

# 数字图像期末复习

## 第一课：简介

1. 灰度数字图像在数据文件中以**八位无符号整型(0-255)**的整数来表示灰度，**0为黑色，255为白色**。
2. 彩色图像最为常用的存储模式为**RGB存储模式**，记为 $f(x, y, z)$ ,  $1 \leq x \leq M, 1 \leq y \leq N, z \in \{1, 2, 3\}$ 。
3. 数字图像由二位元素组成，每一个元素具有特定的位置和灰度值（或颜色值向量），这些元素称为像素。显示设备的总像素数也称为设备的**分辨率**。

## 第二课：图像理论基础与图像分析

### 图片的分类

- 自然图像：**分片光滑、边界分明、纹理清晰**。
- 成像图像：例如CT成像图，是自然信息在某种映射下的表示。
- 抽象图像：往往是某种函数或数据的抽象展示。
- 动态图像：也可以成为视频，是图像附加时间维度的表示。

### BMP（位图）表示

- 图像描述信息：如图像高度和宽度等信息
- 图像数据：顺序存放的连续数据
- BMP格式：
  1. 位图文件头
  2. 位图信息头
  3. 调色板
  4. 图像数据

### 实际图像数据：

- 对于2色值位图，1位表示一个像素颜色，所以一个字节表示8个像素
- 对于16色位图，4位表示一个像素颜色，所以一个字节表示2个像素
- 对于256色位图，一个字节表示1个像素
- 对于真彩色图，三个字节表示1个像素

## 图像的质量

### 对比度

**最大亮度与最小亮度之间的差**，偶尔也会使用求商的方法刻画对比度，最小值不为0。

### 清晰度

清晰度往往代表一种**主观**的感受，一般与图像分辨率、细节丰富程度都有一定的关系。

## 色彩饱和度

色彩饱和度不足的图片会被认为暗淡、不自然，是图像质量较差的一种体现。

## 像素间的关系

- 四邻域：对于  $p(x, y)$  的4邻域  $N_4(p)$  是：  $(x+1, y)$ ,  $(x-1, y)$ ,  $(x, y+1)$ ,  $(x, y-1)$
- $D$ 邻域：对角上的点，用  $N_D(p)$  表示像素  $p$  的  $D$  邻域：  
 $(x+1, y+1)$ ,  $(x+1, y-1)$ ,  $(x-1, y+1)$ ,  $(x-1, y-1)$ 。
- 八邻域：  $N_8(p) = N_4(p) + N_D(p)$

像素间的欧氏距离：  $D_e(p, q) = \sqrt{(x-s)^2 + (y-t)^2}$

- 像素间的  $D_4$  距离：  $D_4(p, q) = |x-s| + |y-t|$
  - 像素间的  $D_8$  距离：  $D_8(p, q) = \max(|x-s|, |y-t|)$
- 

## 整体图像数值指标

1. 图像的大小：  $S = M * N$ ，灰度图像可以用 `numel` 函数直接生成，任意图像都可以用 `size` 获得图像行数和列数。
2. 图像的灰度平均值：  $\bar{f} = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N f(x, y)$ ，表示图像整体的明暗水平，可以用 `mean` 求平均值。
3. 图像灰度方差：表示单张图像的灰度值差异情况。

$$C_{ff} = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x, y) - \bar{f}]^2 = \sigma_{ff}^2$$

4. 图像协方差：表示两图像明暗相对变化的趋势是否一致。

$$C_{fg} = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x, y) - \bar{f}][g(x, y) - \bar{g}]$$

5. 图像间的相关性系数  $\rho_{fg}$
- 

## 图像间的误差比较

### 峰值信噪比 (PSNR)

基于八位无符号整形的定义为：

$$PSNR = 10 \cdot \log_{10} \left( \frac{255^2 \cdot \text{总点数}}{\sum [f(x, y) - g(x, y)]^2} \right) (\text{dB})$$

### 结构相似性 (SSIM)

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

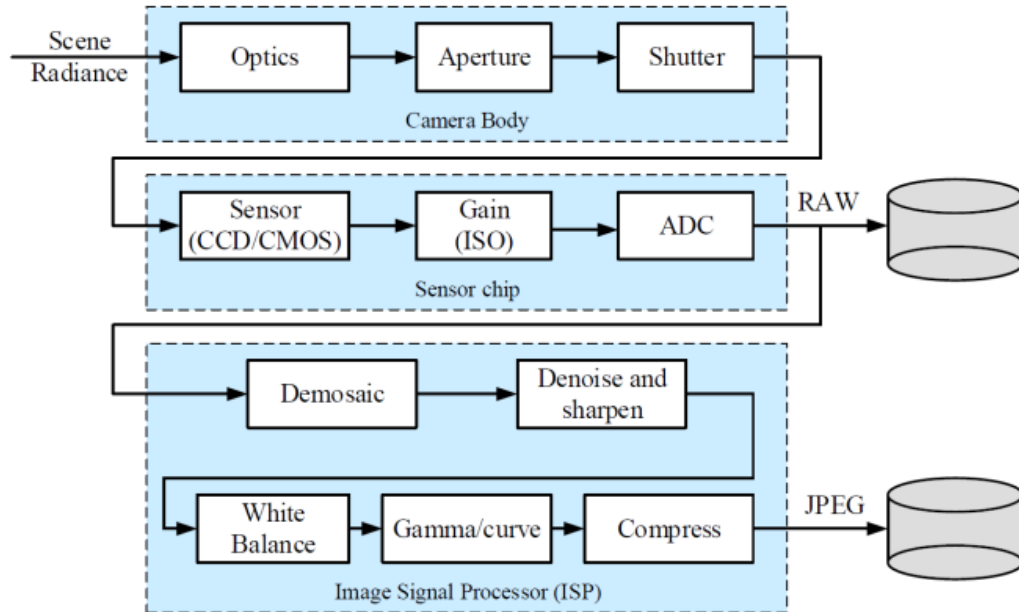
$\mu$  为两图的均值， $\sigma^2$  代表方差， $\sigma_{xy}$  代表协方差， $c_1, c_2$  为防止除零操作的常数，可以设  $c_1 = (0.01L)^2$ ， $c_2 = (0.03L)^2$ ， $L$  为峰值。

---

## 第三课：图像数字化与基础运算

### 图像数字化的大致流程

□ 提高采集技术、分析并预先了解采集数据的特性与问题。下图表示自然图像数字化流程：



### 二维连续傅里叶变换 (fft)

二维连续傅里叶变换可以理解为一维连续傅里叶变换的张量积，或理解按照两个方向分别作一次一维傅里叶变换。

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

其中  $(x, y)$  表示时间域的位置， $(u, v)$  代表频率域的坐标。此定义可以获取二位函数或图像  $f(x, y)$  的频率分析。

逆变换定义为：

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} du dv$$

### 采样定律

奈奎斯特采样定律：如果 2D 信号  $f(x, y)$  的傅里叶频谱  $F(u, v)$  满足

$$F(u, v) = \begin{cases} F(u, v) & |u| \leq U_c, |v| \leq V_c \\ 0 & |u| > U_c, |v| > V_c \end{cases}$$

即傅里叶频谱的支撑集有界，则当采样周期  $\Delta x, \Delta y$  满足  $\frac{1}{\Delta x} = u_s \geq 2U_c$ ， $\frac{1}{\Delta y} = v_s \geq 2V_c$  时，采样信号  $f(m\Delta x, n\Delta y)$  能唯一的恢复原图像信号  $f(x, y)$ 。

当二维连续函数可以完美重构时，可以由下列公式通过有限采样数据进行任一点的函数值重构：

$$f(x, y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m\Delta x, n\Delta y) \frac{\sin \frac{\pi}{\Delta x}(x - m\Delta x)}{\frac{\pi}{\Delta x}(x - m\Delta x)} \cdot \frac{\sin \frac{\pi}{\Delta y}(y - n\Delta y)}{\frac{\pi}{\Delta y}(y - n\Delta y)}$$

连续函数的傅里叶变换值可以由下列函数重构：

$$F_p(u, v) = \Delta x \Delta y \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m\Delta x, n\Delta y) e^{-j2\pi(m\Delta x u + n\Delta y v)}$$

## 图像非统一采样和量化策略

在灰度级变化尖锐的区域，用细腻的采样，在灰度级比较平滑的区域，用粗糙的采样。在边界附近使用较少的灰度级。剩余的灰度级可用于灰度级变化比较平滑的区域，避免假轮廓现象。

## 图像的基本运算

- 点运算：即逐点将原有图像映射到新值。
- 反转变换： $s = (L - 1) - r$

$[0, L - 1]$  为图像的灰度级，使得黑的变白，白的变黑

- 对数变换： $s = c \log(1 + r)$

$c$  是常数， $r \geq 0$ 。有时原图的动态范围太大，**超出某些设备的允许动态范围**，如果直接使用原图则可以丢失一部分细节。故而对原图进行灰度压缩，例如对数变换就能解决问题。

- 幂次变换： $s = cr^\gamma$  例如核磁共振图像需要  $\gamma < 1$  变亮，而航拍需要降低灰度级使图像变暗。

$c$  和  $\gamma$  是正常数， $\gamma < 1$  提高灰度级，在正比函数上方，使得图像变亮。 $\gamma > 1$  降低灰度级，在正比函数下方，使得图像变暗。**(注意这里的基本运算都是 double 形，所以取值为  $[0, 1]$ )**

- 对比度拉伸：提高图像处理时灰度级的动态范围。
- 位平面运算：实例，图像加密  $A = A - \text{mod}(A, 2) + \text{uint8}(B)$ 
  - 通过对特定位提高亮度，改善图像质量
  - 较高位（如前4位）包含大多数视觉重要数据
  - 较低位（如后4位）对图像中的微小细节有作用
  - 分解为位平面，可以分析每一位在图像中的相对重要性

## 图像间的代数运算

- 加法，去除叠加性噪声（堆栈），生成图像叠加效果

$g(x, y) = \alpha f(x, y) + \beta h(x, y)$  其中  $\alpha + \beta = 1$ 。可以得到各种图像合成的效果，也可以用于两张照片的衔接。

- 减法：显示两幅图像的差异，去除不需要的叠加性图案。
- 乘法：图像的局部展示，用二值蒙版，图像与原图像做乘法
- 非：获得反色图或补图  $g(x, y) = 255 - f(x, y)$
- 与：求两个子图像的相交子图
- 或：合并子图像
- 异或：获得相交子图像

## 离散卷积

已知  $a = [a_1, a_2, \dots, a_n]$ ,  $b = [b_1, b_2, \dots, b_m]$ ,  $\text{conv}(a, b)$  将返回  $a * b$  的向量，即

$$(a * b)_k = \sum_{i=1}^{k-1} a_i b_{k-i}, \quad 2 \leq k \leq m + 2$$

$\forall i > n, a_i = 0, \forall j > m, b_j = 0$ 。

$\text{conv}(a, b, 'same')$  将返回一个长度等于  $n$  的卷积结果，即截取上述卷积结果中间  $n$  个元素的结果，**常用于信号或图像处理问题**。若3个选2个，则默认选择后两个。

$\text{conv}(a, b, 'valid')$  将返回一个长度等于  $n - m - 1$  的卷积结果，**即仅包含  $b$  的全部元素参与卷积的中间项**。

$\text{conv2}$  为二维离散卷积的定义形式，在图像卷积特别是深度学习之中，图像的大小一般很大，卷积核的大小很小，所以使用 `same` 关键字的卷积方式往往更为常见

## 图像卷积的延拓条件

- 复制性边界条件，即用边界的值代替外面的值，例如 $[1, 2, 3, 4, 5]$ ，右侧均为5。
- 对称性边界条件，有利于傅里叶变换和预选变化的解释，即12345 54321
- 周期性边界条件最有利于傅里叶变换的解释，但往往不合实际，即 12345 12345

## 第四课：图像的傅里叶变换

求  $f(x, y) = \begin{cases} A & 0 \leq x \leq X, 0 \leq y \leq Y \\ 0 & otherwise \end{cases}$  的傅里叶变换

Proof.

$$\begin{aligned} F(u, v) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) e^{-j2\pi(u\alpha + v\beta)} d\alpha d\beta = A \int_0^X e^{-j2\pi u\alpha} d\alpha \int_0^Y e^{-j2\pi v\beta} d\beta \\ &= A \cdot \frac{e^{-2\pi juX} - e^0}{-2j\pi u} \cdot \frac{e^{-2\pi jvY} - e^0}{-2j\pi v} \\ &= AXY \cdot \frac{e^{j\pi uX} - e^{-j\pi uX}}{2j\pi u} \cdot \frac{e^{j\pi vY} - e^{-j\pi vY}}{2j\pi v} \cdot e^{-j\pi(uX + vY)} \end{aligned}$$

其幅度谱为

$$|F(u, v)| = AXY \left| \frac{\sin(\pi uX)}{\pi uX} \right| \left| \frac{\sin(\pi vY)}{\pi vY} \right|$$

## 二维离散傅里叶变换

- 尺寸为  $M \times N$  的离散图像函数的 DFT

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

- 反变换也可以对  $F(u, v)$  求 IDFT 得到

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

- 实部与虚部
- 模定义为  $|F(u, v)|$ ，相位角。
- 几个基本形式
  - 线性性质
  - 变换可分性

$$DFT[f(x, y)] = F_x\{F_y[f(x, y)]\} = F_y\{F_x[f(x, y)]\}$$

- 周期性

$$f(x, y) = f(x + mM, y + nN), \quad n, m \in \mathbb{Z}$$

- 平均值性质，即低频系数

$$F(0, 0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

- 180度旋转性质

$$DFT\{DFT[f(x, y)]\} = f(-x, -y)$$

## 一维离散傅里叶变换的简单模拟

设  $f(x, y) = \begin{cases} A & 0 \leq x \leq X, \\ 0 & otherwise \end{cases}$ ,  $X \geq N$ , 则

$$F(u) = \frac{1}{N} \sum_{x=1}^N A e^{-j2\pi x u / N} = \begin{cases} A & u = 0, \\ 0 & otherwise \end{cases}$$

即由于圆周上的点互相抵消, 最终结果为 0。

而再设  $f(x, y) = \begin{cases} A & 0 \leq x \leq X, \\ 0 & otherwise \end{cases}$ ,  $X < \frac{N}{2}$ , 且  $X$  为整数, 则

$$F(u) = \frac{1}{N} \sum_{x=1}^X A e^{-j2\pi x u / N}$$

而  $F(u)$  在  $[0, N]$  左右对称。

## 连续与离散傅里叶变换的联系

由于连续傅里叶变换幅度不符合奈奎斯特采样定律, 故而离散傅里叶变换只能近似反应连续傅里叶变换的结果。在 MATLAB 中, 定义

$$F(u, v) = \sum_{x=1}^m \sum_{y=1}^n f(x, y) e^{\frac{-2\pi j(x-1)(u-1)}{m} + \frac{-2\pi j(y-1)(v-1)}{n}}$$

其中  $1 \leq u \leq m$ ,  $1 \leq v \leq n$ , 特别的当  $(u, v) = (1, 1)$  的时候,  $F(1, 1) = \sum_{x=1}^m \sum_{y=1}^n f(x, y)$  称为离散傅里叶变换的

低频系数, 其余系数称为高频傅里叶系数。离散傅里叶逆变换定义如下:

$$F(u, v) = \frac{1}{mn} \sum_{u=1}^m \sum_{v=1}^n F(u, v) e^{\frac{2\pi j(x-1)(u-1)}{m} + \frac{2\pi j(y-1)(v-1)}{n}}$$

同样  $1 \leq x \leq m$ ,  $1 \leq y \leq n$

- 满足周期延拓性质的图像平移后, 傅里叶变换的模保持不变。

## 分片二维离散傅里叶变换

实际的自然图像, 大都在整体区域上不满足恒为常值, 也不满足具有周期纹理状的统一分布。但是往往在局部分片上满足相应形式, 因此, 分片傅里叶变换就有了很大的存在和利用价值。

## 离散卷积定理

将两个函数都用右下补零的方法即可延拓, 则它们的离散卷积为  $f_e(x, y)$  和  $g_e(x, y)$ , 则:

$$f_e(x, y) * g_e(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e(m, n) g_e(x-m, y-n)$$

则卷积定理为:

$$DFT[f_e(x, y) * g_e(x, y)] = F_e(u, v) \cdot G_e(u, v)$$

注意: 进行零延拓不仅仅是为了让傅里叶变换可以直接定义乘法, 而且可以解决卷积的边界问题,  $f(x, y)$  与  $g(x, y)$  一般并不满足周期性, 零延拓可以避免产生交叠误差, 在更大的  $M \times N$  的矩阵大小确保卷积定理成立。

## 离散傅里叶变换的其他性质

- 比例性质:  $f(ax, by) \iff \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right)$
- 频率域卷积性质:  $f_e(x, y) * g_e(x, y) \iff F_e(x, y) \cdot G_e(u, v)$
- 频移性质:  $f(x, y) e^{2\pi j \frac{u_0 x + v_0 y}{N}} \iff F(u - u_0, v - v_0)$
- 差分性质: 结合傅里叶变换的平移性质, 差分的本质就是傅里叶变换乘以某个固定的复数

## 第五课：图像的其他变换

### 一维离散余弦变换

任何实对称函数的傅里叶变换中只含余弦项，余弦变换是傅里叶变换的特例，余弦变换是简化DFT的重要方法。将一个信号通过对折延拓成实偶函数，然后进行傅里叶变换，我们就可用2N点的DFT来产生N点的DCT。

1. 以  $x = -\frac{1}{2}$  为对称轴折叠原来的实序列  $f(n)$
2. 以  $2N$  为周期将其周期延拓，得到

$$f_c(2k\pi + n) = f_c(n), \quad -N \leq n < N$$

为了计算方便，我们取  $n \in [0, 2N - 1]$  的一个周期来进行离散傅里叶变换的计算，其模可以用余弦表示：

$$\begin{aligned} F(u) &= \frac{1}{2N} \sum_{x=0}^{2N-1} f_c(x) \cdot e^{-\frac{\pi j u x}{N}} = \frac{1}{2N} \left( \sum_{x=0}^N f_c(x) \cdot e^{-\frac{\pi j u x}{N}} + \sum_{x=N+1}^{2N} f_c(x) \cdot e^{-\frac{\pi j u x}{N}} \right) \\ &= \frac{1}{2N} \left( \sum_{x=0}^N f_c(x) \cdot e^{-\frac{\pi j u x}{N}} + \sum_{x=N+1}^{2N} f_c(2N-1-x) \cdot e^{-\frac{\pi j u (2N-1-x)}{N}} \right) \\ &= \frac{1}{2N} \sum_{x=0}^N f_c(x) \left( e^{-\frac{\pi j u x}{N}} + e^{-\frac{\pi j u (2N-1-x)}{N}} \right) \\ &= \frac{1}{2N} \sum_{x=0}^N f_c(x) \left( \cos\left(-\frac{\pi u x}{N}\right) + j \sin\left(-\frac{\pi u x}{N}\right) + \cos\left(-\frac{\pi u (2N-1-x)}{N}\right) + j \sin\left(-\frac{\pi u (2N-1-x)}{N}\right) \right) \\ &= \frac{1}{N} e^{-\frac{\pi(2N-1)ju}{2N}} \sum_{x=0}^{N-1} f_c(x) \cdot \cos\left(\frac{\pi j u (2N-1-2x)}{2N}\right) \end{aligned}$$

其逆变换定义为：

$$f(x) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} C(k) F(k) \cos\left[\frac{\pi}{N} \left(x + \frac{1}{2}\right)\right]$$

其中

$$c(k) = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & 1 \leq k \leq N-1 \end{cases}$$

### 二维离散余弦变换

- 二维离散余弦变换同样是可分的
- 离散余弦变换对于图像向量是一个正交变换。变换矩阵的转置即逆变换。

#### MATLAB 版二维离散余弦变换

$$C(u, v) = a(u)b(v) \sum_{x=1}^m \sum_{y=1}^n f(x, y) \cos \frac{(2x-1)(u-1)\pi}{2m} \cos \frac{(2y-1)(v-1)\pi}{2n}$$

其中  $1 \leq u \leq m, 1 \leq v \leq n$ 。其中

$$a(u) = \begin{cases} \frac{1}{\sqrt{m}}, & u = 1 \\ \frac{2}{\sqrt{m}}, & u \neq 1 \end{cases}, \quad b(v) = \begin{cases} \frac{1}{\sqrt{n}}, & v = 1 \\ \frac{2}{\sqrt{n}}, & v \neq 1 \end{cases}$$

当  $(u, v) = (1, 1)$  时， $C(1, 1) = \frac{1}{\sqrt{mn}} \sum_{x=1}^m \sum_{y=1}^n f(x, y)$  称为离散余弦变换的**低频系数**，其余称高频。

### 二维离散沃尔什——哈达玛变换

- 沃尔什函数是一组矩形波，取值为 1 和 -1
- 沃尔什函数有三种排列或编号方式，以哈达玛排列最便于快速计算。
- 采用哈达玛排列的沃尔什函数进行的变换称为沃尔什-哈达玛变换，简称 WHT 或直称哈达玛变换

- 哈达玛矩阵：元素仅由  $+1$  和  $-1$  组成的正交方阵。
- 正交方阵：指它的任意两行（或两列）都彼此正交，或者说它们对应元素之和为零。这一点和余弦变换相同。
- 哈达玛变换要求图像的大小为  $N=2^n$
- 一维哈达玛变换核为

$$g(x, u) = \frac{1}{N} (-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)}$$

其中,  $b_k(z)$  代表  $z$  的二进制表示的第  $k$  位值。

## 哈达玛变换

- 一维哈达玛正变换为:  $H(u) = \frac{1}{N} \sum_{x=0}^{n-1} f(x) (-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)}$
- 一维哈达玛反变换为:  $f(x) = \sum_{u=0}^{n-1} H(u) (-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)}$
- 二维哈达玛正反变换为

$$H(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) (-1)^{\sum_{i=0}^{n-1} [b_i(x) b_i(u) + b_i(y) b_i(v)]}$$

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} H(u, v) (-1)^{\sum_{i=0}^{n-1} [b_i(x) b_i(u) + b_i(y) b_i(v)]}$$

- 二维哈达玛正、反变换也具有相同形式
- 正反变换都可通过两个一维变换实现
- 高阶哈达玛矩阵可以通过如下方法递推求得

$$H_N = \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix}$$

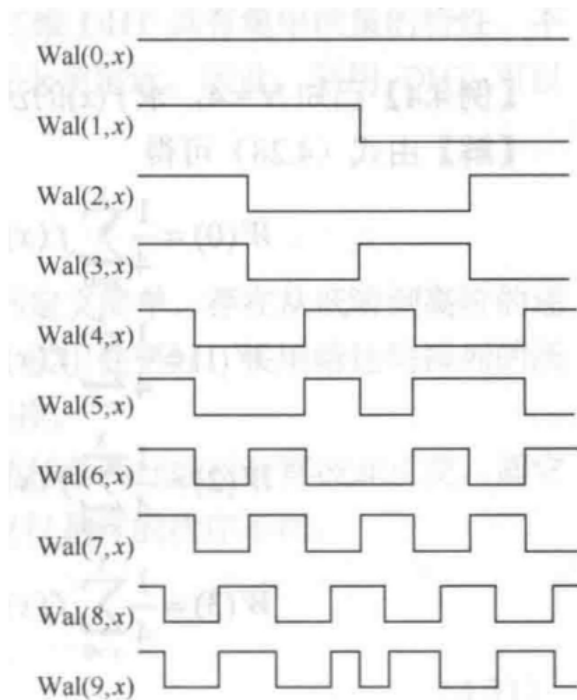
- $N = 2$  的最简单哈达玛矩阵为

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- 每一个  $1, -1$  的交错信号，我们称符号从左到右变化的次数为列率。
- 哈达玛变换矩阵，其列率排列是无规则的。
- 将无序的哈达玛核进行列率的排序，就称为**有序沃尔什 (Walsh) 变换**，其系数可以反映图像频率。列率即为正负号变化的次数，例如  $1 -1 1 -1$ ，其中变换了三次，相应的就会放在  $N = 4$  的最下面。



# 一维离散沃尔什变换的图示



- 二维沃尔什变换和哈达玛变换都是可分的。
- 二维沃尔什变换的矩阵形式可以理解为  $W = \frac{1}{N^2} G f G$
- 二维沃尔什逆变换矩阵形式为  $f = \frac{1}{N^2} G W G$

## 例子

考虑  $N = 4$  的沃尔什变换

哈达玛核为

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

而后我们恰好来练习一下如何计算列率，注意到列率从上到下依次为0,3,1,2，故而排序后得到沃尔什变换

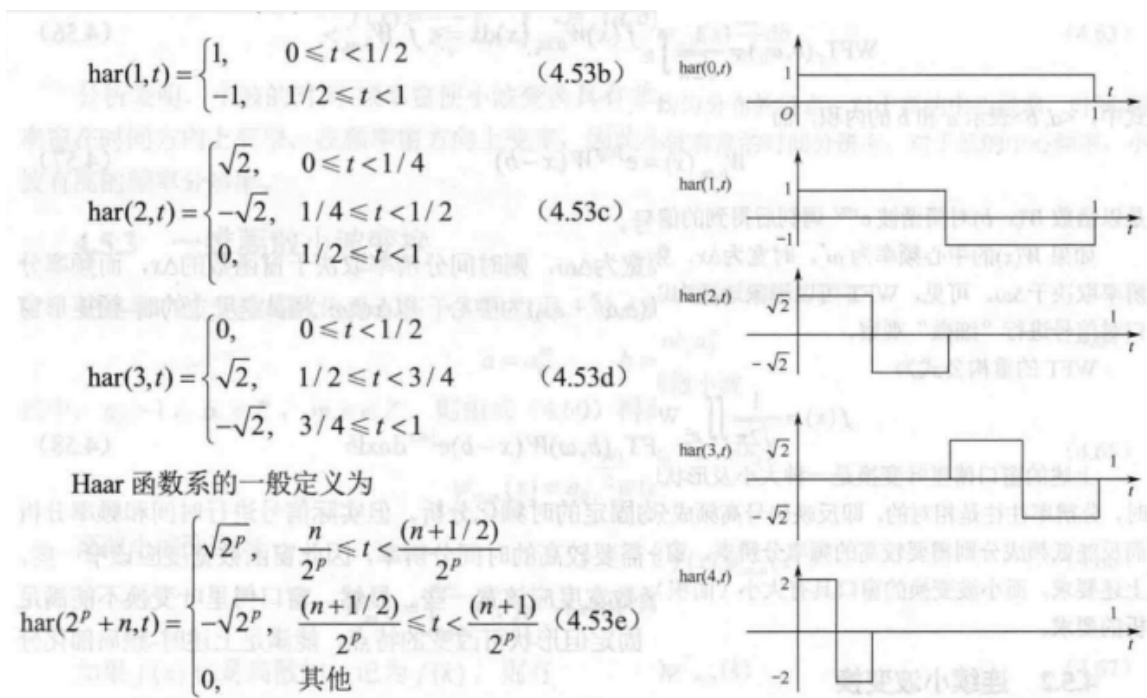
$$G = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

而后我们作用在  $f$  上得到

$$W(0) = \frac{1}{4}[f(0) + f(1) + f(2) + f(3)]$$
$$W(2) = \frac{1}{4}[f(0) + f(1) - f(2) - f(3)]$$

## 哈尔变换

哈尔变换的定义是基于**哈尔函数系**  $\text{har}(2^p + n, t)$ ,  $p = 1, 2, \dots$ ,  $n = 0, 1, \dots, 2^p - 1$  的函数内积来定义的。哈尔函数的参数  $p$  代表收缩情况， $t$  代表整数平移情况。



## 加窗傅里叶变换

即连续函数的**局部傅里叶变换**，设定实值偶函数 $W(x)$ ，通常有紧支集，并定义加窗傅里叶变换为

$$\text{WFT}_f(b, \omega) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(x) W^*(x - b) e^{-j\omega x} dx$$

其中 $\omega$ 为频率域自变量， $b$ 代表平移量。**WFT**也称为**短时傅里叶变换**。离散化之后对一维信号可以用于声音音调识别，对二维图像还可以进行局部分析。

缺点是窗口的大小不能改变，有点事可以平移选择多个窗口分别进行频率分析。光滑性鱿鱼哈尔变换但是变换的结构性不如哈尔变换。

## 一维连续小波变换

基于**加窗傅里叶变换**。为了增加不同的伸缩等级，即可定义面向连续函数的小波变换。

首先定义小波母函数 $\psi(x)$ ，满足条件 $C_\psi = \int_{\mathbb{R}} \frac{|\hat{\psi}(\omega)|}{|\omega|} d\omega < \infty$ 且 $\hat{\Psi}(0) = 0$ ，其中 $\hat{\psi}(\omega)$ 为 $\psi(x)$ 的傅里叶变换。而哈尔变换就是特殊的小波变换，即哈尔函数即为小波母函数。这里ppt表述有误

定义小波母函数后，可定义小波函数族 $\psi_{a,b}(x) = |a|^{-\frac{1}{2}} \psi\left(\frac{x-b}{a}\right)$ ，其中 $a$ 为伸缩因子， $b$ 为平移因子。分别与 $L^2$ 函数 $f$ 计算内积，即定义了如下的小波变换：

$$W_f(a, b) = \langle f, \psi_{a,b}(x) \rangle = |a|^{-\frac{1}{2}} \int_{\mathbb{R}} f(x) \psi^*\left(\frac{x-b}{a}\right) dx$$

这里的等号在ppt中表述有误

而将连续小波变换中的参数 $a, b$ 进行可列个点的取值，即可构成离散小波变换。设 $a = a_0^m$ ， $b = nb_0 a_0^m$ ，其中 $a_0 > 1$ ， $b_0 \in \mathbb{R}$ ， $m, n \in \mathbb{Z}$ 。更常见的以 $a_0 = 2$ ， $b_0 = 1$ 。则小波函数族和小波变换分别表示为：

$$\begin{aligned} \psi_{m,n}(x) &= 2^{-\frac{m}{2}} \psi(2^{-m}x - n) \\ W_{m,n} &= \int_{-\infty}^{\infty} f(x) \psi_{m,n}^*(x) dx = \langle f(x), \psi_{m,n}(x) \rangle \end{aligned}$$

若函数 $f(x)$ 离散定义在整数点，则 $W_{m,n} = \sum_k f(k) \psi_{m,n}^*(k)$ 。这种方式往往理解成 $f(k)$ 定义成了某些狄拉克函数及其平移的线性组合所构成的特殊函数（泛函）。

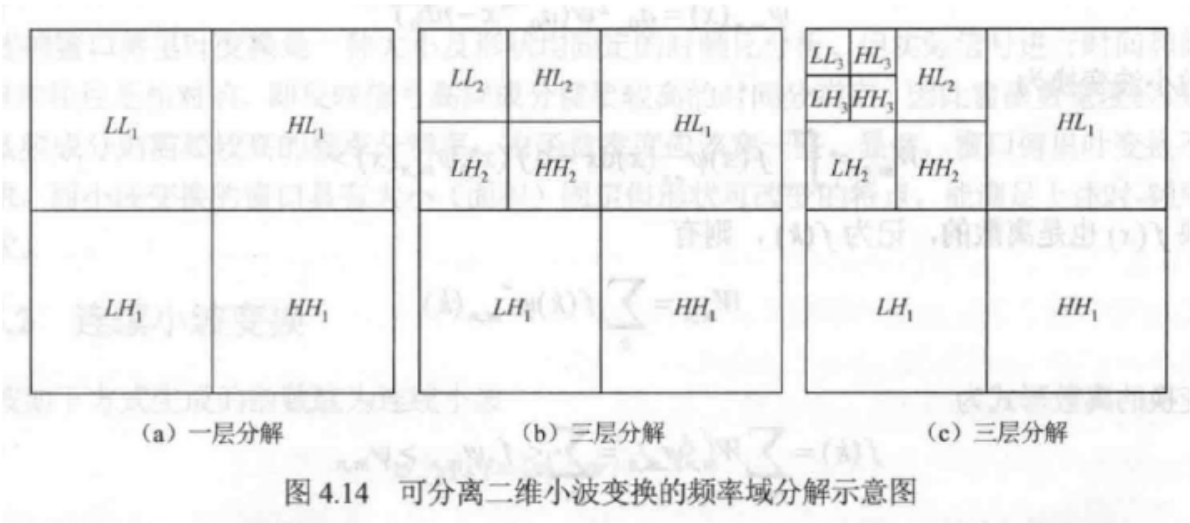
若 $\psi_{m,n}$ 可以构成正交基，逆变换即可定义为

$$f(k) = \sum_{m,n} W_{m,n} \psi_{m,n} = \sum_{m,n} \langle f, \psi_{m,n} \rangle \psi_{m,n}$$

## 二维离散小波变换 DWT

- 二维离散小波变换通常是通过张量积的方式由横、纵两个方向的小波变换结合而成，并可定义其逆变换。
- 二维离散小波变换的离散化形式可以直接理解并应用于图像矩阵的处理。
- 尺度函数的一维形式并不唯一，可以是矩形脉冲函数、山形函数或其他高次的B样条函数等等

□ 二维离散小波可以进行多层分解，一种常见的正交化小波分解，在每一层分解后会进行下采样（信息量减少），其分解结构图如下：



## 第六课：图像的数字水印与编码

基于位平面运算的数字水印 鲁棒性不好 水印容易被简单的滤波算法模糊化甚至消除

### 衡量标准

- 不可见性：不会引起明显的降低质量，不易被察觉
- 鲁棒性：经历多种信号处理过程中，数字水印仍能保持完整性或被准确识别

### 分类

- 水印的特性：鲁棒数字水印和脆弱数字水印
- 水印的检测过程：明文水印和盲水印
- 水印内容：有意义水印和无意义水印
- 隐藏位置：时域水印、频率域水印、时空域数字水印

### 一般步骤

嵌入、传播和抽取

### LSB 方法

任何一幅图片都具备一定的容噪性，像素数据的最低有效位（LSB）对人眼的视觉影响很小，秘密信息就隐藏在图像每一个像素的最低位或次低位，实现不可见性。

对于256色灰度图像，隐藏在第八位（最低有效位）

把水印分别嵌入到图像像素的不同位，越低位嵌入，人眼越难识别

DCT 域加入水印方法

先计算DCT，然后将水印叠加到DCT域中幅值最大的前k系数上(不包括直流分量)，通常为图像的低频分量。若DCT系数的前k个最大分量表示为  $D = \{d_i\}$ ，水印服从高斯分布的随机实数序列  $W = \{w_i\}$ ，那么水印的嵌入算法为：

$$D_i = d_i(1 + \alpha w_i)$$

用新的系数做反变换即可得到水印图像  $I_w$ 。

考虑  $\alpha = 2$  的情况

```
Ac = dct2(A), Bc = dct2(B);
ind = abs(Ac) >= 100;
Ac(ind) = Ac(ind) + 2 * Bc(ind);
A_revised = idct2(Ac);
B_revised = dct2(uint8(A_revised)) - Ac;
```

基于DWT的数字水印方法

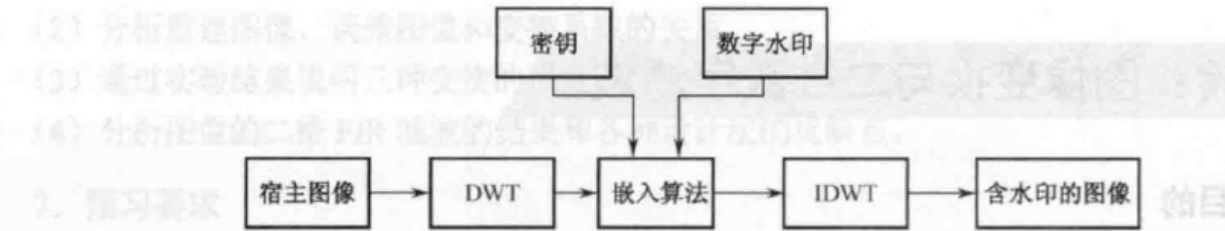


图 4.21 小波域水印嵌入的一般框图

图 4.22 所示为小波域水印检测的一般框图。

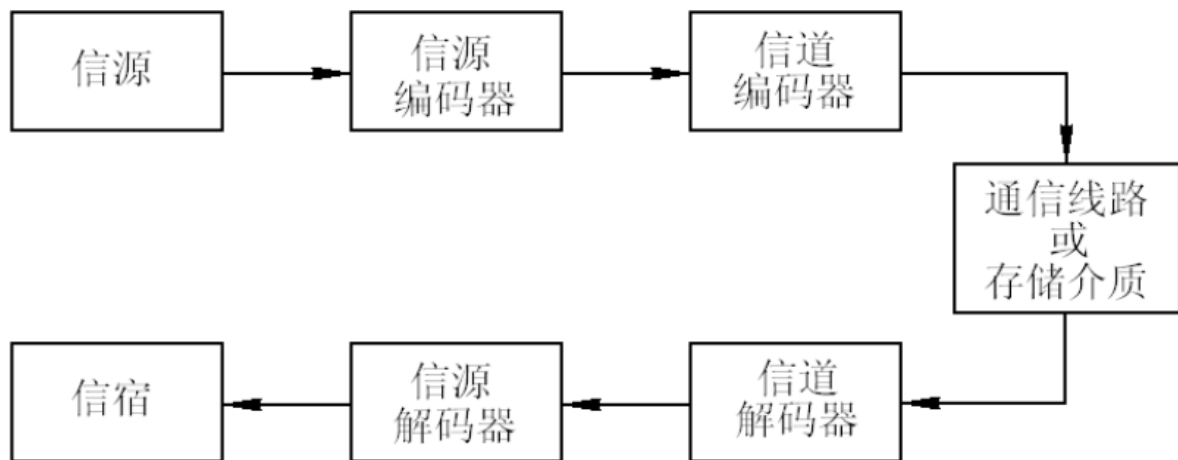


图 4.22 小波域水印检测的一般框图

数据编码与数据压缩

数据编码：信息保密、信息的压缩存储与传输

数据压缩：以较少的数据量表示信源以原始形式所代表的信息，目的在于节省存储空间、传输时间、信号频带和发送能量等。



## 信息熵

- 代表信源所含的平均信息量。
- 若信源编码的熵大于信源的实际熵，则信源中的数据一定存在冗余度冗余数据的去除不会减少信息量。
- 信息量与数据量的关系可由下式表示

$$I = D - du$$

其中,  $I, D, du$  分别表示信息量、数据量和冗余量

## 图像数据量的完整表示

$$V = w \cdot h \cdot d / 8$$

$V, w, h, d$  分别表示图像数据量 (字节, byte, B)、图像宽度 (像素数, pel)、图像高度 (像素数, pel)、图像深度 (位, bit)

图像种类	图像参数	数据量
二值传真图像	A4 (210 × 297 mm) 大小、1728 × 2376 × 2色分辨率	501 KB
灰度图像	512×512, 8 bit灰度等级	256 KB
VGA图像	640 × 480 × 256色	300 KB
CIF视频图像	352 × 288 × 256色, 亮度取样率为3 MHz, 亮度和两色差按4 : 1 : 1取样, 亮色量化位数共12 bit, 帧频29.97, 按1 s计算	4.3 MB
HDTV亮度信号	1280 × 720, 量化位数为8 bit, 帧频30 Hz, 按1 s计算	52.7MB

## 压缩编码的可行性

一般图像中存在着以下数据冗余因素:

- 编码冗余信息编码存在大量的重复子结构或其他冗余结构
- 像素间的相关性形成的冗余自然图像普遍呈分片光滑状, 可以“成片”表示
- 视觉特性和显示设备引起的冗余
- 人眼对小的误差不敏感, 用于有损压缩

## 图像熵:

假定图像的灰度级为  $x_i$ , 且  $x_i$  出现的概率为  $p(x_i)$ , 则图像熵可以定义为

$$H(x) = - \sum_{i=1}^M p(x_i) \log_2 p(x_i)$$

图像熵值在  $p(x_i) = 1/M$  时达到最大值  $\log_2 M$ , 在所有灰度全部取相同值的时候达到最小值 0, 因此熵值反映了图像或编码分配的平均程度。而当  $M = 2^L$  时, 可以定义平均码长

$$R(x) = \sum_{i=1}^M p(x_i) L_i$$

$L_i$  表示分配给灰度级  $x_i$  的码字的长度。(例如用Huffman编码等)

## 技术指标

- 图像冗余度与编码效率: 图像冗余度  $r = 1 - \frac{H(x)}{R(x)}$ , 而定义  $\eta = \frac{H(x)}{R(x)} = \frac{1}{1+r}$  为编码效率。
- 压缩比: 定义为平均码长的比例  $C_r = \frac{\bar{r}_b}{\bar{r}_c}$ , 也可以理解为总码长的比例, 或者是总的空间占有的比例。
- 客观保真度评价: 用于有损压缩, 均方误差以及信噪比SNR, 峰值信噪比PSNR。

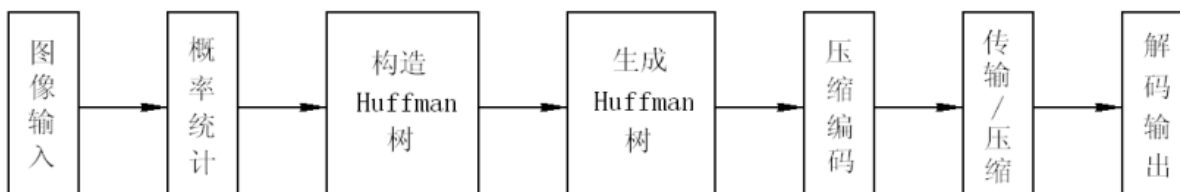
## Huffman 编码

优点:

- 实现Huffman编码的基础是统计源数据集中各信号的概率分布。
- Huffman编码在无失真的编码方法中效率优于其他编码方法(平均码长最短), 是一种最佳变长码, 其平均码长接近于熵值。

缺点:

- 当信源数据成分复杂时, 庞大的信源集致使Huffman码表较大, 码表生成的计算量增加, 编译码速度相应变慢
- **不等长编码**致使硬件译码电路实现困难。上述原因致使Huffman编码的实际应用受到限制



## 香农编码

Shannon提出了将信源符号依其概率降序排列, 用符号序列累积概率的二进制表示作为对信源的唯一可译编码。

其应用于图像编码的步骤如下:

1. 将N个灰度级  $x_i$  按其概率递减进行排列。
2. 求概率分布  $p_i$  的第  $i$  个灰度级的二进制位数  $n_i$ 。

$$-\log_2 p_i \leq n_i \leq -\log_2 p_i + 1$$

3. 计算与  $p_i$  相对应的累积概率  $P_i$ , 把与  $P_i$  相对应的二进制码和接下去与  $p_k$  ( $k > i$ ) 相应的码相比较, 前面的  $n_i$  位至少有一位以上的数字是不同的。

# Shannon 编码原理

Shannon 编码是一种基于信息论的编码方法，它根据每个符号的概率分布来确定码字的长度。一般来说，概率较高的符号会分配较短的码字，概率较低的符号会分配较长的码字，以达到压缩数据的目的。下面是具体的步骤和示例推导。

## Shannon 编码步骤：

1. 计算累积概率：
- 对于每个符号，计算其出现概率的累积概率  $P_i$ 。
2. 确定位数：
- 根据符号的概率  $p_i$ ，计算其码字的长度  $L_i = \lceil -\log_2 p_i \rceil$ 。
3. 生成码字：
- 对于每个符号  $x_i$ ，确定其码字是累积概率  $P_i$  的前  $L_i$  位二进制表示。

## 具体示例推导

假设有一信源，其符号集  $x_i$  及其概率分布如下：

符号 $x_i$	概率分布 $p_i$	累积概率 $P_i$	位数 $L_i$	编码
$x_1$	0.40	0	2	00
$x_2$	0.20	0.40	3	011
$x_3$	0.12	0.60	4	1001
$x_4$	0.11	0.72	4	1011
$x_5$	0.09	0.83	4	1101
$x_6$	0.08	0.92	4	1110

### 1. 计算累积概率 $P_i$ ：

累积概率  $P_i$  是符号  $x_i$  及之前所有符号出现概率的累积和。我们从第一个符号开始逐一计算：

- $P_1 = 0$
- $P_2 = 0 + 0.40 = 0.40$
- $P_3 = 0.40 + 0.20 = 0.60$
- $P_4 = 0.60 + 0.12 = 0.72$
- $P_5 = 0.72 + 0.11 = 0.83$
- $P_6 = 0.83 + 0.09 = 0.92$

### 2. 确定位数 $L_i$ ：

对于每个符号，位数  $L_i$  通过计算其概率  $p_i$  的负对数并取上整数：

- $L_1 = \lceil -\log_2 0.40 \rceil = 2$
- $L_2 = \lceil -\log_2 0.20 \rceil = 3$
- $L_3 = \lceil -\log_2 0.12 \rceil = 4$
- $L_4 = \lceil -\log_2 0.11 \rceil = 4$
- $L_5 = \lceil -\log_2 0.09 \rceil = 4$
- $L_6 = \lceil -\log_2 0.08 \rceil = 4$

### 3. 生成码字：

对于每个符号  $x_i$ ，其码字是累积概率  $P_i$  的前  $L_i$  位二进制表示：

- $P_1 = 0$ ，前2位为 "00"
- $P_2 = 0.40$ ，前3位为 "011"



- $P_3 = 0.60$ , 前4位为 "1001"
- $P_4 = 0.72$ , 前4位为 "1011"
- $P_5 = 0.83$ , 前4位为 "1101"
- $P_6 = 0.92$ , 前4位为 "1110"

Shannon 编码通过考虑符号概率的累积概率生成对应的码字，旨在使得平均码长尽可能接近熵值，从而达到较高的编码效率。

## 算术编码

算术编码是一种无损数据压缩算法，在信源各符号概率接近的条件下，算术编码是一种优于Huffman编码的方法。它的主要思路是利用实数（二进制小数）的方法来记录一组数据流。传输完成后再将实数恢复为数据流。

### 算术编码的步骤：

1. **初始化区间：**
  - 算术编码通常以  $[0, 1)$  设定为初始区间，并假设概率空间宽度  $A_0 = 1$ , 下限  $C_0 = 0$ , 上限  $D_0 = 1$ 。
2. **根据符号概率更新区间：**
  - 根据符号出现的概率分配子区间  $[p_{il}, p_{ih})$ ，对应概率值  $p_{ih} - p_{il}$ 。
  - 对每一位符号  $s_n$ ，更新概率空间的下限和上限值：
$$C_n = C_{n-1} + A_{n-1} \cdot p_{nl}$$
$$D_n = C_{n-1} + A_{n-1} \cdot p_{nh}$$
  - 此后将区间  $[C_n, D_n)$  设为新的当前区间，重复迭代。

### 算术编码示例分析

我们使用给定的信源概率分布  $X = \begin{bmatrix} 0 & 1 \\ \frac{2}{5} & \frac{3}{5} \end{bmatrix}$  对二进制序列 01011 进行算术编码。下面是具体的步骤和分析。

#### 信源概率分布

- 符号 0 出现的概率  $p(0) = \frac{2}{5}$
- 符号 1 出现的概率  $p(1) = \frac{3}{5}$

#### 编码步骤

我们根据序列 01011 来进行编码：

Step	$x$	$s$	下限 $C$	上限 $D$
1	0	0	0	$\frac{2}{5}$
2	1	$0 + \frac{2}{5} \times \frac{2}{5} = \frac{4}{25}$	$\frac{4}{25}$	$\frac{6}{25}$
3	0	$\frac{2}{5} + 0 \times \frac{6}{25} = \frac{4}{25}$	$\frac{4}{25}$	$\frac{12}{125}$
4	1	$\frac{4}{25} + \frac{2}{5} \times \frac{12}{125} = \frac{124}{625}$	$\frac{124}{625}$	$\frac{36}{625}$
5	1	$\frac{124}{625} + \frac{2}{5} \times \frac{36}{625} = \frac{692}{3125}$	$\frac{692}{3125}$	$\frac{108}{625}$

#### 具体步骤详解

1. **初始区间：**
  - 初始化区间为  $[0, 1)$
  - 第一步编码符号 0，对应的概率为  $p(0) = \frac{2}{5}$
  - 更新区间：
    - 下限：0
    - 上限：  $\frac{2}{5}$
2. **第二步编码符号 1：**



- 当前区间:  $[0, \frac{2}{5})$
- 符号 1 的累积概率区间:  $[0.4, 1)$
- 更新区间:
  - 下限:  $0 + \frac{2}{5} \times \frac{2}{5} = \frac{4}{25}$
  - 上限:  $0 + \frac{2}{5} \times \frac{3}{5} = \frac{6}{25}$

### 3. 第三步编码符号 0:

- 当前区间:  $[0, \frac{2}{5})$
- 符号 0 的累积概率区间:  $[0, \frac{2}{5})$
- 更新区间:
  - 下限:  $\frac{2}{5} + 0 \times \frac{6}{25} = \frac{4}{25}$
  - 上限:  $\frac{4}{25} + \frac{2}{5} \times \frac{12}{125} = \frac{12}{125}$

### 4. 第四步编码符号 1:

- 当前区间:  $[0, \frac{2}{5})$
- 符号 1 的累积概率区间:  $[0.4, 1)$
- 更新区间:
  - 下限:  $\frac{4}{25} + \frac{2}{5} \times \frac{12}{125} = \frac{124}{625}$
  - 上限:  $\frac{4}{25} + \frac{2}{5} \times \frac{36}{625} = \frac{36}{625}$

### 5. 第五步编码符号 1:

- 当前区间:  $[0, \frac{2}{5})$
- 符号 1 的累积概率区间:  $[0.4, 1)$
- 更新区间:
  - 下限:  $\frac{124}{625} + \frac{2}{5} \times \frac{36}{625} = \frac{692}{3125}$
  - 上限:  $\frac{108}{625}$

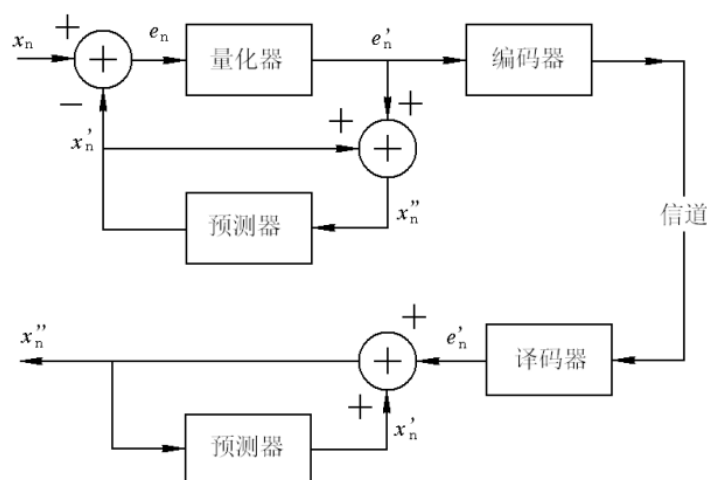
## 第七课：图像压缩

### 预测编码

在某种模型的指导下，根据过去的样本序列推测当前的信号样本值，然后用实际值与预测值之间的误差值进行编码。如果模型与实际情况符合得比较好且信号序列的相关性较强，则误差信号的幅度将远远小于样本信号。

#### 差分脉冲编码调制/Differential Pulse Code Modulation/DPCM

#### DPCM 系统组成结构



## 线性预测编码 有损

此处 ppt 存在问题

假设经扫描后的图像信号  $x(t)$  是一个**均值为零**、方差为  $\sigma^2$  **平稳随机过程**。线性预测就是选择  $a_i$  ( $i = 1, 2, \dots, N-1$ ) 使得预测值为  $x'_n = \sum_{i=1}^{N-1} a_i x_i$ ，并且使得差值的均方值最小。即 MSE 均方误差  $E\{e_n^2\} = E\{(x_n - x'_n)^2\}$ 。

- 最小 MSE 准则，可以令对  $a_i$  的 Partial 为 0，即  $\frac{\partial E(e_n^2)}{\partial a_i} = 0$
- 定义  $x_i$  与  $x_j$  的自相关函数

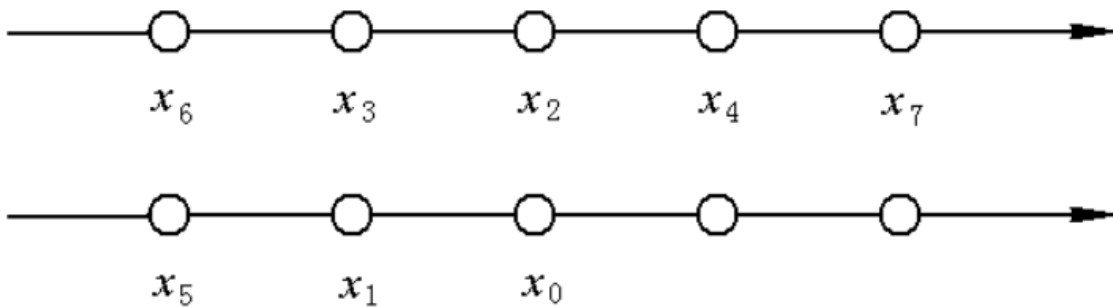
$$R(i, j) = E\{x_i, x_j\} = R(i - j) = R(j - i)$$

写成矩阵形式为 Yule-Walker 方程组  $R(i) = \sum_{k=1}^{N-1} a_k R(k-1)$

$$\begin{bmatrix} R(0) & R(1) & \cdots & R(N-2) \\ R(1) & R(0) & \cdots & R(N-3) \\ \vdots & \vdots & \cdots & \vdots \\ R(N-2) & R(N-3) & \cdots & R(0) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{N-1} \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(N-1) \end{bmatrix}$$

图像的相关性越强，压缩效果越好。而当某个阶数已经使得  $E\{e_N, e_{N-1}\} = 0$  时即使再增加预测点数，压缩效果也不可能继续提高。

若  $\{x_i\}$  为平稳的  $m$  阶 Markov 过程，则  $m$  阶线性预测器就是在 MMSE 意义下的最佳预测期。



## 预测器方案

- 前值预测：用同一行的临近像素来预测， $\hat{x} = x_1$
- 一维预测：如上图的  $x_1$ ,  $x_5$
- 二维预测
- 三维预测：用于视频压缩，还会使用前几帧的同位置或临近像素值进行预测。

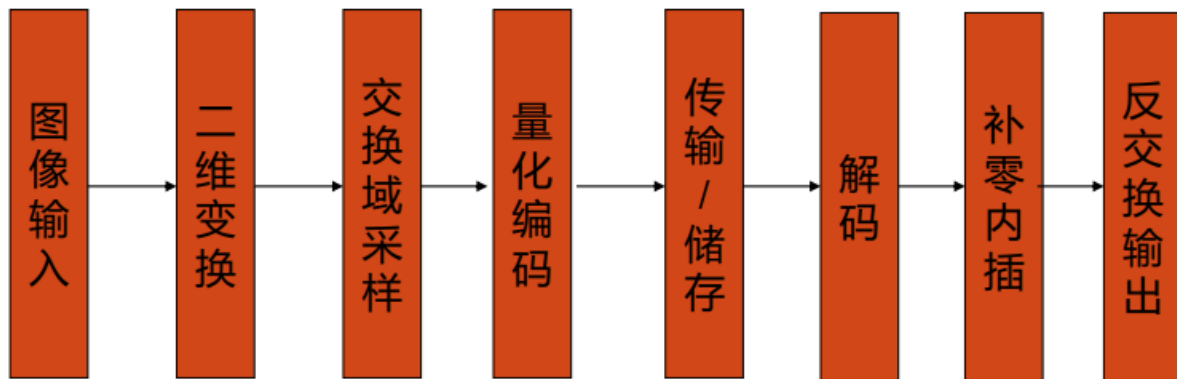
## 自适应预测编码

- 自适应预测：预测参数根据信号的统计特性来确定，以达到最佳预测，适用于信号不够平稳的情况
- 预测编码的优点：直观快捷、便于实现
- 预测编码的缺点：压缩比不够高

## 变换编码

通过数学变换可以改变信号能量的分布，从而压缩信息量。

以傅里叶变换的概念说明合理的变换可以改变信号能量分布的基本原理。



## 变换编码的实现

在变换编码中有以下几个问题值得注意：

- 图像变换方法的选取
- 子图像大小的选取
- 常用的图像编码方法
  - 区域编码
  - 阈值编码（DCT 系数小于阈值则记为0）
  - 混合编码

## 整数小波变换 无损

传统的 DCT 与经典小波变换在图像变换后会产生浮点数，因此必须进行量化处理。产生了失真。

新一代的整数小波变换（又叫第二代小波变换）采用提升方法能够实现整数变换，因而能够实现图像的无损压缩，显然它是一种很适合于医学等图像的压缩方法。

新的静态图像压缩标准 JPEG2000 中采用了基于提升方法的整数小波变换。

### 1. 分裂

将一原始信号序列  $s$  按偶数和奇数分成两个较小，互不相交的小波子集  $s_{j-1}$  和  $d_{j-1}$

### 2. 预测

定义一个预测算子  $P$ ，用  $P(s_{j-1})$  来预测  $d_{j-1}$ ，这样可以用相邻的偶数序列来预测奇数序列。也可以用平均值代替中点值。

### 3. 更新

构造一个  $U$  算子来更新  $s$ ，使之保持原有的数据集的某些特性。

逆变换则反向运算，先更新  $s$  再更新  $d$  即可。

## 二值图像编码

只用白（0）和黑（1）的两个灰度级称之为二值图像。

## 跳跃空白编码 无损

用于0较多的图。将图像的每一条扫描线分成若干等长的段，每段有  $m$  个像素，一般  $m = 8 - 12$ 。

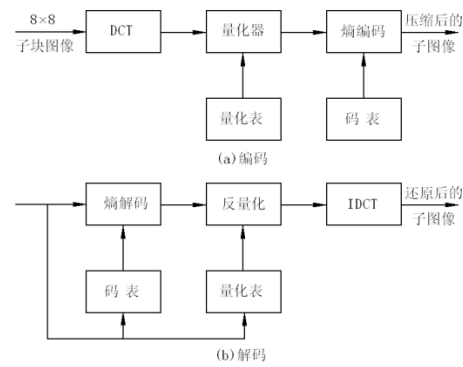
扫描线段可能出现的两种情况：

1. 全是0像素，这种线段称为 空白块 blank，常表示背景成分，编码时空白块用码字0表示
2. 全是1像素或者由01组成：用1加直接编码表示

游程长度编码RLC 无损

它改变连续出现相同字符的表达方式，以降低码长。一个符号代表数据，一个符号代表串长。参考数据库中此部分。

JPEG 有损 & 无损



每个单独的彩色图像分量的编码算法：

- 1. 带压缩图像分成若干个8 × 8的子块，作基于子块的DCT
- 2. 根据最佳视觉特性构造量化表，设计自适应量化器并对DCT的频率系数进行量化
- 3. 为了增加连续的0系数的个数，对量化后的数据进行Z字形重排
- 4. Huffman码作变字长熵编码器进行编码，进一步压缩

彩色空间转换问题——YCbCr表示

YCbCr颜色空间将颜色信息分为亮度（Y）和色度（Cb和Cr）分量：

- Y（亮度）：表示图像的亮度信息，范围通常是16到235。
- Cb（蓝色差）：表示颜色信息中蓝色与亮度的差异。
- Cr（红色差）：表示颜色信息中红色与亮度的差异。

量化—最佳的亮度量化表和色度量化表，Z字形重排

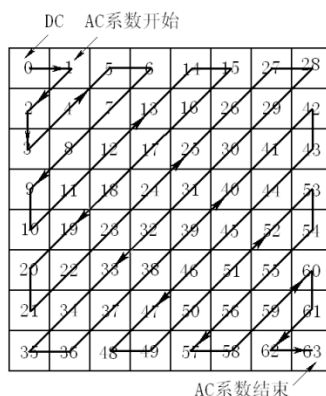
亮度量化表

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

色度量化表

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

## Z字形排列（系数频率渐低）



### JPEG的视频应用

- 尽管JPEG标准是基于彩色静止图像压缩而提出的，但对于压缩视频的帧内图像也十分有效。每秒可处理2700万个像素的单片JPEG编/解码器芯片，已能实时处理常规电视图像。
- M-JPEG（Motion-JPEG，MPEG）技术即运动静止图像压缩技术。
- 这种技术广泛应用于可精确到帧编辑和多层图像处理的非线性编辑领域。
- M-JPEG的压缩和解压缩是**对称的**，可由相同的硬件和软件实现。

### 压缩标准

JPEG2000还将彩色静态画面采用的JPEG编码方式与2值图像采用的JBIG编码方式（本次课不细讲，一种**无损**编码）统一起来，成为对应各种图像的通用编码方式。

### JPEG主要特征

**JPEG 2000放弃了以DCT为主的区块编码方式。采用以小波转换为主的多解析编码方式**

1. 高压率率。
2. 无损压缩和有损压缩。
3. 渐进传输。
4. 感兴趣区域压缩。
5. 码流的随机访问和处理。
6. 容错性。
7. 开放的框架结构。
8. 基于内容的描述。

### 基本框架

1. 对原始图像进行 DWT 离散小波变换
2. 变换后的小波系数进行量化
3. 对量化后的数据熵编码
4. 最后形成输出码流

解码器是编码器的逆过程

## 第八课：图像的模式识别

### 主成分分析

对数据进行中心化，再分析自身的协方差矩阵，即 $(A - \bar{A})(A - \bar{A})^T$ 进行特征值分解，该矩阵为对称半正定矩阵，取其最大的2-3个特征值，并获取对应特征向量上的系数，即可完成数据的降维操作。

## KNN方法

观察未知标签图像最接近的若干图像是什么标签，可以完成对未知图标签的简单估计，该方法称为  $k$  临近（KNN）方法。

## SVM 支持向量机

更具鲁棒性。找出训练集的一个分割超平面，有很多个超平面时，需要确定一个鲁棒性最强的，即离两类的点的距离最远的。

而SVM模型通常通过下列优化问题求得：

$$\min_{\vec{w}, \vec{b}} \quad \frac{1}{2} \|\vec{w}\|^2$$
$$s.t. \quad y_i(\vec{w}^T \vec{x}_i + \vec{b}) \geq 1$$

## 深度学习

在最近的十几年，深度学习几乎统治了图像的模式识别。它所使用的非线性变换，进行多层的（**卷积+非线性激活函数**）的混合操作，从而可以得到很多抽象但也难以解释数据。这些数据往往还会进行**下采样或筛选（池化）**，成为上图中的预处理数据，最终以**线性拟合**（全连接层）的方式和输出对接。卷积核的定义需要训练。

### 神经网络 NN 模型

- 由若干个神经元组成的单向传播的网络
- NN中的每个神经元通常是简单的处理单元，采用一个或多个输入并产生输出。每个输入都带有一个权重，用来修改输入的强度，之后神经元将所有修改过后的输入加在一起，并通过激活函数（activation function）计算要传递给下一层神经元的输出。

### 激活函数

- ReLU  $\max(0, x)$
- Sigmoid  $\sigma(x) = 1/(1 + e^{-x})$

### 池化层

池化操作一般都伴随着数据的筛选和减少，常见的池化方法包括取最大值（最常见）、取最小值、取平均值等。

### LeNet

LeNet最初的定义比较简单，其结构为**卷积-池化-卷积-池化-卷积-全连接**的层级结构。后续的神经网络在层级上会更加丰富，或使用了不同的卷积定义，或使用了不同的池化（数据筛选）方法。结构先走向复杂，又趋于稳定。

## 第九课：图像增强（上）

使用图像处理算法，使观察者能够更方便的观察出图像的灰度色彩变化、整体结构框架、边界纹理细节等信息的处理方式。

## 技术分类

- 空间域：
  - 点运算：通常建立图像灰度/颜色的一种映射，使图像根据原有灰度或颜色变成一种新的对应灰度或颜色，达到更改整体亮度/对比度/颜色风格的效果
  - 线性/仿射运算：点乘和卷积，前者灰度增强，后者平滑或锐化增强。
- 频率域增强：频率域增强往往是先用DFT或DCT转化到频率域，然后重复上面的操作，最后逆变换。

## 灰度增强

对比度的放缩、暗光图的增强，或对局部区域个别灰度窗口进行强化显示的增强

前述的傅里叶变换模取对数就是一种对比度减小的图像增强，旨在显示更多的系数区分度。*防止屏幕不兼容导致色彩出现问题？*

## 朴素方法和标定方法

设理想真实的图像为  $f(i, j)$ ，实际获得的含噪声的为  $g(i, j)$ ，则有  $g(i, j) = e(i, j)f(i, j)$ 。 $e(i, j)$  为使理想图像发生畸变的比例因子。知道了  $e(i, j)$  就可以求出不失真图像。

标定系统失真系数的方法：采用一幅灰度级为常数  $C$  的图像成像，若经成像系统的实际输出为  $g_c(i, j)$ ，则有  $g_c(i, j) = e(i, j)C$ ，可以得到比例因子  $e(i, j) = g_c(i, j)C^{-1}$ 。而后实际图像校正后即可恢复。

但是乘了一个系数，校正之后可以出现**溢出现象**。灰度级值可能超过某些记录器件或显示设备输入信号的动态可范围，需再作适当的灰度变换，最后对变换后的图像进行量化。

经灰度级校正后的图像灰度值不一定在原降质图像的量化值上，因此须对变换后的图像重新进行量化。朴素模型推广为 **Retinex模型**

$$I(x, y) = R(x, y) \times L(x, y)$$

其中  $L(x, y)$  为入射图像， $R(x, y)$  为反射分量， $I(x, y)$  为人眼观察到的反射图。 $R(x, y)$  对应前述  $g(i, j)$ 。

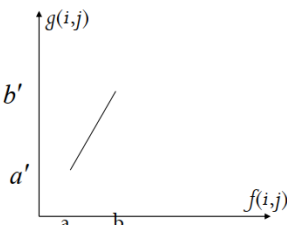
## 灰度变换 扩展或压缩动态范围

为了将图像灰度级的整个范围或一段范围扩展或压缩到记录或显示设备的动态范围，使图像变得更加清晰、图像上的特征更加明显。

灰度变换可分为**线性变换**，**分段线性变换**、**非线性变换**

### 线性变换

即一次函数进行插值

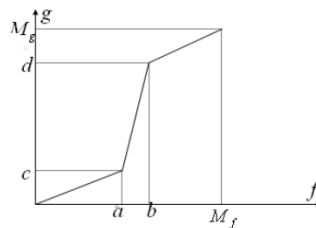
$$g(i, j) = a' + \frac{b' - a'}{b - a} [f(i, j) - a]$$


## 分段线性灰度变换 突出目标区间

突出感兴趣的目标或灰度区间，相对抑制那些不感兴趣的灰度区间。常用的是三段线性变换

$g(i, j)$ 与 $f(i, j)$ 的变换公式为：

$$g(i, j) = \begin{cases} \left(\frac{c}{a}\right) f(i, j), & 0 \leq f(i, j) < a \\ \frac{d-c}{b-a} \cdot (f(i, j) - a) + c, & a \leq f(i, j) < b \\ \left[\frac{M_g - d}{M_f - b}\right] \cdot (f(i, j) - b) + d, & b \leq f(i, j) \leq M_f \end{cases}$$



仔细调整折线拐点的位置及控制分段直线的斜率，可以对图像的任一灰度区间进行拉伸或压缩。在某些应用场合，可将过黑或过白的灰度级压缩成一个灰度级。

## 非线性灰度变换 改变对比度

利用非线性函数，例如指数与对数进行变换。利用对数变换降低对比度，指数函数提高对比度

$$g(i, j) = a + c \log(1 + |f(i, j)|)$$

$$g(i, j) = b \cdot e^{c(f-a)}$$

## Gamma 变换 (幂次变换)

根据指数函数进行变换  $y = x^\gamma$ ，对于普遍暗光的图可以用  $\gamma < 1$  实现亮度提升，过量的图利用  $\gamma > 1$ 。注意这里应该用 `double` 型，即在  $[0, 1]$  之间的元素，否则这里的  $\gamma$  应取反。

全暗图在亮度校正后，色彩饱和度往往不是很足，而且会存在比较明显的噪声或不平滑特征。

## 图像的直方图变换 灰度间距拉开或分布均匀

直方图变换后可使图像的灰度间距拉开或使灰度分布均匀，从而增大对比度，使图像细节清晰，达到增强的目的。

### • 直方图均衡化

- 通过对原图像进行某种变换，使得图像的直方图变为均匀分布的直方图。
- 灰度级连续的灰度图像：当变换函数是原图像直方图累积分布函数（CDF）时，能达到直方图均衡化（灰度范围  $[0, 1]$ ）的目的。
- 对于离散图像，用频率来代替概率。利用频率构成的分布列计算累积分布函数，也可以达到直方图均衡化（灰度范围  $[0, 1]$ ）的目的。

$k$	0	1	2	3	4	5	6	7
$r_k$	0	1/7	2/7	3/7	4/7	5/7	6/7	1
$n_k$	790	1023	850	656	329	245	122	81
$p_r(r_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
$s'_k$	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1
$s''_k$	1/7	3/7	5/7	6/7	6/7	1	1	1
$s_k$	1/7	3/7	5/7		6/7			1
$n_{s_k}$	790	1023	850		985			448
$p_s(s_k)$	0.19	0.25	0.21		0.24			0.11

- 求变换函数，即求累积分布函数  $s'_k$

$$s'_0 = T(r_0) = \sum_{j=0}^0 p_r(r_j) = \frac{790}{64 * 64} = 0.19$$

类似的

$$s'_1 = T(r_1) = \sum_{j=0}^1 p_r(r_j) = \frac{790 + 1023}{64 * 64} = 0.44$$

后文同理

- 计算等间隔的近似灰度值  $s''_k$

输出图像灰度是等间隔的，且与原图像灰度范围一样取 8 个等级，即要求最终的值  $s_k = \frac{k}{7}$ 。需要对  $s'_k$  进行重新量化后加以修正，而后我们发现输出灰度级仅为五个级别。

$$s_0 = 1/7 \quad s_1 = 3/7 \quad s_2 = 5/7 \quad s_4 = 6/7 \quad s_7 = 1$$

而后计算对应每个  $s_k$  的  $n$

### • 直方图规定化



## 图像的同态增晰 消除不均匀照度的影响而不损失图像细节

图像由于物体受到不均匀的照度，其灰度级动态范围很大，黑与白形成强烈反差，而感兴趣目标的灰度级范围很小，分不清目标的灰度层次和细节，图像上对应照度暗的部分，其细节较难分辨。

使用灰度线性变换无法解决这类图像的增强问题：扩展灰度级虽可提高目标的反差，但会使动态范围更大；压缩灰度级，虽可减少动态范围，但目标的灰度层次和细节就会更看不清，

同态增晰:目的是**消除不均匀照度的影响而不损失图像细节。即减弱反射分量增强照明分量**

图像是物体对照明光的反射，自然景物图像是由两个分量乘积组成的，即照明函数和反射函数的乘积。**反射分量**反映图像的实际内容（细节，纹理，边缘等），随图像细节不同在空间上做快速变化，其频谱落在**空间高频区域**。而**照明分量**在空间上均具有缓慢变化的性质，其频谱落在空间**低频区域**。因此可通过**傅里叶变换**将两者分开，进行同态滤波。

### 算法 本质：对数图像频率域的高通滤波

#### 1. 对数运算

$$z(x, y) = \ln f(x, y) = \ln i(x, y) + \ln r(x, y)$$

#### 2. 傅里叶变换

$$z(u, v) = I_1(u, v) + R_1(u, v)$$

#### 3. 选择同态滤波特性 $H(u, v)$ ，**乘以 DFT 结果**。衰减 $I_1$ 以压缩 $i(x, y)$ 分量的变换范围，提升 $R_1(u, v)$ 以增强 $r(x, y)$ 细节分量的对比度。

#### 4. 傅里叶反变换

#### 5. 指数运算进行还原

高通滤波器也可以理解成图像减去其低通滤波结果。减少高频系数的成为低通滤波器。

### 另一种算法

在对数运算时加入高斯核函数

$$\ln R(x, y) = \ln I(x, y) - \ln(I(x, y) * F(x, y))$$

其中， $F(x, y)$  为高斯核函数， $I(x, y)$  为输入图， $R(x, y)$  为图像反射图，即进行时间域的一种均值滤波（也相当于频率域高斯核低通滤波）

使用高斯核做滤波后进行模糊化，从而可以大致确定图像的分片亮度情况。而后就可以得到  $R(x, y)$  再进行上述的线性或非线性增强。

### 反色去雾法

先将暗图用反色变成雾图，再用去雾算法（一般用 Retinex 理论或图像熵来解决），而后反色复原。

## 第十课：图像增强（中）

### 区域增强

区域增强算法包括**平滑算法**和**锐化算法**。低通滤波可对图像进行平滑处理（去掉高频部分）；高通滤波可对图像进行锐化处理（去掉低频部分）。除此之外，还有一种频率域带通滤波器（去掉太低和太高频的部分），往往可以获得某些特殊的图像特征，但在图像处理之中使用的不算多。

在时间域中也可以直接进行多种平滑算法，例如邻域平均法、中值滤波和边界保持类滤波。

### 图像噪声

数字图像要经过**采集**、处理、存储、传输等一系列加工变换。

#### 1. 噪声分类

- 产生原因：**外部噪声**和内部噪声
- 统计特性是否随时间变化（一般用于视频或批量图像）：平稳噪声和非平稳噪声

- 噪声幅度随时间分布形状来定义：**高斯噪声**、瑞利噪声、泊松噪声等
- 噪声频谱形状来分：**白噪声**、 $1/f$  噪声、三角噪声
- 按噪声和信号之间的关系：加性噪声、乘性噪声、其他噪声
  - 加性噪声： $f + n$  如高斯噪声、热噪声、散弹噪声、量化噪声
  - 乘性噪声： $f + nf$  扫描光栅、胶片颗粒

泊松噪声就属于既不是加性噪声也不是乘性噪声的噪声。椒盐噪声、CT成像金属噪声也不是这两类噪声。

## 高斯噪声 加性噪声

$g = f + n$ ,  $n$  由服从高斯分布的样本构成

## 椒盐噪声

一种结构比较简单的冲击噪声，其含义为在图像中随机选取一定比例的像素点并将其灰度更改为黑色的0或白色的255，以达到完全破坏这些像素的效果。其余像素则保持原值不变。

## 泊松噪声

用泊松分布近似每一个像素的光子数，参数  $\lambda = kf \approx np$ ,  $f$  为图像的理想亮度， $k$  为一较大的亮度，表示光子数与亮度之间的比值。而后设  $X \sim P(\lambda)$ ，获取  $x$  的一个样本，实际亮度表示为  $g = x/k$ 。

故而由绝对噪声强度来讲， $k$  一定时，**越亮的地方噪声越强**。

当  $f$  一定时， $k$  越大则  $D(g)$  越小（对应近似二项分布就是样本越多频率越接近于概率），这也就说明总体光强越大，则泊松噪声影响越小。在包括CT成像等工程或医学成像之中，强光成像质量更好。**减弱放射光强则会增强成像噪声**。

泊松噪声没有很好的理论去除方法，一种简单的操作是使用Anscombe变换，将原始图  $f$  映射成  $\sqrt{f}$  或  $\sqrt{f \pm 0.375}$ （峰值为1），可以将噪声变为近似高斯噪声，去噪后再做逆变换。

## 均值滤波

空间域均值滤波：大部分的噪声都可以看作是随机信号，对图像的影响可以看作是孤立的。某一像素，如果它与周围像素点相比，有明显的不同，则该点被噪声感染了。设当前待处理像素为  $f(m, n)$ ，给出一个大小为  $3 \times 3$  的九点均值滤波处理模板。

设处理后的图像为  $g(m, n)$ ，则九点均值滤波可以描述为

$$g(m, n) = \begin{cases} \frac{1}{9} \sum_{i \in Z} \sum_{j \in Z} f(m+i, n+j) & \left| f(m, n) - \frac{1}{9} \sum_{i \in Z} \sum_{j \in Z} f(m+i, n+j) \right| > \varepsilon \\ f(m, n) & otherwise \end{cases}$$

其中  $Z = \{-1, 0, 1\}$ ,  $\varepsilon$  为门限，若  $\varepsilon = 0$ ，等价于卷积操作，也称为**低通空间滤波**。而可以等价于

$$g(m, n) = \sum_{r=-k}^k \sum_{s=-l}^l f(m-r, n-s) H(r, s)$$

上述  $H(r, s)$  称为掩模。

常见掩模举例：（注意观察系数和，必须为1，且系数非负）

$$H_1 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$H_2 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$H_3 = \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$H_4 = \frac{1}{2} \begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix}$$

均值滤波不适用于反差很大的椒盐噪声

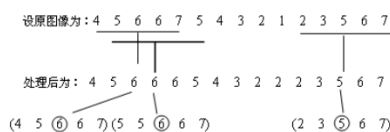
## 中值滤波

邻域平均法在去噪时使得边界也变得模糊。而中值滤波是用非线性的处理方法，在去噪的同时可以兼顾到边界信息的保留。选一个含有奇数点的窗口  $W$ ，将这个窗口在图像上扫描，把该窗口中所含的像素点按灰度级的升（或降）序排列，取位于中间的灰度值，来代替该点的灰度值。

m-2	m-1	m	m+1	m+2
-----	-----	---	-----	-----

◆ 常用的窗口还有方形、十字形、圆形和环形等。

图4.9 一维窗口



		(m-2,n)		
		(m-1,n)		
(m,n-2)	(m,n-1)	(m,n)	(m,n+1)	(m,n+2)
		(m+1,n)		
		(m+2,n)		

(m-1,n-1)	(m,n-1)	(m+1,n-1)
(m-1,n)	(m,n)	(m+1,n)
(m-1,n+1)	(m,n+1)	(m+1,n+1)

中值滤波对于消除孤立点和线段的干扰十分有用。特别是对于二进噪声尤为有效，对于消除高斯噪声的影响效果不佳。对于一些细节较多的复杂图像，还可以多次使用不同的中值滤波，然后通过适当的方式综合所得的结果作为输出，这样可以获得更好的平滑和保护边缘的效果。对于椒盐噪声的效果不错。

## K临近均值滤波器

在  $m \times m$  的窗口中，属于同一集合类的像素，它们的灰度值将高度相关。被处理的像素（对应于窗口中心的像素）可以用窗口内与中心像素灰度最接近的  $k$  个邻近像素的平均灰度来代替。

1. 作一个  $m \times m$  的作用模板。
2. 在其中选择K个与待处理像素的灰度差为最小的像素。
3. 用这K个像素的灰度均值替换掉原来的值。

## 加权的临近均值滤波——双边滤波

双边滤波是高斯核均值滤波的升级，普通的高斯核滤波权值仅仅与图像像素距离有关，即：

$$GB[I]_p = \sum_{q \in S} G_{\sigma}(\|p - q\|) I_q$$

其中高斯核函数定义为：

$$G_{\sigma}(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

双边滤波则将权值同时考虑距离相似度和灰度相似度，并进行归一化，定义如下：

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(I_p - I_q) I_q$$

## 非局部均值滤波

定义邻域窗口（图像小块patch），通过比较小块整体的相似性来确定权值。为了避免算法复杂度过高，非局部算法也会在一个有限的搜索窗口内，进行小块匹配。

## K临近中值滤波器——边界保持

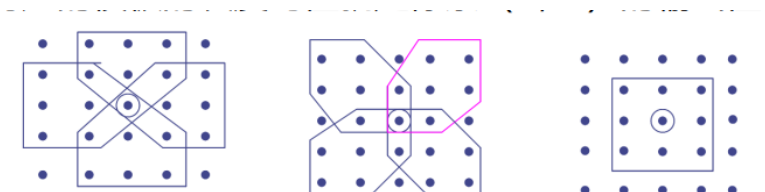
在K近旁中值滤波器中，不选k个邻近像素的平均灰度来代替，而选k个邻近像素的中值灰度来代替。

## 最小均方差滤波器

对图像上待处理的像素  $(m, n)$  选它的  $5 \times 5$  邻域。

在此邻域中采用下图所示的模板。

计算各个模板的均值和方差，按方差排序，最小方差所对应的模板的灰度均值就是像素  $(m, n)$  的输出值



### ● 计算步骤如下：

- (1) 按图做出9个模板，计算出各自的方差。
- (2) 选出方差为最小的模板。
- (3) 用该模板的灰度均值代替原像素的灰度值。

- 以方差作为各个邻域灰度均匀性的测度。
- 若邻域含有尖锐的边缘，灰度方差必定很大，而不含边缘或灰度均匀的邻域，方差就很小，那么最小方差所对应的邻域就是灰度最均匀邻域。
- 通过这样的平滑既可以消除噪声，又能够不破坏邻域边界的细节

## DNCNN

DNCNN是2017年提出的一种基于深度学习和神经网络的端对端高斯噪声去噪模型。MATLAB提供了预训练网络（利用数据集训练完成），可以简单快速的完成去噪

总结

滤波方法	优点	缺点	适用噪声类型
均值滤波	计算简单，适合实时处理	模糊图像细节	高斯噪声、热噪声、量化噪声
中值滤波	保留边缘，去除孤立噪声点效果好	对大面积平滑区域效果不佳	椒盐噪声、冲击噪声
K-临近均值滤波	保留更多细节，适合边缘丰富的图像	计算复杂度高	高斯噪声、热噪声
K-临近中值滤波	保留细节，去除孤立噪声点效果好	计算复杂度高	椒盐噪声、冲击噪声
双边滤波	保留边缘，平滑噪声	计算复杂度高	高斯噪声、热噪声
非局部均值滤波	保留图像细节，去噪效果好	计算复杂度极高	高斯噪声、热噪声
最小均方差滤波	既能去噪又能保留边界细节	计算复杂	高斯噪声、热噪声
DNCNN	去噪效果好，利用深度学习模型	需要大量计算资源和训练数据	高斯噪声、复杂噪声

第十一课：图像增强（下）与图像填充

图像锐化

加强图像中景物的边缘和轮廓，使模糊图像变得更清晰。图像模糊的实质：图像受到平均或积分运算，可采用逆运算，可对图像微分或对离散图像差分运算，可以使模糊图像的质量得到改善。

从频域角度，图像模糊是高频分量受到衰减，可采用**高通滤波器**使图像清晰。（减少或去除低频系数）

边缘和轮廓常常位于图像中灰度突变的地方，因而可以直观的想到用灰度的差分对边缘和轮廓进行提取。

一阶差分

我们常用绝对值或最大值运算代替平方与平方根运算。而数字图像中，微分可以用差分（向前差分）代替

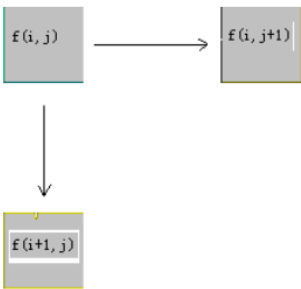


图4.15 沿x和y方向的一阶差分

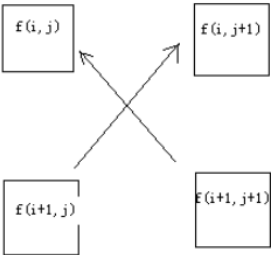


图4.16 罗伯茨差分

差分算子

算子	$\mathcal{D}_x$	$\mathcal{D}_y$	特点
Roberts	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	定位准确，但对噪声敏感

算子	$G_x$	$G_y$	特点
Prewitt	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	用像素点上、下、左、右邻点的灰度差，在边缘处达到较大值检测边缘，去掉部分伪边缘， <b>对噪声具有平滑作用</b> 。与 Roberts 相比，减低了对噪声的敏感程度
Sobel	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	与上述算子相比，对于 <b>像素的定位响应做了加强</b> 。因此效果更好， <b>提取的图像轮廓</b> 有时并不能令人满意
Isotropic Sobel	$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$	又称为各向同性 Sobel 算子。和 Sobel 算子相比，它的 <b>定位加权系数更为准确</b> ，在检测不同方向的边时准确度的确一致
Kirsch	$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	共有 8 个方向算子，这里列的是水平和垂直两个方向算子，有较好的 <b>抑制噪声</b> 的作用。其他 6 个方向算子参见 8.14 (Kirsch 锚状边缘检测)

一阶差分的锐化方法

- 1. 使其输出图像的各点等于该点处的梯度。即  $g(i, j) = \nabla f(i, j)$ ，若取绝对值则相当于一种边界检测。
- 2. 对梯度值超过某阈值  $T$  的像素选用梯度值，而小于  $T$  时选用原像素点值

$$g(i, j) = \begin{cases} \nabla f(i, j), & \nabla f(i, j) \geq T \\ f(i, j), & \text{其他} \end{cases}$$

- 3. 对梯度超过  $T$  的像素选用固定灰度  $L_G$  代替，而小于  $T$  时选用原像素点值，边界清晰而不突兀

$$g(i, j) = \begin{cases} L_G, & \nabla f(i, j) \geq T \\ f(i, j), & \text{其他} \end{cases}$$

- 4. 将梯度值超过  $T$  的像素选用梯度值，而小于  $T$  时选用固定的灰度。该方法用于弱化背景信息，强化边界显示。
- 5. 将梯度值超过某阈值的选用固定灰度，而小于某阈值的选用固定灰度，用于图像背景和边界的分类。

二阶差分——Laplace算子

对于数字图像而言，基于中心二阶差分，拉普拉斯算子可以简化为：

$$g(i, j) = 4f(i, j) - f(i + 1, j) - f(i - 1, j) - f(i, j + 1) - f(i, j - 1)$$

也可以表示为卷积的形式，即

$$g(i, j) = \sum_{r=-k}^k \sum_{s=-l}^l f(i - r, j - s) H(r, s)$$

而  $k = 1, l = 1$  时矩阵为

$$H_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

增强算子（保留原图基本信息），即做完差分后加一次原像素，即  $H_1$  的中心元素 +1 即可

广义的高通滤波器

以上两个模板的卷积都会得到均值0的结果，如果希望得到平均灰度的保持，并强化边界的锐化效果，应当让模板的中心系数加1即可



## 图像复原的来源——图像退化

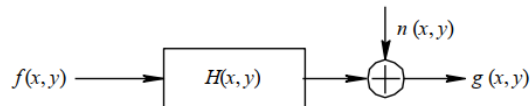
图像在形成、传输和记录过程中，由于受到多方面的影响，造成图像质量的**退化 (degradation)**

### 图像复原的目的

在研究图像退化原因的基础上，以退化图像为依据，根据一定的先验知识，建立一个退化模型，然后用相反的运算，以恢复原始景物图像

- 图像复原要明确规定**质量准则**，衡量接近原始景物图像的程度。（PSNR、SSIM、brisque等）
- 图像复原**模型**
  - 用连续数学或离散数学处理
  - 实现可以在空间域卷积或频率域相乘

- 图像退化的一般模型如图6.1所示。



- 输入图像 $f(x, y)$ 经过一个**退化系统**或退化算子 $H(x, y)$ 后产生的退化图像 $g(x, y)$ 可以表示为：

$$g(x, y) = H[f(x, y)] \quad (6.1)$$

- 如果仅考虑**加性噪声**的影响，则退化图像可表示为：

$$g(x, y) = H[f(x, y)] + n(x, y) \quad (6.2)$$

- 退化的图像是由成像系统的退化加上额外的系统噪声而形成的。
- 若已知  $H(x, y)$  和  $n(x, y)$ ，图像复原是在退化图像的基础上，作逆运算，得到  $f(x, y)$  的一个最佳估计
- “最佳估计”而非“真实估计”。由于存在可能导致图像复原的病态性
- 逆问题可以存在多个解

### 高斯噪声去噪的数学模型

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

$n(x, y)$  为服从某种正态分布，若  $f(x, y)$  为无任何意义的图像，例如随机雪花图像，就无法恢复。利用  $f(x, y)$  的片段光滑性或其他性质，可以使用滤波或等价的最优化模型进行简单的去噪。

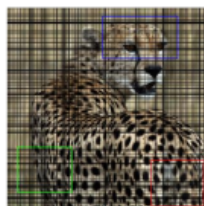
### 图像填充问题

图像填充问题描述为图像的部分像素已知（为真实值或有少量噪声干扰），剩余像素完全未知的退化模型，其任务为对未知像素值给出合理估计。

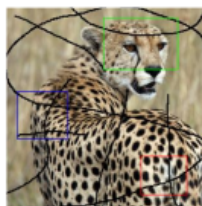
数学模型上讲，将图像拉成  $m \times n$  的向量表示后，原图为  $f$ ，待填充图为  $g$ ，其退化模型为  $P_{\Lambda} f = g$ ， $g$  的部分元素为 0，其余部分与  $f$  相同， $P_{\Lambda}$  为对角矩阵。



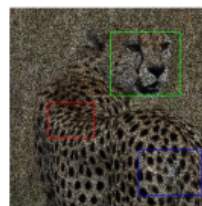
原始图像



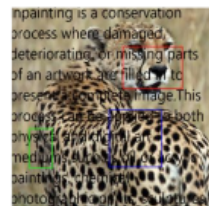
行列缺失



涂鸦缺失



散点缺失



文字缺失

## 均值滤波在填充中的应用

均值滤波由于会将椒盐点或丢失点的无意义值加入平均值，从而导致填充结果出现巨大误差，因而在滤波时，应当**筛选邻域内有效的像素点**进行平均值的计算，完成滤波操作。

同时，可以对滤波算法剪枝，**仅计算无效点的滤波结果**。这样就无需在滤波后恢复真实值了。

部分无效像素附近的有效像素偏少，或临近图像边界，导致平均值出现偏差，因而恢复效果**略不如中值滤波**。

## 反复小窗口滤波

过大的滤波窗口容易使多个区域混淆，导致平均结果远远不符合预期。小滤波窗口可以确保填充区域边缘较为准确，但无法影响到内部填充区域。

利用**反复滤波+恢复真实值**的方法，自外向内一层一层对填充区域进行估计与更新，可以获得相对更加理想的效果。

## 非局部匹配与图像填充

非局部匹配也是图像填充的重要方法，利用当前为止像素点的邻域内的有效值形成“不完整小块”，与搜索窗口内其他“不完整小块”进行比较，最终利用**最接近小块，或加权平均**的结果来确定填充结果，是流程化填充方法的主要思想。

对于比较粗的填充区域，往往需要规定合理的填充顺序，普遍这种顺序定义为**从外向内**的填充顺序。

---

# 第十二课：图像反卷积与其他恢复问题

## 图像反卷积问题

- 图像反卷积问题亦称之为图像去模糊问题。该问题恢复的图像往往是原图像和某个模糊核算子进行卷积所得到的模糊图像，有时可能附带一些噪声。
- 在不考虑噪声干扰的情况下，如果模糊核已知，不论采用哪一种边界条件（零延拓、复制边界条件、周期边界条件、对称边界条件等），**图像的卷积模糊操作本质上是一种线性变换**。将清晰图像视作  $m \times n$  维向量  $f$ ，模糊图像视作  $m \times n$  维向量  $g$ ，则退化模型为  $Af = g$ 。
- 注意：矩阵  $A$  是一个  $(m \times n) \times (m \times n)$  的大型矩阵，并不是模糊核本身。当模糊核明显小于图像时， $A$  为稀疏矩阵（思考为什么）。可以注意到  $A$  通常是可逆矩阵，但图像模糊效果伴随着整数量化的误差，从而为反卷积带来了一定的难度。

这是因为在这种情况下，模糊核仅影响图像中的局部区域，而不是整个图像。具体来说，每个像素的模糊只会影响它周围一个小范围的像素。因此，矩阵  $A$  的非零元素只出现在与这些局部模糊相关的行和列中，其余大部分元素都为零，从而使得  $A$  成为稀疏矩阵。

原图  $f(x, y)$  为  $A \times B$  大小，模糊核  $h(x, y)$  为  $C \times D$  大小。而后用零延拓将图像和模糊核函数扩展成  $M = A + C - 1$  和  $N = B + D - 1$  个元素的周期函数，则

- 以零延拓为例，将图像和模糊核函数扩展成  $M=A+C-1$  和  $N=B+D-1$  个元素的周期函数。

$$f_e(x, y) = \begin{cases} f(x, y) & 0 \leq x \leq A-1 \text{ 且 } 0 \leq y \leq B-1 \\ 0 & \text{其他} \end{cases}$$

$$h_e(x, y) = \begin{cases} h(x, y) & 0 \leq x \leq C-1 \text{ 且 } 0 \leq y \leq D-1 \\ 0 & \text{其他} \end{cases}$$

- 则输出的降质数字图像为

$$g_e(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e(m, n) h_e(x-m, y-n)$$

可以发现这种形式与之前所讲的空间域滤波/卷积并无区别，也就对应了线性变换形式  $g = Af$ 。



## 矩阵 $A$ 的托普利茨矩阵表示形式

要理解图中的矩阵  $A$  的托普利茨 (Toeplitz) 矩阵表示形式，我们可以从以下几个方面来分析：

### 1. 托普利茨矩阵的定义

托普利茨矩阵是一种特殊形式的矩阵，其中每一条从左到右下的对角线上的元素都是相同的。具体来说，如果矩阵  $T$  是一个  $n \times n$  的托普利茨矩阵，那么它满足如下条件：

$$T_{i,j} = T_{i+1,j+1}$$

这意味着矩阵中的每个对角线上的元素都相等。

### 2. 图中矩阵 $A$ 的结构

图中展示的矩阵  $A$  是一个由多个子矩阵  $H_j$  组成的大矩阵。每个子矩阵  $H_j$  都是一个托普利茨矩阵。具体来说，矩阵  $A$  可以表示为：

$$A_{M \times M} = \begin{bmatrix} H_0 & H_{M-1} & H_{M-2} & \cdots & H_1 \\ H_1 & H_0 & H_{M-1} & \cdots & H_2 \\ H_2 & H_1 & H_0 & \cdots & H_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ H_{M-1} & H_{M-2} & H_{M-3} & \cdots & H_0 \end{bmatrix}$$

可以看出，矩阵  $A$  中的每个元素本身又是一个托普利茨矩阵  $H_j$ 。

### 3. 子矩阵 $H_j$ 的结构

每个子矩阵  $H_j$  的具体结构如下：

$$H_j = \begin{bmatrix} h_e(j, 0) & h_e(j, N-1) & h_e(j, N-2) & \cdots & h_e(j, 1) \\ h_e(j, 1) & h_e(j, 0) & h_e(j, N-1) & \cdots & h_e(j, 2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_e(j, N-1) & h_e(j, N-2) & h_e(j, N-3) & \cdots & h_e(j, 0) \end{bmatrix}$$

这里  $h_e(j, k)$  是模糊核的元素，这些元素满足托普利茨矩阵的性质，即每个对角线上的元素都是相同的。

### 4. 稀疏矩阵

由于模糊核通常比图像小很多，模糊核只会影响图像的局部区域，因此矩阵  $A$  中大多数元素是零，这使得  $A$  是一个稀疏矩阵。稀疏矩阵的性质使得在存储和计算时可以更加高效。

图中的矩阵  $A$  是一个大型的块托普利茨矩阵，其中每个块  $H_j$  本身也是一个托普利茨矩阵。由于模糊核的局部影响，矩阵  $A$  中包含大量的零元素，使其成为稀疏矩阵。这种矩阵结构在处理大规模图像处理问题时具有重要的实际意义。观察可知  $A$  转置仍然代表一种卷积操作，但卷积核相比原卷积核分别进行了一次上下翻转与左右翻转。傅里叶变换相当于取共轭。

## 该方法对于整数舍入的脆弱性

其取值会相比严格的数值卷积结果有小的误差，但仅仅这些误差就可以使恢复结果变得完全崩溃，主要原因就是**除法对于误差的脆弱性**，使得恢复图像的傅里叶变换存在明显误差。

## 使用反问题求解方法

反问题指的是从观测数据中推断出系统或过程的原始状态。具体来说，在图像处理中的反卷积问题中，我们试图从模糊图像中恢复出清晰图像。这个问题可以表示为：

$$Af = g$$

其中， $A$  表示模糊矩阵， $f$  表示原始图像的向量形式， $g$  表示模糊图像的向量形式。反问题的核心任务是从已知的  $g$  和  $A$  中求解出未知的  $f$ 。

### 反问题方法的优势

- 通用性**：反问题方法适用于各种边界条件，无论是零边界、复制边界还是周期性边界条件。这使得该方法在不同的应用场景下都能够得到良好的效果。
- 鲁棒性**：反问题方法对整数误差具有鲁棒性。即使在处理过程中存在一些小的误差，反问题方法也能有效地恢复出原始图像。

3. **数值稳定性**: 反问题方法可以通过引入正则化技术 (如广义最小残差法) 来提高数值计算的稳定性, 避免因噪声或误差导致的解不稳定现象。

## 逆滤波复原问题

**非约束复原**, 根据对退化系统  $H$  和噪声  $n$  的了解, 已知退化图像  $g$  的情况下, 在一定的最小误差准则下得到原始图像  $f$  的估计。

逆滤波是最早使用的一种**无约束复原**方法, 本质上就是前述傅里叶变换的除法版本成功地应用于航天器传来的退化图像。

□ 设  $n = g - Hf$  时, 设计模型使  $\|n\|_2^2 = n^T n$  最小, 即模型目标函数  $J(\hat{f}) = \|g - H\hat{f}\|_2^2$  最小。

□ 对上述函数关于  $\hat{f}$  求偏导, 得  $\frac{\partial(J(\hat{f}))}{\partial \hat{f}} = -2H^T(g - H\hat{f}) = 0$ , 根据前述内容, 模糊模型为一个可逆方阵  $H$ , 则利用移项、合并同类项等方法, 可得  $\hat{f} = (H^T H)^{-1} H^T g = H^{-1} g$ 。相当于无视了噪声本身的存在, 直接进行反卷积操作。

无视了噪声本身的存在

### 约束复原方法

约束复原除了对降质系统的PSF有所了解外, 还需要对原始图像和外加噪声的特性有先验知识。

根据不同领域的要求, 有时需要对  $f$  和  $n$  作一些特殊的规定, 使处理得到的图像满足某些条件。

**约束最小二乘法复原问题** 令  $Q$  为  $f$  的线性算子, 要设法寻找一个最优估计  $\hat{f}$ , 使其形式为  $\|Q\hat{f}\|^2$  的服从约束条件  $\|g - H\hat{f}\|^2 = \|n\|^2$  的函数最小化问题, 而  $\|Q\hat{f}\|^2$  被称为**正则项**,  $Q$  称为线性正则变换, 用以刻画恢复图像的某些内在性质, 而另一个约束项  $\alpha (\|g - H\hat{f}\|^2 - \|n\|^2)$  称为**保真项**, 用以刻画图像噪声的信息。

而目标函数为  $J(\hat{f}) = \|Q\hat{f}\|^2 + \alpha (\|g - H\hat{f}\|^2 - \|n\|^2)$  最小, 这里的  $\alpha$  为保真项的惩罚强度, 对上述函数关于  $\hat{f}$  求偏导并令其等于 0, 可以得到

$$\hat{f} = (H^T H + \gamma Q^T Q)^{-1} H^T g$$

其中  $\gamma = \alpha^{-1}$ , 可以成为正则项惩罚强度。

### 维纳滤波

维纳滤波 (Wiener Filtering) 是一种经典的线性滤波方法, 用于在信号处理和图像处理领域中去除噪声和恢复原始信号。其目标是通过最小化信号和噪声之间的均方误差来实现最佳的信号恢复。

从图片中的公式开始解释:

- 首先定义了两个矩阵:
  - $R_f = E\{ff^T\}$  代表信号  $f$  的自相关矩阵。
  - $R_n = E\{nn^T\}$  代表噪声  $n$  的自相关矩阵。
- 定义  $Q^T Q$  矩阵为  $R^{-1} R_n$ , 其中  $R$  是信号和噪声无关的假设下的相关矩阵。**信噪不相关**
- 根据这个定义, 维纳滤波的解形式为:

$$\hat{f} = (H^T H + \gamma R_f^{-1} R_n)^{-1} H^T g$$

其中:

- $H$  是系统矩阵 (或卷积矩阵)。
- $\gamma$  是常数, 用于调整滤波器的强度。
- $g$  是观测到的信号。

4. 假设  $M = N$ ,  $S_f$  和  $S_n$  分别为图像和噪声的功率谱, 得到:

$$\hat{F}(u, v) = \left[ \frac{H^*(u, v)}{|H(u, v)|^2 + \gamma \left[ \frac{S_n(u, v)}{S_f(u, v)} \right]} \right] G(u, v)$$

这个公式描述了维纳滤波在频域中的实现形式, 其中:

- $H(u, v)$  是频域中的传递函数。
- $H^*(u, v)$  是传递函数的共轭。
- $G(u, v)$  是观测到的信号的频谱。

维纳滤波的基本思想是在频域中调整信号的频谱, 以最小化信号恢复中的误差。通过使用信号和噪声的统计特性 (自相关函数或功率谱), 维纳滤波能够有效地去除噪声, 同时保留尽可能多的原始信号信息。

- $\gamma = 1$  时, 系统函数  $H_w(u, v)$  是维纳滤波器的传递函数, 常根据噪声强度设置

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

- 如果  $\gamma = 0$ , 即系统变成单纯的去卷积滤波器, 传递函数即为  $H^{-1}$
- 若  $\gamma$  为可调整的其他参数, 此时为参数化维纳滤波器

平滑度约束最小平方滤波

设置不同的  $Q$  来达到反卷积效果, 例如拉普拉斯算子的离散卷积核的高通滤波来进行近似等。

图像盲反卷积

即既不知道清晰图像  $f$ , 也不知道退化矩阵  $A$ 。

现在的相机“防抖”技术一般是通过**传感器**大致获取模糊核的值, 或通过连续拍摄**多帧照片** (短视频) 并结合三维重构的方法来避免模糊。

目前主流的单图盲反卷积方法有两种, 一种是结合一些对模糊核的正则约束, **同时求解模糊核与清晰图像**。另一种则是**先对模糊核进行大致估计**, 在用普通反卷积算法恢复

图像去雾

暗通道算法:

$$I(x) = J(x)t(x) + A(1 - t(x))$$

其中  $I(x)$  为雾化图,  $J(x)$  为真实图,  $t(x)$  为透射率,  $A$  为大气光强度。先估计  $A$  的值, 而后对  $J(x)$  加上必要的正则约束即可解出真实图  $J(x)$ 。

即物体反射的光由于雾的效果, 只能反射出  $t(x)$  的光强, 而后雾也有自身的反射, 在CMOS中收到的即为两部分叠加之和。

归纳总结

方法	基本情况	优点	缺点
逆滤波 (无约束)	使用已知退化系统 $H$ 和噪声 $n$ , 通过傅里叶变换求解原始图像 $f$ 的估计。	对退化图像有较好的复原效果。	对噪声非常敏感, <b>恢复效果容易受到噪声影响。</b>
约束最小二乘法	在对降质系统和噪声有先验知识的基础上, 设法找到最优估计。	可以在噪声环境中有效恢复图像, 对噪声有一定的鲁棒性。	需要对噪声和原始图像的特性有较好的先验知识, 计算复杂度较高。
维纳滤波	基于信号和噪声的统计特性, 最小化均方误差进行图像恢复。	对信噪比变化有较好的适应性, 能有效去除噪声并保留信号。	需要知道信号和噪声的功率谱, 计算复杂度较高。

方法	基本情况	优点	缺点
平滑度约束 最小平方滤波	使用拉普拉斯算子作为正则项，对图像进行高通滤波，达到去模糊效果。	可以在高噪声环境中有效恢复图像，减少伪影和噪声影响。	需要合理选择正则化参数，计算复杂度较高。
盲反卷积	在未知模糊核和原始图像的情况下，通过估计模糊核和清晰图像进行恢复。	不需要预先知道模糊核，可以处理更复杂的模糊情况。	对初始模糊核估计敏感，计算复杂度高，恢复效果不如有关先验知识的情况。
参数化维纳滤波	通过调整参数 $\gamma$ 控制滤波器的强度，实现不同程度的去模糊效果。	可以灵活调整滤波器参数，适应不同的噪声环境和图像情况。	参数调整较为复杂，恢复效果依赖于参数选择。

这些反卷积方法各有优缺点，具体使用时需要根据实际情况选择合适的方法。如果需要详细介绍或代码实现，可以参考课程PPT中的具体章节。

## 第十三课：正则化模型与优化算法（上）

### 最小二乘拟合

最小二乘函数  $\|A\vec{x} - \vec{b}\|_2$  最小当且仅当

$$\|A\vec{x} - \vec{b}\|_2^2 = \vec{x}^T A^T A \vec{x} - \vec{x}^T A^T \vec{b} - \vec{b}^T A \vec{x} + \vec{b}^T \vec{b}$$

最小，而后对  $\vec{x}$  求导，而后令偏导数均为0即可得到

$$2A^T A \vec{x} - A^T \vec{b} - A^T \vec{b} = 0$$

即

$$A^T A \vec{x} = A^T \vec{b}$$

而当  $R(A) = n$  时，最小二乘解唯一，并且有等价表达式  $\vec{x} = (A^T A)^{-1} A^T \vec{b}$ 。

### 梯度下降法

参考运筹学中的方法，计算  $\nabla f$  而后代入  $\vec{x}^k$  即可，即  $\vec{x}^{k+1} = \vec{x}^k - \delta \cdot \nabla f(\vec{x}^k)$  即可。而后对  $\delta$  求偏导数即可。

### 梯度下降法解决图像反卷积问题

由于计算  $A^{-1}$  往往比较困难，但是  $A^T$  却可以通过翻转模糊核的情况来进行卷积模拟，因此可以用梯度下降法高效完成反卷积的模拟。

### 共轭梯度法解决图像反卷积问题

直接求解反问题  $(A^T A)\vec{x} = A^T \vec{b}$ ，而  $A^T A$  正定矩阵符合要求。

### 岭回归与正则约束

对拟合系数进行必要的约束，使其绝对值不会太大，避免系数膨胀或对部分错误/离群信息过拟合。也可以防止最小二乘解不唯一时，任意解膨胀的情况。

$$\min_{a_1, a_2, \dots, a_{n+1}} \|y - (a_1 x^n + \dots + a_n x + a_{n+1})\|_2^2 + \lambda \|a\|_2^2$$

其中  $a = [a_1 \ a_2 \ \dots \ a_{n+1}]$  表示拟合系数，而也可以等价描述为

$$\min_{\vec{x}} \frac{1}{2} \|A\vec{x} - \vec{b}\|_2^2 + \frac{\lambda}{2} \|\vec{x}\|_2^2$$

而第一项称为**保真项**，使得误差尽可能小，第二项称为**正则项**，控制  $\vec{x}$  的向量规模。

由于目标函数可微，也可以类似于前面的最小二乘问题，用梯度下降法求解，也可以令梯度等于零得到最优解。

图像梯度的引入与 Tikhonov 正则模型

对于大多数的自然图像而言，其图像的颜色或灰度值大都是不等于0甚至不接近于0的。因此，前述的正则模型很可能仅会将图像暗度降低，对于图像的噪声控制或其他方面帮助往往并不大。但是，对于大部分分片光滑的图像而言，他们的图像差分算子得到的结果，却可以满足区域内像素的差分结果普遍接近于0。

Tikhonov 模型也是一种 L2 正则化模型

- 正则化模型即考虑 惩罚项 的范数  $\|\cdot\|$ ？
- L0正则化：直接限制非零参数个数，难以优化。
- L1正则化：通过绝对值惩罚项实现特征选择，产生稀疏模型。
- L2正则化：通过平方和惩罚项缩小参数值，防止过拟合。
- Elastic Net正则化：结合L1和L2的优点，兼顾稀疏性和参数缩减。

其最大的区别是正则的向量不是原始的图像向量，而是计算梯度之后的向量，梯度的计算可以使用简单的向前或中心差分算子，亦可以使用包括Sobel算子在内的偏导数计算算子：

$$\min_{\vec{x}} \frac{1}{2} \|A\vec{x} - \vec{b}\|_2^2 + \frac{\lambda}{2} \|\nabla \vec{x}\|_2^2$$

Tikhonov模型的求解可以与岭回归模型一样，通过显示表达式

$$\vec{x} = (A^T A + \lambda \nabla^T \nabla)^{-1} (A^T \vec{b})$$

直接解出，该模型同样可以使用共轭梯度法解出。

该模型求解的时候，正则项不再以将图像灰度接近于 0 为目标，而是以“**弱化图像噪声、边界及纹理信息**”为主要目的。λ 的增大意味着更强力的去噪，也更容易对图像信息造成破坏，直到将图像磨平成几乎等于常数的灰图。

正则化模型总结

正则化类型	优点	缺点	适用场景
L2 正则化（岭回归）	防止过拟合，解决共线性问题	无法产生稀疏模型，所有特征都参与计算	图像去噪、图像恢复
Tikhonov 正则化	考虑图像梯度，适用于图像去噪和恢复。 <b>“弱化图像噪声、边界及纹理信息”</b>	计算复杂度较高，参数选择较为复杂	图像处理、图像去噪

第十四课：正则化模型与优化算法（下）