

# 数字图像知识重点

---

## 绪论和基础

### 图像的分类

---

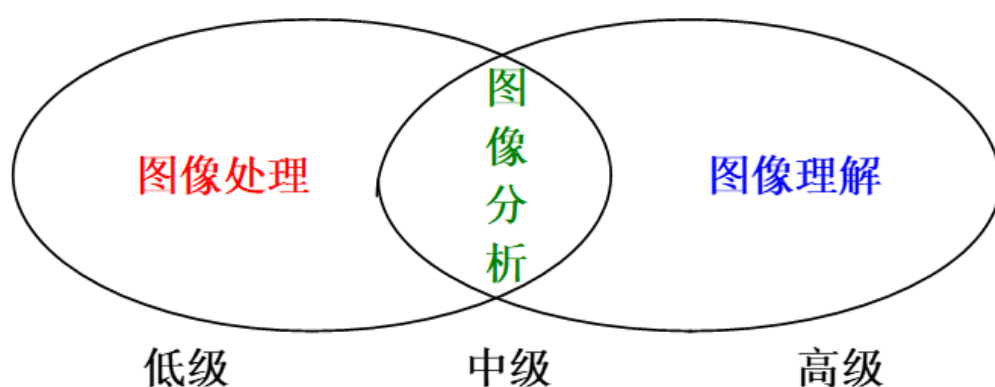
- 根据空间坐标和亮度的连续性

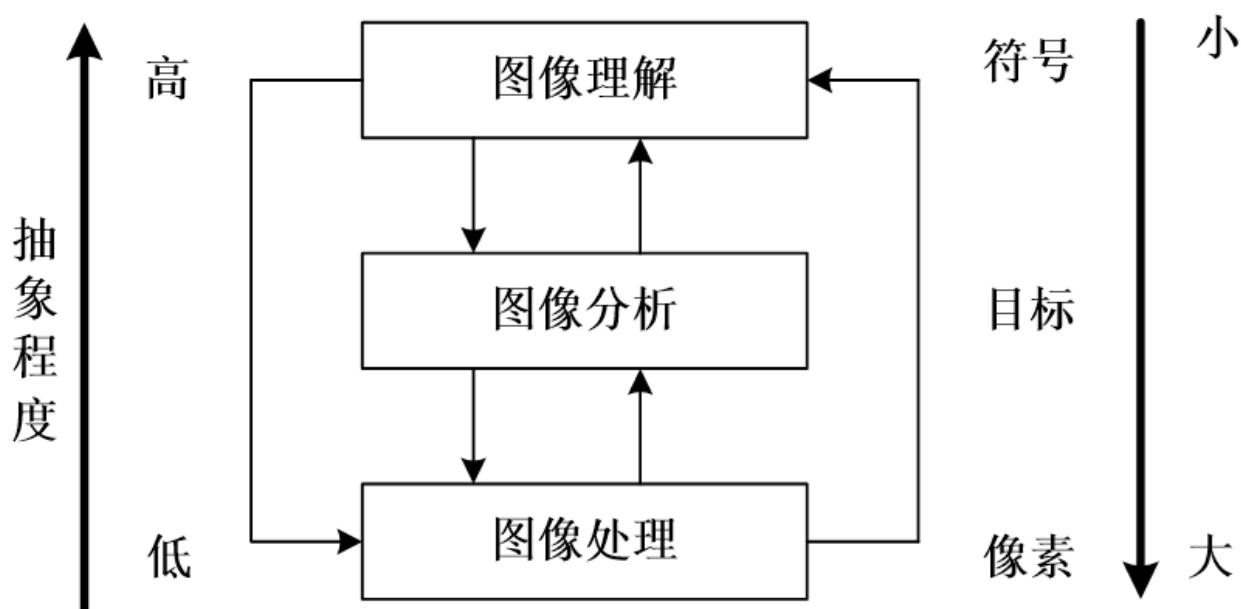
**模拟图像：**空间坐标和亮度都连续变化

**数字图像：**空间坐标和亮度都不连续

### • 图像处理的层次

从计算机处理的角度可以由高到低将数字图像分为三个层次。这三个层次覆盖了图像处理的所有应用领域。





1. 图像处理：对图像进行各种加工，以改善图像的视觉效果；

(1) 强调图像之间进行的变换；

(2) 图像处理是一个**从图像到图像的过程**。

2. 图像分析：对图像中感兴趣的目标进行提取和分割，获得目标的客观信息（特点或性质），建立对图像的描述；

(1) 以观察者为中心研究客观世界；

(2) 图像分析是一个**从图像到数据的过程**。

3. 图像理解：研究图像中各目标的性质和它们之间的相互联系；得出对图像内容含义的理解及原来客观场景的解释；

以客观世界为中心，借助知识、经验来推理、认识客观世界，属于高层操作（符号运算）。

## 颜色模型的基本概念

各种表示颜色的方法，称做颜色模型。目前使用最多的是面向机器（如显示器、摄像机、打印机等）的RGB模型和面向颜色处理（也面向人眼视觉）的HSI（HSV）模型。

## 人眼成像机理和视觉特性

人眼的机理与照相机类似：

(1) 瞳孔：透明的角膜后是不透明的虹膜，虹膜中间的圆孔称为瞳孔，其直径可调节，控制进入人眼内之光通量(照相机光圈作用)。

(2) 晶状体：瞳孔后是一扁球形弹性透明体，其曲率可调节，以改变焦距，使不同距离的图在视网膜上成象(照相机透镜作用)。

(3) 视细胞：视网膜上集中了大量视细胞，分为两类：

锥状细胞：白昼视觉或亮视觉，在强光下检测亮度和颜色；

杆(柱)状细胞：暗视觉或微光视觉，在弱光下检测亮度，无色彩感觉。

其中，每个锥状视细胞连接着一个视神经末梢，故分辨率高，分辨细节、颜色；多个杆状视细胞连接着一个视神经末梢，故分辨率低，仅分辨图的轮廓。

## 人的视觉特性

### 图像“黑”“白”（“亮”、“暗”）对比参数

- 对比度:  $c = B_{\max} / B_{\min}$  ,
- 相对对比度:  $cr = (B - B_0) / B_0$

### 人眼亮度感觉范围

- 总范围很宽 ( $c = 10^8$ )
- 人眼适应某一环境亮度后，范围限制
  - 适当平均亮度下:  $c = 10^3$
  - 很低亮度下:  $c = 10$

**同时对比度**：人眼对亮暗程度所形成的“黑”“白”感觉具有相对性，即按对比度 $c$ 感觉物体亮度对比。

**马赫带 (Mach Band) 效应**：马赫带效应的出现，是因为人眼对于图像中不同空间频率具有不同的灵敏度，而在空间频率突变处就出现了“欠调”或“过调”。

**人眼亮度感觉之应用**：目的是为人眼观看效果的图像处理：若一幅原图像经过处理，恢复后得到重现图像，重现图像的亮度不必等于原图像的亮度，只要保证二者的对比度及亮度层次（灰度级）相同，就能给人以真实的感觉。

## 图像数字化

- 采样

- 量化
  - 目前非特殊用途的图像均为8bit量化，即用0-255描述“黑-白”。
  - 在3bit以下的量化，会出现伪轮廓现象

## 灰度直方图

灰度直方图是灰度级的函数，描述的是图像中该灰度级的像素个数。即：横坐标表示**灰度级**，纵坐标表示图像中该灰度级出现的**个数**。

## 图像几何变换

### 空间变换

- 掌握平移、缩放、旋转变换的方法
- 坐标系、齐次坐标
- 了解仿射、剪切、透视

### 灰度插值

- 最近邻
- 双线性

## 图像频域变换

### 频域变换的主要思路和意义

#### 主要思路：

1. **傅里叶变换 (Fourier Transform, FT)**：最常用的频域变换技术，将图像的像素值转换为不同的频率成分。图像中的每个像素点都对应于一个特定的频率和相位，这些信息共同描述了图像的频率内容。
2. **离散傅里叶变换 (Discrete Fourier Transform, DFT)**：由于数字图像是离散的，因此通常使用DFT来处理数字图像。DFT是FT的离散版本，适用于数字计算。
3. **快速傅里叶变换 (Fast Fourier Transform, FFT)**：为了提高DFT的计算效率，FFT算法被开发出来。FFT是一种高效的DFT算法，可以快速计算出图像的频域表示。
4. **频域滤波**：在频域中，可以对图像进行滤波处理，以增强或抑制特定频率的成分。这包括低通滤波器（去除高频噪声）、高通滤波器（增强边缘和细节）、带通滤波器（保留特定频率

范围的成分) 等。

5. **逆傅里叶变换**: 处理完频域数据后, 可以通过逆傅里叶变换 (Inverse Fourier Transform, IFT) 将图像从频域转换回空间域, 以观察处理效果。

**意义:**

1. **特征提取**: 频域变换可以帮助识别图像中的周期性模式和特征, 这对于图像分析和识别非常重要。
2. **噪声降低**: 通过频域滤波, 可以有效地去除图像中的噪声, 尤其是那些在空间域中难以识别的周期性噪声。
3. **图像压缩**: 频域变换是许多图像压缩算法 (如JPEG2000) 的基础, 通过去除人眼不敏感的高频成分来减少图像的数据量。
4. **图像增强**: 通过增强图像的特定频率成分, 可以改善图像的视觉效果, 如增强边缘和细节。
5. **图像分析**: 频域变换可以揭示图像的内在结构, 有助于进行图像分析和理解。
6. **多分辨率分析**: 频域变换与小波变换结合, 可以进行多分辨率分析, 这对于图像的多尺度处理非常有用。
7. **图像复原**: 在图像退化模型已知的情况下, 频域变换可以用来恢复图像的原始状态, 例如去除模糊。
8. **模式识别**: 频域特征对于模式识别和机器学习算法来说是一种重要的特征表示, 可以用于训练分类器。

**傅立叶变换基本表达式、主要性质**

**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**

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

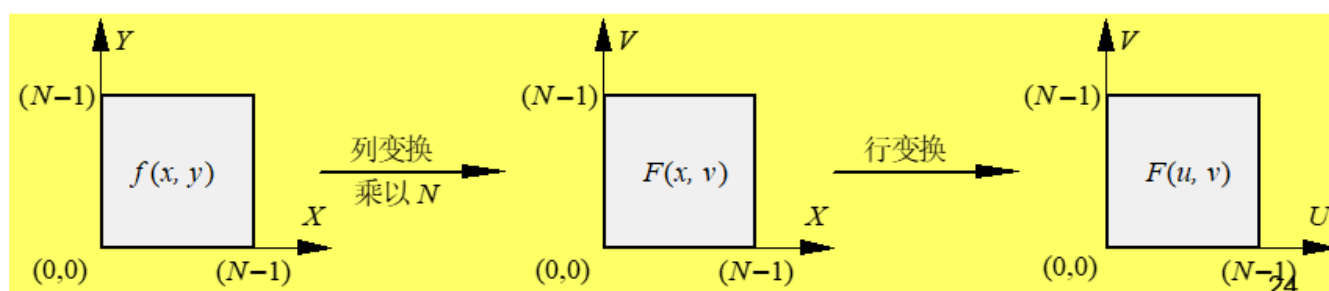
**性质1：可分离性**

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

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

**1次2-D  $\Rightarrow$  2次1-D**

**$O(N^4)$  减为  $O(N^3)$**



## 性质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：

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

## 性质3：周期性和共轭对称性

### 离散的傅里叶变换和它的反变换具有周期为 $N$ 的周期性：

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

### 如果 $f(x, y)$ 是实函数，则它的傅里叶变换具有共轭对称性：

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

## 性质4：旋转性质

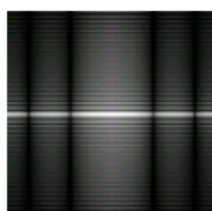
在极坐标系下有： $f(x, y) = f(r, \theta) \Leftrightarrow F(\omega, \varphi)$

如果 $f(x, y)$ 被旋转 $\theta_0$ ，则 $F(u, v)$ 被旋转同一角度：

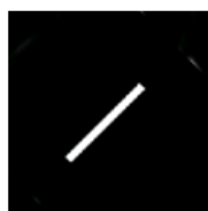
$$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$$



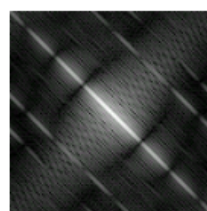
简单图像



傅里叶谱



旋转后图像



旋转后图像的傅里叶谱

## 性质5：分配率

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

## 性质6：尺度缩放

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

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



**性质7:  $F(0,0)$ 与图像均值的关系:**

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

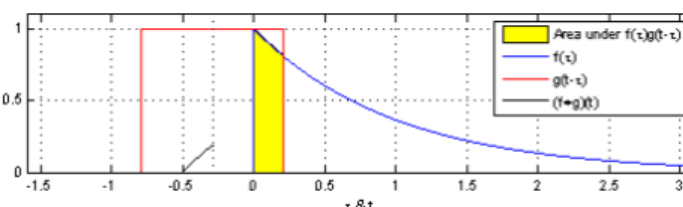
**同时有:**  $F(0,0) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)$

**因此有:**  $\bar{f}(x,y) = F(0,0)$

**即 $F(0,0)$ 等于图像灰度均值。**

**性质8: 卷积与相关定理:**

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(z)g(x-z)dz$$



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

**对于二维有:**  $f(x,y) * g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(p,q)g(x-p,y-q)dpdq$

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

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

31

**了解DCT变换**

- 离散余弦变换是**实值变换**，计算复杂度适中，又具有可分离性，还有快速算法，变换后有很少的非零元素，所以被广泛地用在图像数据压缩编码算法中，如JPEG、MPEG-1、MEPG-2、

和H.261等压缩编码国际标准都采用了离散余弦变换编码算法；

- 其变换核为实数的余弦函数，因而DCT的计算**速度比DFT快得多**。
- DCT矩阵的左上角代表低频分量，右下角代表高频分量

## 图像增强

### 空域增强

- 灰度变换
  - 直接灰度变换
  - 基于直方图的增强（直方图均衡化/规定化）
- 空域滤波
  - 均值滤波、中值滤波、边界保持平滑滤波
  - 线性锐化滤波、非线性锐化滤波
- 频域增强
  - 低通滤波、高通滤波、带通带阻滤波、同态滤波

计算方法，滤波器模板，方法特点，方法之间的区别，**看图会应用**

### 线性锐化滤波

线性锐化滤波主要基于模板卷积进行，其核心思想是通过增强图像中的高频成分来突出边缘和细节。这种滤波方法的输出像素是输入像素及其邻域像素的线性组合。常见的线性锐化滤波方法包括：

1. **拉普拉斯算子**：这是一种二阶微分算子，用于检测图像中的边缘。通过将原图像与拉普拉斯变换后的图像叠加，可以得到锐化后的图像。拉普拉斯算子对噪声敏感，因此更适合无噪声图像。
2. **梯度算子**：包括Roberts算子、Prewitt算子和Sobel算子等。这些算子通过计算图像中像素的梯度（即边缘的斜率）来突出边缘。Sobel算子对邻域内像素位置加权，可以较好地进行边缘检测。

### 非线性锐化滤波

非线性锐化滤波不依赖于像素值的线性组合，而是基于某种非线性映射关系。这种方法在一定程度上能克服线性滤波器的不足，尤其是在处理非高斯噪声或信号频谱与噪声频谱混叠时。常见的非线性锐化滤波方法包括：

1. **中值滤波**：这是一种非线性滤波器，通过将每个像素值替换为其邻域内的中值来实现锐化。中值滤波适用于去除椒盐噪声（黑白点噪声），同时保持边缘信息。
2. **边缘保持滤波**：这类滤波器旨在减少噪声的同时保持边缘信息。它们通常基于非线性映射，将特定的噪声近似地映射为零，而保留信号的重要特征。

## 边界提取及表达

Roberts、Prewitt、Sobel算子的模板、特点

### LoG算子的主要构成

1. **高斯滤波 (Gaussian Filtering)**：LOG算子首先对图像进行高斯滤波，这是一种平滑处理，用于降低图像中的噪声。高斯滤波器的数学表达式为一个高斯函数，它对图像中的像素值进行加权平均，权重由像素距离中心像素的距离决定，距离越远，权重越小。
2. **拉普拉斯算子 (Laplacian Operator)**：在高斯滤波之后，LOG算子接着使用拉普拉斯算子来检测图像中的边缘。拉普拉斯算子是一个二阶导数算子，用于检测图像中灰度值变化最大的地方，即边缘和角点。
3. **零交叉检测 (Zero Crossing)**：LOG算子通过检测滤波结果的零交叉来获得图像或物体的边缘。零交叉是指在拉普拉斯滤波后图像中，灰度值从正变负或从负变正的点，这些点通常对应于边缘的位置。

### Canny算法的主要流程、优点

主要流程包括以下几个步骤：

1. **高斯滤波 (Gaussian Smoothing)**：使用高斯滤波器对图像进行平滑处理，以减少噪声的影响。
2. **梯度计算 (Gradient Calculation)**：计算图像中每个像素点的梯度强度和方向。这通常通过计算图像的一阶导数（梯度幅度）和二阶导数（梯度方向）来实现。
3. **非极大值抑制 (Non-Maximum Suppression)**：在梯度方向上，对梯度幅度进行非极大值抑制，以消除边缘检测中的杂散响应，使得边缘更加清晰。
4. **双阈值检测 (Double-Threshold Detection)**：应用双阈值方法来确定真实的和潜在的边缘。这涉及到设置两个阈值，一个高阈值和一个低阈值，以区分强边缘和弱边缘。

5. **边缘跟踪和滞后 (Edge Tracking and Hysteresis)**：通过滞后技术来跟踪边缘，将与强边缘相连的弱边缘确定为真实边缘，而孤立的弱边缘则被抑制。

## 优点

1. **高准确性**：Canny算法能够准确地检测出图像中的边缘，并且对边缘像素进行细致的定位。
2. **抑制噪声**：通过使用高斯滤波器平滑图像，Canny算法能够有效地抑制噪声干扰，提高边缘检测的准确性。
3. **边缘连接**：Canny算法采用了双阈值的方法，将边缘像素分为强边缘和弱边缘，并通过边缘连接算法将弱边缘与强边缘进行连接，得到连续的边缘线条。
4. **参数可调性**：Canny算法的参数包括高斯滤波器的大小、双阈值的设置等，这些参数可以根据具体应用进行调整，以得到满足需求的边缘检测结果。
5. **鲁棒性**：Canny算法具有较高的鲁棒性，能够处理各种噪声类型的图像。
6. **有效性**：Canny算法能够有效地抑制噪声，同时保持边缘信息，并且能够检测到细小的边缘。

## Hough变换检测直线/圆的思路和方法流程

### Hough变换检测直线的思路和方法流程

#### 思路：

Hough变换是一种特征提取技术，用于在图像中检测简单形状，如直线、圆等。其核心思想是将图像空间中的问题转化为参数空间的问题。对于直线检测，Hough变换通过将图像中的点映射到参数空间（通常是 $\rho$ - $\theta$ 空间），来寻找在参数空间中形成交点的直线。

#### 方法流程：

1. **边缘检测**：首先对输入图像进行边缘检测，得到二值化的边缘图像。
2. **参数空间量化**：在参数空间（ $\rho$ - $\theta$ 空间）中，为每个可能的直线参数（ $\rho$ 和 $\theta$ ）分配一个累加器。
3. **累加器填充**：对于图像中的每个边缘点，计算所有可能的直线参数（ $\rho$ 和 $\theta$ ），并在对应的累加器单元中增加计数。
4. **寻找峰值**：在累加器中寻找高于某个阈值的峰值点，这些峰值点对应于图像中可能的直线参数。
5. **提取直线**：根据累加器中的峰值点，提取对应的直线参数，并将其映射回图像空间，得到直线的方程。

## Hough变换检测圆的思路和方法流程

### 思路：

对于圆检测，Hough变换的思想是将图像中的点映射到参数空间（通常是 $x_0, y_0, r$ 空间），其中 $x_0$ 和 $y_0$ 是圆心坐标， $r$ 是半径。通过在参数空间中寻找高投票值的点来确定圆的存在。

### 方法流程：

1. **边缘检测**：与直线检测类似，首先对图像进行边缘检测，得到二值化的边缘图像。
2. **参数空间量化**：为每个可能的圆参数（ $x_0, y_0, r$ ）分配一个累加器。
3. **累加器填充**：对于图像中的每个边缘点，计算所有可能的圆参数，并在对应的累加器单元中增加计数。
4. **寻找峰值**：在累加器中寻找高于某个阈值的峰值点，这些峰值点对应于图像中可能的圆参数。
5. **提取圆**：根据累加器中的峰值点，提取对应的圆参数，并将其映射回图像空间，得到圆的方程。
6. **极大值抑制**：对临界点进行平均处理，以得到最终的圆心和半径。

## 链码的定义，循环首差链码的计算

## 多边形逼近的两种基本思路

## 阈值/区域分割

## OTSU法思路、算法

## OTSU算法总结：

1. 计算图像的归一化直方图 $p_i, i=0, 1, \dots, L-1$
2. 遍历所有灰度级作为阈值 $T$ ，灰度级处于 $[0, T]$ 的像素分为 $C_0$ , 处于 $[T+1, L-1]$ 的像素分为 $C_1$
3. 依次计算：

$$\begin{aligned}\mu_0 &= \sum_{i=0}^T ip_i / w_0 & \mu_1 &= \sum_{i=T+1}^{L-1} ip_i / w_1 & \sigma_B^2 &= w_0 w_1 (\mu_1 - \mu_0)^2 \\ w_0 &= \sum_{i=0}^T p_i & w_1 &= \sum_{i=T+1}^{L-1} p_i = 1 - w_0\end{aligned}$$

4. 记录下最大类间方差所对应的 $T^*$
5. 如果不唯一，可取多个 $T^*$ 的平均值
6. 用 $T^*$ 作为最优阈值进行分割

### 一维熵阈值法的概念、算法

所谓灰度图像的一维熵最大，就是选择一个阈值，使图像用这个阈值分割出的两部分的一阶灰度统计的信息量最大

#### 一维熵阈值法的概念

一维熵阈值法是一种基于信息熵理论的图像分割方法。其核心思想是选择一个阈值，使得前景和背景的熵之和最大，即图像的总熵最大。当每一类像素中的像素在此类中趋向均匀分布时，此时这部分熵最大，当前景和背景两类像素都趋向均匀分布时，前景的熵和背景的熵最大，由系统熵的累加性，此时整幅图像的熵最大，即熵最大时将前景和背景区分开来。

#### 一维熵阈值法的算法

一维熵阈值法的算法步骤如下：

1. **计算灰度直方图**：首先计算图像的灰度直方图，得到每个灰度级的概率分布。
2. **归一化灰度直方图**：将灰度直方图归一化，得到概率直方图。
3. **计算累加直方图**：计算归一化灰度直方图的累加直方图。



4. **计算各个灰度级的熵**：对于每个灰度级，计算其熵值。
5. **寻找最大熵阈值**：遍历所有可能的阈值，对于每个阈值，计算前景和背景的熵，然后计算总熵。选择使得总熵最大的阈值作为分割阈值。
6. **阈值处理**：使用找到的阈值对图像进行二值化处理。

## 区域生长的基本思路与方法

### ◆ 像素集合的区域生长

1. 根据图像的不同应用选择一个或一组种子，它或者是最亮或最暗的点，或者是位于点簇中心的点；
2. 选择一个描述符（生长条件）；
3. 从该种子开始向外扩张，首先把种子像素加入结果集合，然后不断将与集合中各个像素连通（4或8邻域）、且满足描述符的像素加入集合；
4. 上一过程进行到不再有满足条件的新结点加入集合为止（停止条件）。

## 区域分裂合并的基本思路与方法

分裂-合并的区域分割方法，利用**图像数据塔式的层次结构**，将图像划分成不相交的区域，以某一检测准则从四叉树数据结构的任一层开始，对区域进行分裂或合并。并逐步改善区域划分的性能，直到最后将图像分成数量最少的均匀区域为止。

区域分裂合并是一种图像分割算法，它与区域生长法相似，**但不需要预先指定种子点**，而是根据一致性准则来分裂或合并区域。这种方法特别适用于复杂场景图像的分割。其基本思路是：

1. **初始化**：将整个图像作为一个初始区域。
2. **分裂**：按照某种一致性准则（如灰度、纹理等），将区域不断分裂成更小的子区域，直到满足分裂条件的区域不再存在。
3. **合并**：检查相邻的子区域，如果它们满足合并条件（如灰度相似、纹理一致等），则将它们合并成一个较大的区域。
4. **迭代**：重复分裂和合并的过程，直到无法进一步分裂或合并为止。

## 图像形态学

- 集合论相关概念
- 膨胀、腐蚀、开、闭运算的计算
- 学会选择应用

## 彩色图像分割

### 分水岭算法的原理、特点

原理：

- 任何的灰度级图像都可以被看做是一个地形图
- 假设我们在每个区域最小值位置地方打个洞,让水以均匀的速度上升,从低到高淹没整个地形.当处在不同的汇聚盆地中的水将要聚合在一起时,修建大坝将阻止聚合,最后得到的水坝边界就是分水岭的分割线.

特点：

- 优点
  - 速度较快
  - 在层次分割方法中效果好
  - 能分割出封闭的区域轮廓
- 缺点
  - 容易“过分割”
  - 没有自顶向下的信息

### MeanShift思路、方法、特点

#### MeanShift算法的方法

1. **初始化**：选择一个初始窗口，并计算窗口内样本点的核密度估计。
2. **迭代移动**：将窗口中心移动到核密度估计的梯度方向，即向密度增加最快的方向移动。
3. **更新窗口**：更新窗口位置，重复步骤2，直到窗口中心不再变化或变化小于某个阈值，此时认为窗口已经收敛到一个局部密度极大值点。



4. **模式检测和分割完成**：当所有窗口都完成收敛后，位置相近的点会聚集成簇，这些簇表示图像中不同的区域或对象，每个簇的中心即为局部密度的峰值，从而实现图像的分割。

### MeanShift算法的特点

#### 优点：

1. **无需预设簇数量**：MeanShift算法能够自适应地识别数据中的密度峰值，无需预设簇的数量，适合处理复杂或不规则的图像区域。
2. **对噪声和复杂纹理具有鲁棒性**：通过利用局部密度信息，MeanShift对包含噪声或复杂纹理的图像具有良好的鲁棒性，适合多种分割任务。
3. **考虑颜色和空间信息**：在高维特征空间（如颜色+位置）下的应用使其能够在图像分割中同时考虑颜色和空间信息，提高分割质量。
4. **MS的结果通常比较稳定**（受初始值的影响不是很大）

#### 缺点：

1. **计算复杂度高**：尤其在处理大规模图像时，由于每个像素点都需与其邻域进行密度估计，计算成本较高。
2. **对带宽参数敏感**：分割结果依赖于核带宽大小的选择，带宽过大或过小都会影响分割效果。因此，带宽的合理设定对分割精度至关重要。
3. **对维度增加敏感**：在高维特征空间应用时，MeanShift对维度增加较为敏感，适应性和效率可能受到影响。

### 归一化图割的基本思想，准则，快速计算理论，优缺点

#### 基本思想：

归一化图割是一种基于图论的图像分割方法。其基本思想是将图像映射成一个加权图，其中图像的像素点作为图的顶点，邻接像素之间的关系作为图的边，边的权重基于像素间的相似性。通过最小化一个能量函数来实现图像的分割，这个能量函数通常包括数据项和平滑项，分别对应于像素点与其邻接点的相似度和差异度

准则：归一化割的目标是找到一个割，使得割的边权重之和最小

$$\text{minimize} \quad Ncut(A,B) = \frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)}$$

#### 特点：

1. 能够有效集成图像像素颜色、纹理等特征信息及空间位置信息

- 2.具有良好的平衡划分性能，分割性能较好
- 3.分割类别个数事先指定，利于场景划分
- 4.权值矩阵维数为图像中像素的个数，特征系统维数太高，无法处理大尺度的图像
- 5.计算复杂性高，不利于实时应用

快速计算理论：

$$\text{minimize} \quad Ncut(A,B) = \frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)}$$

is given by the following eigen-system

$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}Z = \lambda Z$$

$$D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = L_{sym}$$

对称归一化拉普拉斯

$$W_{ij} = W(x_i, x_j)$$

$$D_{ii} = \sum_j W_{ij}$$

- 最小化 归一化图割 准则，看成是图的谱聚类问题
- 转化成对上述标准形式求特征值的问题：最小特征值是0
- 用第二小的特征值（最小的非零特征值）及特征向量，对图进行划分（分成两个子图）
- 判断是否还要继续细分下去（用递归的方式）

28

#### 算法步骤

1. **构建图**：将图像像素视为图的节点，根据像素间的相似性构建边的权重矩阵W。
2. **计算度量矩阵**：计算度量矩阵D，其对角线元素是每个节点的度（即与该节点相连的边的权重之和）。
3. **求解特征值问题**：求解上述特征值问题，找到最小的特征值及其对应的特征向量。
4. **图像分割**：使用第二小的特征值（最小的非零特征值）及其特征向量对图像进行分割，将图像划分为两个子图。
5. **递归分割**：判断是否需要继续细分。如果需要，对每个子图重复上述步骤，直到满足停止条件

## 基于图划分的阈值法基本原理

计算高维权值矩阵耗时  
权值矩阵太大无法存储

影响阈值方法的效率和实现

W矩阵基于图像灰度级构建，而不是像素数量，以减少维度，大大减小计算时间

阈值与图割思路结合：

对每一个设定阈值 $T(0 \leq T \leq 255)$ 计算 $Ncut(A, B)$

最小的 $Ncut$ 对应的阈值 $T$ 为最佳阈值

$Ncuts$ 割的是像素和像素  
(找像素之间的边界)；  
阈值+GC割的是灰度  
(找的是灰度阈值)！

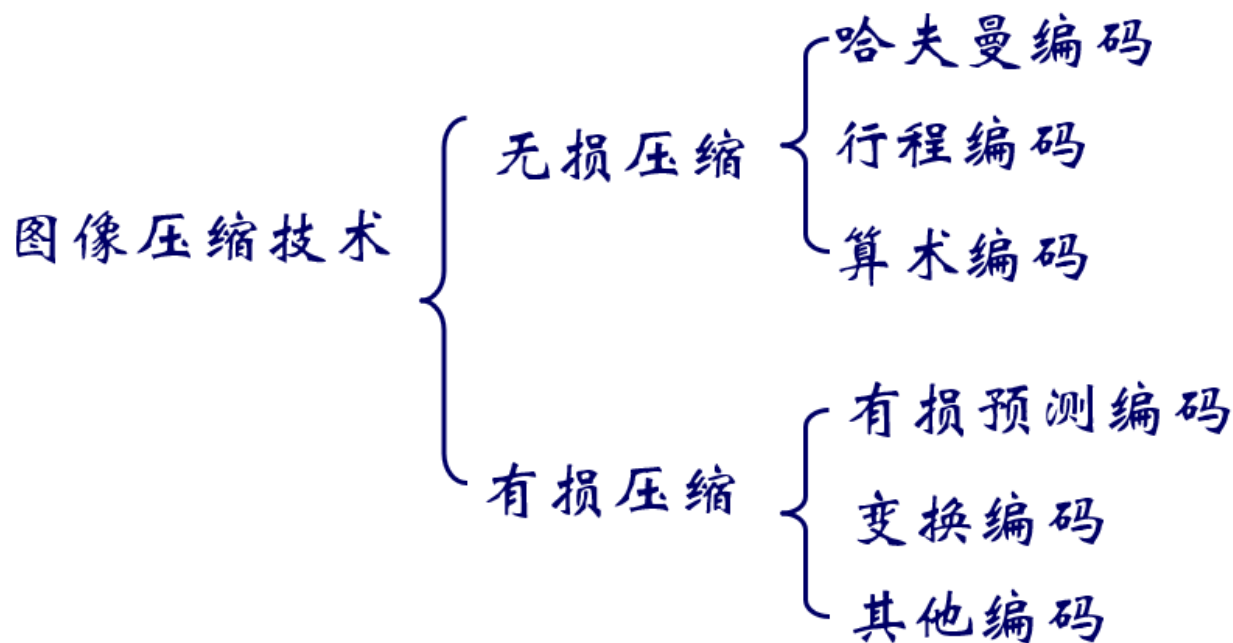
通过归一化图割作为一种准则，对选阈值的结果，从图的谱聚类角度去评价

40

## 图像压缩编码

### 基本概念

- 图像冗余
  - 编码冗余
  - 像素冗余
  - 心理视觉冗余
- 方法分类



- 保真度

- 客观保真度

**保真度准则：**

总体误差为：

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]$$

均方根误差为：

$$e_{rms} = \left[ \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \right]^{1/2}$$

均方信噪比：

$$SNR_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2}$$

- 主观保真度

- 压缩比

一般定义为原始图像每个像素的平均比特数 $n_1$ 和编码后每个像素的平均比特数 $n_2$ 的比值。

$$C_R = \frac{n_1}{n_2}$$

$$\text{数据冗余 } R = 1 - 1/C$$

### 熵编码&变长编码

图像的熵  $H$  是表示其各个灰度级比特数的统计平均值——即平均码字

熵编码是一种无损数据压缩技术，它基于信息论中的熵概念来减少表示数据所需的位数。熵编码的目标是为出现概率高的符号分配较短的编码，而为出现概率低的符号分配较长的编码，从而实现整体数据的压缩。

例如：

输入数据	$W_1$	$W_2$	$W_3$	$W_4$
概率	1/2	1/4	1/8	1/8

图像的熵：
$$H = -\sum_{k=1}^4 p_k \log_2 p_k = 1\frac{3}{4} \text{ (bit)}$$

采用等长编码：

$W_1$	$W_2$	$W_3$	$W_4$
00	01	10	11

平均码长  $R=2(\text{bit})$ ，编码效率  $\eta=87.5\%$ ， $R_d=12.5\%$

采用不等长编码：

$W_1$	$W_2$	$W_3$	$W_4$
0	10	110	111

平均码长  $R=7/4(\text{bit})$ ，编码效率  $\eta=100\%$ ， $R_d=0\%$

## 2. 平均码字长度 Average Codeword Length

设  $N_k$  为数字图像第  $k$  个码字  $c_k$  的长度(二进制数的位数), 其对应出现的概率为  $p_k$ , 则该数字图像的码字平均长度定义为:

$$\bar{N} = \sum_{k=1}^M N_k p_k \quad (\text{bit})$$

## 3. 编码效率 Coding Efficiency

定义数字图像编码的效率为:  $\eta = \frac{H}{\bar{N}} * 100\%$

## 4. 冗余度 Redundancy

定义图像编码的冗余度为:  $R_d = 1 - \eta$

33

如图计算熵可知道平均码字长度, 也就是**压缩的潜力**

在  $\bar{N} \geq H$  情况下总可以设计出某种无失真编码方法, 若  $\bar{N}$  接近于  $H$ , 则说明码编的较好, 称为最佳编码。若要求编码结果  $\bar{N} < H$ , 则必然要丢失信息而引起图像失真。

$$\eta = \frac{H}{\bar{N}} * 100\%$$

Huffman编码计算方法

**Huffman** 编码的原则是概率大的信息用短码，而概率小的信息用长码，即：

若： $p_1(w_1) > p_2(w_2) > \dots \dots > p_M(w_M)$

则取： $N_1(c_1) < N_2(c_2) < \dots \dots < N_M(c_M)$

**Huffman** 编码的编码步骤：

- ①将信源符号（图像的灰度等级）按概率由大到小排列，概率相同的可以任意放
- ②将两个最小概率相加，形成新的概率集合，并按①的原则重新排队
- ③重复②的过程，直到只剩下两个概率为止
- ④分配码字进行编码，原则是从后到前，左0右1(或左1右0)

#### Huffman编码特点

- Huffman编码**构造程序是明确的，但编出的码不是唯一的**，其原因之一是两个概率分配码字“0”和“1”是任意选择的（大概率为“0”，小概率为“1”，或者反之）。第二原因是在排序过程中两个概率相等，谁前谁后也是随机的。这样编出的码字就不是唯一的。
- Huffman编码结果，码字不等长，**平均码字最短，效率最高**，但码字长短不一，实时硬件实现很复杂（特别是译码），而且在抗误码能力方面也比较差。
- Huffman编码的信源概率是2的负幂时，效率达100%，但是对等概率分布的信源，产生定长码，效率最低，因此编码效率与信源符号概率分布相关，故Huffman**编码依赖于信源统计特性**，编码前必须有信源这方面的先验知识，这往往限制了哈夫曼编码的应用。
- Huffman编码只能用近似的整数位来表示单个符号，**而不是理想的小数**，这也是Huffman编码**无法达到最理想的压缩效果**的原因。

#### 算术编码的问题

- 需要增加消息结束指示符号（知道消息有多长）
- 计算机表示小数的精度是有限的



- 缩放策略和舍入策略
  - 缩放：将子区间重新归一化到[0,1)
  - 舍入：根据有限精度截断
- 解码端必须收到所有数位后才能解码（不能边收边解）
- 对位错误敏感

## DCT

- 有损，压缩率较高
- 忽略视觉不敏感的部分
- 采用DCT变换，得到高频和低频分量，采用不同的方法进行表述

### 正变换：

$$F_c(\mu, \nu) = \frac{2}{\sqrt{MN}} c(\mu) c(\nu) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{\pi}{2N}(2x+1)\mu\right] \cos\left[\frac{\pi}{2M}(2y+1)\nu\right]$$

### 逆变换：

$$f(x, y) = \frac{2}{\sqrt{MN}} \sum_{\mu=0}^{M-1} \sum_{\nu=0}^{N-1} c(\mu) c(\nu) F_c(\mu, \nu) \cos\left[\frac{\pi}{2N}(2x+1)\mu\right] \cos\left[\frac{\pi}{2M}(2y+1)\nu\right]$$

其中：

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

## 图像特征提取

### SIFT