

# 第3章 图像分析— 3.1 图像分割

信控学院 蔡利梅

在对图像的研究和应用中，人们往往仅对图像中的某些目标感兴趣，这些目标通常对应图像中具有特定性质的区域。

### **图像分割(Image Segmentation)**

指把一幅图像分成不同的具有特定性质区域的图像处理技术，将这些区域分离提取出来，以便进一步提取特征和理解。

### **如何实现图像分割？**

基于区域相似性，从内部出发，找到具有区域共性的所有像素。

基于边界不连续性，找到区域的边界从而确定不同的区域。

### 3.1.1传统分割方法

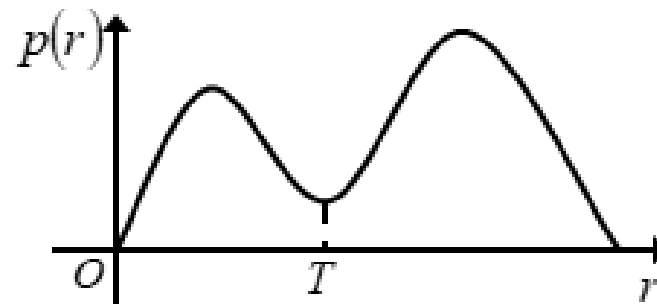
#### ■ 阈值分割

$$g(x, y) = \begin{cases} 1 & f(x, y) \geq T \\ 0 & f(x, y) < T \end{cases}$$

阈值T的选取决定了二值化效果的好坏。

#### □ 基于灰度直方图的阈值选择

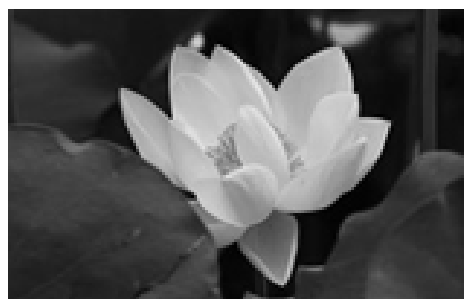
若图像的灰度直方图为**双峰分布**，表明图像的内容大致为两部分，分别为灰度分布的两个山峰附近。**选择阈值为两峰间的谷底对应灰度值**



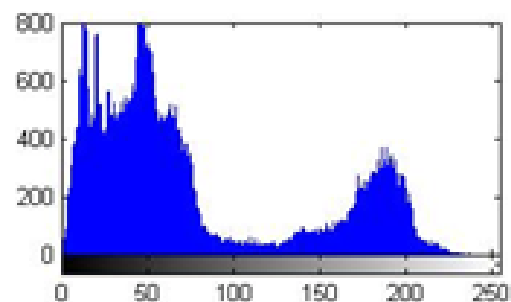
**适用于**图像中前景与背景灰度差别明显，且各占一定比例的情形，是一种**特殊**的方法。若图像整体直方图不具有双峰或多峰特性，可以考虑局部范围内应用。

## □ 基于灰度直方图的阈值选择

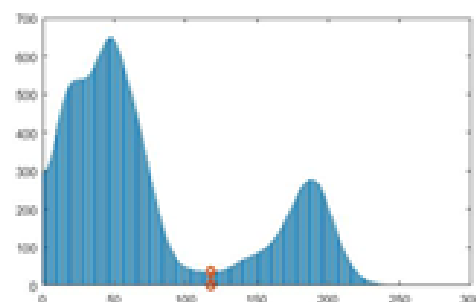
原图



分割图  
 $T=118$



灰度直方图



平滑后的直  
方图和波谷

## □ 最优阈值方法

确定某个**准则函数**，当准则函数**取最优**时对应的阈值为最佳阈值

**最大类间方差法**  
**(OTSU)**

类内方差

$$\sigma_m^2 = p_o \cdot \sigma_o^2 + p_B \cdot \sigma_B^2$$

类间方差

$$\sigma_b^2 = p_o \times (\mu_o - \mu)^2 + p_B \times (\mu_B - \mu)^2$$

$$\sigma_o^2 = \frac{1}{p_o} \sum_{i=0}^T p_i (i - \mu_o)^2$$

$$\sigma_B^2 = \frac{1}{p_B} \sum_{i=T+1}^{L-1} p_i (i - \mu_B)^2$$

$$p_i = \frac{n_i}{M \times N}, i = 0, 1, 2, \dots, L-1$$

$$\mu_o = \frac{1}{p_o} \sum_{i=0}^T i \times p_i$$

$$\mu_B = \frac{1}{p_B} \sum_{i=T+1}^{L-1} i \times p_i$$

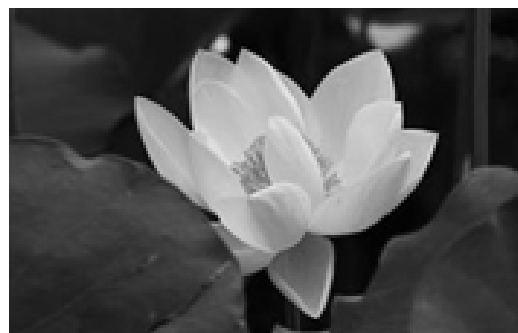
$$p_o = \sum_{i=0}^T p_i \quad p_B = \sum_{i=T+1}^{L-1} p_i$$

$$\mu = p_o \times \mu_o + p_B \times \mu_B$$

使得**类内方差最小**或**类间方差最大**、或者**类内和类间方差比值最小**的阈值  $T$  为最佳阈值。

## 最大类间方差法 (OTSU)

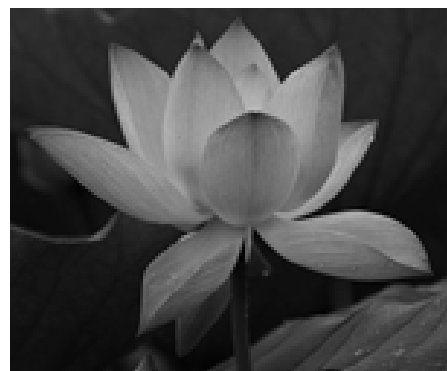
原图



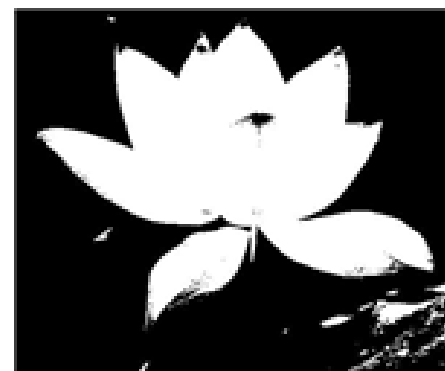
$T=109$



原图



$T=80$



## MR图像的脑图像分割算法

将图像分为局部子区域，每个子区域的灰度分布用高斯分布模拟，整个区域的灰度分布为高斯分布的和，准则函数为真实分布和模拟分布的平方误差和，最小化该函数确定高斯分布的最优参数

$$h_{model}(r) = \sum_{i=1}^n \alpha_i e^{-(r-\mu_i)^2/2\sigma_i^2}$$

$$J = \sum_r [h_{model}(r) - h_{region}(r)]^2$$

- Frank, R J, Grabowski, T J, Damasio, H. Voxelve percentage tissue segmentation of human brain magnetic resonance images(abstract),1995
- D.W. Marquardt. An algorithm for least-squares estimation of non-linear parameters. Journal of the Society for Industrial and Applied Mathematics, 11:431-444,1963.

## ■ 边界分割

通过检测区域的边界轮廓来实现图像分割的方法

### □ 分割过程

**边界检测：**通过各种边缘检测算子从图像中抽取边缘线段

**边界改良：**执行各种改良边缘的处理：如边界细化、边缘闭合等

**边界跟踪：**跟踪边缘形成边界曲线

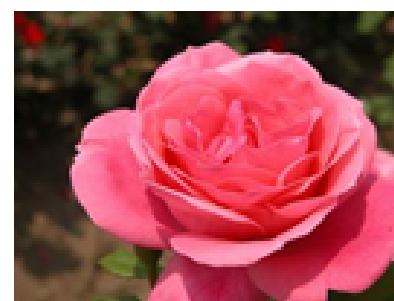


## □ 边界检测

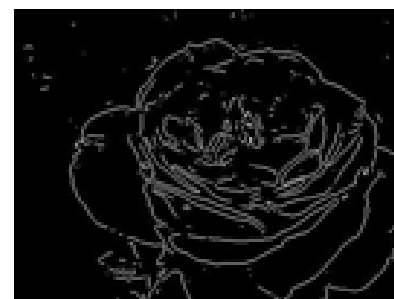
通过各种边缘检测算子从图像中抽取边缘线段

**边缘检测算子：**Roberts、Sobel、Prewitt、LOG、Canny、拉普拉斯等微分算子

微分算子检测图像中灰度变化的位置，可能位于目标和背景之间，也可能是目标内部、背景内部，并没有获取完整的区域边界



原图



边界图像

## □ 边界改良

目标的部分边界与相邻部分背景相近或相同时，提取出的目标区域**边界线会出现断点、不连续或分段连续**等情况；有噪声干扰时，也会使**轮廓线断开**。

要提取目标区域时，应使不连续边界闭合。

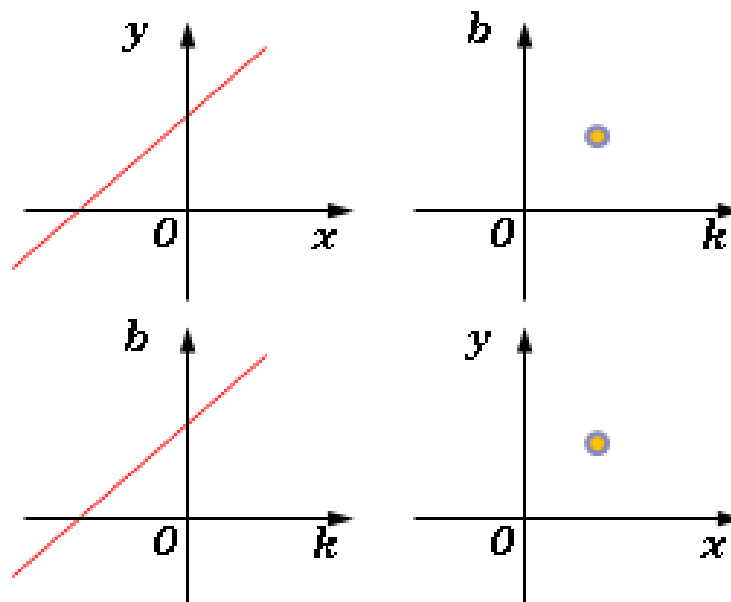
方法多种多样：Hough变换、基于梯度的边界闭合技术、数学形态学等

## Hough变换

检测图像中直线和曲线的方法

以截距式方程为例： $y = kx + b$

$xy$ 空间一条确定的直线对应一组确定的数据 $k$ 、 $b$



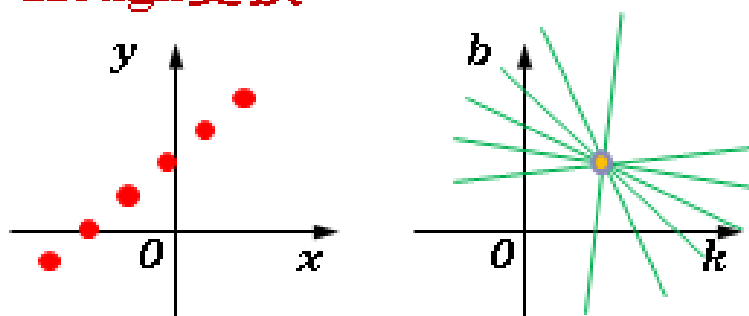
$xy$ 空间一条确定的直线对应 $kb$ 参数空间一个点

$kb$ 空间一条确定的直线对应一组确定的数据 $x$ 、 $y$

$kb$ 空间一条确定的直线对应 $xy$ 空间一个点

直线变形为关于 $k$ 和 $b$ 的直线： $b = -xk + y$

## Hough变换

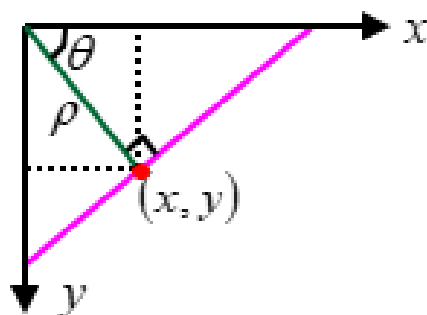


综上所述， $xy$ 空间一条直线上的 $n$ 个点，对应参数 $kb$ 空间经过一个公共点的 $n$ 条直线

**结论：**对于原图中直线上的每一点，在参数空间确定一条直线，直线的交点对应的 $kb$ 即为原图中直线的参数

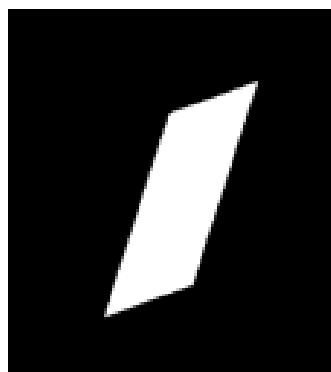
**交点的检测：**在参数空间确定直线时，直线所经过点的值累加1，累加值最大的点为交点

极坐标形式

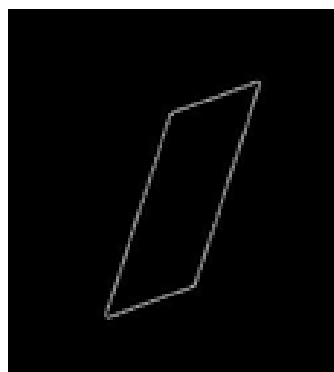


$$\begin{cases} x = \rho \cos \theta \\ y = \rho \sin \theta \end{cases}$$

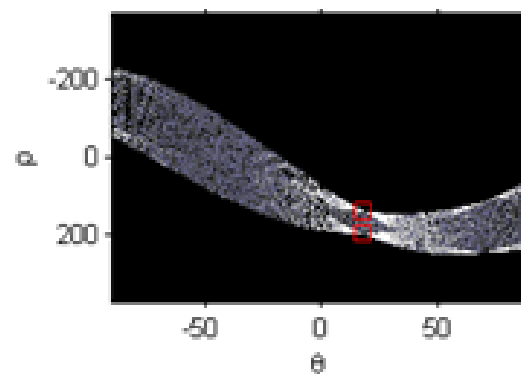
$$\rho = x \cos \theta + y \sin \theta$$



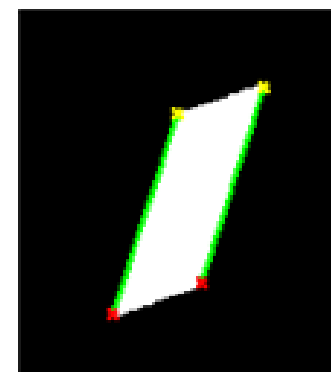
原始图像



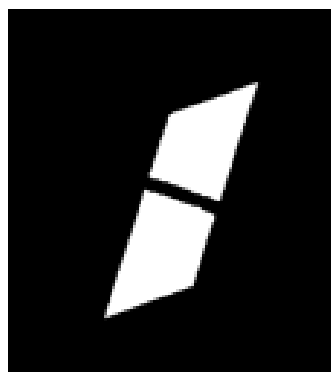
边界图像



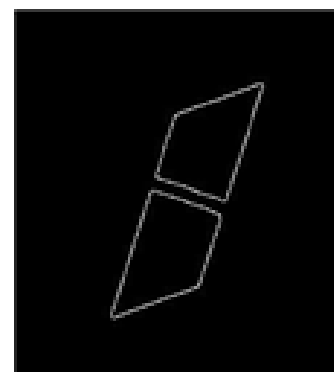
$\rho\theta$ 参数空间



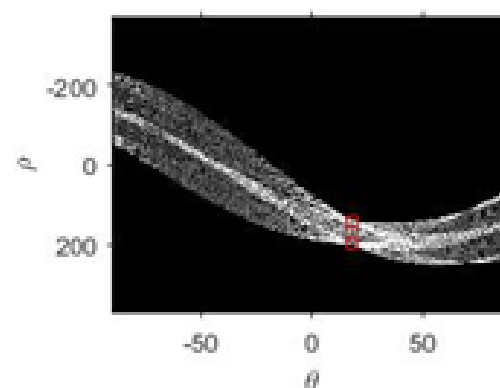
检测结果



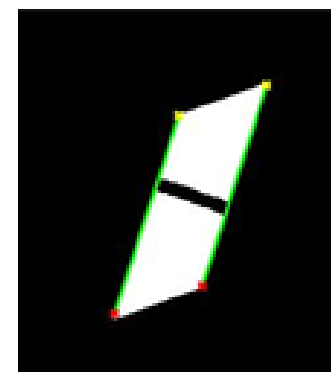
原始图像



边界图像



$\rho\theta$ 参数空间



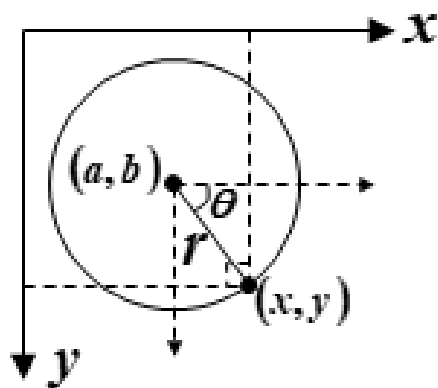
检测结果

## Hough变换

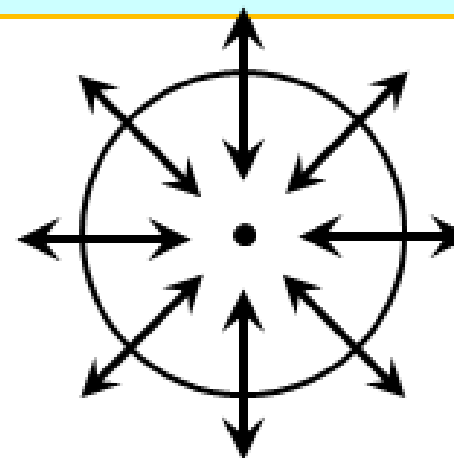
圆方程  $(x-a)^2 + (y-b)^2 = r^2$

- $xy$ 空间一个圆对应三维参数空间一个点  $(a, b, r)$
- $xy$ 空间圆上一个点  $(x, y)$  对应参数空间一条曲线
- $xy$ 空间圆上  $n$  个点对应参数空间  $n$  条相交于一点的曲线

对于原图中每一点，在参数空间确定一条曲线，经过曲线最多的点为原图中圆的参数



$$\begin{cases} x = a + r \cos \theta \\ y = b + r \sin \theta \end{cases}$$



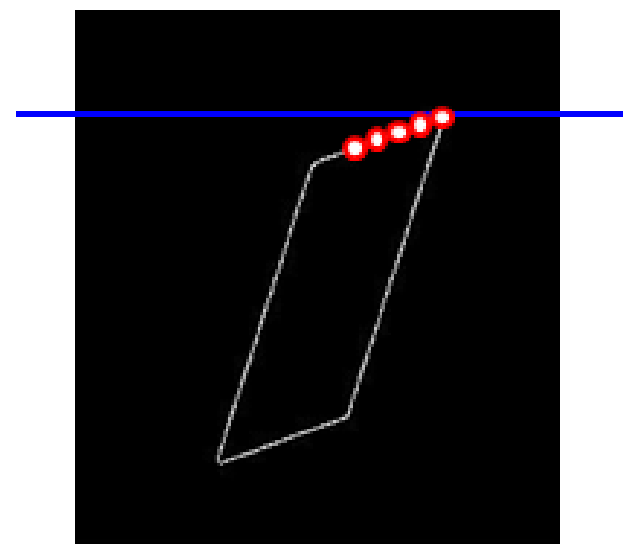
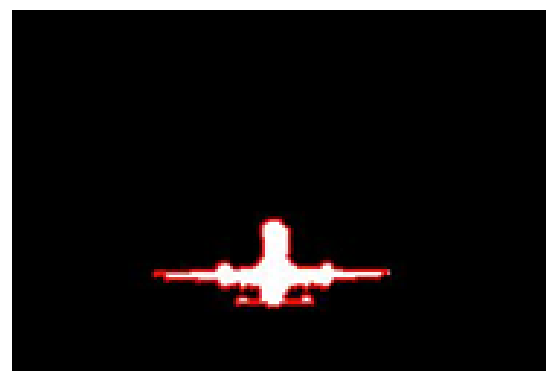
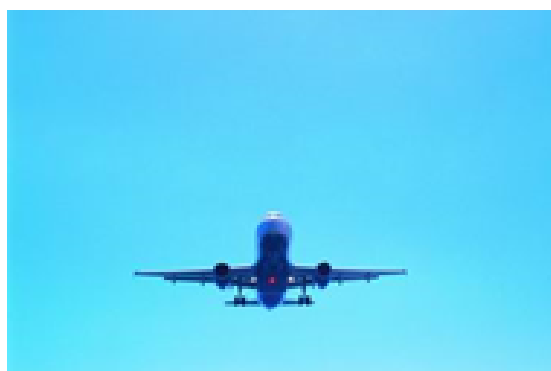
## Hough变换

Hough变换可以推广到具有解析形式  $f(x,a)=0$  的任意曲线， $x$ 表示图像点， $a$ 表示参数向量

- 初始化参数空间  $A[a]$
- 对每个边缘像素  $x$  确定  $a$ ，使得  $f(x,a)=0$ ，并令  $A[a] += 1$
- $A$  的局部最大值对应图像中曲线参数

## □ 边界跟踪

- ◆ 根据某些严格的“**探测准则**”找出目标物体轮廓上的像素，即确定边界的起始搜索点；
- ◆ 再根据一定的“**跟踪准则**”找出目标物体上的其他像素，直到符合跟踪终止条件。
- ◆ **由二维图像变为一维的点序列**





## ■ 区域分割

一般认为，**同一个区域内的像素点具有某种相似性**，如灰度、颜色、纹理等，区域分割即是**根据特定区域与其他背景区域特性上的不同进行图像分割**的技术。

关键技术：

- 明确区域特性，确定相似性准则，与具体问题有关
- 区域分割算法，快速、准确找到区域内像素

## □ 区域生长

- ◆ 确定每个区域区别于其他区域的特征，生成相似性判据；
- ◆ 在每个要分割的区域内确定一个**种子点**，在种子像素周围邻域内，判断**是否有与种子像素相似的像素**，若有，将新的像素包含在区域内，并作为新的种子继续生长，直到没有满足条件的像素点时**停止生长**。

种子点为：(2,2)

相似性准则：

灰度值差小于2

邻域选择：4邻域

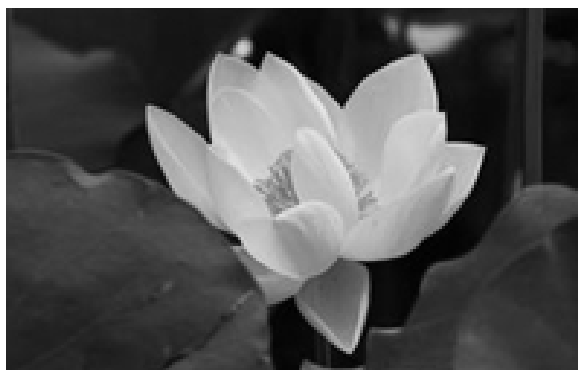
|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 0 | 4 | 6 | 5 | 1 |
| 1 | 0 | 4 | 6 | 6 | 2 |
| 0 | 1 | 5 | 5 | 5 | 1 |
| 0 | 0 | 5 | 6 | 5 | 0 |
| 0 | 0 | 1 | 6 | 0 | 1 |
| 1 | 0 | 1 | 2 | 1 | 1 |



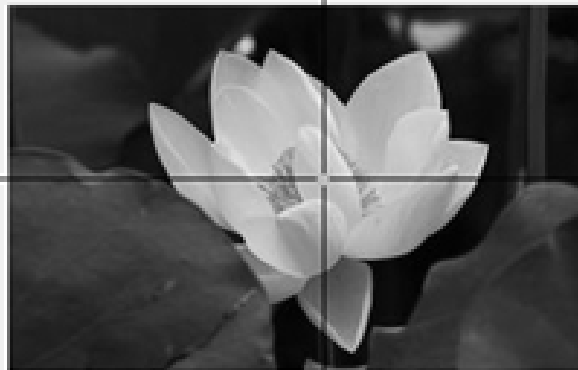
|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 1 | 2 | 2 | 2 | 1 |
| 1 | 1 | 2 | 2 | 2 | 1 |
| 1 | 1 | 2 | 2 | 2 | 1 |
| 1 | 1 | 2 | 2 | 2 | 1 |
| 1 | 1 | 1 | 2 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

## □ 区域生长

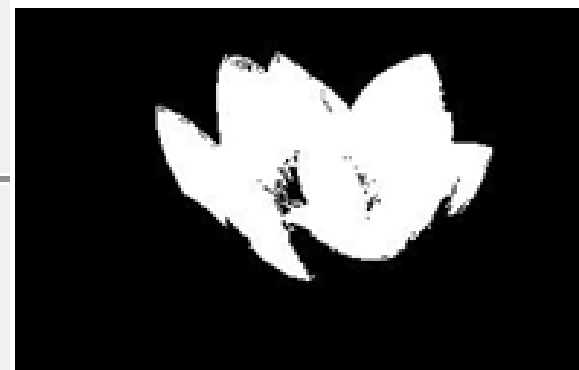
对图像进行区域生长。交互式选取种子，生长准则采用“待测像素点与区域的平均灰度差小于40”，8邻域范围生长，停止生长条件为区域饱和。



原图



交互式选取种子



区域生长

## □ 区域合并

## ◆ 算法实现

若图像已经分为若干个小区域，合并具有相似性的相邻区域；

- ✓ 图像的初始区域分割
- ✓ 确定相似性准则
- ✓ 判断图像中的相邻区域是否满足相似性准则，相似则合并，重复这一步骤，直到没有区域可以合并为止。

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 0 | 4 | 6 | 5 | 1 |
| 1 | 0 | 4 | 6 | 6 | 2 |
| 0 | 1 | 5 | 5 | 5 | 1 |
| 0 | 0 | 5 | 6 | 5 | 0 |
| 0 | 0 | 1 | 6 | 0 | 1 |
| 1 | 0 | 1 | 2 | 1 | 1 |



|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 0 | 4 | 6 | 5 | 1 |
| 1 | 0 | 4 | 6 | 6 | 2 |
| 0 | 1 | 5 | 5 | 5 | 1 |
| 0 | 0 | 5 | 6 | 5 | 0 |
| 0 | 0 | 1 | 6 | 0 | 1 |
| 1 | 0 | 1 | 2 | 1 | 1 |

## □ 区域合并 ◆ 实例 通过区域合并将图像分割为两个区域

✓ 初始化：每个像素为一个小区域；相似性准则：相邻区域灰度均值差 $\leq T$ ；左上角第一个点设为区域1，其余为0，表示未标记。

✓ 第一次扫描图像：从左到右，从上到下，判断每一点与其左上、上、左邻点的灰度距离，三个距离中最小的若符合合并规则，将对应邻点的标记赋予当前点；不符合规则的点，则赋予当前点新的标记。

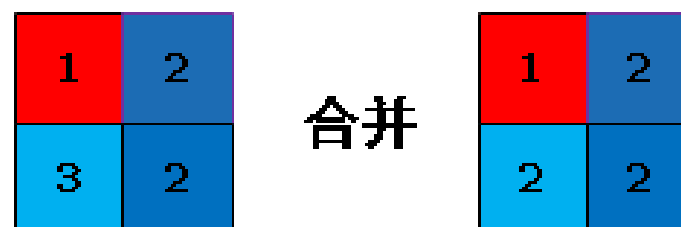
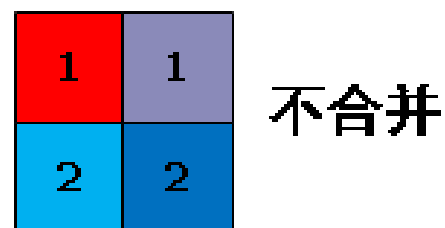
|   |   |
|---|---|
| 1 | 1 |
| 2 | 2 |

|   |   |
|---|---|
| 1 | 1 |
| 2 | 3 |

## □ 区域合并

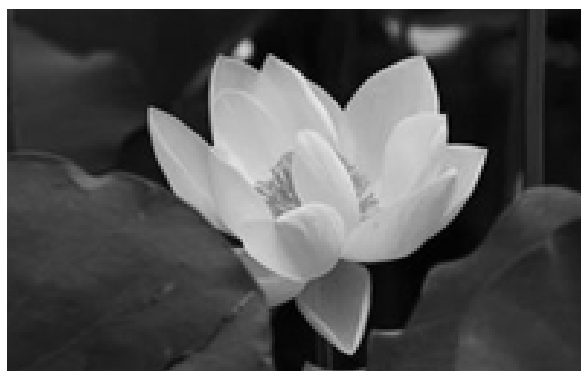
### ◆ 实例

- ✓ 再次扫描图像：若某一像素点上、左邻点标记不一致，但当前点和其中一个邻点标记一致，则判断两个区域是否是同一个，若是，则将两个区域标记修改为较小的一个，即再次区域合并。



## □ 区域合并

### ◆ 实例



原图



合并阈值80



合并阈值100

一种自下而上的方法，某些区域一旦合并，即使与后来的区域相似性并不好，也无法去除。

## □ 区域分裂

### ◆ 原理

检验整个区域是否具有**一致性**，不具有时，分裂为几个小区域；再检测小区域的一致性，不具有时进一步分裂；重复这个过程直到每个区域都具有**一致性**。

### ◆ 关键技术

- ✓ **一致性准则**，根据图像的具体情况、分割的依据确定；
- ✓ **分裂方法**，如何分裂区域为小区域，应尽可能使分裂后的子区域都具有**一致性**。通常采用把区域分割成固定数量、小区域大小相等的方法，如一分为4



## □ 区域分裂

### ◆ 示例

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 2 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 6 | 7 | 1 | 0 | 0 | 1 |
| 1 | 6 | 7 | 5 | 6 | 7 | 1 | 1 |
| 0 | 7 | 6 | 6 | 6 | 0 | 1 | 1 |
| 0 | 7 | 6 | 5 | 7 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

采用区域分裂实现图像分割

✓ 初始化及准则、方法的确定

**初始区域分割：**认为整幅图像为一个区域

**准则的确定：**区域内最大与最小灰度值之差 $\leq 2$

**分裂方法的确定：**一分为四

## □ 区域分裂

### ◆ 示例

✓ 开始分割

区域参数计算:

$\max=7, \min=0$

判断是否分裂:

$\max-\min=7>2$ , 分裂

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| ① |   |   |   | ② |   |   |   |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 2 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 6 | 7 | 1 | 0 | 0 | 1 |
| 1 | 6 | 7 | 5 | 6 | 7 | 1 | 1 |
| 0 | 7 | 6 | 6 | 6 | 0 | 1 | 1 |
| 0 | 7 | 6 | 5 | 7 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| ③ |   |   |   | ④ |   |   |   |

## □ 区域分裂

### ◆ 示例

✓ 对每个小区域重复操作

$$\max_1 - \min_1 = 7 > 2$$

$$\max_2 - \min_2 = 7 > 2$$

$$\max_3 - \min_3 = 7 > 2$$

$$\max_4 - \min_4 = 7 > 2$$

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 2 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 6 | 7 | 1 | 0 | 0 | 1 |
| 1 | 6 | 7 | 5 | 6 | 7 | 1 | 1 |
| 0 | 7 | 6 | 6 | 6 | 0 | 1 | 1 |
| 0 | 7 | 6 | 5 | 7 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

## □ 区域分裂

### ◆ 示例

✓ 依此类推，直至所有的区域都不能再分裂

分析：

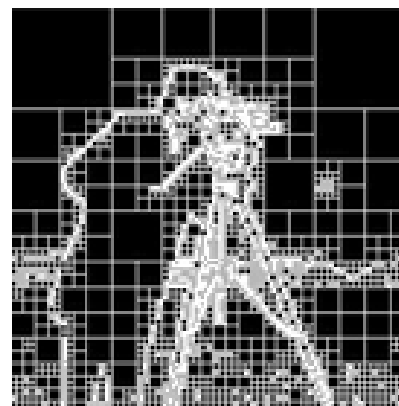
- ✓ 图像为方图，宽高为2的幂
- ✓ 分成的块太多

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 2 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 6 | 7 | 1 | 0 | 0 | 1 |
| 1 | 6 | 7 | 5 | 6 | 7 | 1 | 1 |
| 0 | 7 | 6 | 6 | 6 | 0 | 1 | 1 |
| 0 | 7 | 6 | 5 | 7 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

## □ 区域分裂

### ◆ 实例

原始图像



四叉树分解

分析：

一种自上而下的方法，某些区域一旦分裂，即使其中的部分小区域具有相似性，也只能被分割在不同的区域。

## □ 区域分裂合并

### ◆ 原理

结合合并、分裂方法，将原图分成若干个子块，检测子块是否具有**一致性**，不具有则分裂该子块；若某些子块具有**相似性**，则合并这些子块

### ◆ 步骤

分裂合并同时进行，或先分裂后合并的方法

## □ 区域分裂合并

### ◆ 示例

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 6 & 7 & 1 & 0 & 0 & 1 \\ 1 & 6 & 7 & 5 & 6 & 7 & 1 & 1 \\ 0 & 7 & 6 & 6 & 6 & 0 & 1 & 1 \\ 0 & 7 & 6 & 5 & 7 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

采用区域分裂合并实现图像分割

✓ 分裂：采用前例中的过程

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 6 & 7 & 1 & 0 & 0 & 1 \\ 1 & 6 & 7 & 5 & 6 & 7 & 1 & 1 \\ 0 & 7 & 6 & 6 & 6 & 0 & 1 & 1 \\ 0 & 7 & 6 & 5 & 7 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

## □ 区域分裂合并

### ◆ 示例

✓ 合并：相似性准则：相邻区域灰度均值差小于2

|      |   |      |  |      |   |      |  |
|------|---|------|--|------|---|------|--|
| 0.75 |   | 0.75 |  | 0.75 |   | 0.5  |  |
| 0    | 0 | 6.25 |  | 1    | 0 | 0.75 |  |
| 1    | 6 |      |  | 6    | 7 |      |  |
| 0    | 7 | 5.75 |  | 6    | 0 | 0.5  |  |
| 0    | 7 |      |  | 7    | 1 |      |  |
| 0.75 |   | 0.75 |  | 1    |   | 0.5  |  |

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 |
| 1 | 2 | 2 | 2 | 2 | 2 | 1 | 1 |
| 1 | 2 | 2 | 2 | 2 | 1 | 1 | 1 |
| 1 | 2 | 2 | 2 | 2 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |



## □ 聚类分割

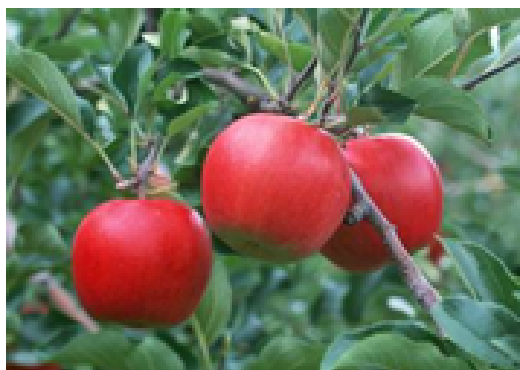
聚类是模式识别中对特征空间中数据进行分类的方法，取“物以类聚”的思想，把某些向量聚集为一组，每组具有相似的值。

把图像分割看做**对像素进行分类**的问题，把像素表示成**特征空间的数据**，采用**聚类算法**划分数据为不同类别，对应原图则是实现对像素的分组，分组后利用“连通成分标记”找到连通区域。

**如何把像素表示成特征空间中的点**：用向量来代表像素或像素周围邻域，向量的元素为与像素相关的特征，根据图像的具体情况，判断待分割区域的共性来设计。

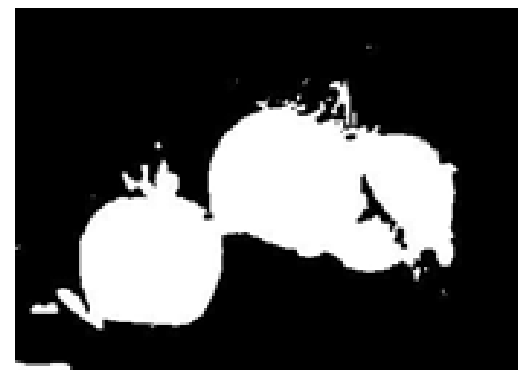
## □ 聚类分割

实例：对一幅苹果图像，利用色彩信息，实现聚类分割



原始图像

色调值作为聚类的数据；  
初始聚类中心选择  
0、1/6、2/6、3/6、  
4/6、5/6；

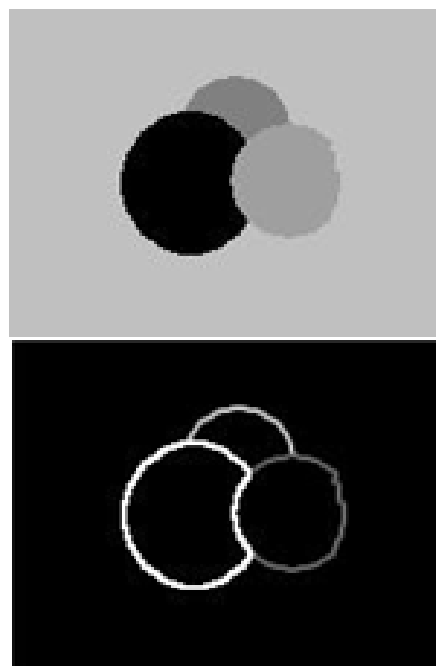
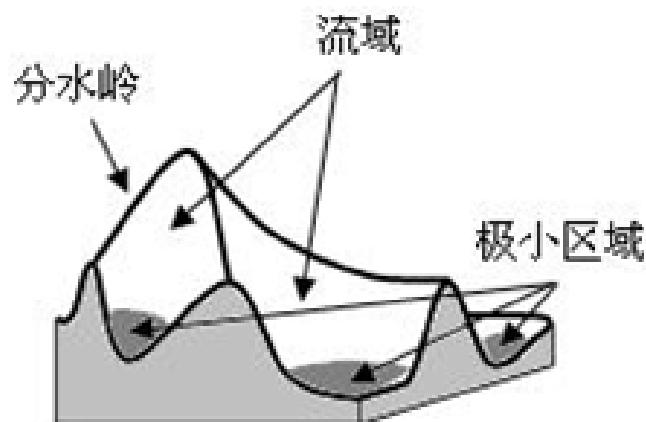


K均值聚类分割

## □ 分水岭分割

基于地形学概念的分割方法，实现可采用数学形态学的方法

### ◆ 地形图与梯度图像

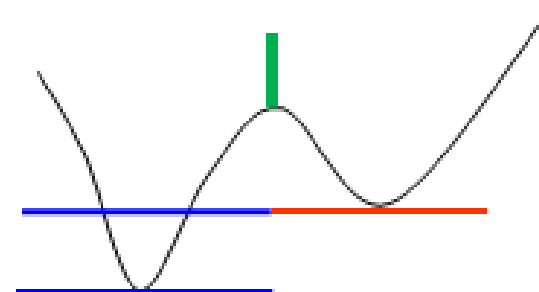


梯度图像中各区域内部对应极小区域，**边界对应高灰度，即分水岭**。确定分水岭，即确定区域边界，实现分割

## □ 分水岭分割

### ◆ 涨水法

设水从谷底上涌，水位逐渐升高。若水位高过山岭，不同流域的水就会汇合。在不同流域中的水面将要汇合到一起时，在中间筑起一道堤坝，阻止水汇合，且堤坝高度随着水面上升而增高。当所有山峰都被淹没时，露出水面的只剩下堤坝，且将整个平面分成了若干个区域。



## □ 分水岭分割

### ◆ 相关定义

原图像:  $f(x, y)$ , 梯度图像:  $g(x, y)$ ,

梯度图像中的极小区域:  $M_1, M_2, \dots, M_r$ , 流域:  $C(M_i)$ ,

谷底和山峰:  $\min g(x, y)$ ,  $\max g(x, y)$

涨水从谷底开始, 单灰值递加, 第 $n$ 步时水深为 $n$ , 定义集合:

$$T(n) = \{(x, y) | g(x, y) < n\}$$

水深为 $n$ 时, 流域所对应水平面区域:

$$C_n(M_i) = C(M_i) \cap T(n)$$

第 $n$ 步流域溢流部分的并:  $C(n)$

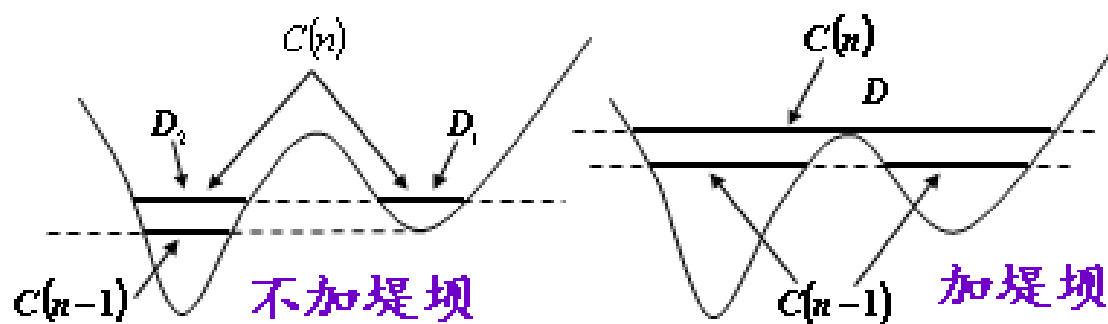
## □ 分水岭分割

### ◆ $C(n)$ 的递归计算

**初始:**  $C(\min + 1) = T(\min + 1)$

**$n = n + 1$ :** 确定  
 $T(n)$ 中的连通成分  
 $D_i$ , 由 $D_i$ 和  
 $C(n-1)$ 的交集状况  
确定 $C(n)$

$C(n-1) \cap D_i$ 有三种情况



$$C(n-1) \cap D_1 = \emptyset$$

$C(n-1) \cap D_2 = C(n-1)$ 的一个连通成分

$C(n-1) \cap D_3 = C(n-1)$ 的多个连通成分

## □ 分水岭分割

### ◆ 示例

对  $f$  进行分水岭分割

$$f = \begin{bmatrix} 3 & 3 & 3 & 1 & 1 & 1 \\ 3 & 1 & 3 & 1 & 1 & 1 \\ 3 & 3 & 3 & 1 & 1 & 1 \\ 1 & 1 & 1 & 3 & 3 & 3 \\ 1 & 1 & 1 & 3 & 1 & 3 \\ 1 & 1 & 1 & 3 & 3 & 3 \end{bmatrix}$$

✓ 采用Prewitt  
梯度算子计算  
梯度图像

$$g = \begin{bmatrix} 4 & 2 & 6 & 6 & 0 & 0 \\ 2 & 0 & 4 & 6 & 0 & 0 \\ 6 & 4 & 0 & 4 & 6 & 6 \\ 6 & 6 & 4 & 0 & 4 & 6 \\ 0 & 0 & 6 & 4 & 0 & 2 \\ 0 & 0 & 6 & 6 & 2 & 4 \end{bmatrix}$$

✓ 求梯度图像最大最小值

$$\begin{aligned} \min &= 0 \\ \max &= 6 \end{aligned}$$

涨水过程中:  $n=1\sim7$

## □ 分水岭分割

✓  $n=1$

$$T(1) = \{(x, y) \mid g(x, y) < 1\}$$

$C(1) = T(1)$  极小区域三个:  $M_1, M_2, M_3$

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 4 | 2 | 6 | 6 | 0 | 0 |
| 2 | 0 | 4 | 6 | 0 | 0 |
| 6 | 4 | 0 | 4 | 6 | 6 |
| 6 | 6 | 4 | 0 | 4 | 6 |
| 0 | 0 | 6 | 4 | 0 | 2 |
| 0 | 0 | 6 | 6 | 2 | 4 |

✓  $n=2$

$$T(2) = \{(x, y) \mid g(x, y) < 2\}$$

三个连通成分  $D_i, i = 1, 2, 3$

$$D_1 \cap C(1) = C_1(M_1) \quad D_2 \cap C(1) = C_1(M_2)$$

$$D_3 \cap C(1) = C_1(M_3)$$

$$C(2) = C(1) + D_{i=1,2,3} = C(1)$$

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 4 | 2 | 6 | 6 | 0 | 0 |
| 2 | 0 | 4 | 6 | 0 | 0 |
| 6 | 4 | 0 | 4 | 6 | 6 |
| 6 | 6 | 4 | 0 | 4 | 6 |
| 0 | 0 | 6 | 4 | 0 | 2 |
| 0 | 0 | 6 | 6 | 2 | 4 |



## □ 分水岭分割

✓  $n=3$

$$T(3) = \{(x, y) \mid g(x, y) < 3\}$$

三个连通成分  $D_i, i = 1, 2, 3$

$$D_1 \cap C(2) = C_2(M_1) \quad D_2 \cap C(2) = C_2(M_2)$$

$$D_3 \cap C(2) = C_2(M_3) \quad C(3) = C(2) + D_{i=1,2,3}$$

✓  $n=4$

$$T(4) = \{(x, y) \mid g(x, y) < 4\}$$

同  $n=3$  时一致，求连通成分没有变化

$$C(4) = C(3)$$

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 4 | 2 | 6 | 6 | 0 | 0 |
| 2 | 0 | 4 | 6 | 0 | 0 |
| 6 | 4 | 0 | 4 | 6 | 6 |
| 6 | 6 | 4 | 0 | 4 | 6 |
| 0 | 0 | 6 | 4 | 0 | 2 |
| 0 | 0 | 6 | 6 | 2 | 4 |

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 4 | 2 | 6 | 6 | 0 | 0 |
| 2 | 0 | 4 | 6 | 0 | 0 |
| 6 | 4 | 0 | 4 | 6 | 6 |
| 6 | 6 | 4 | 0 | 4 | 6 |
| 0 | 0 | 6 | 4 | 0 | 2 |
| 0 | 0 | 6 | 6 | 2 | 4 |

## □ 分水岭分割

✓  $n=5$   $T(5) = \{(x, y) \mid g(x, y) < 5\}$

一个连通成分  $D$

$$D \cap C(4) = C_4(M_1) \cup C_4(M_2) \cup C_4(M_3)$$

属于**第三种情况**，三个极小区域即将连通；在  $D$  中**加堤坝**，黑色底纹的点，其余阴影点为  $C(5)$

✓  $n=6$   $T(6) = \{(x, y) \mid g(x, y) < 6\}$

同  $n=5$  时一致，求连通成分没有变化  $C(6) = C(5)$

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 4 | 2 | 6 | 6 | 0 | 0 |
| 2 | 0 | 4 | 6 | 0 | 0 |
| 6 | 4 | 0 | 4 | 6 | 6 |
| 6 | 6 | 4 | 0 | 4 | 6 |
| 0 | 0 | 6 | 4 | 0 | 2 |
| 0 | 0 | 6 | 6 | 2 | 4 |

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 4 | 2 | 6 | 6 | 0 | 0 |
| 2 | 0 | 4 | 6 | 0 | 0 |
| 6 | 4 | 0 | 4 | 6 | 6 |
| 6 | 6 | 4 | 0 | 4 | 6 |
| 0 | 0 | 6 | 4 | 0 | 2 |
| 0 | 0 | 6 | 6 | 2 | 4 |

## □ 分水岭分割

✓  $n=7$   $T(7) = \{(x, y) | g(x, y) < 7\}$

一个连通成分  $D$   $D \cap C(6) = C_6(M_1) \cup C_6(M_2) \cup C_6(M_3)$

属于**第三种情况**，三个极小区域即将连通；在  $D$  中**加堤坝**，黑色底纹的点，其余阴影点为  $C(7)$

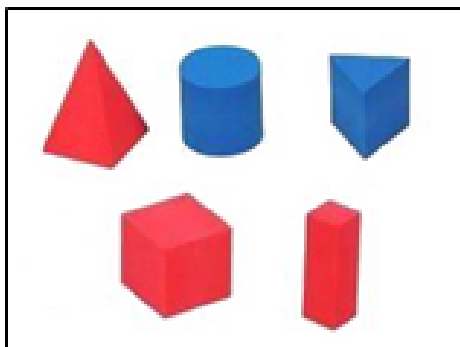
至此，所有流域均被淹没，只剩下分水岭露于水面上，分割完成，用编号1、2、3表示最后分割出来的区域，用0表示分水岭，则分割结果为：

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 4 | 2 | 6 | 6 | 0 | 0 |
| 2 | 0 | 4 | 6 | 0 | 0 |
| 6 | 4 | 0 | 4 | 6 | 6 |
| 6 | 6 | 4 | 0 | 4 | 6 |
| 0 | 0 | 6 | 4 | 0 | 2 |
| 0 | 0 | 6 | 6 | 2 | 4 |

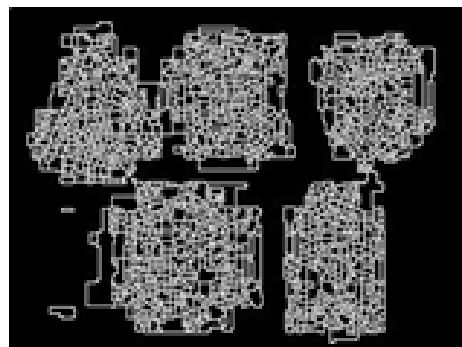
|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 2 | 2 | 2 | 0 | 1 | 1 |
| 2 | 2 | 2 | 0 | 1 | 1 |
| 2 | 2 | 2 | 0 | 0 | 0 |
| 0 | 0 | 0 | 2 | 2 | 2 |
| 3 | 3 | 0 | 2 | 2 | 2 |
| 3 | 3 | 0 | 2 | 2 | 2 |

## □ 分水岭分割

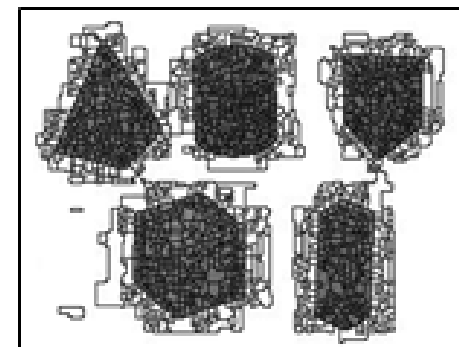
产生过分割现象



原始图像



分水岭



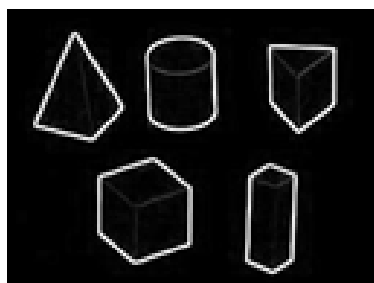
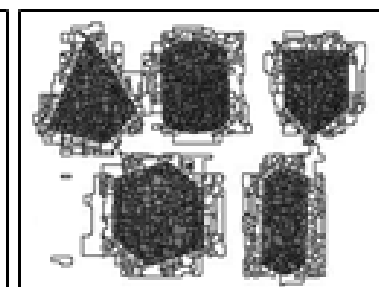
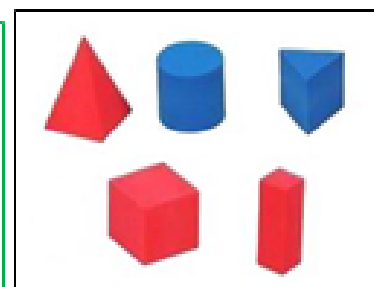
分割结果

**过分割现象产生原因：**梯度噪声、量化误差及目标内部细密纹理的影响，导致许多局部的“谷底”和“山峰”，形成很多小区域。

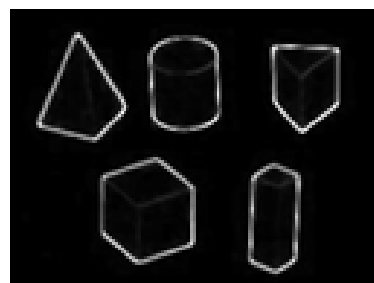
## □ 分水岭分割

**减少过分割：**加入预处理和后处理

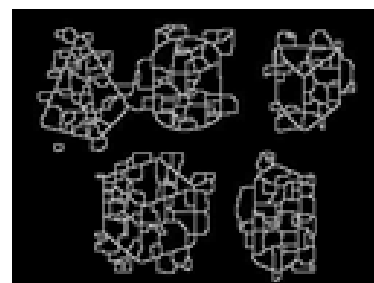
- ✓ 滤波减弱噪声干扰；
- ✓ 增强图像中的轮廓；
- ✓ 合并一些较小的区域等。



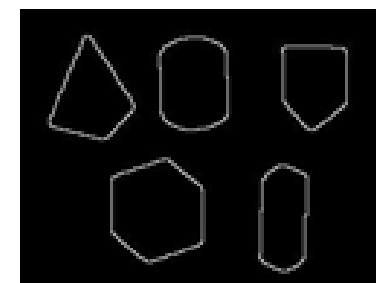
梯度图像



中值滤波



分水岭分割



区域合并

**其他改进方法：**

- ✓ Meyer F. and Beucher S. Morphological segmentation.
- ✓ Dobrin B. P., Viero T. and Gabbouj M. Fast watershed algorithms: Analysis and extensions.

### 3.1.2新的分割思路

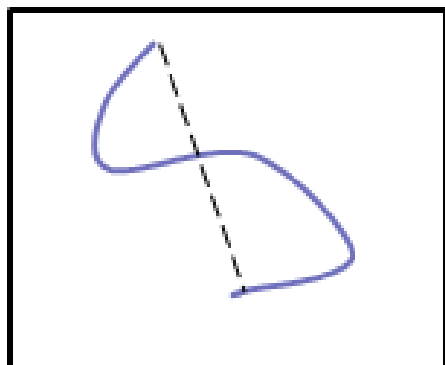
#### ■ 图像分割的新需求

能适应图像数据集的增长、单个图像数据大小的增长、数据维数的增长

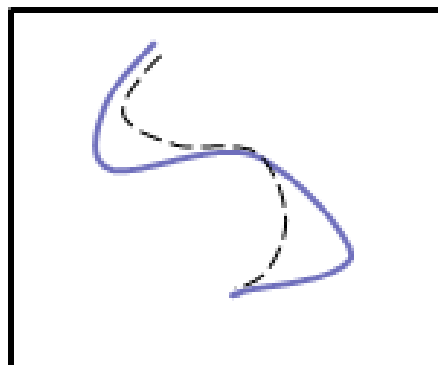
- 均值移位分割 (Mean Shift Image Segmentation, 核密度估计)
- 活动轮廓模型分割 (Active Contour Models Image Segmentation)
- 几何变形模型-水平集和测地活动轮廓 (偏微分)
- 模糊分割
- 基于图论的分割
- 针对3D图像的分割

## ■ 活动轮廓模型

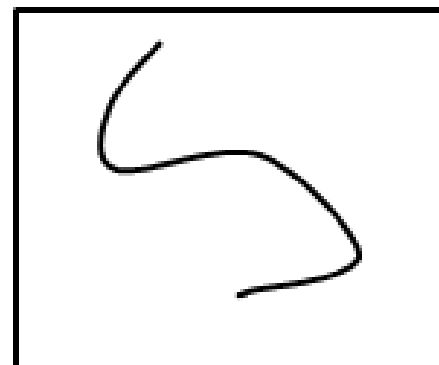
也称Snake，能量样条曲线，形状及在图像中的位置决定其能量，局部能量最小对应于想要的图像属性。



指定Snake的估计形状和开始位置，在期望的轮廓附近



向合适解的位置移动，即能量函数最小化迭代



真实轮廓对应能量函数最小化

## □ 能量函数

$$E_{snake} = \int_0^1 E_{snake}[v(s)] ds = \int_0^1 \{ E_{int}[v(s)] + E_{image}[v(s)] + E_{con}[v(s)] \} ds$$

轮廓点的坐标向量 $[x(s), y(s)]$ ,  
 $s \in [0, 1]$ , 归一化的弧长参数

因弯曲产生的  
内部能量

从图像中得  
到的能量

外部的约束  
力

**内部能量**

$$E_{int} = \alpha(s) \left| \frac{dv}{ds} \right|^2 + \beta(s) \left| \frac{d^2v}{ds^2} \right|^2$$

弹性势能：当外力使曲线伸展时，  
产生弹性势能使其收缩。  
 $\alpha(s)$ 弹力系数，控制曲线的弹性

弯曲势能：曲线上各点的曲率， $\beta(s)$   
强度系数，控制曲线的刚性。  
当曲线为圆，弯曲势能最小



## 来自图像数据的能量

一般通过计算图像的灰度、边缘等特征获得，用来吸引曲线到目标的边缘轮廓等

吸引到目标轮廓  $E_{image} = E_{edge} = -|\nabla f(x, y)|^2$   $E_{image} = E_{edge} = -|\nabla [G_{\sigma}(x, y) * f(x, y)]|^2$

多项加权综合  $E_{image} = w_{line} E_{line} + w_{edge} E_{edge} + w_{term} E_{term}$

三项加权和决定snake被吸引到轮廓线、边缘还是端点

$E_{line} = f(x, y)$ ,  $w_{line}$ 的符号指定了snake偏向亮线或暗线

$E_{term}$ 反应轮廓线端点和角点对snake活动的可能影响，平滑图像的轮廓曲率

## 来自外部的约束能量 $E_{con}$

由用户指定或来自于其他更高层处理，可以让snake**朝着或背离某些指定的特征**，比如，snake到达局部能量极小值，但高层处理判断为错误，在该处产生一个能量峰值区域，迫使snake离开去找另一个局部极小值

### □ 能量函数求极小

$$E_{snake} = \int_0^1 \left[ \alpha(s) |v'(s)|^2 + \beta(s) |v''(s)|^2 \right] + E_{image}[v(s)] ds$$

使用变分法，应用欧拉-拉格朗日方程，得

$$\alpha(s)v''(s) - \beta(s)v'''(s) - \nabla E_{image} = 0$$

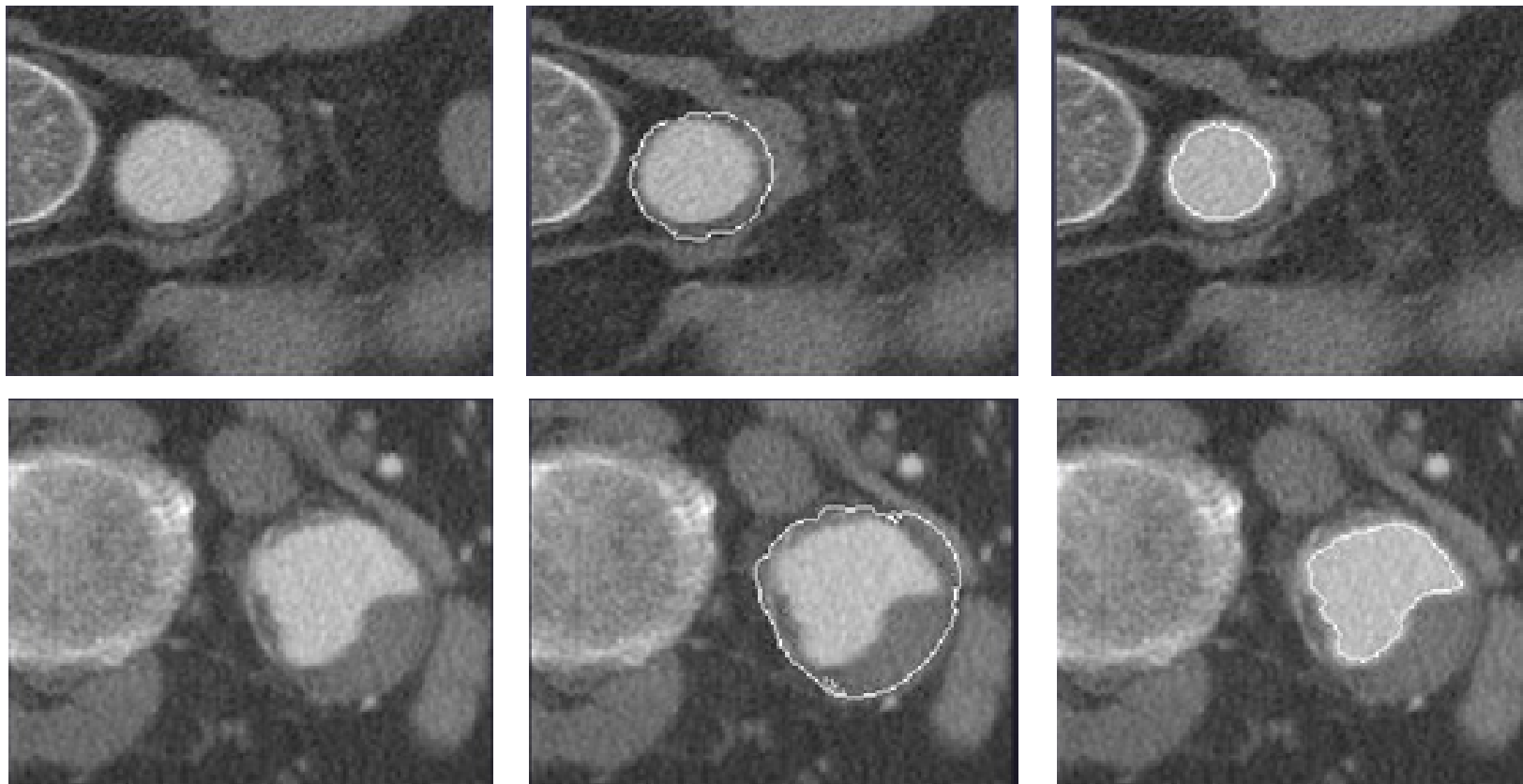
可看作力的平衡方程，每一项是由各个能量项产生的力，曲线在这些力的共同作用下变形

**弹性力**，使曲线收缩，若无其他力，将使曲线收缩为一个点

**弯曲力**，使曲线平滑，若无其他力，将使曲线成为一个圆

**外力**，将曲线吸引到感兴趣的目标轮廓上

## □ 应用



## □ 存在的问题

- ◆ 对初始位置敏感，需要依赖其他机制将snake放在感兴趣的图像特征附近
- ◆ 搜索凹形边界时存在问题，无法步入凹形区
- ◆ 解欧拉-拉格朗日公式的解数值不稳定
- ◆ 由于曲线是参数模型，变形过程中无法自由改变曲线的拓扑结构，提取轮廓时必须明确目标个数或增加控制条件

## □ 改进思路

**Berger M.O. and Mohr R. Towards autonomy in active contour models. In 10<sup>th</sup> International Conference on Pattern Recognition, 1990**

**蛇行增长，解决了最初方法的数值不稳定**

**Cohen L. D. On active contour models and balloons. CVGIP-Image Understanding, 1991** **引入膨胀力，解决了对初始位置敏感的问题**

**Xu C. and Prince J.L. Snakes, Shapes, and Gradient Vector Flow. IEEE Transaction on Image Processing, 1998**

**梯度矢量流，GVF场，解决凹形边界分割问题**

**McInerney T. and Terzopoulos D. Topologically adaptabel snakes. In 5th International Conference on Computer Vision, 1995**

**T-snake模型，在曲线变形过程中可以很好的处理曲线的拓扑结构变化**