

# 1. 绪论

## 采样量化与图像质量的关系

- 1) 空间位置离散化 - - - 采样，对于空间上的细分程度
  - 2) 性质空间离散化（灰度） - - 量化，对于灰度等级上的细分程度
- 采样越多，图像质量越好  
采样等级差会出现棋盘模式  
量化等级差会出现虚假轮廓

## 图像概念

灰度图像：图像只有明暗程度的变化而没有色彩的变化，最简单的是二值图像，只有两种灰度；

伪彩色图像：是指经过伪彩色处理而形成的彩色图像。其像素值是所谓的索引值，是按照灰度值进行彩色指定的结果，其色彩并不一定忠实于外界景物的真实色彩；

假彩色图像：是指遥感多波段图像合成的彩色图像；

真彩色图像：是忠实于外界景色的色彩的图像，其像素一般是颜色的真实值。

静止图像：图像的内容不随时间变化。

活动（连续）图像：前一帧和后一帧的图像内容随时间发生变化。

矢量图像：由描述像素点阵的一组数据刻画的图像。

点阵图像(光栅图像)：由像素点阵所组成的图像。

彩色图像存储：RGB，一个像素点24位

## 打印

### 1. 灰度图像

半调技术：将一个像素区域细分多个单元，每个单元输出黑或白，形成不同灰度级

### 2. 彩色图像

减色模型：青，品红，黄，黑；CMYK模型

## 2、人眼的视觉特性

### 特性

1. 亮度恒定：具有对数性质。只要对比度保持一定，亮度即使在很宽范围内变动，人的亮度感受也是相同的
2. 亮度区分能力有限：使用韦伯比，作为区分的光斑亮度与背景的比值，越大说明越差。最小为1。

### 成像模型

#### 几何成像模型

1. 图像坐标变换

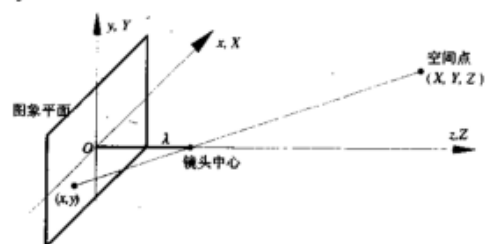
摄像机坐标( $x_x, y_y, z_z$ )与图像平面坐标之间 ( $x, y, z$ )的关系是 $x_x = x$ ;  $y_y = y$ ,  $z = 0$ ;

摄像机的 $z$ 也是成比例的投射过去

- 利用投影变换矩阵

$$c_h = Pw_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/\lambda & 1 \end{bmatrix} \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} = \begin{bmatrix} kX \\ kY \\ kZ \\ -kZ/\lambda + k \end{bmatrix}$$
$$c = [x \quad y \quad z]^T = \begin{bmatrix} \frac{\lambda X}{\lambda - Z} & \frac{\lambda Y}{\lambda - Z} & \frac{\lambda Z}{\lambda - Z} \end{bmatrix}^T = [-0.2 \quad -0.4 \quad -0.6]^T$$

空间坐标 $X, Y, Z$ ，摄像机坐标 $x, y, z$ ，图像坐标 $x, y$



投影成像示意图

- 2.

#### 亮度成像模型

同态滤波时用到

### 图像基础

#### 像素之间的关系

1. 四邻域，链接，联通
2. 八邻域，链接
3. M链接

可以直着联，也可以斜着联，同时出现时优先4链接

#### 灰度直方图

统计特性，直方图

## 图像代数运算

### 1. 加

去噪，特性

两幅图像相加，可平滑噪声等

要提高信噪比多少倍，就要有多少个图像叠加

### 2. 减

去除固定背景，发现两幅图像之间是否有运动变化，梯度

### 3. 乘

二值模板图像与原图像做乘法，实现掩模

操作

### 4. 除

用于校正图像灰度阴影

## 图像几何运算

### 1. 空间变化

#### i. 平移矩阵

矩阵形式

#### (1) 平移变换

将空间点  $(X, Y, Z)$  平移到  $(X', Y', Z')$

$$\begin{cases} X' = X + X_0 \\ Y' = Y + Y_0 \\ Z' = Z + Z_0 \end{cases}$$

写成齐次变换矩阵形式：

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\text{即 } V' = TV$$

#### ii. 旋转矩阵

### (3) 旋转变换:

空间点:

以X轴顺时针旋转 $\alpha$ :

$$\text{旋转矩阵 } R_{\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

以Y轴顺时针旋转 $\beta$ :

$$\text{旋转矩阵 } R_{\beta} = \begin{bmatrix} \cos \beta & 0 & -\sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

以Z轴顺时针旋转 $\gamma$ :

$$\text{旋转矩阵 } R_{\gamma} = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 & 0 \\ -\sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

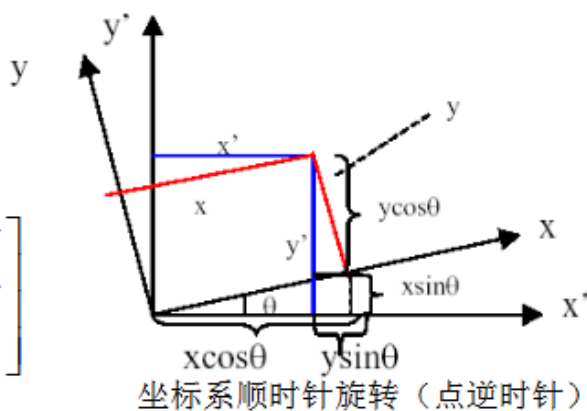
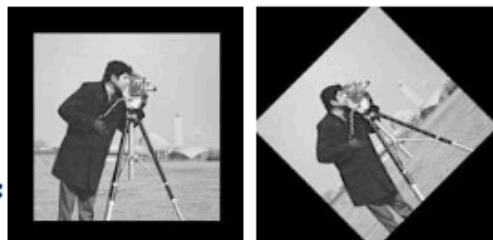
将空间点A绕空间任一点B旋转需要3个变换:

- (1) 平移B到原点
- (2) 将点A绕原点旋转
- (3) 平移点B回到原始位置

图像上点:

绕原点顺时针旋转 $\theta$ 角:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



空间点:

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$V' = SV$$

图像上点:

a. 水平镜像

设:  $X' = -x, y' = y$

用齐次矩阵表示:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

b. 垂直镜像

设:  $X' = x, y' = -y$

用齐次矩阵表示:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

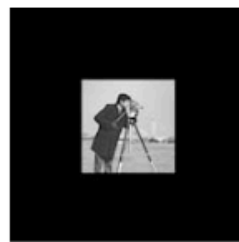
c. 放缩变换

$x$ 方向放缩 $c$ 倍,  $y$ 方向放缩 $d$ 倍

设:  $x' = cx, y' = dy$

用齐次矩阵表示:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} c & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



例:  $S = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

图像尺度变换矩阵:

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

图像尺度变换逆矩阵:

$$S^{-1} = \begin{bmatrix} 1/S_x & 0 & 0 \\ 0 & 1/S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

## 2. 仿射变换

是空间变换的组合, 可以直接矩阵相乘以达到效果

## 灰度插值

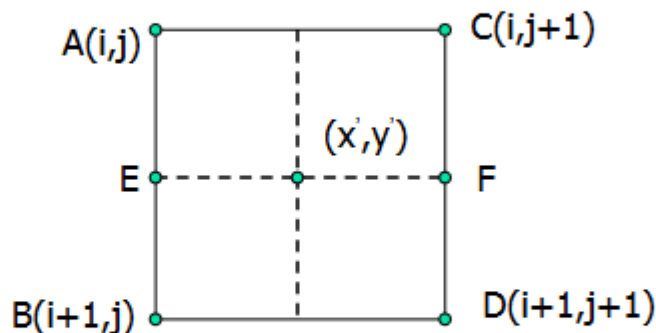
### 1. 最近邻插值效果不好

产生马赛克效应

## 2. 双线性插值

### (2) 双线性插值（一阶插值）

已知正方形的4个顶点，求正方形内部的点，有双线性方程： $f(x, y) = ax + by + cxy + d$



E点的灰度：

$$g(E) = (x' - i)[g(B) - g(A)] + g(A)$$

$$g(F) = (x' - i)[g(D) - g(C)] + g(C)$$

$$g(x', y') = (y' - j)[g(F) - g(E)] + g(E)$$

根据距离四个点的距离做加权平均

## 3. 失真矫正 可能不考

# 3 图像空间域增强

## 点处理模式

点处理模式，输出与输入相关

### 1. 灰度变换

#### i. 线性变换：

线性拉伸，分段拉伸

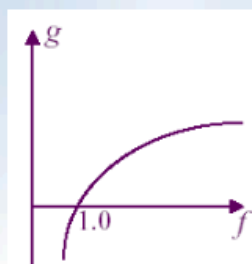
#### ii. 非线性变换

## 2. 非线性变换

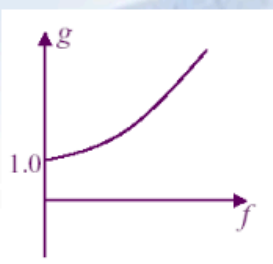
原始图像 $f(i, j)$ 的灰度范围为 $[a, b]$ ，可以通过自然对数变换到区间 $[a', b']$ 上，从而求得图像 $f'(i, j)$ ：

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

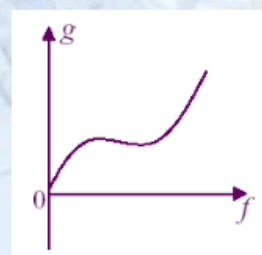
这一变换扩展输入图像中低灰度区的对比度，而压缩高灰度值。



a) 对数变换



b) 指数变换



c) 一般非线性变换

15

### 2. 直方图增强

#### i. 均衡化

### 直方图均衡化算法：

(1) 统计图像中各灰度级像素个数  $n_k$

(2) 计算直方图：  $p_r(r_k) = \frac{n_k}{n}$

(3) 计算累积直方图：  $s_k = T(r_k) = \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{n}$

(4) 取整  $S_k = \text{int}[(L-1)s_k + 0.5]$ ;

(5) 确定映射对应关系：  $k \rightarrow S_k$ ;

(6) 对图像进行增强变换(  $k \rightarrow S_k$  ).

其中 $L$ 是灰度等级，  $n$ 是图像总像素数

首先计算各个灰度值的累计概率函数，之后根据这个值乘以总灰度个数，四舍五入即为新的灰度

## ii. 规定化

分别计算原本图像的累计直方图，和规定的累计直方图

- a. SML单映射：为每一个原始灰度找到一个唯一映射，标准是找距离自己最近的一个
- b. GML组映射：为每一个目标灰度找到一组映射，标准是找到最近的原始灰度，之后将包括他及其前面没有被分配的都包含进去

## 邻域处理

与输入邻域想关

### 1. 平滑：去噪，领域平均，模糊

- i. 邻域平均滤波器：用于消除细小噪声，图像会模糊
- ii. 中值滤波器：消除脉冲噪声，较好地保护边缘轮廓信息
- iii. 中值滤波器适用于椒盐噪声污染的图像，平滑滤波适用于高斯噪声污染的噪声

### 2. 锐化边缘

#### i. 一阶微分算子

所有算子系数和为0：Sobel, Canny

若是平滑的系数和为1

#### ii. 二阶：

拉普拉斯算子

## 4 图像变换

### 傅里叶变换性质

一维：  $F(u) = \int_{-\infty}^{\infty} f(x)e^{-2\pi jux} dx$

离散：  $F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x)e^{-2\pi jux/N}$

二维：  $ux \rightarrow ux+vy$

#### 1. 离散的分线性

先做y方向的，在做x方向的与直接二维相同

#### 2. 周期性：

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

#### 3. 共轭对称性

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

$$|F(u,v)| = |F(-u,-v)|$$

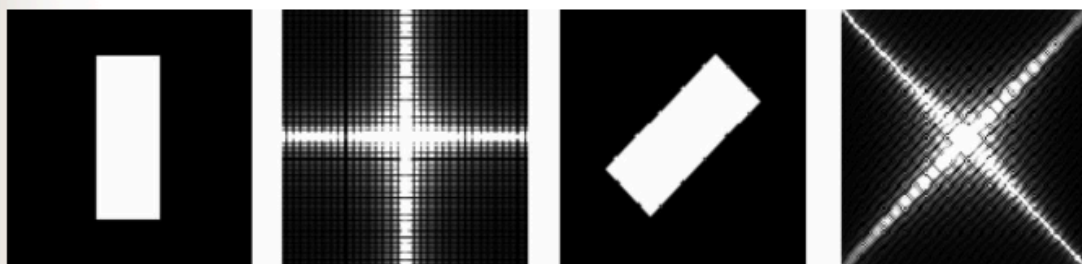
#### 4. 旋转性，加法性



#### 4. 旋转性:

$$f(r, \theta) \Leftrightarrow F(\omega, \phi)$$

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



#### 5. 加法定理: $F[f_1(x,y) + f_2(x,y)] = F[f_1(x,y)] + F[f_2(x,y)]$

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

#### 5. 尺度变换 (相似性定理):

$$f(ax, by) = \frac{1}{ab} F(u/a, v/b)$$

#### 6. 卷积定义与计算

##### (1). 卷积积分(连续): 如果函数 $y(t)$ 满足下列关系式

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)d\tau = x(t) * h(t)$$

#### 7. 卷积定理

$$(1) 1D \text{情况下: } f_e(x) * g_e(x) = \frac{1}{M} \sum_{m=0}^{M-1} f_e(m)g_e(x-m)$$

$$(2) 2D \text{情况下: } f_e(x,y) * g_e(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e(m,n)g_e(x-m,y-n)$$

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

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

(3)卷积的应用: 去除噪声; 特征增强

$$=F(0,0)/N$$

## 9. 相关定理

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

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

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

$$\text{自相关: } f(x,y) \circ f(x,y) \Leftrightarrow |F(u,v)|^2$$

- **相关:** 如果函数  $z(t)$  满足下列关系式

$$z(t) = \int_{-\infty}^{+\infty} x^*(\tau) h(t+\tau) d\tau = x(t) \circ h(t)$$

则称函数  $z(t)$  为函数  $x(t)$  和  $h(t)$  的相关函数

39

- **相关定理:** 如果  $x(t)$  和  $h(t)$  的傅里叶变换分别为  $X(f)$  和  $H(f)$ , 则  $x(t)$  和  $h(t)$  的相关的傅里叶变换为  $X^*(f)H(f)$ 。即

$$\int_{-\infty}^{+\infty} x^*(\tau) h(t+\tau) d\tau \Leftrightarrow X^*(f)H(f)$$

其中,  $X^*(f)$  为  $X(f)$  的复共轭

卷积在不同周期要周期扩展到同一周期

## FFT

### 傅立叶变换:

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j \frac{2\pi ux}{N}}$$

$$= \frac{1}{N} \sum_{x=0}^{N-1} f(x) \left( \cos \frac{2\pi ux}{N} - j \sin \frac{2\pi ux}{N} \right)$$

## 一维快速Fourier变换:

## 一、快速Fourier变换的推导

$$\text{令: } w_N^{\mu x} = \exp(-j \frac{2\pi \mu x}{N})$$

$$\begin{aligned} \text{则: } F(\mu) &= \frac{1}{N} \sum_{x=0}^{N-1} f(x) w_N^{\mu x} \\ &= \frac{1}{2} \left[ \frac{2}{N} \sum_{x=0}^{N/2-1} f(2x) w_N^{2\mu x} + \frac{2}{N} \sum_{x=0}^{N/2-1} f(2x+1) w_N^{\mu(2x+1)} \right] \\ &\stackrel{\text{M}}{\underset{\text{N}}{\triangle}} \frac{1}{2} \left[ \frac{1}{M} \sum_{x=0}^{M-1} f(2x) w_M^{\mu x} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) w_M^{\mu x} w_N^{\mu} \right] \\ &= \frac{1}{2} [F_e(\mu) + w_N^{\mu} F_o(\mu)] \quad \left( \text{分成奇数项和偶数项之和} \right) \end{aligned}$$

## DCT离散余弦变换

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

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & u = 1, 2, \dots, N-1 \end{cases}$$

偶数点产生的是实偶，变成偶函数。使用FFT实现DCT  
借助傅立叶变换计算余弦变换的步骤：

## 借助傅立叶变换计算余弦变换的步骤:

- 1)把  $f(x)$  延拓成  $f_e(x)$  , 长度为 $2N$  ;
- 2)求 $f_e(x)$  的 $2N$  点的FFT;
- 3)对 $u$ 各项乘上对应的因子 $\sqrt{2} \exp\left(-j \frac{\pi u}{2N}\right)$ ;
- 4)取实部, 并乘上因子 $\sqrt{\frac{1}{N}}$  ;
- 5)取 $F(u)$ 的前 $N$ 项, 即为 $f(x)$ 的余弦变换。

## woshi变换

### 核函数

相当于求 $f(x)$ 的平均值, 只不过加的权重为-1的 $b$ 次幂,  $b$ 为 $x$ 的正序,  $u$ 的逆序二进制乘积之和

### 1. 1D情况下:

$$\text{核函数 } g(x, u) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{b_i(x)b_{n-1-i}(u)} \quad N=2^n$$

$b_i(x)$  是 $x$ 的二进制的第 $i$ 位

Walsh正变换:

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

Walsh反变换:

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

# Hadamard

核函数中的x与u是同向的了

## 4. Hadamard变换的矩阵形式:

$$1D : N=2 \text{ 时 } H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$N=4 \text{ 时 } H_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix}$$

$$H_{2N} = \frac{1}{\sqrt{2N}} \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

$$T=AFA \quad A = \frac{1}{\sqrt{N}} H_N$$

沿Hadamard矩阵的列方向符号变换的次数通常叫做**列率/序率**。—> 递增

1D下排序后变换核:

$$g(x, u) = \frac{1}{N} (-1)^{\sum_{j=0}^{n-1} b_j(x) p_j(u)}$$

$$\left. \begin{aligned} p_0(u) &= b_{n-1}(u) \\ p_1(u) &= b_{n-1}(u) + b_{n-2}(u) \\ &\vdots \\ p_{n-1}(u) &= b_1(u) + b_0(u) \end{aligned} \right\}$$

## kl变换

特征矢量作为基函数, 均值为0, 协方差  
协方差矩阵为每个元素减去这一行的均值

## 数字特征的计算：

$$m_X \approx \frac{1}{M} \sum_{i=1}^M X_i$$

$$C_X \approx \frac{1}{M} \sum_{i=1}^M (X_i - m_X)(X_i - m_X)^T \approx \frac{1}{M} \sum_{i=1}^M X_i X_i^T - m_X m_X^T$$

A的每一行都是特征向量

## K-L变换的计算步骤：

1. 求协方差矩阵  $C_X$  ；
2. 求协方差矩阵的特征值  $\lambda_i$  ；
3. 求相应的特征向量  $e_i$
4. 用特征向量  $e_i$  构成变换矩阵A，求  $Y = A(X - m_X)$  。

## 5 图像变换域增强

### 滤波器

1. 理想低通滤波：去除小噪声
  - i. D0半径内的频率分量无损通过，圆外的频率分量会被滤除。  
起到平滑作用，去除小噪声；
  - ii. 若滤除的高频分量中含有大量的边缘信息，会发生图像边缘模糊现象。
  - iii. 产生振铃现象
2. 巴特沃斯低通：去处虚假轮廓



## 2. 巴特沃思低通滤波器 (BLPF)

$n$ 阶巴特沃思低通滤波器的传递函数为:

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

或者:

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

模糊程度减少, 无振铃

尾部含有较多的高频, 对噪声的平滑效果不如ILPF。

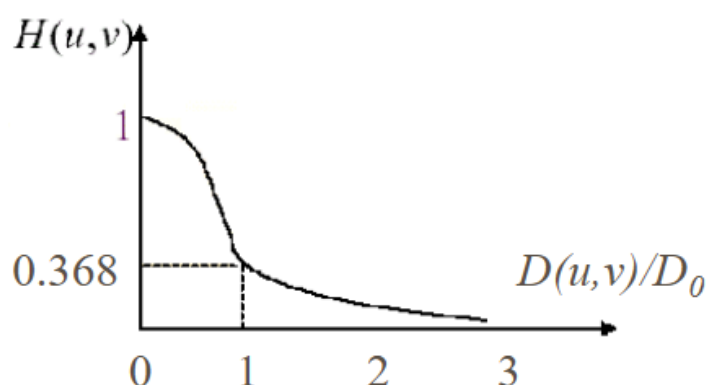
### 3. 指数低通滤波器

#### ❖ 3. 指数低通滤波器 (ELPF)

$$H(u,v) = e^{-\ln \frac{1}{\sqrt{2}} \left[ \frac{D(u,v)}{D_0} \right]^n} \quad \text{或} \quad H(u,v) = e^{-\left[ \frac{D(u,v)}{D_0} \right]^n}$$

☑ 有更加平滑的过渡带, 平滑后的图像没有跳跃现象

☑ 与BLPF相比, 衰减更快, 经过ELPF滤波的图像比BLPF处理的图像更模糊一些



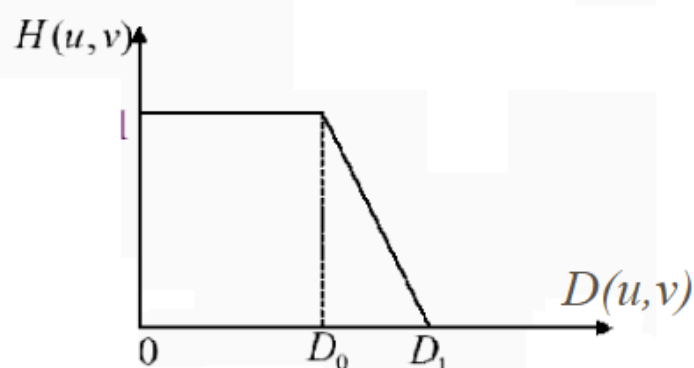
二阶称为高斯低通滤波器, 无振铃效应

### 4. 梯形低通滤波器

$$H(u,v) = \begin{cases} 1 & D(u,v) < D_0 \\ \frac{D(u,v) - D_1}{D_0 - D_1} & D_0 \leq D(u,v) \leq D_1 \\ 0 & D(u,v) > D_1 \end{cases}$$

其中  $D_0 < D_1$ 。

一般情况下，定义  $D_0$  为截止频率。



振铃效应较轻

5. 高通滤波器

反过程

6. 高频增强滤波

**通常采用高频增强滤波： $H_e(u,v) = H(u,v) + c$**

高通图像加上原图

7. 同态滤波：去除乘性噪声



图像：
$$f(x, y) = i(x, y)r(x, y)$$

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

再取傅立叶变换：
$$Z(u, v) = I(u, v) + R(u, v)$$

用传递函数 $H(u, v)$ 进行滤波处理：

$$S(u, v) = H(u, v)Z(u, v)$$

取傅立叶反变换，便可得空间域输出 $s(x, y)$

最后，滤波后的结果为：

$$g(x, y) = e^{s(x, y)}$$

## 6 图像恢复与重建

重建不考

**恢复：**

**逆滤波**

放大噪声。当退化图像的噪声较小，即轻度降质时，采用逆滤波复原的方法可以获得较好的结果。

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

放大噪声的原因： $H(x, y)$ 的幅值随着 $u, v$ 离原点的距离的增加而迅速下降，而噪声的幅值变化则较平缓  
方法：利用 $F^{\wedge}(u, v) = G(u, v) / H(u, v)$ 求估计值， $f^{\wedge}(x, y) = F^{-1} [F^{\wedge}(u, v)]$

**维纳滤波**

不会放大噪声

当 $D$ 为对角阵，分块循环矩阵

$$H = WDW^{-1}$$
$$H^T = WD^*W^{-1}$$

因此：

$$W^{-1}\hat{f} = (DD^* + \gamma A^{-1}B)^{-1}D^*W^{-1}g$$

写成频域形式为：

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

其中 $p_f(u,v), p_n(u,v)$ 分别是 $f(x,y)$ 和 $n(x,y)$ 的功率谱。

$\gamma = 1$ 时， $\hat{F}(u,v)$ 称为维纳滤波器

$\gamma \neq 1$ 时，称为参数维纳滤波器

当没有噪声时， $p_n(u,v)=0$ ，为逆滤波

## 运动模糊

传递函数的推导：

若物体不运动时:

$$\text{成像: } g(x,y) = \int_0^T f(x,y) dt \quad \text{其中曝光时间为} T$$

当成像传感器与被摄景物之间存在足够快的相对运动时, 所摄取的图像就会出现“运动模糊”, 运动模糊是场景能量在传感器拍摄瞬间 ( $T$ ) 内在像平面上的非正常积累。

物体作匀速直线运动时:  $x_0(t)$ 、 $y_0(t)$  分别是景物在  $x$ 、 $y$  方向的运动量

$$\text{模糊图像: } g(x,y) = \int_0^T f[x-x_0(t), y-y_0(t)] dt$$

上式进行傅立叶变换得到:

$$\begin{aligned} G(u,v) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x,y) \exp(-j2\pi(ux+vy)) dx dy \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left\{ \int_0^T f[x-x_0(t), y-y_0(t)] dt \right\} \exp(-j2\pi(ux+vy)) dx dy \\ &= F(u,v) \int_0^T \exp[-j2\pi(ux_0(t)+vy_0(t))] dt \end{aligned}$$

35



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

$$\text{则有 } G(u,v) = H(u,v) \cdot F(u,v)$$

$H(u,v)$  为运动模糊的传递函数。

## 7 图像压缩

有损无损压缩

平均码长不可能小于信息熵

### 无损

#### 编码冗余

##### 1. huffman编码

(1) B2码: 信息位两位, 标志位1位

信息为用C表示, 总格式为CxxCxx, xx长度为B2中的2. 之后可以按照概率从小到大编码, 每个C后面有 $2^2$ 种, 从一个C到多个C编码

(2) 截断Huffman码：对M个出现概率最大的符号进行Huffman编码，第M个Huffman码作为前缀码

(3) 二元平移码：将所有符号概率由大到小排列，然后分成各有7个符号的n块，将第一块作为参考块。

nbit平移码， $2^n$  中编码，留一个当前缀，一般为全1，之后每 $2^{n-1}$ 个编码后在前面加一个前缀。

(4) Huffman平移码：将所有符号概率由大到小排列，然后分成各有7个符号的n块，将第一块 $a_1...a_7$  与 $a_8'$ 进行Huffman编码

与上一种方法类似，只不过编码为变长的哈夫曼编码而不是自然码，一般预留00作为前缀

2. 算数编码：对一串数据用一个编码表示，编码本身确定在0和1之间的1个实数区间。

3. 游程编码

假设0,1交替出现.

### ■ 举例说明：

aaaa bbb cc d eeeeee ffffff (共 $22 \times 8 = 176$  bits)

→ 4a3b2c1d5e7f (共 $12 \times 8 = 96$  bits)

### ■ 在传真中的应用：

传真件中一般都是白色比较多，而黑色相对比较少。所以可能常常会出现如下的情况：

600W 3b 100w 12b 4w 3b 200w

**定长游程编码：**上面的行程编码所需用的字节数为：

因为： $512 < 600 < 1024$

所以：计数值必须用10bit来表示

$10 \times 7 = 70\text{bit}$

**变长游程编码：** 白色：10bit，黑色：4bit

600W 3b 100w 12b 4w 3b 200w

所需字节数为：  $4 \times 10 + 3 \times 4 = 52\text{bit}$

比原来的方式 $10 \times 7 = 70\text{bit}$ 减少了18bit

36

4. LZW编码

P为上一个码，C为当前输入。PC不在表中，输出P， $P=C$ ，添加PC到表，否则 $P=PC$

## LZW编码算法具体步骤:

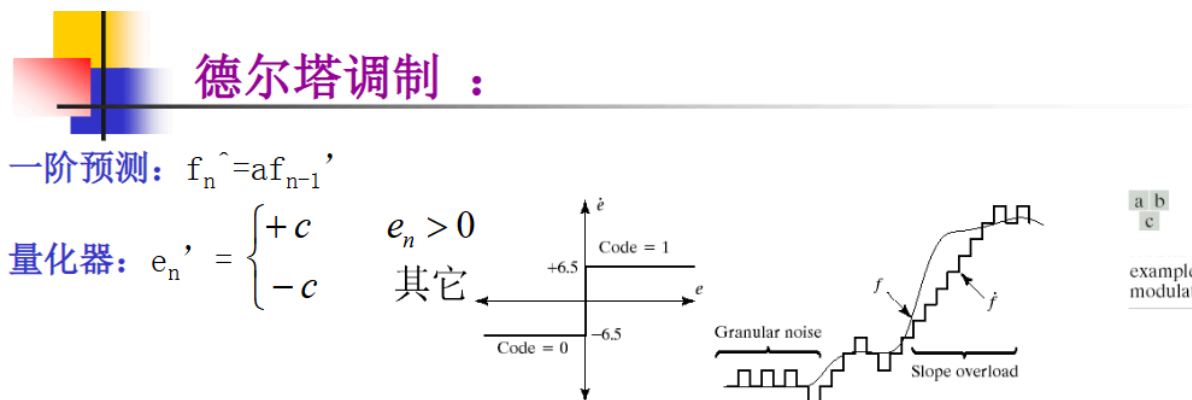
- 步骤1:** 将词典初始化为包含所有可能的单字符,  
当前前缀P初始化为空;
- 步骤2:** 当前字符C的内容为输入字符流中的下一个字符;
- 步骤3:** 判断P+C是否在词典中
- (1) 如果“是”, 则用C扩展P, 即让 $P=P+C$ ;
  - (2) 如果“否”, 则
    - ① 输出当前前缀P的码字到码字流;
    - ② 将P+C添加到词典中;
    - ③ 令前缀 $P = C$  (即现在的P仅包含一个字符C);
- 步骤4:** 判断输入字符流中是否还有码字要编码
- (1) 如果“是”, 就返回到步骤2;
  - (2) 如果“否”
    - ① 把当前前缀P的码字输出到码字流;
    - ② 结束。

37

## 像素间相关性冗余

### 1. 预测编码

- i. 无损预测
- ii. 有损



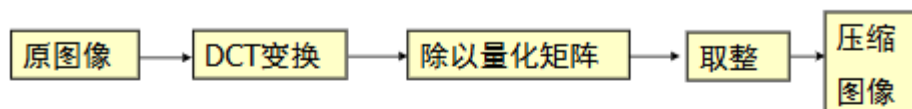
Input		Encoder				Decoder		Error
$n$	$f$	$\hat{f}$	$e$	$\hat{e}$	$\hat{f}$	$\hat{f}$	$\hat{f}$	$[f - \hat{f}]$
0	14	—	—	—	14.0	—	14.0	0.0
1	15	14.0	1.0	6.5	20.5	14.0	20.5	-5.5
2	14	20.5	-6.5	-6.5	14.0	20.5	14.0	0.0
3	15	14.0	1.0	6.5	20.5	14.0	20.5	-5.5
·	·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·	·
14	29	20.5	8.5	6.5	27.0	20.5	27.0	2.0
15	37	27.0	10.0	6.5	33.5	27.0	33.5	3.5
16	47	33.5	13.5	6.5	40.0	33.5	40.0	7.0
17	62	40.0	22.0	6.5	46.5	40.0	46.5	15.5
18	75	46.5	28.5	6.5	53.0	46.5	53.0	22.0
19	77	53.0	24.0	6.5	59.6	53.0	59.6	17.5
·	·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·	·

有损预测编码系统特性: 会引起斜率过载、界线繁忙、颗粒噪声和轮廓噪声

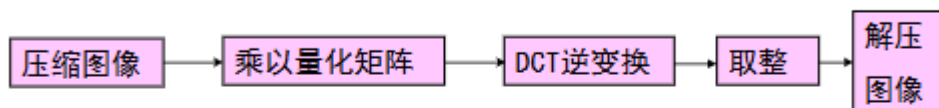
### 2. 变化编码

## DCT变换编码方法:

### 1) 编码过程:



### 2) 解码过程:



70

例:

原图像为:  $F = \begin{bmatrix} 59 & 60 & 58 & 57 \\ 61 & 59 & 59 & 57 \\ 62 & 59 & 60 & 58 \\ 59 & 61 & 60 & 56 \end{bmatrix}$   $\xrightarrow{\text{DCT变换}}$   $D_1 = \begin{bmatrix} 120.5 & 119.5 & 118.5 & 114.0 \\ -0.27 & -0.65 & -1.58 & 0.38 \\ -2.50 & 1.50 & -0.50 & -1.00 \\ 0.65 & -0.27 & 0.11 & 0.92 \end{bmatrix}$

Huffman: 42bits

除以量化矩阵, 取整

$C = \begin{bmatrix} 16 & 11 & 11 & 16 \\ 12 & 12 & 14 & 19 \\ 14 & 13 & 16 & 24 \\ 14 & 17 & 22 & 29 \end{bmatrix}$   $\xrightarrow{\text{除以量化矩阵, 取整}}$   $D = \begin{bmatrix} 8 & 11 & 11 & 7 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

Huffman: 28bits

71

## 视觉相关性冗余

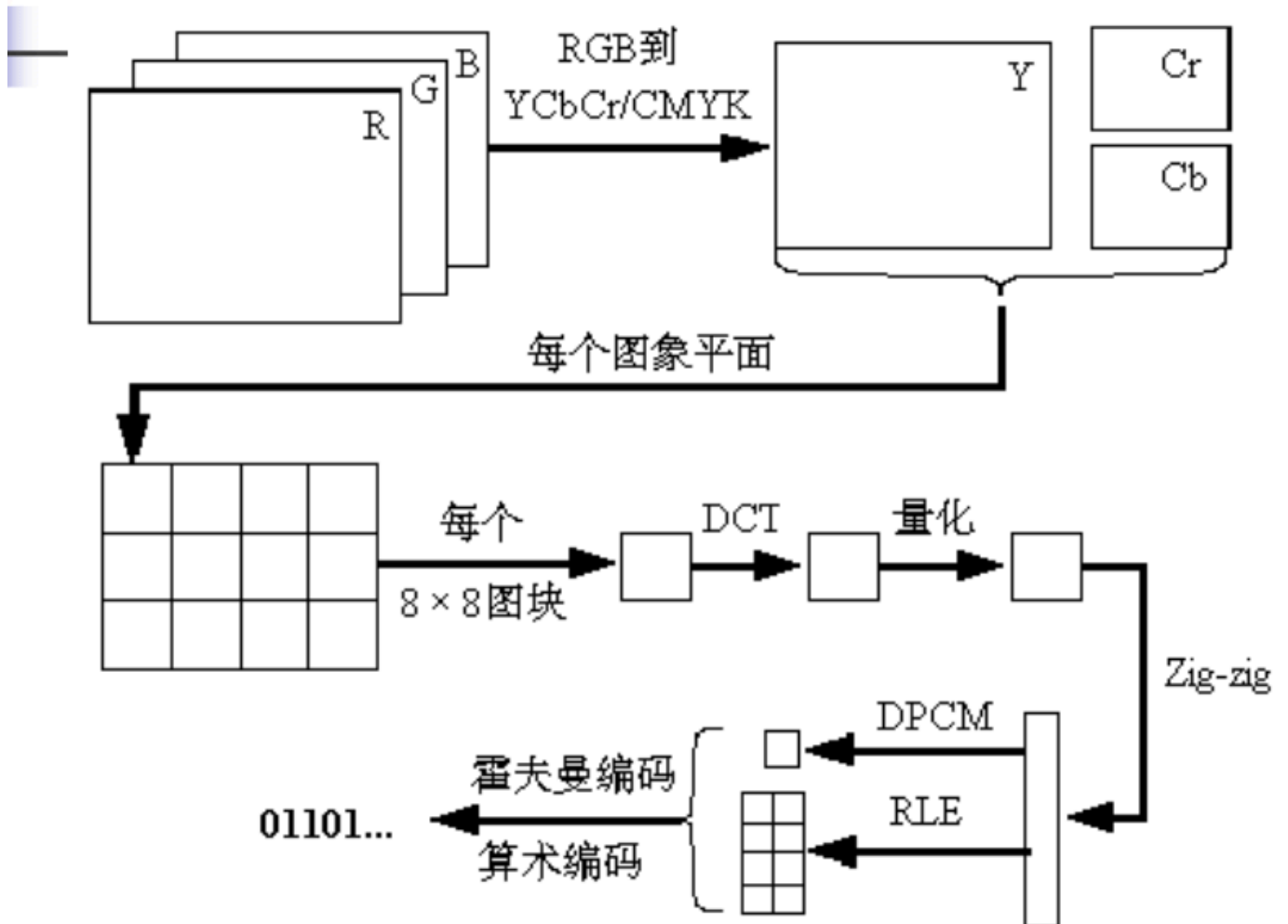
## 压缩标准

JPEG压缩标准: 产生分块效应

JPEG压缩编码算法的主要计算步骤如下:

1. 正向离散余弦变换(FDCT)。
2. 量化(quantization)。
3. Z字形编码(zigzag scan)。

4. 使用差分脉冲编码调制(differential pulse code modulation, DPCM)对直流系数(DC)进行编码。
5. 使用行程长度编码(run-length encoding, RLE)对交流系数(AC)进行编码。
6. 熵编码(entropy coding)



## 8、图像分割

### 灰度分割

#### 1. 直方图分割

使得类间方差最大，使用自动阈值划分，适合于图像直方图是双峰情况

算法实现为遍历灰度，找到使得类间方差最大的

具体：加权计算所有类的均值与总均值差的平方的和，即类间方差



假设图像的前景较暗，并且图像的大小为 $M \times N$ ，图像中像素的灰度值小于阈值 $T$ 的像素个数记作 $N_0$ ，像素灰度大于阈值 $T$ 的像素个数记作 $N_1$ ，则有：

$$\omega_0 = N_0 / (M \times N) \quad (1)$$

$$\omega_1 = N_1 / (M \times N) \quad (2)$$

$$N_0 + N_1 = M \times N \quad (3)$$

$$\omega_0 + \omega_1 = 1 \quad (4)$$

$$\mu = \omega_0 * \mu_0 + \omega_1 * \mu_1 \quad (5)$$

$$g = \omega_0 * (\mu_0 - \mu)^2 + \omega_1 * (\mu_1 - \mu)^2 \quad (6)$$

式(5)代入式(6)，得到等价公式： $g = \omega_0 * \omega_1 * (\mu_0 - \mu_1)^2$  (7)

用遍历的方法可以得到类间方差最大的阈值 $T$ 。

## 边界分隔

边界判断边缘

1. hough变换：检测已知形状的曲线，如寻找直线

## 区域的分割方法

1. 区域生长

具体步骤：

- (1)选取或确定一组能正确代表所需区域的种子像素
- (2)确定在生长过程中能将相邻像素包括进来的准则
- (3)制定让生长停止的条件或规则

2. 分割合并

- (1) 对任一个区域 $R_i$ ，如果 $P(R_i) = \text{FALSE}$ 就将其分裂成不重叠的4等份，即内部不是纯色
- (2) 对相邻的2个区域 $R_i$ 和 $R_j$ ，如果 $P(R_i \square R_j) = \text{TRUE}$ ，就将它们合并
- (3) 如果进一步的分裂或合并都不可能，则结束

## 纹理的分割方法

1. 灰度共生矩阵：

灰度共生矩阵为概率矩阵： $P_{ij}$  其中 $P_{ij}$ 是距离为 $d$ 的两个像素，一个像素的灰度值为 $i$ ，另一个像素灰度值为 $j$ 的情况在整幅图像中出现的频率。

会给出 $d$ 与方向。注意0度，距离为1的话， $(3, 3)$  会算为2次



# 二值图像的形态学

## 1. 膨胀

简单向外扩充一个像素，将与物体边界接触的背景像素合并到物体中

## 2. 腐蚀

将一个物体沿边界减小的过程，在物体的周边减少一个像素

## 3. 开

腐蚀运算后再进行膨胀运算的组合运算

## 1. 删除小物体；

## 2. 将物体拆分为小物体；

## 3. 平滑大物体边界而不明显改变它们的面积；

## 4. 闭

膨胀运算后再进行腐蚀运算

## 1. 填充物体的小洞；

## 2. 连接相近的物体；

## 3. 平滑物体的边界而不明显改变它们的面积

# 9、目标表达和描述

## 边界表达

### 链码

用边界的方向作为编码依据。为简化边界的描述。一般描述的是边界点集

## 1. 链码起点归一化：依一个方向循环，使所构成的自然数最小

## 2. 链码旋转归一化：通过使用链码的首差（一阶差分）代替码本身的方式。当前位减去前一位。

e.g: 10103322-> 33133030

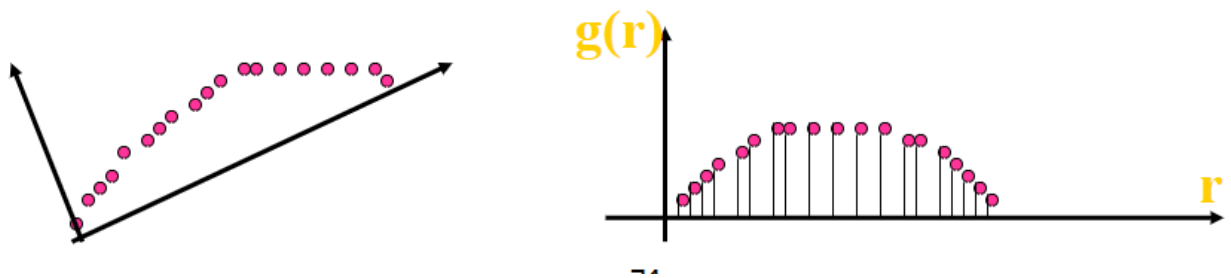
### 边界描述

## 1. 形状数

最小循环首差链码

## 2. 边界矩

把边界当作函数 $g(r)$ ，将 $g(r)$ 的线下面积看成一个直方图



### 3. 骨架

循环执行两个算法

邻域的和为2到6,且0到1转换次数为1,且右三角,下三角乘积均为0



p9	p2	p3
p8	p1	p4
p7	p6	p5

p9	p2	p3
p8	p1	p4
p7	p6	p5

p9	p2	p3
p8	p1	p4
p7	p6	p5

#### ■ 基本操作1

对于满足以下四个条件的边界点打标记准备删除:

(a)  $2 \leq N(p_1) \leq 6$  其中  $N(p_1)$  是点  $p_1$  的邻域中 **1** 的个数,  
即:  $N(p_1) = p_2 + p_3 + \dots + p_9$

(b)  $S(p_1) = 1$

其中  $S(p_1)$  是按  $p_2, p_3, \dots, p_9$  顺序, 0-1 转换的个数

(c)  $p_2 * p_4 * p_6 = 0$  ( $p_2$ 、 $p_4$ 、 $p_6$  至少有一个 **0**)

(d)  $p_4 * p_6 * p_8 = 0$  ( $p_4$ 、 $p_6$ 、 $p_8$  至少有一个 **0**)

i.

前面相同,改成左三角,上三角

## 基本操作2

条件(a)、(b)与操作1相同

条件(c)、(d)改为:

$$c') \quad p_2 * p_4 * p_8 = 0$$

$$d') \quad p_2 * p_6 * p_8 = 0$$

p9	p2	p3
p8	p1	p4
p7	p6	p5

p9	p2	p3
p8	p1	p4
p7	p6	p5

举例:

0	0	0
0	1	1
0	1	1

1	1	0
1	1	0
0	0	0

打标记准备删除

46

ii.

直到不能执行

## 区域描述

### 1. 欧拉数

$$E = C - H$$

C: 区域内的连通组元

H: 区域内的孔数

例如B的欧拉数为1-2, i的为2-0, d为1-1

### 2. 区域矩

矩的定义同概统  $\sum P(x)(x - \mu)^n$

区域矩为图像平面的区域不变矩,

■  $f(x,y)$  的  $p+q$  阶中心矩定义为:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x,y)$$

$$\text{其中 } \bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

## 归一化的中心矩:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad p, q = 0, 1, 2, \dots$$

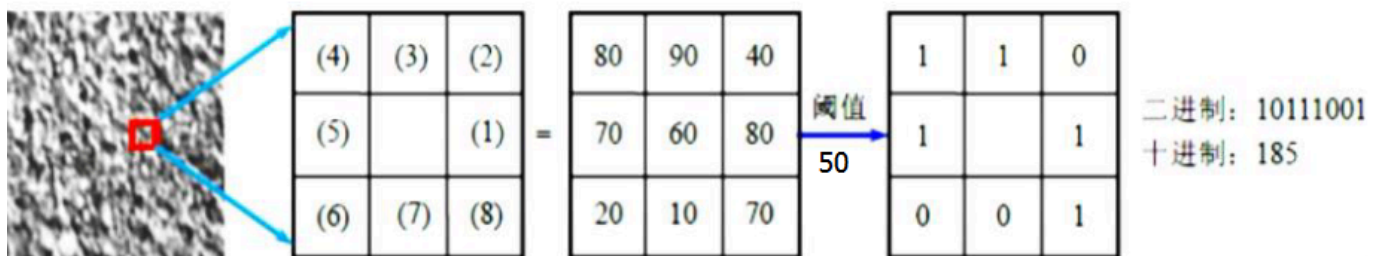
$$\gamma = \frac{p+q}{2} + 1 \quad p, q = 2, 3, 4, \dots$$

矩对平移、旋转、尺度具有不变性

### 3. 局部二值模式

对一个像素的3×3邻域的像素按顺序阈值化，并将结果看作一个二进制数（也可转为十进制），并作为中心像素的标号，所以一个256中，其中58种为有效的。

- 有**256个（2<sup>8</sup>）**不同标号，统计图像每个像素标号得到的直方图可进一步用作整个区域的纹理描述符



判断有效，即均匀模式的依据。从0到1，或者从1到0的变换总共最多2次。  
所有非均匀算作一种，故一共59种符号

## 10 彩色图像处理

### RGB, CMYK

$$[R, G, B] = [1, 1, 1] - [C, M, Y]$$

# HSI

- H定义颜色的波长，称为色调
- S表示颜色的深浅程度，称为饱和度
- I表示强度或亮度，与物体反射率成正比

反映了人的视觉对色彩的感觉，人类观察彩色的方式

## 彩色增强

中值滤波:

采用简化排序：将三个分量相加得到一个标量值

缺点：滤波输出图像中产生彩色“渗色”

## 彩色平衡

将其中一项作为基准，灰色应该三个分量相同的值。之后根据这两个点解两个方程组，算出这两个分量的偏移，之后应用这个偏移来达到平衡的目的

- 1) 从图像中选出两点颜色为灰色的点
- 2) 设以G分量为基准，匹配R和B分量
- 3)

$$F_1 = (R_1, G_1, B_1) \quad F_2 = (R_2, G_2, B_2)$$



$$F_1^* = (G_1, G_1, G_1) \quad F_2^* = (G_2, G_2, G_2)$$

40

3) 由

$$R_1^* = k_1 * R_1 + k_2$$

求出:  $k_1$ 和 $k_2$

$$R_2^* = k_1 * R_2 + k_2$$

$$B_1^* = l_1 * B_1 + l_2$$

求出:  $l_1$ 和 $l_2$

$$B_2^* = l_1 * B_2 + l_2$$

4) 用

$$R(x, y)^* = k_1 * R(x, y) + k_2$$

$$B(x, y)^* = l_1 * B(x, y) + l_2$$

$$G(x, y)^* = G(x, y)$$

## 彩色补偿

某些应用中, 目标是分离出主要或完全是颜色不同的各种类型的物体

左乘 $A_1$ 的逆再左乘 $A_2$

## 2. 彩色补偿算法:

1) 在图像上找到主观视觉看是纯红、绿、蓝的三个点。  
(如果可以根据硬件知道频段的覆盖则无须这样做)。

$$\begin{array}{l} p_1 = (R_1, G_1, B_1) \\ p_2 = (R_2, G_2, B_2) \\ p_3 = (R_3, G_3, B_3) \end{array} \quad \text{它们的理想值为:} \quad \begin{array}{l} p_1^* = (R^*, 0, 0) \\ p_2^* = (0, G^*, 0) \\ p_3^* = (0, 0, B^*) \end{array}$$

### 2) 计算 $R^*$ , $G^*$ , $B^*$ 的值

考虑到彩色补偿之后图像的亮度不变, 对 $R^*$ ,  $G^*$ ,  $B^*$ 的计算如下:

$$\begin{aligned} R^* &= 0.30 \cdot R_1 + 0.59 \cdot G_1 + 0.11 \cdot B_1 \\ G^* &= 0.30 \cdot R_2 + 0.59 \cdot G_2 + 0.11 \cdot B_2 \\ B^* &= 0.30 \cdot R_3 + 0.59 \cdot G_3 + 0.11 \cdot B_3 \end{aligned}$$

57

### 3) 构造变换矩阵

将所取到的三个点的RGB值分别构成彩色补偿前及补偿后的两个矩阵 $A_1$ 和 $A_2$ 。

$$A_1 = \begin{bmatrix} R_1 & R_2 & R_3 \\ G_1 & G_2 & G_3 \\ B_1 & B_2 & B_3 \end{bmatrix} \quad A_2 = \begin{bmatrix} R^* & 0 & 0 \\ 0 & G^* & 0 \\ 0 & 0 & B^* \end{bmatrix}$$

#### 4) 进行彩色补偿

设

$$S(x, y) = \begin{bmatrix} R_s(x, y) \\ G_s(x, y) \\ B_s(x, y) \end{bmatrix} \quad F(x, y) = \begin{bmatrix} R_F(x, y) \\ G_F(x, y) \\ B_F(x, y) \end{bmatrix}$$

分别为新、旧图像的像素值

■ 则：

$$S(x, y) = C^{-1} * F(x, y)$$

其中：  $C = A_1 * A_2^{-1}$

#### 3. 彩色补偿的作用：

可以通过不同的颜色通道提取不同的目标物。

59

## 彩色压缩

### YCBCR格式

### 色度图

(1)色度图中每点对应一种颜色

(2)色度图中边界上的点代表纯色即饱和度最大

(3)两个端点按比例可合成另一种颜色

(4)3个端点连成的三角形中的任意颜色都可由这3个色组成