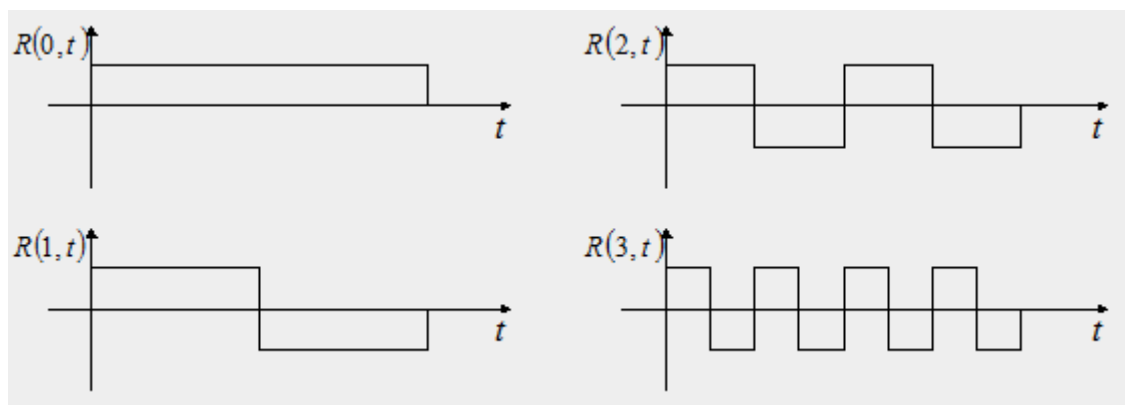
快速傅里叶变换

- N 点序列依次分解成一系列短序列，然后，求出这些短序列的离散傅里叶变换，以此来**减少乘法运算**。

沃尔什变换

Rademacher 函数

$$R(u, x) = \text{Sgn}(\cos 2^u \pi x)$$



正变换核

正变换核

$$N = 2^n$$

$$h(x, u) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}$$

$b_k(z)$: z 的 **二进制表达** 中的第 k 位

如 $n = 3$

对 $z = 6$ (110_2)

有 $b_0(z) = 0$, $b_1(z) = 1$, $b_2(z) = 1$

正变换
$$W(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}$$

反变换
$$f(x) = \sum_{u=0}^{N-1} W(u) \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}$$

傅里叶变换

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp(-j2\pi ux/N)$$

沃尔什变换

$$W(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}$$

- 沃尔什变换意义

沃尔什变换是实现图像变换的重要方法之一。它是一种对应二维离散的数字变换，**大大提高运算速度**；

变换核是值为+1或-1的有序序列。这种变换只需要做**加法或者减法运算**，不需要向傅里叶变换那样做复数乘法运算，所以能提高计算机的运算速度，**减少存储容量**。这种变换同时有快速算法，能进一步提高运算速度。

哈达玛变换

正变换核：
$$h(x, u) = \frac{1}{N} (-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)}$$

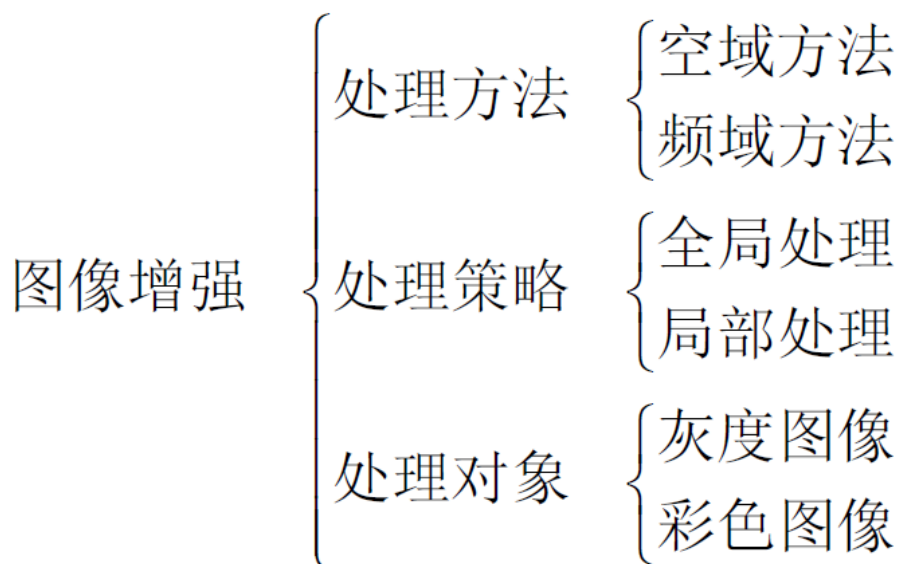
$b_k(z)$ ： z 的二进制表达中的第 k 位

正变换：
$$H(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) (-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)}$$

图像增强

- 常用的图像增强技术根据其处理所进行的空间不同，可分为基于**空域**的方法和基于**频域**的方法；

在图像处理中，空域是指由像素组成的空间。空域增强方法指直接作用于像素的增强方法，可表示为： $g(x, y) = EH[f(x, y)]$ ；



空域点处理变换

点处理的几类方法

1. 将 $f(\cdot)$ 中的每个像素按 EH 操作直接变换以得到 $g(\cdot)$
2. 借助 $f(\cdot)$ 的直方图进行变换
3. 借助一系列图像间的操作进行变换

关键在于设计合适的映射函数

1. 人为设计映射函数，试探其处理效果
2. 从改变图像整体的灰度分布出发，设计一种映射函数，使变换后图像灰度直方图达到或接近预定的形

直接灰度变换

线性变换

对输入图像灰度做线性扩张或压缩，映射函数为一个直线方程

$$g(x, y) = af(x, y) + b$$

a 相当于变换直线的斜率， b 相当于截距

$$b = 0; \begin{cases} a > 1 & \text{对比度扩张} \\ a < 1 & \text{对比度减弱} \\ a = 1 & \text{相当于复制} \end{cases}$$

分段线性处理

与线性变换类似，都是对输入图像的灰度对比度进行拉伸，只是对**不同灰度范围**进行**不同的映射处理**

$$g(x,y)=\begin{cases} r_1 f(x,y) & 0 < f(x,y) < f_1 \\ r_2 [f(x,y) - f_1] + a & f_1 < f(x,y) < f_2 \\ r_3 [f(x,y) - f_2] + b & f_2 < f(x,y) < f_3 \end{cases}$$

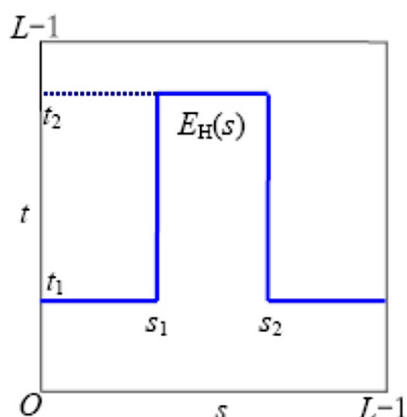
对数变换

扩张数值较小的灰度范围，压缩数值较大的灰度范围。这种变换符合人的视觉特性。（注意要加1）

$$g(x,y) = C \log(1 + f(x,y))$$

灰度切分

将某个灰度值范围变得比较突出，而降其余灰度值变为某个低灰度值



直方图处理

直方图均衡化

直方图均衡化的实质——减少图像的灰度级以换取对比度的加大

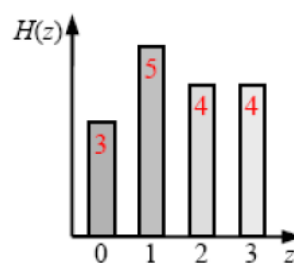
灰度统计直方图：1-D的离散函数，提供了图像像素的灰度值分布情况

计算： 设置一个有 L 个元素的数组，对原图的灰度值进行统计。

$$H(k) = n_k / n$$

0	1	2	3
1	2	3	1
2	3	1	0
3	1	0	2

(a)



(b)

- 直方图均衡化是通过对原图像进行某种变换，使原图像的灰度直方图修正为均匀分布的直方图的一种方法。
- 直方图均衡化方法的基本思想是对在图像中像素个数多的灰度级进行展宽，而对像素个数少的灰度级进行缩减，从而达到清晰图像的目的。
- 原始图像的灰度值 s 在 $0 \sim 1$ 之间，即 $0 \leq s \leq 1$ ， $p_s(s)$ 为原始图像灰度分布的概率密度函数。直方图均衡化处理实际上就是寻找一个灰度变换函数 T ，使变换后的灰度值 $t = T(s)$ ， t 也归一化为 $0 \leq t \leq 1$ ，即建立 s 与 t 之间的映射关系，要求处理后图像灰度分布的概率密度函数 $p_t(t) = 1$ ，即期望所有灰度级出现的概率相同

- 为使变换后的灰度级仍保持从黑到白的次序不变，且变换后的像素灰度在允许的范围内，必须规定：
 - 在 $0 \leq s \leq 1$ 内， $T(s)$ 为单调递增函数；
 - 在 $0 \leq s \leq 1$ 内，有 $0 \leq T(s) \leq 1$ 。

➤ 直方图均衡化

$$\int_{s_j}^{s_j+\Delta s} p_s(s) ds = \int_{t_j}^{t_j+\Delta t} p_t(t) dt$$

当 $\Delta s \rightarrow 0, \Delta t \rightarrow 0$, 有:

$$p_s(s) ds = p_t(t) dt$$

由于 $t = T(s)$, $p_t(t) = 1$, 则:

$$p_s(s) ds = dT(s)$$

最终得到的直方图均衡化的灰度变化函数为:

$$t = T(s) = \int_0^s p_s(s) ds$$

离散形式

$$t_k = T(s_k) = \sum_{i=0}^k p_s(s_i) = \sum_{i=0}^k \frac{n_i}{n}$$

代码示例:

```
% Histogram Equalization
function I_eq = im_histeq(I)

% Convert to grayscale image
if ndims(I) == 3
    I = rgb2gray(I);
end

% Compute the histogram of I
pmf = im_hist(I);
% cumulative distribution function
csm = cumsum(pmf);
s = round(csm * 255);

% Apply the result to image
I_eq = I;

for i = 0:255
    I_eq(I == i) = s(i + 1);
end
end
```

实例:

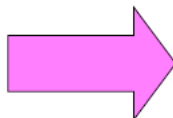
- 设R、S分别为原图像和处理后的图像。

1. 求出原图R的灰度直方图，设为H

➤ 直方图均衡化

1	3	9	9	8
2	1	3	7	3
3	6	0	6	4
6	8	2	0	5
2	9	2	6	0

R 原图像



0	3
1	2
2	4
3	4
4	1
5	1
6	4
7	1
8	2
9	3

H 灰度直方图

2. 求出图像 R 的**总像素个数** (M 、 N 分别为图像的长和宽)

$$NR = M \times N$$

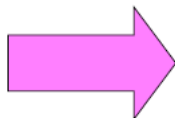
3. 计算**每个灰度级**的像素个数在整个图像中**所占的百分比**

$$HR(I) = H(i)/NR$$

➤ 直方图均衡化

0	3
1	2
2	4
3	4
4	1
5	1
6	4
7	1
8	2
9	3

H 灰度直方图



0	0.12
1	0.08
2	0.16
3	0.16
4	0.04
5	0.04
6	0.16
7	0.04
8	0.08
9	0.12

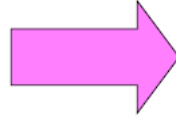
HR 灰度级像素百分比

4. 计算图像各灰度级的累积分布 HS **累加**

$$HS(i) = \sum_{k=0}^i HR(k)$$

➤ 直方图均衡化

0	0.12
1	0.08
2	0.16
3	0.16
4	0.04
5	0.04
6	0.16
7	0.04
8	0.08
9	0.12



0	0.12
1	0.20
2	0.36
3	0.52
4	0.56
5	0.60
6	0.76
7	0.80
8	0.88
9	1.00

HR 灰度级像素百分比

HS 累积分布

5. 求出新图像 P 的灰度值, L 为灰度级个数10 (本题中), 更新对应像素点的灰度值

$$S = \text{int}[(L - 1) \times HS(i) + 0.5], i = 1, 2, 3 \dots$$

4) 求出新图像P的灰度值

$$S = \text{int}[(L - 1) \cdot HS(i) + 0.5] \quad i = 0, 1, 2, \dots$$

式中, L 为灰度级个数

$$HS = [0.12, 0.20, 0.36, 0.52, 0.56, 0.60, 0.76, 0.80, 0.88, 1.00]$$

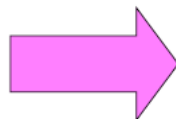
$$9 \cdot HS = [1.08, 1.80, 3.24, 4.68, 5.04, 5.40, 6.84, 7.20, 7.92, 9.00]$$

四舍五入

新图 [1, 2, 3, 5, 5, 5, 7, 7, 8, 9]

原图 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

1	3	9	9	8
2	1	3	7	3
3	6	0	6	4
6	8	2	0	5
2	9	2	6	0



2	5	9	9	8
3	2	5	7	5
5	7	1	7	5
7	8	3	1	5
3	9	3	7	1

R 原图像

P 新图像

- **优缺点：**由于直方图均衡化增加了图像灰度的动态范围，所以也增加了图像的对比度，反映在图像上就是图像有较大的反差，许多细节更加清晰。
- 但它的具体增强效果不易控制，处理的结果总是得到全局均衡化的直方图

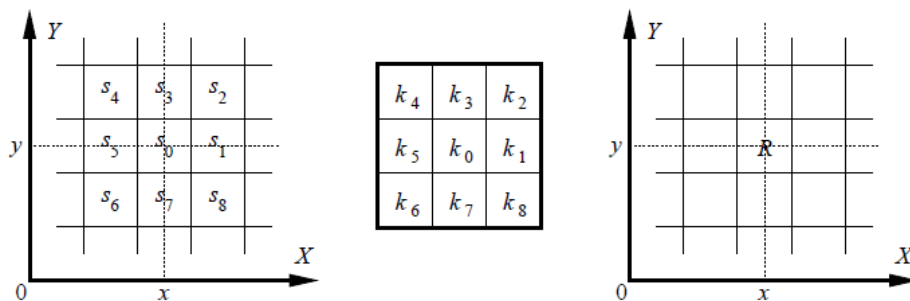
直方图规定化（匹配）（要再看！！）

假设 $p_r(r)$ 是原始图像灰度分布的概率密度函数， $p_z(z)$ 是希望得到的图像的灰度概率密度函数。

空域滤波

- 借助模板进行邻域操作完成的，根据操作特点分为**线性滤波**和**非线性滤波**两类；而根据滤波效果又分为**平滑滤波**和**锐化滤波**；
- 在图像中任意一点 (x, y) ，滤波器的响应 $g(x, y)$ 是滤波器系数与由该滤波器包围的像素的乘积之和。

$$g = k_0 s_0 + k_1 s_1 + \cdots + k_8 s_8$$



图像噪声

- **定义：**图像在摄取或传输时所受到的**随机干扰信号**。不可预测的，只能用概率统计方法来认识的随机误差；
- **描述：**可以借用随机过程及其概率分布函数和概率密度函数；
- **特征：**均值、方差和相关函数等。

- **加性噪声：**噪声和图像信号的强度不相关。设 f 为理想图像， n 为噪声， g 为实际输出的图像：

$$g = f + n$$

加性噪声又可分为**脉冲噪声**和**高斯噪声**

- **乘性噪声：**噪声和图像信号的强度相关：

$$g = f + fn$$

椒盐噪声（脉冲噪声）

- 受噪声干扰的图像像素以**50%**的相同概率等于图像灰度的最大（表现为使相应的像素点变白）或最小（表现为使相应的像素点变黑）的可能取值。



原始图像



椒盐噪声干扰的图像

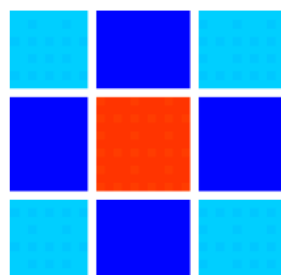
均值滤波

- 空间域
- 依据：相邻像素间存在很高的空间相关性，而噪声是统计独立的。则可用像素邻域内各像素的灰度平均值代替该像素原来的灰度值，实现图像的平滑。

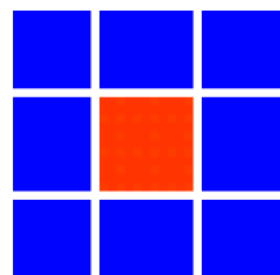
设有一幅 $N \times N$ 的图像 $f(x, y)$ ，若平滑图像为 $g(x, y)$ ，则有：

$$g(x, y) = \frac{1}{M} \sum_{(i, j) \in s} f(i, j)$$

式中， s 为 (x, y) 邻域内像素坐标的集合； M 为 (x, y) 邻域内像素的总数。常用的邻域为**4-邻域**或**8-邻域**



4-邻域



8-邻域

- 性能分析：如果图像中的噪声是加性噪声，窗口内各点噪声是独立同分布的，则经过上述平滑后，信号与噪声的含量比可提高 **M** 倍。
 - 优点：算法简单、处理速度快；
 - 缺点：局部平滑法在降低噪声的同时**会使图像产生模糊**，特别在边缘和细节处。而且邻域越大，在去噪能力增强的同时**模糊程度越严重**。原因是它对所有的像素点都是**同等对待**，在将噪声点抑制的同时，将景物的边缘点也抑制了

加权均值滤波器

$$H_1 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad H_2 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad H_3 = \frac{1}{2} \begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix}$$

- 一般，处于模板中心位置的像素比其它像素的权值要大，即把**中心点加强为最高**。又因为对角项离中心比正交方向相邻的像素更远，它的重要性要比与中心直接相邻的四个像素低，所以随着**离中心点距离的增加权值减小**。这样，经过加权均值滤波后，可**减小平滑处理中的模糊**。

中值滤波

- 因为噪声（如椒盐噪声）的出现，使该点像素比周围的像素亮（暗）许多；
- 如果在某个模板中，对像素进行**由小到大的重新排列**，那么最亮或者最暗的点**一定被排在两侧**；
- 取模板中排在**中间位置上的**像素的灰度值**替代待处理像素的灰度值**，就可以达到滤除噪声的目的。
- 中值滤波对**离散阶跃信号**，**斜坡信号**不产生影响；
- 对于离散的脉冲信号，当其**连续出现的次数小于窗口尺寸的一半**时，将被抑制掉，否则将不受影响。
- 对于**椒盐噪声**，**中值滤波效果比均值滤波效果好**。

原因

椒盐噪声只在画面中的部分点上**随机出现**，所以根据中值滤波原理可知，通过数据排序的方法，将图像中未被噪声污染的点替代噪声点的值的概率比较大，因此中值滤波对椒盐噪声的抑制效果很好，同时画面的清晰度基本保持。

因为**椒盐噪声的均值不为0**，所以均值滤波不能很好地去除椒盐噪声点。

- 对于**高斯噪声**，**均值滤波效果比中值滤波效果好**。

原因

高斯噪声是以随机大小的幅值**污染所有的点**，因此无论怎样进行数据选择，得到的**总是被污染的值**，所以中值滤波对高斯噪声的抑制效果不是很好；

因为**正态分布的均值为0**，所以均值滤波可以消除高斯噪声。

边界保持平滑滤波

经过平滑滤波处理后，图像就会**变得模糊**

- 原因：目标物之间存在**边界**，而边界点与噪声点有一个共同的特点是，都**具有灰度的阶跃特性**。所以平滑处理会**同时将边界也处理了**。
- 解决方法**：首先判别当前像素**是否为边界上的点**，如果是，则不进行平滑处理；如果不是，则进行平滑处理。

超限像素平滑法

将 $f(x, y)$ 和其**邻域平均** $g(x, y)$ **差的绝对值**与选定的**阈值** T 进行比较，根据比较结果决定点 (x, y) 的最后灰度 $g'(x, y)$ ，其表达式为：

$$g'(x, y) = \begin{cases} g(x, y) & |f(x, y) - g(x, y)| > T \\ f(x, y) & |f(x, y) - g(x, y)| \leq T \end{cases}$$

如果某个像素的灰度值和其邻域像素的**平均值之差达到了一定水平**，则判断该像素为**噪声**，继而用其**邻域像素的平均值**取代这一像素值，否则，像素值不变。

- **优点**：对抑制椒盐噪声比较有效，对保护仅有微小灰度差的细节及纹理也有效。
- **缺点**：随着邻域增大，去噪能力增强，但模糊程度也变大。
- T 太大，噪声消除不干净； T 太小，易使图像模糊。一般选用 3×3 窗口

锐化滤波器

- **边缘** 是图像中特性分布的不连续处，占据图像的**高频成分**
- 图像锐化就是要增强图像的边缘或轮廓，属于**高通滤波**
- 图像锐化旨在增强图像频谱中的高频部分，就相当于从原图像中**减去**它的低频分量，即原始图像经平滑处理后所得的图像

$$g(x, y) = f(x, y) - \bar{f}(x, y)$$

$$g(x, y) = A \times f(x, y) - \bar{f}(x, y)$$

当 $A = 1$ 时，上式：

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

平滑窗口 锐化模板

非线性锐化滤波器

微分法

- 考察正弦函数 $\sin 2\pi ax$ ，它的微分为 $2\pi a \cos 2\pi ax$ 。微分后频率不变，幅度上升 $2\pi a$ 倍。
可见，频率越高，幅度增加就越大。这表明**微分是可以加强高频成份**的。而边缘与轮廓一般位于**灰度突变**的地方，占据图像的高频成份

最常用的微分方法是 **梯度法**

最常用的微分方法是**梯度法**。对于图像 $f(x, y)$ ，在 (x, y) 处的梯度定义为：

$$\nabla f(x, y) = \begin{bmatrix} f'_x \\ f'_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix}$$

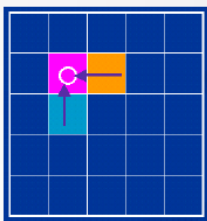
梯度是一个矢量，其大小和方向分别为：

$$|\nabla f(x, y)| = \sqrt{f'^2_x + f'^2_y} = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2}$$

$$\theta = \tan^{-1}(f'_y / f'_x) = \tan^{-1}\left(\frac{\partial f(x, y)}{\partial y} / \frac{\partial f(x, y)}{\partial x}\right)$$

在 (x, y) 点处的梯度的方向是 $f(x, y)$ 在该点灰度变化率最大的方向。

对于数字图像处理而言，常用到梯度的大小，因此把**梯度的大小习惯称为“梯度”**，并且一阶偏导数采用一阶差分近似表示，即：



$f(x, y)$	$f(x, y+1)$
$f(x+1, y)$	

-1
1

-1	1
----	---

典型梯度算子

$$f'_x = f(x+1, y) - f(x, y)$$

$$f'_y = f(x, y+1) - f(x, y)$$

以左上角为零点

为更适合计算机实现，采用**绝对差分算法**：

$$\begin{aligned} |\nabla f(x, y)| &= |f'_x| + |f'_y| \\ &= |f(x+1, y) - f(x, y)| + |f(x, y+1) - f(x, y)| \end{aligned}$$

梯度法的特点：

- 图像的边缘上，像素灰度有陡然变化，梯度值**很大**；
- 图像灰度变化平缓区域，梯度值**很小**；
- 等灰度区域，梯度值为**零**。

-1	
	1

	-1
1	

Roberts算子

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Prewitt算子

Prewitt 锐化边缘的同时减少噪声的影响

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Sobel算子

对4邻域采用加权的方法计算差分。同*Prewitt*算子相比，*Sobel*算子对于像素的位置的影响做了加权，因此效果更好

生成不同的增强图像

(1)梯度图像直接输出

$$g(x, y) = \nabla f(x, y)$$

特点：增强的图像边缘轮廓比较突出，灰度平缓变化部分的梯度小，很黑。



(2)加阈值的梯度输出

$$g(x, y) = \begin{cases} \nabla f(x, y), & \nabla f(x, y) \geq T \\ f(x, y), & \text{其它} \end{cases}$$

特点：适当选取阈值T，可使明显的边缘轮廓得到突出，又不会破坏原来灰度变化比较平缓的背景。



(3)边缘灰度规定化输出

$$g(x, y) = \begin{cases} L_G, & \nabla f(x, y) \geq T \\ f(x, y), & \text{其它} \end{cases}$$

特点：将明显边缘用一固定的灰度级 L_G 来表现。



(4)边缘灰度规定化输出

$$g(x, y) = \begin{cases} \nabla f(x, y), & \nabla f(x, y) \geq T \\ L_B, & \text{其它} \end{cases}$$

特点：将背景用一个固定的灰度级 L_B 来表现。



(5)二值图像输出

$$g(x, y) = \begin{cases} L_G, & \nabla f(x, y) \geq T \\ L_B, & \text{其它} \end{cases}$$

特点：将明显边缘和背景分别用灰度级 L_G 和 L_B 。



Laplacian 算子

Laplacian 算子是一个二阶微分算子

对数字图像而言，二阶偏导数用二阶差分近似，有：

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

0	1	0
1	-4	1
0	1	0

Laplacian算子

Laplacian增强算子为：

$$\begin{aligned} g(x, y) &= f(x, y) - \nabla^2 f(x, y) \\ &= 5f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] \end{aligned}$$

其对应的模板为：

0	-1	0
-1	5	-1
0	-1	0

Laplacian增强算子

频域增强

卷积定理

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

增强图

$$g(x, y) = \mathcal{T}^{-1}[H(u, v)F(u, v)]$$

步骤

- (1) 计算图象的变换
- (2) 在频域滤波
- (3) 反变换回图象空间

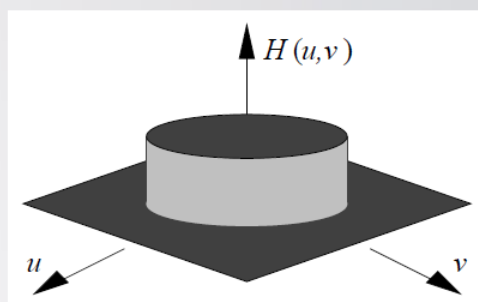
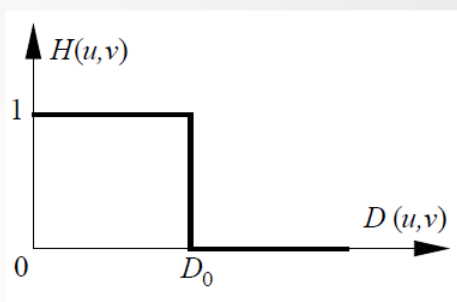
频域滤波

低通，高通，带通/带阻，同态

低通滤波

理想低通滤波器

理想是指小于 D_0 的频率可以完全不受影响地通过滤波器，而大于 D_0 的频率则完全通不过



$H(u, v)$:转移/滤波函数

$$H(u, v) = \begin{cases} 1 & \text{如 } D(u, v) \leq D_0 \\ 0 & \text{如 } D(u, v) > D_0 \end{cases}$$

D_0 : 截断频率 (非负整数)

$D(u, v)$ 是从点 (u, v) 到频率平面原点的距离

$$D(u, v) = (u^2 + v^2)^{1/2}$$

理想低通滤波器的模糊

- 理想低通滤波所产生的“振铃”现象在2-D图象上表现为一系列同心圆环
- 圆环半径反比于截断频率

巴特沃斯低通滤波器

巴特沃斯低通滤波器

截断频率

使 H 降到最大值某个百分比的频率

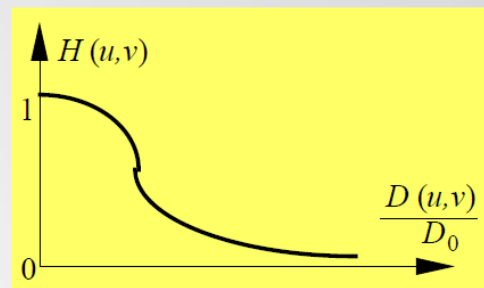
在 $D(u, v) = D_0$ 时

- $H(u, v) = 0.5$

或

- $H(u, v) = \frac{1}{\sqrt{2}}$

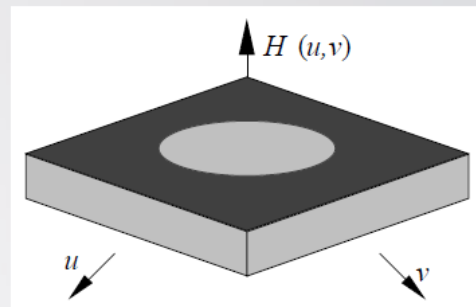
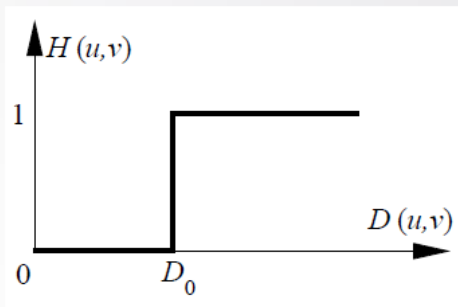
$$H(u, v) = \frac{1}{1 + [D(u, v) / D_0]^{2n}}$$



- 它的通带与阻带之间没有明显的不连续性，因此在抑制噪声的同时，图像边缘的模糊程度大大减小，没有振铃效应产生
- 尾部含有较多的高频成分，对噪声的平滑效果不如理想低通滤波器

形状与低通滤波器的形状正好相反

$$H(u, v) = \begin{cases} 0 & \text{如 } D(u, v) \leq D_0 \\ 1 & \text{如 } D(u, v) > D_0 \end{cases}$$



巴特沃斯高通滤波器

形状与巴特沃斯低通滤波器的形状正好相反

截断频率

使 H 升到最大值某个百分比的频率

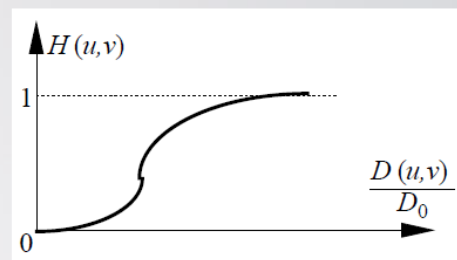
在 $D(u, v) = D_0$ 时

- $H(u, v) = 0.5$

或

- $H(u, v) = \frac{1}{\sqrt{2}}$

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



傅里叶变换: $G(u, v) = H(u, v)F(u, v)$

高频增强转移函数: $H_e(u, v) = k \times H(u, v) + c$

高频增强输出图的傅里叶变换:

$$G_e(u, v) = k \times G(u, v) + c \times F(u, v)$$

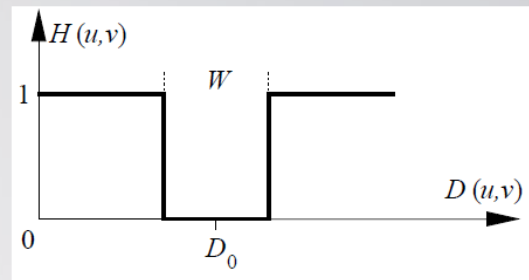
反变换回去:

$$g_e(u, v) = k \times g(u, v) + c \times f(u, v)$$

带通和带阻滤波

带阻滤波器

阻止一定频率范围
(允许其它频率范围)



$$D(u, v) = \left[(u - u_0)^2 + (v - v_0)^2 \right]^{1/2}$$

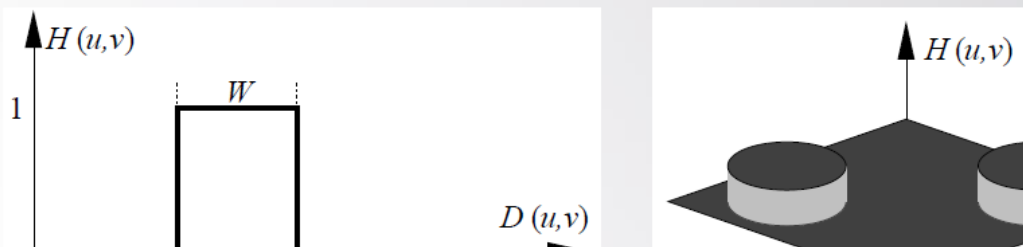
$$H(u, v) = \begin{cases} 0 & \text{如 } D(u, v) \leq D_0 \\ 1 & \text{如 } D(u, v) > D_0 \end{cases}$$

带通滤波器

与带通滤波器互补

允许一定频率范围（阻止其它频率范围）

$$H_P(u, v) = -[H_R(u, v) - 1] = 1 - H_R(u, v)$$



○同态滤波

把图像亮度 $f(x, y)$ 看成是由入射分量（入射到景物上的光强度） $i(x, y)$ 和反射分量（景物反射的光强度） $r(x, y)$ 组成。

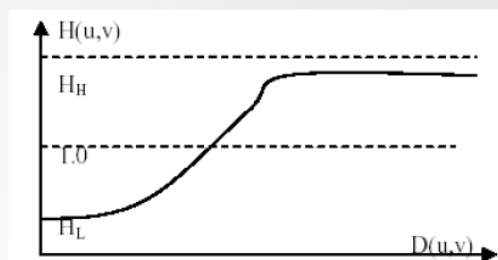
- 同态滤波是一种在频域中同时进行图像**对比度增强**和**压缩图像亮度范围**的滤波方法；
- 其基本思想是**减少入射分量** $i(x, y)$ ，并同时**增加反射分量** $r(x, y)$ 来改善图像 $f(x, y)$ 的显示效果；
- $i(x, y)$ 在空间上变化缓慢，其频谱集中在**低频段**， $r(x, y)$ 反映图像的细节和边缘，其频谱集中在**高频段**。

对成像模型两边取对数： $\ln f(x, y) = \ln i(x, y) + \ln r(x, y)$

两边取傅里叶变换： $F(u, v) = I(u, v) + R(u, v)$

用一频域函数 $H(u, v)$ 处理 $F(u, v)$ ：

$$H(u, v)F(u, v) = H(u, v)I(u, v) + H(u, v)R(u, v)$$



反变换到空域： $h_f(x, y) = h_i(x, y) + h_r(x, y)$

两边取指数：

$$g(x, y) = \exp|h_f(x, y)| = \exp|h_i(x, y)| \cdot \exp|h_r(x, y)|$$

代码

```
% Homomorphic Filtering
function I_homo = homo_filter(I, rH, rL, c, D0)

% Convert to grayscale image
if ndims(I) == 3
    I = rgb2gray(I);
end
I = double(I);

% turn the multiplicative components into additive components
lg_I = log(I+1);

% fourier transform
F = fft(lg_I);

% use gaussian high-pass filter
I_gf = GHPF(I, rH, rL, c, D0);
IF = I_gf.*F;

% Inverse fourier transform
ln_if = ifft(IF);
intermediate = exp(ln_if)-1;

% Normalization
intermediate = im_norm(intermediate);

% Image retrieval
I_homo = uint8(round(intermediate*255));
end

% Gaussian high-pass filter.
function H = GHPF(I, rH, rL, c, D0)
% Convert to grayscale image
if ndims(I) == 3
    I = rgb2gray(I);
end
[M, N] = size(I);
H = zeros(M, N);

% get the midpoint of the image
m = floor(M/2);
n = floor(N/2);
% Calculate gaussian high-pass filter H(u,v)
for x = 1:M
    for y = 1:N
        D2_uv = (x-m)^2+(y-n)^2;
        H(x,y) = (rH-rL)*(1-exp(-c*(D2_uv/(D0^2))))+rL;
    end
end
end

% Normalizes the input image I.
function I_norm = im_norm(I)
```

```

I = double(I);
I_max = max(I, [], 'all');
I_min = min(I, [], 'all');
% Normalize
I_norm = (I - I_min)/(I_max-I_min);
end

```

图像恢复和重建

图像退化的典型表现为**图像模糊、失真、有噪声**

图像恢复vs 图像增强

- 相同之处：改进输入图像的视觉质量
- 不同之处：
 - 图像增强**借助人的视觉系统特性**，以取得较好的视觉结果（不考虑退化原因）
 - 图像恢复根据相应的**退化模型**和知识重建或恢复原始的图像（考虑退化原因）

退化模型和对角化

噪声是一种**外部干扰**

- 信噪比 SNR

$$SNR = 10 \log_{10} \left(\frac{V_s^2}{V_n^2} \right)$$

- 常见噪声
 - 热噪声：白噪声（频率覆盖整个频谱）
 - 高斯噪声（幅度符合高斯分布）
 - 闪烁噪声：具有反比于频率（1/f）的频谱
 - 粉色噪声（在对数频率间隔内有相同的能量）
 - 发射噪声：高斯分布（电子运动的随机性）

高斯噪声

$$p(z) = \frac{1}{\sqrt{2\pi} \sigma} \exp \left[-\frac{(z - \mu)^2}{2\sigma^2} \right]$$

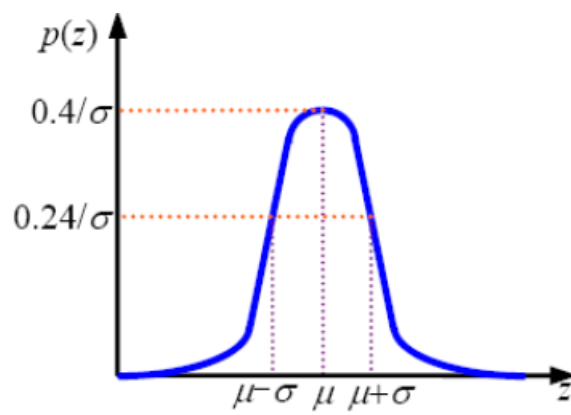


图 8.1.5 一个高斯噪声的概率密度函数

均匀噪声

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

$$\mu = (a+b)/2$$

$$\sigma^2 = (b-a)^2 / 12$$

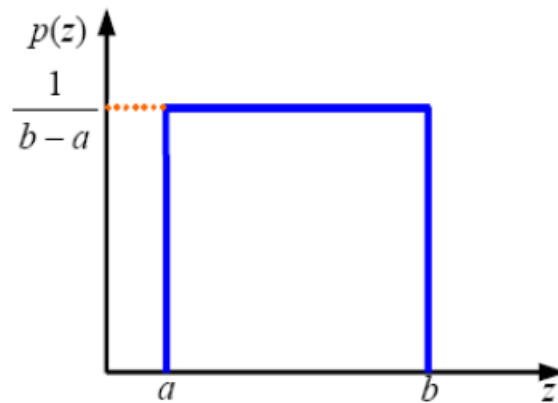


图 8.1.6 一个均匀噪声的概率密度函数

脉冲噪声

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

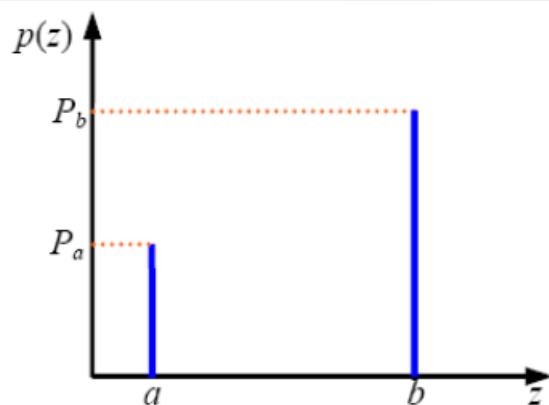


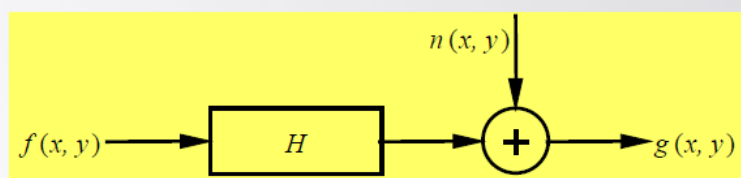
图 8.1.7 一个脉冲噪声的概率密度函数

- 噪声脉冲可以是正的或负的
- 一般假设a和b都是“饱和”
- **双极性脉冲噪声** 也称椒盐噪声

退化模型

- **H: 退化过程**
- **$n(x, y)$: 加性噪声 (统计特性已知)**

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



加性噪声: 在给定 $g(x, y)$ 和代表退化的 H 的基础上得到对 $f(x, y)$ 的某个近似

退化H的性质

(1) 线性:

$$H[k_1 f_1(x, y) + k_2 f_2(x, y)] = k_1 H[f_1(x, y)] + k_2 H[f_2(x, y)]$$

(2) 相加性 ($k_1 = k_2 = 1$) :

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

(3) 一致性 ($f_2(x, y) = 0$) :

$$H[k_1 f_1(x, y)] = k_1 H[f_1(x, y)]$$

(4) 位置 (空间) 不变性:

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

退化过程

注意周期 M 的范围

1-D退化过程

卷积 $f(x)$ 和 $h(x)$: 采样 \rightarrow 2个数组 A 和 B

为避免卷积周期重叠: $M \geq A + B - 1$

$$f_e(x) = \begin{cases} f(x) & 0 \leq x \leq A-1 \\ 0 & A \leq x \leq M-1 \end{cases}$$

$$h_e(x) = \begin{cases} h(x) & 0 \leq x \leq B-1 \\ 0 & B \leq x \leq M-1 \end{cases}$$

$$g_e(x) = \sum_{m=0}^{M-1} f_e(m) h_e(x-m) \quad x = 0, 1, \dots, M-1$$

用矩阵表示

$$g_e(x) = \sum_{m=0}^{M-1} f_e(m) h_e(x-m) \quad x = 0, 1, \dots, M-1$$

$$g = Hf = \begin{bmatrix} g_e(0) \\ g_e(1) \\ \vdots \\ g_e(M-1) \end{bmatrix} = \begin{bmatrix} h_e(0) & h_e(-1) & \dots & h_e(-M+1) \\ h_e(1) & h_e(0) & \dots & h_e(-M+2) \\ \vdots & \vdots & \ddots & \vdots \\ h_e(M-1) & h_e(M-2) & \dots & h_e(0) \end{bmatrix} \begin{bmatrix} f_e(0) \\ f_e(1) \\ \vdots \\ f_e(M-1) \end{bmatrix}$$

根据周期性

$$h_e(x) = h_e(x + M)$$

轮换矩阵

$$H = \begin{bmatrix} h_e(0) & h_e(-1) & \dots & h_e(-M+1) \\ h_e(1) & h_e(0) & \dots & h_e(-M+2) \\ \vdots & \vdots & \ddots & \vdots \\ h_e(M-1) & h_e(M-2) & \dots & h_e(0) \end{bmatrix}$$

推广到二维

$$f_e(x, y) = \begin{cases} f(x, y) & 0 \leq x \leq A-1 \quad \text{和} \quad 0 \leq y \leq B-1 \\ 0 & A \leq x \leq M-1 \quad \text{或} \quad B \leq y \leq N-1 \end{cases}$$

$$h_e(x, y) = \begin{cases} h(x, y) & 0 \leq x \leq C-1 \quad \text{和} \quad 0 \leq y \leq D-1 \\ 0 & C \leq x \leq M-1 \quad \text{或} \quad D \leq y \leq N-1 \end{cases}$$

块轮换矩阵（每块都轮换标注）

$$g = Hf + n = \begin{bmatrix} H_0 & H_{M-1} & \dots & H_1 \\ H_1 & H_0 & \dots & H_2 \\ \vdots & \vdots & \ddots & \vdots \\ H_{M-1} & H_{M-2} & \dots & H_0 \end{bmatrix} \begin{bmatrix} f_e(0) \\ f_e(1) \\ \vdots \\ f_e(MN-1) \end{bmatrix} + \begin{bmatrix} n_e(0) \\ n_e(1) \\ \vdots \\ n_e(MN-1) \end{bmatrix}$$