

第八章：图像分割

中国科学技术大学
电子工程与信息科学系



图像分割

- 图像分割定义
- 阈值分割
- 区域生长法
- 分裂合并方法
- 分水岭算法
- 聚类分割算法
- 主动轮廓分割
- Graph Cut



图像分割

- 图像分割定义
- 阈值分割
- 区域生长法
- 分裂合并方法
- 分水岭算法
- 聚类分割算法
- 主动轮廓分割
- Graph Cut



图像分割的定义

- 令集合 R 代表整个图像区域，对 R 的分割可看做将 R 分成若干个满足下述条件的非空的子集（子区域）

R_1, R_2, \dots, R_n :

1. $\bigcup_{i=1}^n R_i = R$
2. 对 $\forall i \neq j$, 有 $R_i \cap R_j = \emptyset$
3. 每个子区域 R_i 都是联通的；
4. 对于各个子区域，有**均匀性测度度量** $P(\cdot)$ 为真；但对其中任意两个和两个以上相邻子区域之并，其均匀性测度度量 P 为假（**避免过度分割**）。即：

$$P(R_i) = \text{TRUE}, \text{ 且 } P(R_i \cup R_j) = \text{FALSE}$$



图像分割

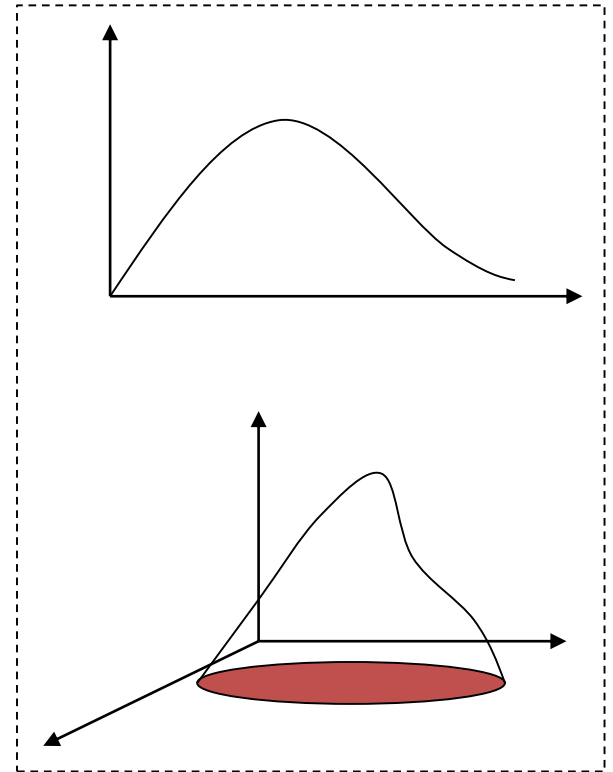
- 图像分割定义
- 阈值分割
- 区域生长法
- 分裂合并方法
- 分水岭算法
- 聚类分割算法
- 主动轮廓分割
- Graph Cut

阈值分割 (Thresholding)

- 取阈值法是以图像**直方图**为依据，选定阈值，再逐个对像素作判决

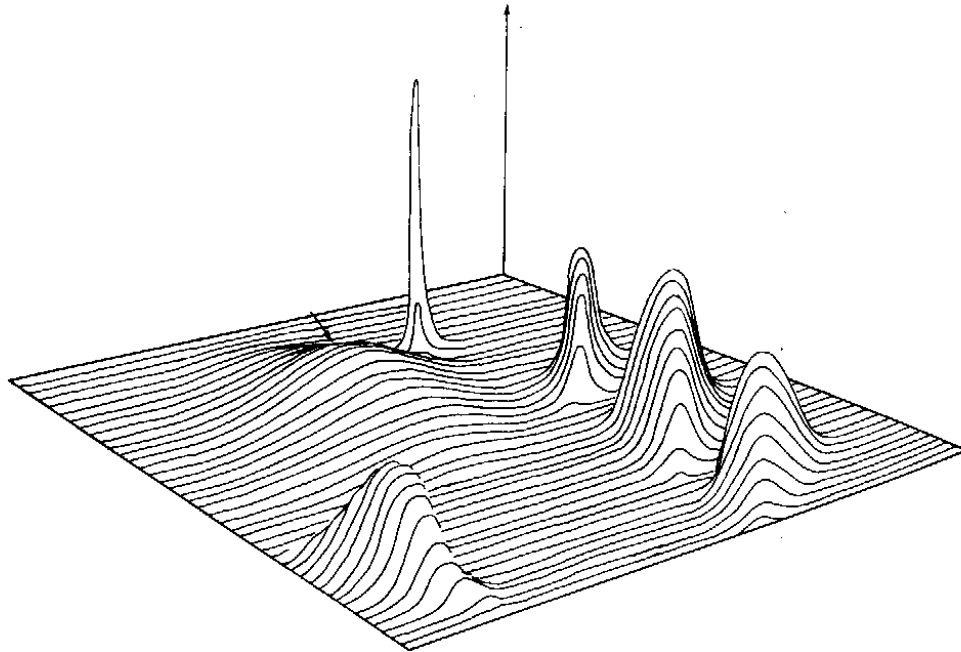
- 图像直方图可以是：
 - 单个特征的一维直方图
 - ✓ 如灰度直方图
 - 多个特征的多维直方图
 - ✓ 如两个波段组成的二维直方图

- **特征**可以是灰度，也可以是其他值

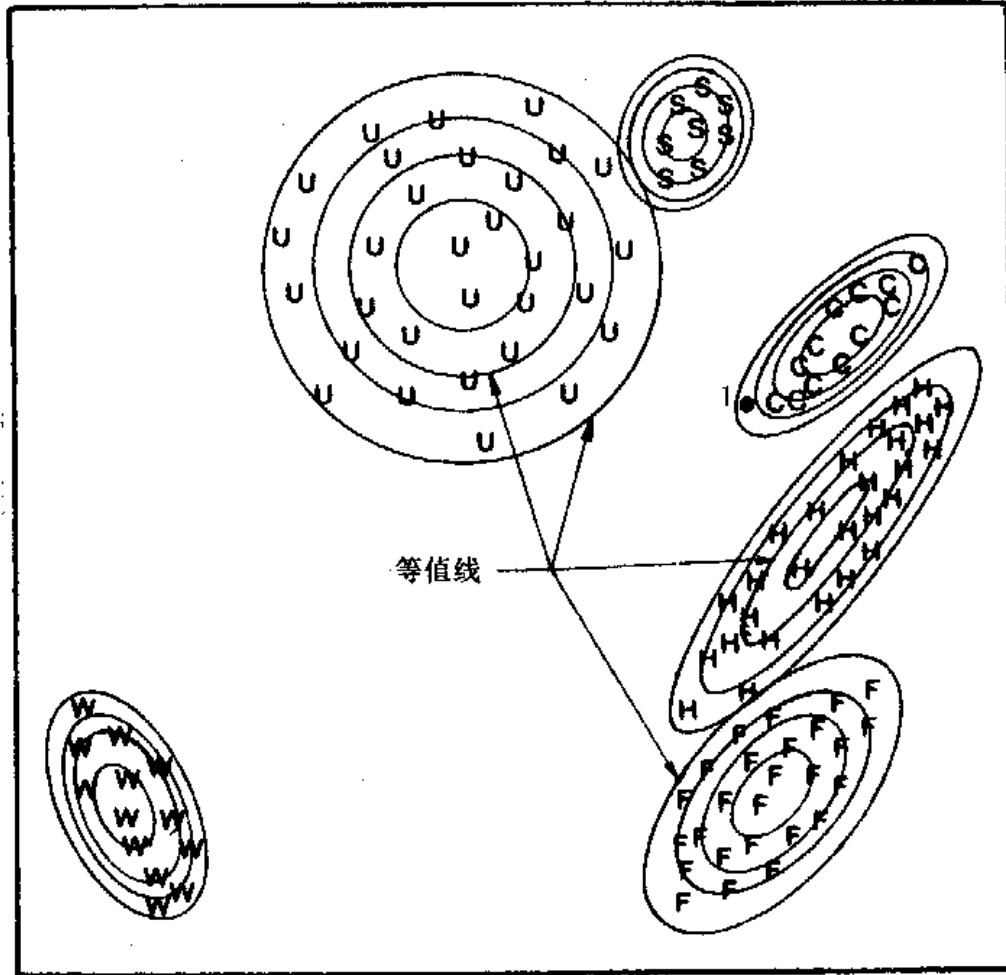


特征直方图

- 图像的特征直方图可以是：
 - 灰度直方图
 - 梯度直方图
 - 纹理直方图
 -
- 多个特征组成多维直方图



基于二维直方图的分割



原理和分类

□ 图像模型：

- 图像由具有**单峰测度**(灰度, 颜色, 纹理)分布的**目标**和**背景**组成
 - ✓ 目标: Object
- 目标或背景内部的相邻像素间的测度值高度相关
 - ✓ 均匀性测度度量是真
- 目标和背景**交界处**两边的像素测度值差异大

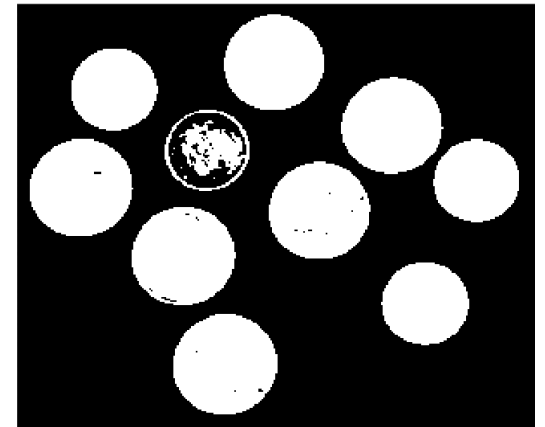
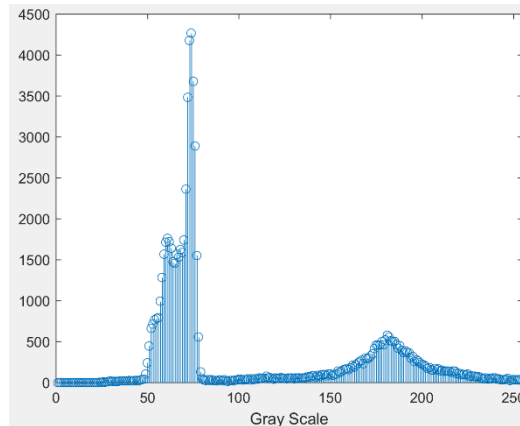


原理和分类

□ 单阈值分割图像

- 对灰度图（取值在 g_{\min} 和 g_{\max} 之间）确定一个灰度阈值 T
($g_{\min} < T < g_{\max}$)

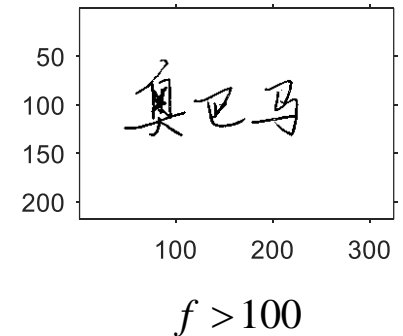
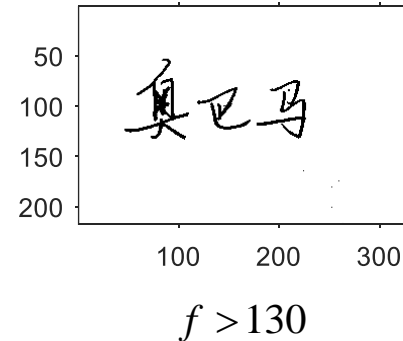
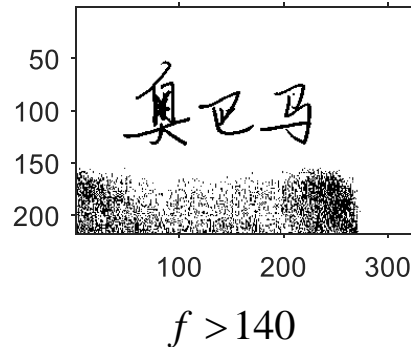
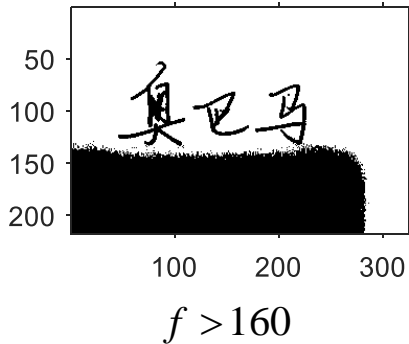
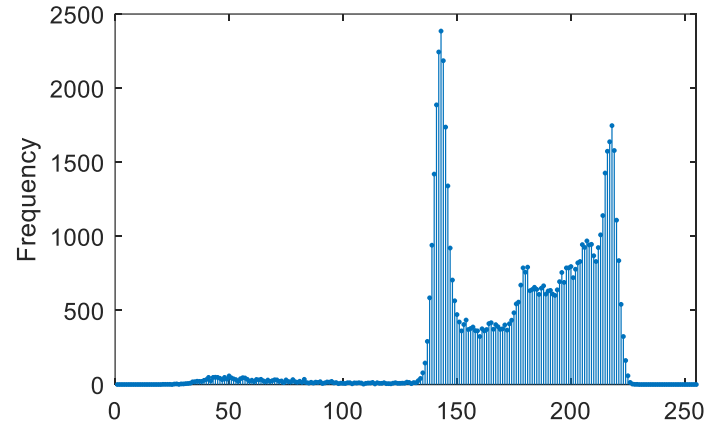
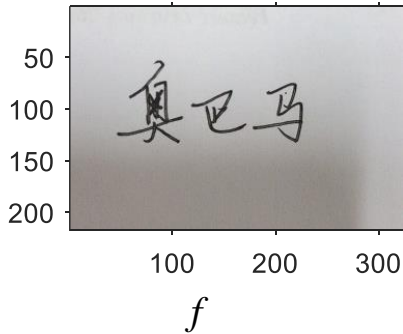
$$g(x, y) = \begin{cases} 1 & \text{如 } f(x, y) > T \\ 0 & \text{如 } f(x, y) \leq T \end{cases}$$



原理和分类

□ 实例：生成电子签名

- 课后作业：做一份自己的电子签名

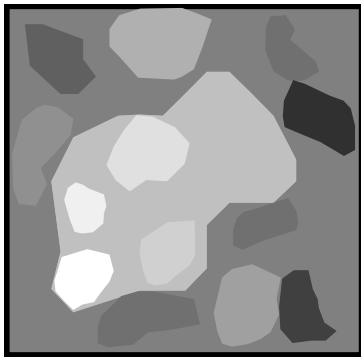


原理和分类

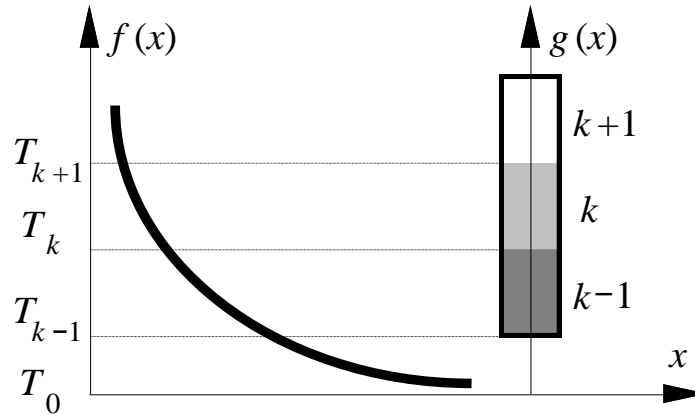
□ 多閾值分割图像

- 确定一系列分割閾值

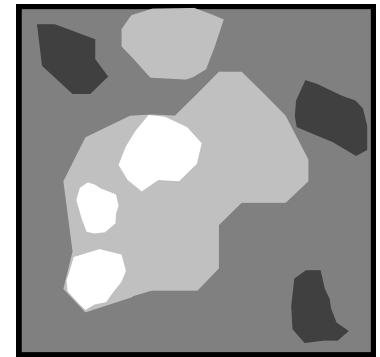
$$g(x, y) = k \quad \text{如} \quad T_k < f(x, y) \leq T_{k+1} \quad k = 0, 1, 2, \dots, K$$



(a)



(b)



(c)



原理和分类

□ 阈值分割方法分类

$$T = T [x, y, f(x, y), p(x, y)]$$

(1) 依赖像素的（全局）阈值方法：

- 仅根据 $f(x, y)$ 来选取阈值

(2) 依赖区域的（局部）阈值方法：

- 根据 $f(x, y)$ 和 $p(x, y)$ 来选取阈值， $p(x, y)$ 表示像素 (x, y) 的邻域

(3) 依赖坐标的（动态）阈值方法：

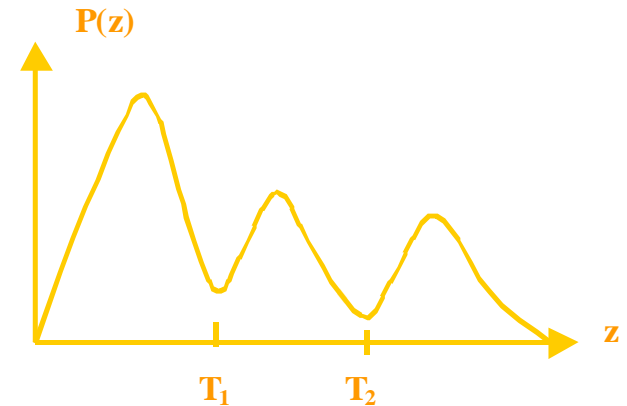
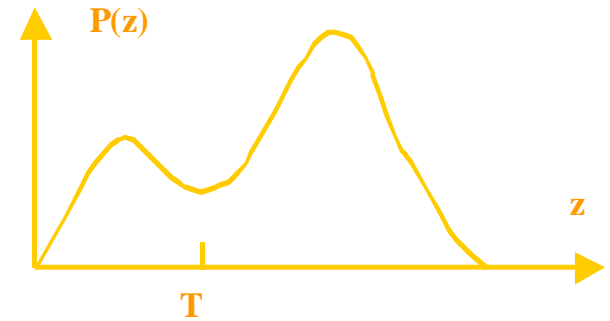
- 除根据 $f(x, y)$ 和 $p(x, y)$ 来选取，还与 x, y 有关

□ 前两种阈值也称为固定阈值

(1) 依赖象素的阈值选取

适用于直方图为双峰或多峰的图象

- 单一阈值
 - 取一阈值，把图像分成目标和背景两部分
- 多个阈值
 - 取几个阈值将图象分成若干个目标和背景几部分
- 阈值宜取在双峰或多峰直方图的谷点。



极小值点阈值： $\frac{\partial h(z)}{\partial z} = 0$ 和 $\frac{\partial^2 h(z)}{\partial z^2} > 0$



最优阈值的选取

可以证明，在对象的分布近似为正态分布时，阈值选取在直方图的谷点，分割误差最小。

$$p_1(z) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(z-\mu_1)^2}{2\sigma_1^2}\right]$$

$$p_2(z) = \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(z-\mu_2)^2}{2\sigma_2^2}\right]$$

$$E(T) = P_1 E_1(T) + P_2 E_2(T)$$

$$E_2(T) = \int_{T_r}^{\infty} p_1(z) dz$$

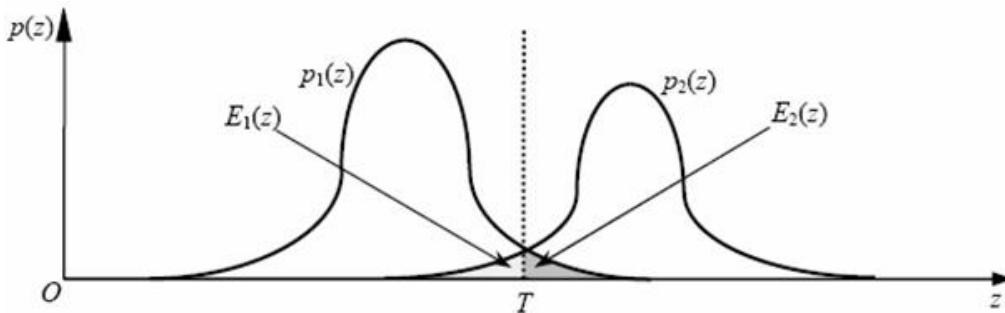
$$E_1(T) = \int_{-\infty}^{T_r} p_2(z) dz$$

(P_1, P_2 是先验概率)

为求使该误差最小的阈值，令 $\frac{\partial E(T)}{\partial T} = 0 \Rightarrow P_1 \times p_1(T) = P_2 \times p_2(T)$

当 $\sigma_1 = \sigma_2 = \sigma$ 时

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_2}{P_1}\right)$$



$p(z)$ 的参数可根据最小均方的方法借助直方图得到

$$e_{ms} = \frac{1}{n} \sum_{i=1}^n [p(z_i) - h(z_i)]^2$$



Otsu's 方法 (1979)

□ 基本思想

- 假设图像像素可分为两类，选择分割阈值，使得类内方差最小。

□ 形式化表达

- 给定一个图像 I ，计算其归一化的直方图 $P(i)$
 - ✓ 归一化：直方图向量中所有元素值之和为1
- 选择阈值 t ，将像素分为两类，每类概率：

$$q_1(t) = \sum_{i=1}^t P(i) \qquad q_2(t) = \sum_{i=t+1}^T P(i)$$

- 每类像素的灰度均值：

$$\mu_1(t) = \sum_{i=1}^t \frac{i \cdot P(i)}{q_1(t)} \qquad \mu_2(t) = \sum_{i=t+1}^T \frac{i \cdot P(i)}{q_2(t)}$$

- Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66.



Otsu's 方法

□ 形式化表达

每类像素的灰度方差：

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad \sigma_2^2(t) = \sum_{i=t+1}^T [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

■ 加权类内方差： $\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$

■ 最优阈值选择（穷举法）： $\tau = \arg \min_t \sigma_w^2(t)$

■ 二值化图像分割： $B(i, j) = \begin{cases} 1 & I(i, j) \geq \tau \\ 0 & I(i, j) < \tau \end{cases}$

(1) 依赖象素的阈值选取

□ 由直方图凹凸性确定的阈值

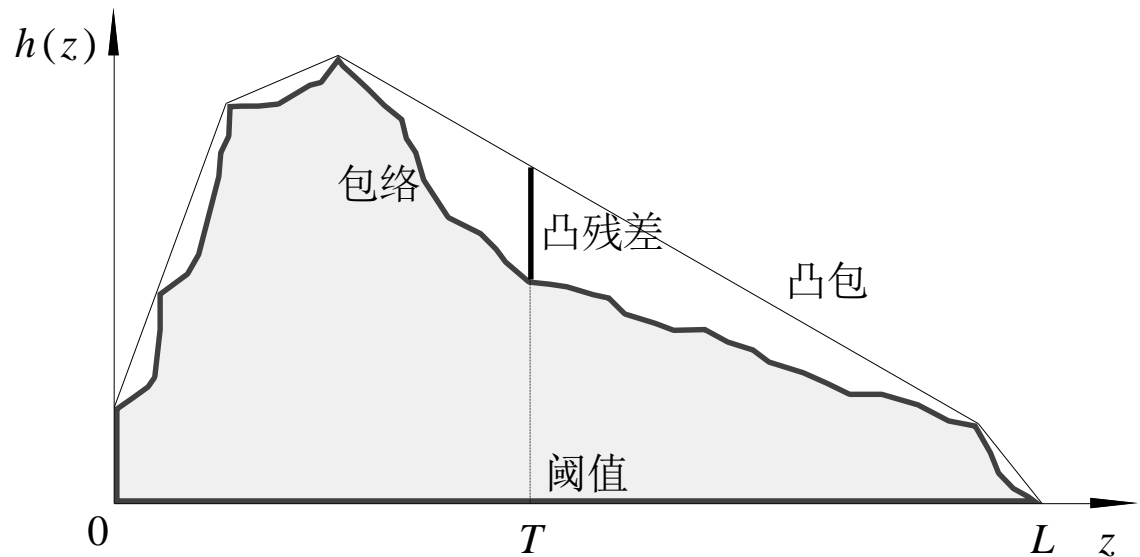
直方图的一个峰淹没在另一个峰旁的缓坡里

□ 直方图的包络

→ 区域凸包

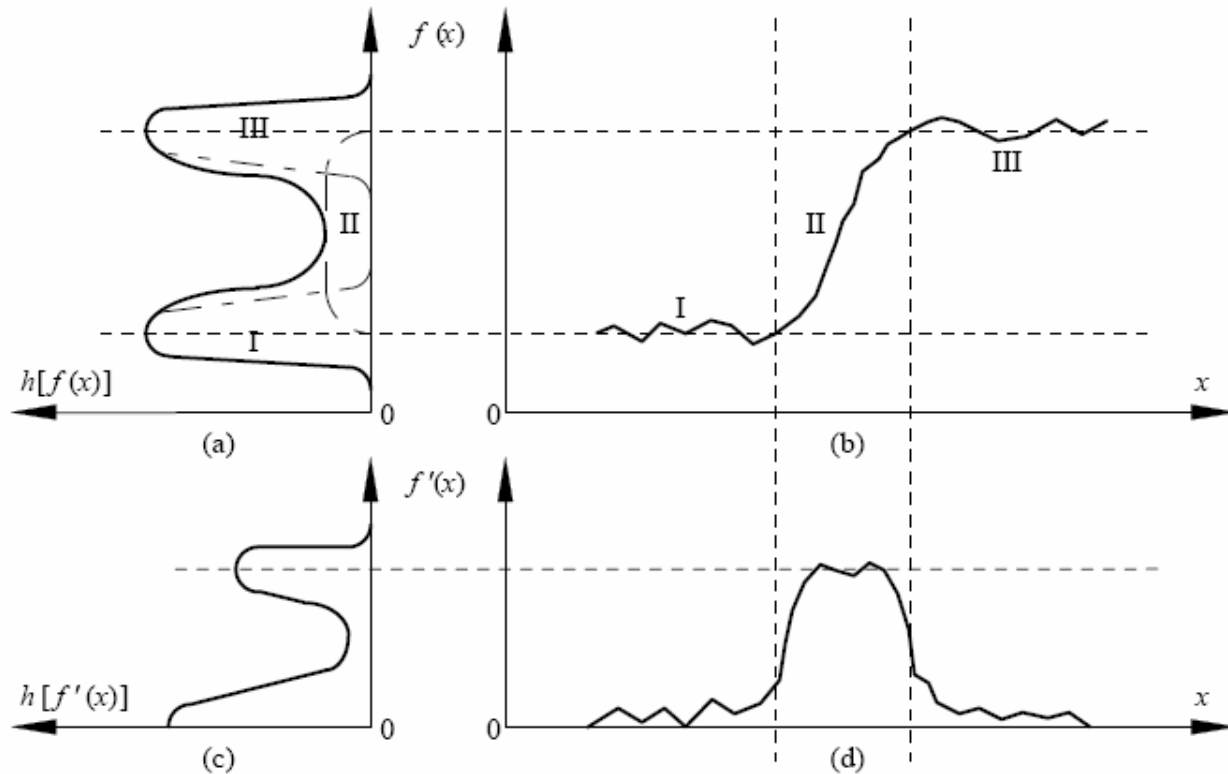
→ 最大凸残差

→ 分割阈值



(2) 依赖区域的阈值选取

- 根据 $f(x, y)$ 和 $p(x, y)$ 来选取阈值, $p(x, y)$ 表示像素 (x, y) 的邻域
- 考虑一个特例: 图象 f 只有一行像素





(2) 依赖区域的阈值选取

□ 直方图变换

■ 仅利用像素灰度可能出现的问题：

✓ 灰度直方图的谷被填充

□ 借助邻域性质，改造原来的直方图

1. 获得低梯度值像素的直方图

✓ 峰之间的谷比原直方图深

2. 获得高梯度值像素的直方图

✓ 峰由原直方图的谷转化而来

(2) 依赖区域的阈值选取

□ 直方图变换

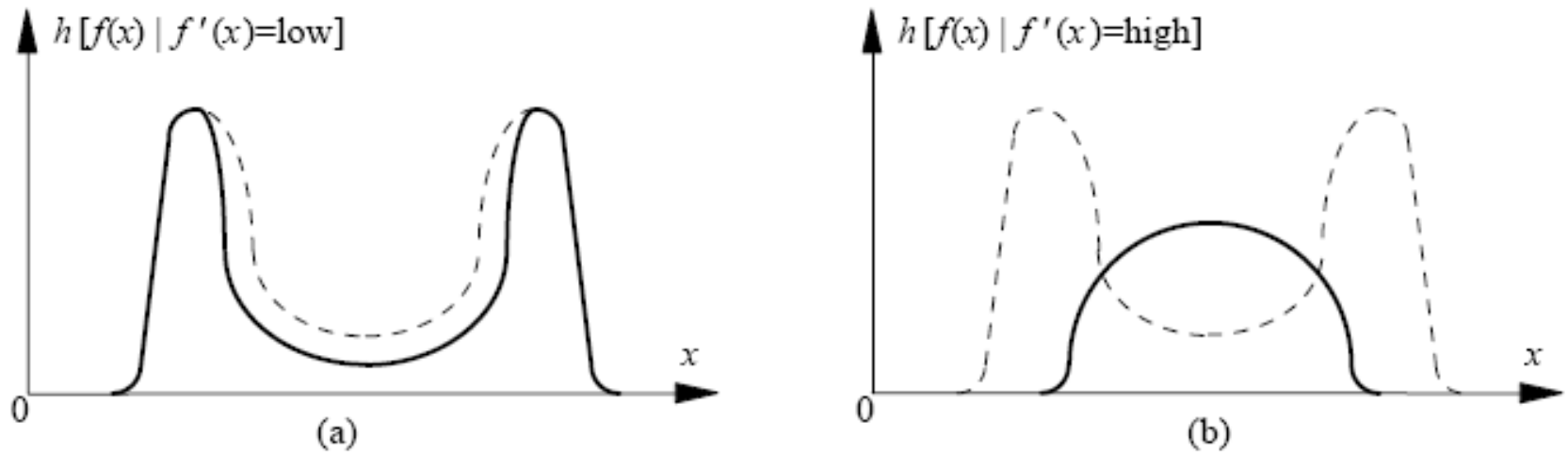
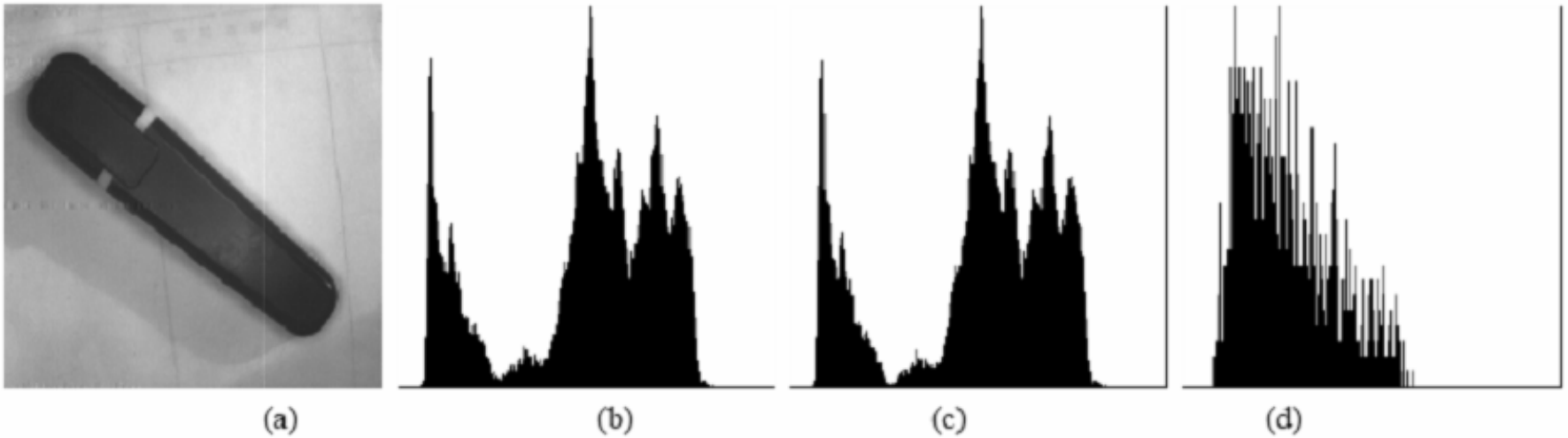


图 4.4.6 变换直方图示例

变换直方图实例

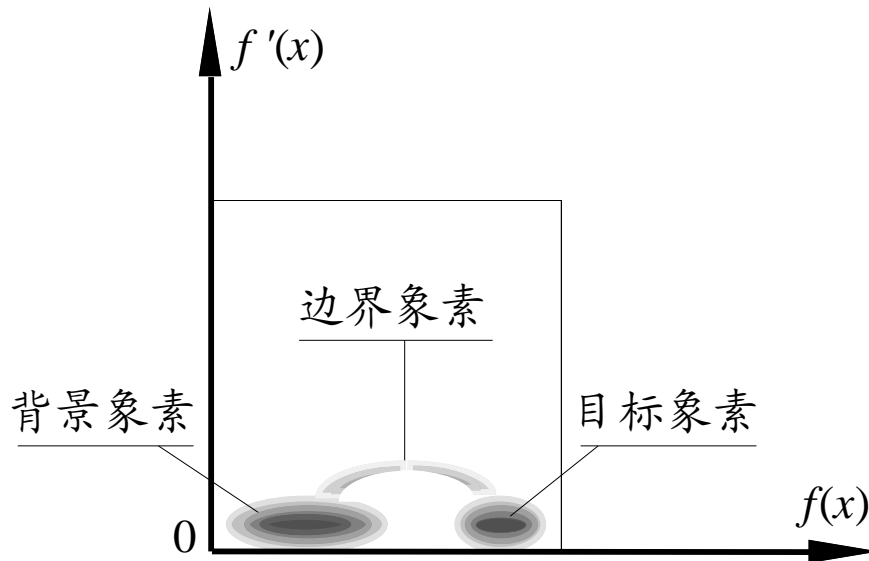


(a) 原图像； (b) 为其直方图； (c) 具有低梯度像素的直方图； (d) 具有高梯度像素的直方图

(2) 依赖区域的阈值选取

□ 灰度-梯度散射图

2-D直方图



2-D图象

1个轴是灰度值轴

1个轴是梯度值轴

示例

目标，背景聚类

(2) 依赖区域的阈值选取

□ 灰度-梯度散射图

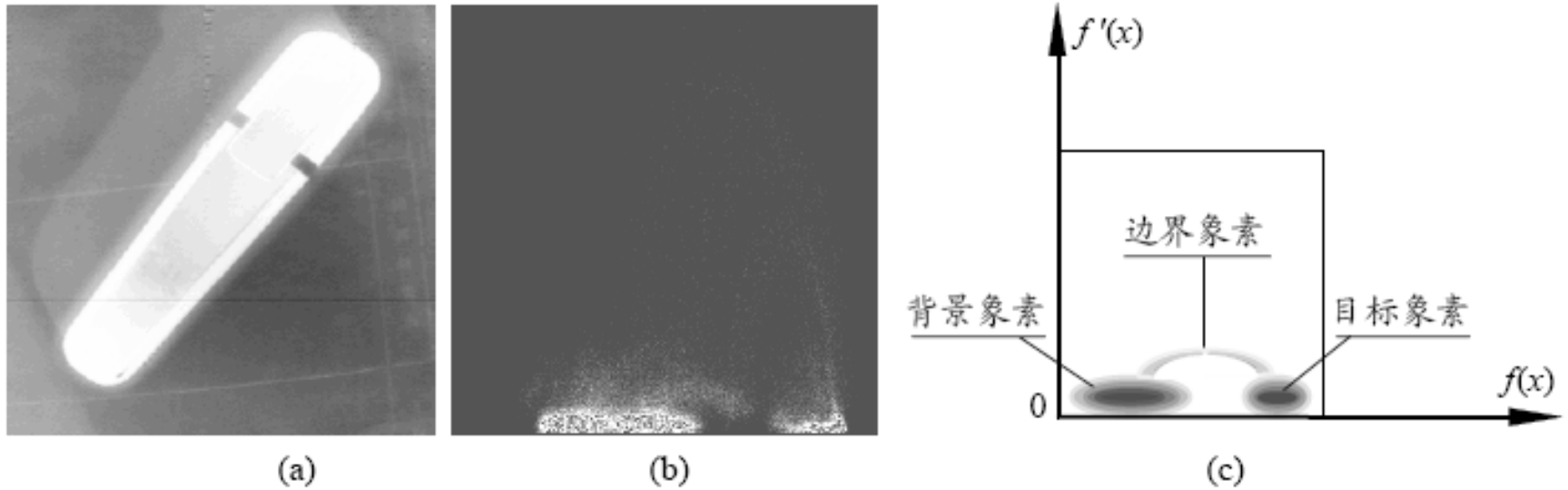
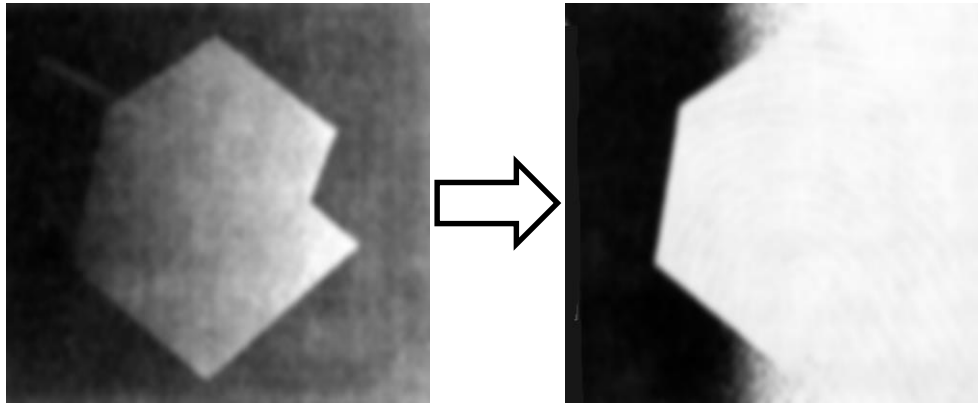


图 4.4.8 灰度和梯度散射图

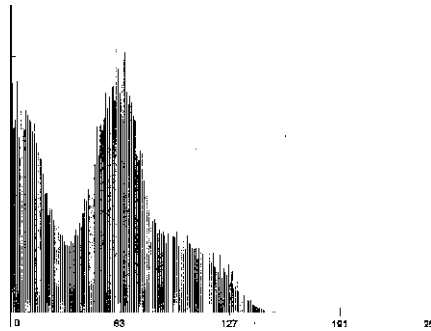
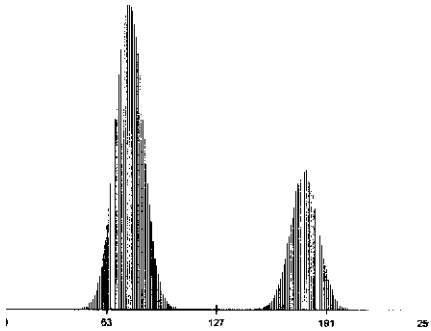
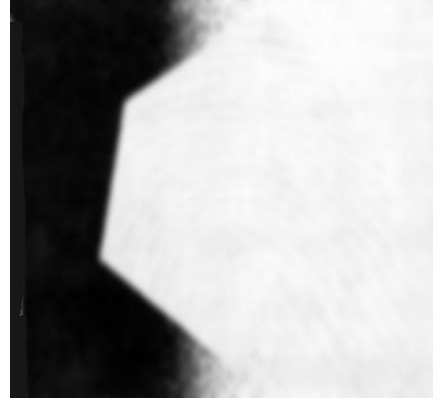
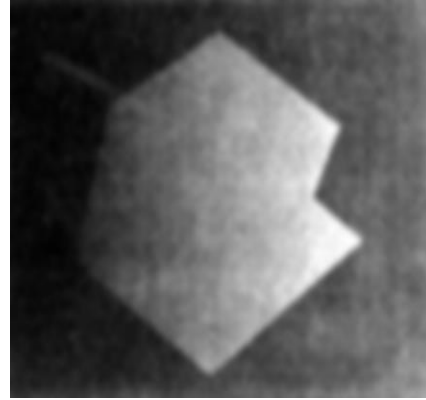
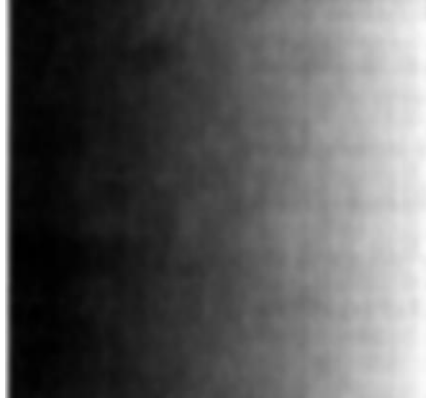
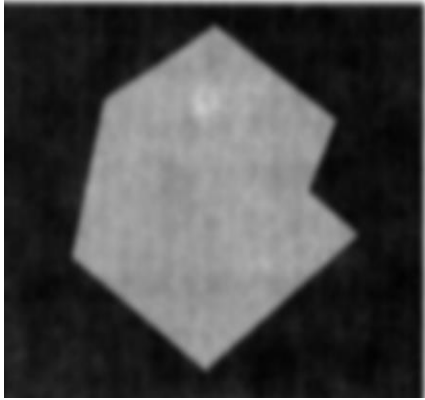
(3) 依赖坐标的阈值选取

- 全局阈值不能兼顾图像各处的情况
 - 解决方案：用与坐标相关的一系列阈值来对图像分割
 - 阈值除根据 $f(x, y)$ 和 $p(x, y)$ 有关，还与 x, y 有关



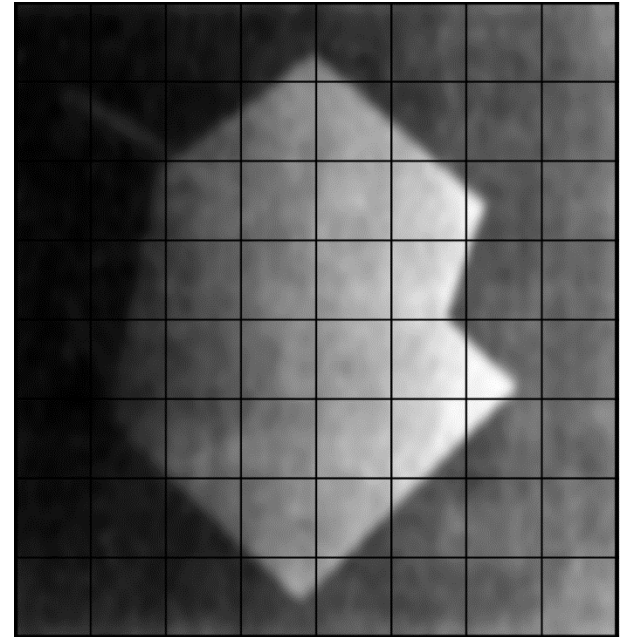
- 基本思想/思路：为每个像素计算一个分割阈值
 1. 将图像分解成一系列子图像
 2. 对每个子图像计算一个阈值
 3. 用这些子图像阈值，对每个像素的阈值进行插值
 4. 用插值结果（阈值曲面）进行分割

光照不均匀对分割的影响

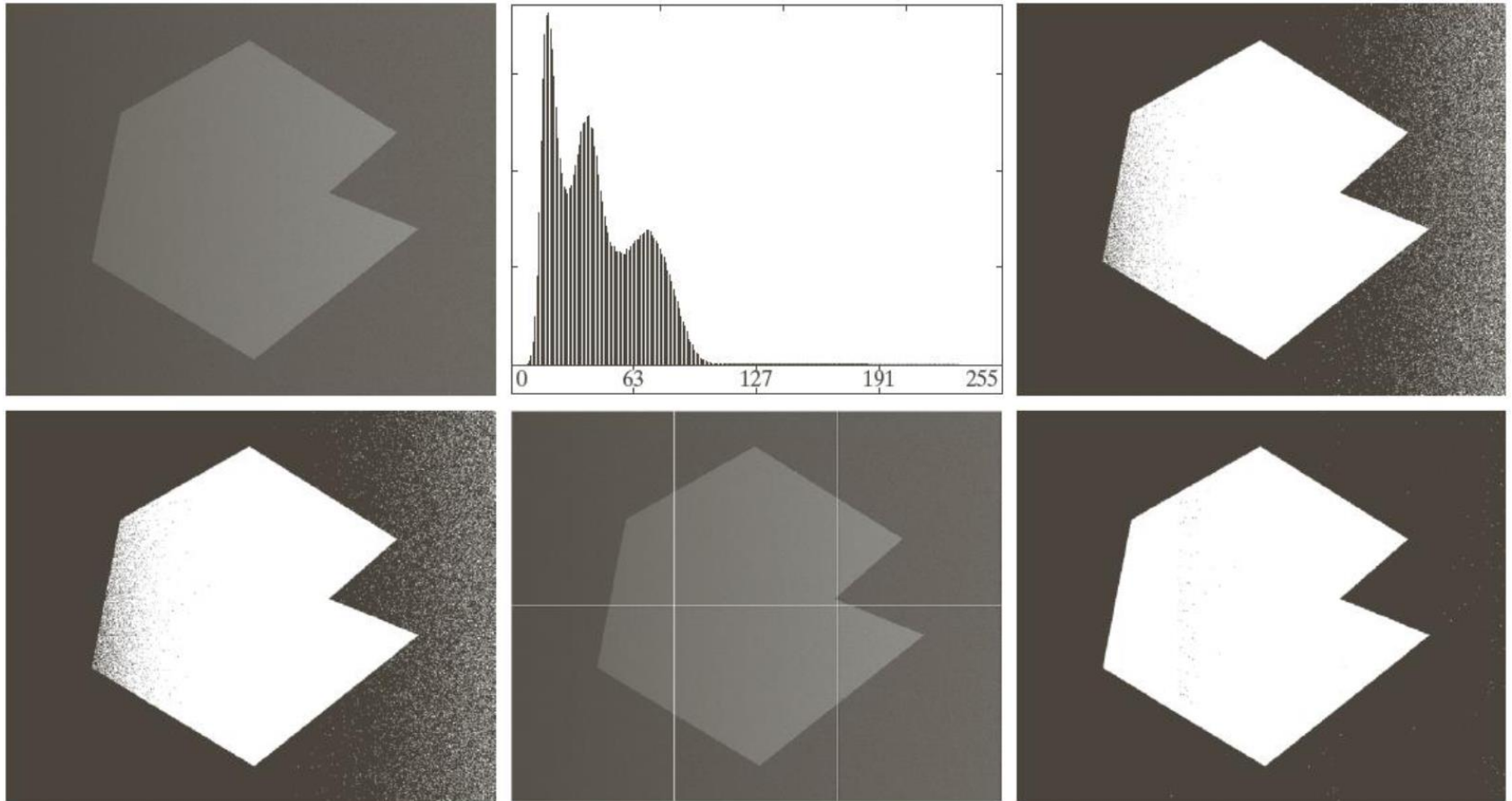


变化阈值法

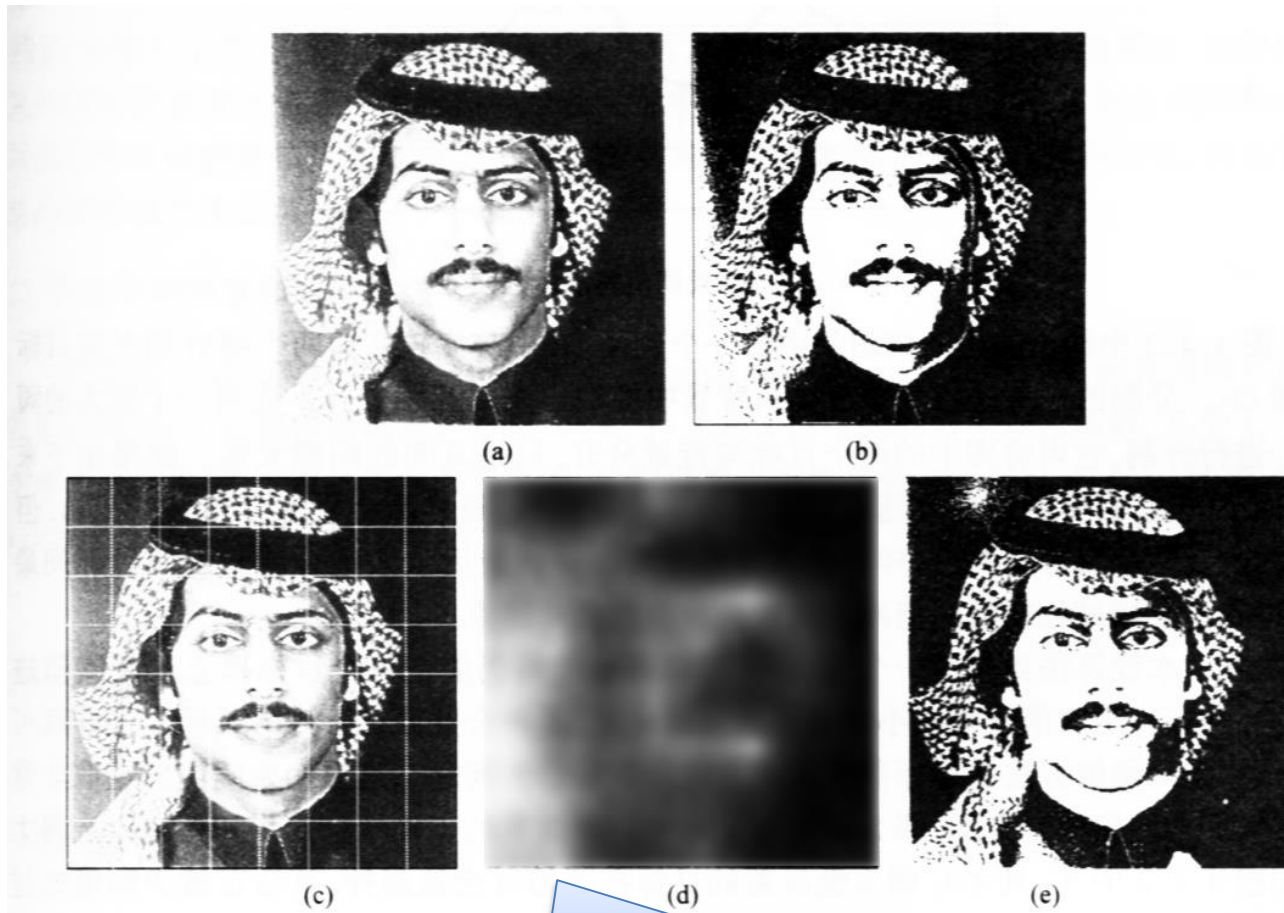
1. 将整幅图像分成一系列互相之间有50%重叠的子图像
 - 如何确定子图像的大小?
2. 做出每个子图像的直方图
3. 检测各个子图像的直方图是否为双峰的。如是，则采用最优阈值法确定一个阈值；否则，不进行处理
 - 如何判断双峰分布？方差
4. 根据对直方图为双峰的子图像得到的阈值，通过**插值**得到所有子图像的阈值
5. 根据各子图像的阈值，再通过**插值**得到所有像素的阈值，然后对图像进行分割



依赖坐标的阈值选取



分块取阈值一例



阈值图像，每个像素都有一个对应阈值



图像分割

- 图像分割定义
- 阈值分割
- 区域生长法
- 分裂合并方法
- 分水岭算法
- 聚类分割算法
- 主动轮廓分割
- Graph Cut



区域生长法

□ 基本思想：

- 均匀性度量准则：将相似像素组合起来构成区域

□ 基本步骤：

1. 选择区域的种子像素
2. 确定将相邻像素包括进来的准则
3. 制定生长停止的规则

□ 讨论：

1. 种子像素的选取
 - ✓ 聚类中心
 - ✓ 交互选取
2. 生长准则（依赖具体应用）



区域生长法

□ 生长示例

1. 根据直方图选取聚类中心的象素为种子
2. 根据与种子象素灰度差 ($< T$) 判断是否生长
3. 根据图象边缘确定生长何时终结

原始图

1	0	4	7	5
1	0	4	7	7
0	1	5	5	5
3	0	5	6	5
3	3	5	6	4

$T = 3$

1	1	5	5	5
1	1	5	5	5
1	1	5	5	5
1	1	5	5	5
1	1	5	5	5

$T = 2$

1	1	5	7	5
1	1	5	7	7
1	1	5	5	5
3	1	5	5	5
3	3	5	5	5

$T = 7$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1



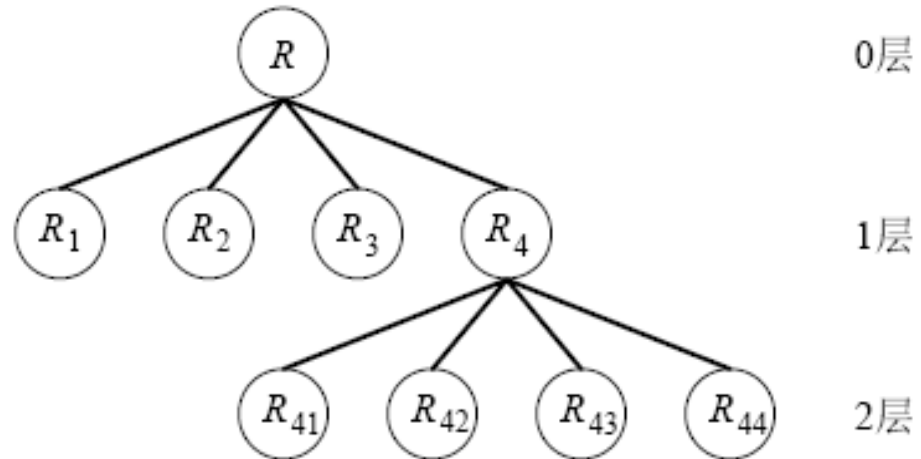
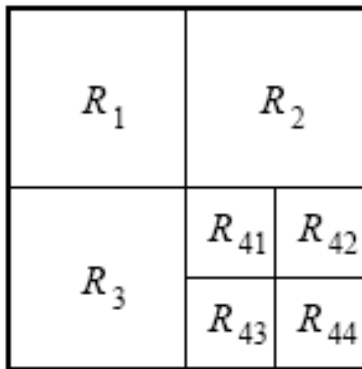
图像分割

- 图像分割定义
- 阈值分割
- 区域生长法
- **分裂合并方法**
- 分水岭算法
- 聚类分割算法
- 主动轮廓分割
- Graph Cut

分裂合并

□ 主要步骤

- 先把图像分成任意大小且不重叠的区域
- 然后再（根据准则）合并或分裂这些区域
- 迭代进行直到实现分割



图像的四叉树表达法



分裂合并

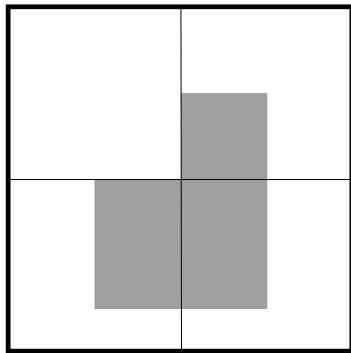
- 令 R 代表整个图像区域， P 代表逻辑谓词
- 把 R 连续地分裂成越来越小的 $1/4$ 的正方形子区域 R_i ，并且始终使 $P(R_i) = \text{TRUE}$
 1. 对任一个区域 R_i ，如果 $P(R_i) = \text{FALSE}$ ，就将其分裂成不重叠的四等分
 2. 对相邻的两个区域 R_i 和 R_j ，如果 $P(R_i \cup R_j) = \text{TRUE}$ ，就将它们合并起来
 3. 如果进一步的分裂或合并都不可能了，则结束

分裂合并

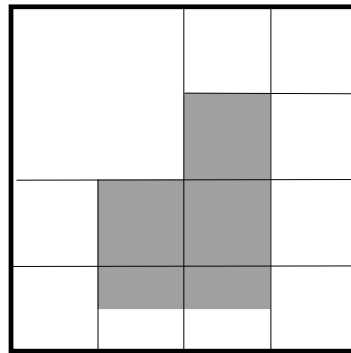
示例（四叉树）：分裂

分裂

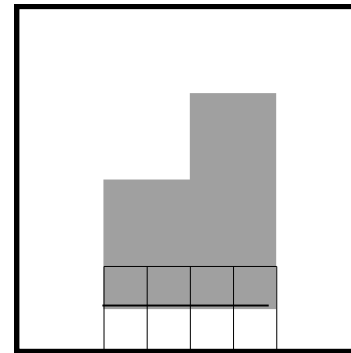
合并



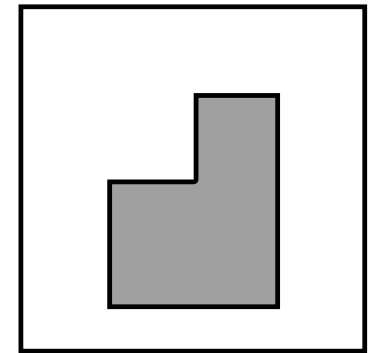
(a)



(b)



(c)



(d)



分—合图像分割方法

1. 确定**均匀性测度**，构造四分树结构
2. 选择初始分割层（一般为中间某一层）
3. 分裂处理
 - 从中间层开始，计算各块均匀性测度。对于均匀性测度为假的那些块，一分为四，重新编码。重复进行，直到各块的均匀性测度为真。
4. 合并处理
 - 从同一中间层开始，测试**同属于一个父节点的四块**，如果它们之并的均匀性测度为真，则合并这四块为一块。重复进行，直至不再存在可以合并的那些块。
5. 组合处理
 - 使用该数据编码判断位置，对相邻的大小不一，或者虽然大小一样，但不能合并为一个父节点的区域，进行均匀性测度测试，合并均匀测度度量为一对区域。反复重复这一运算，直到不再存在可以合并的区域。
6. 小区处理
 - 清除小区等整理工作。

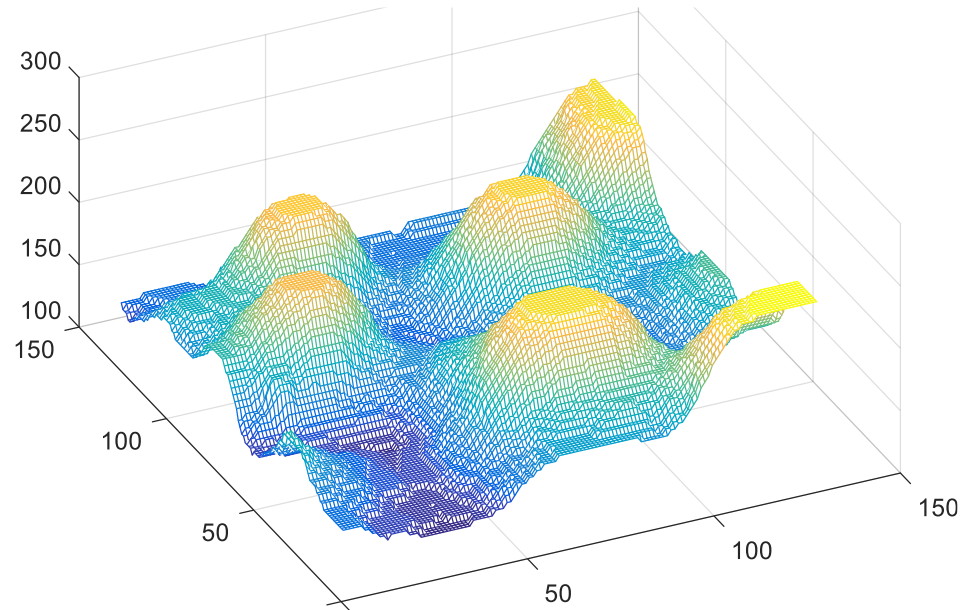
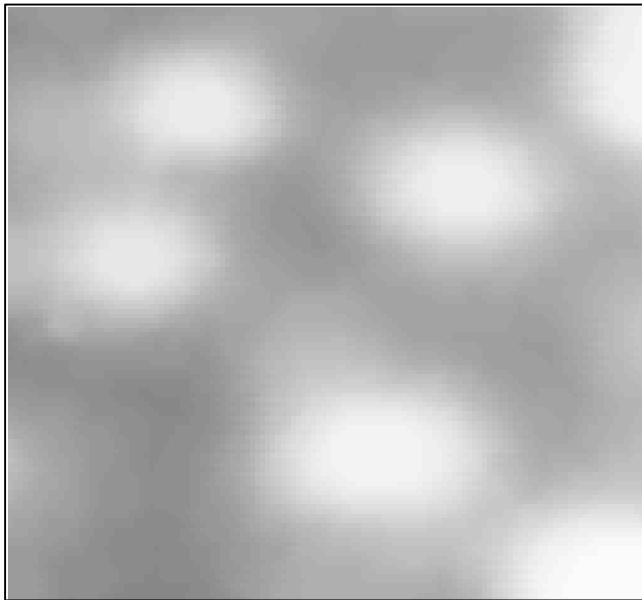


图像分割

- 图像分割定义
- 阈值分割
- 区域生长法
- 分裂合并方法
- 分水岭算法
- 聚类分割算法
- 主动轮廓分割
- Graph Cut

分水岭分割算法

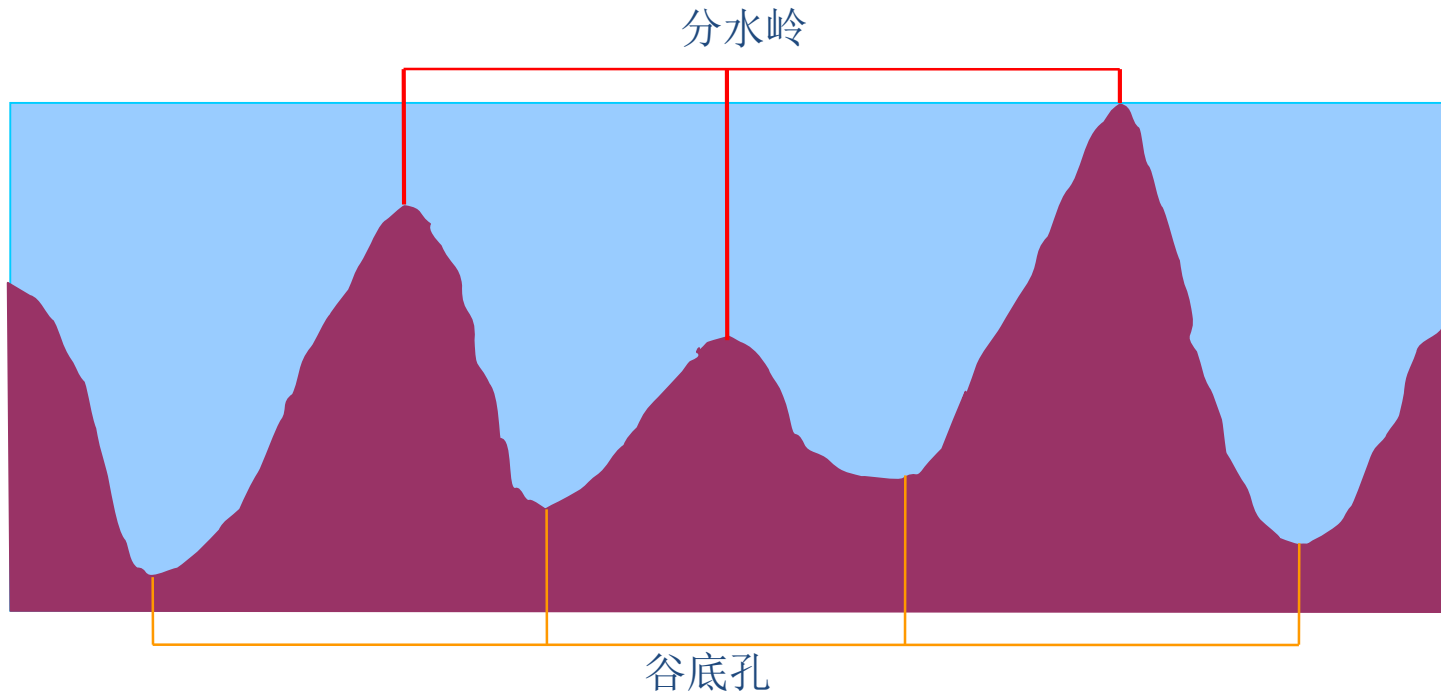
- 分水岭 (watershed, 也称分水线/水线)
- 把图象看成3-D地形的表示, 即2-D的地基 (对应图像空间) 加上第3维的高度 (对应图像灰度)
- 图像的**梯度图**也可以视为3D地形
 - 图像梯度图3D地形中的**峰岭**对应目标的边界



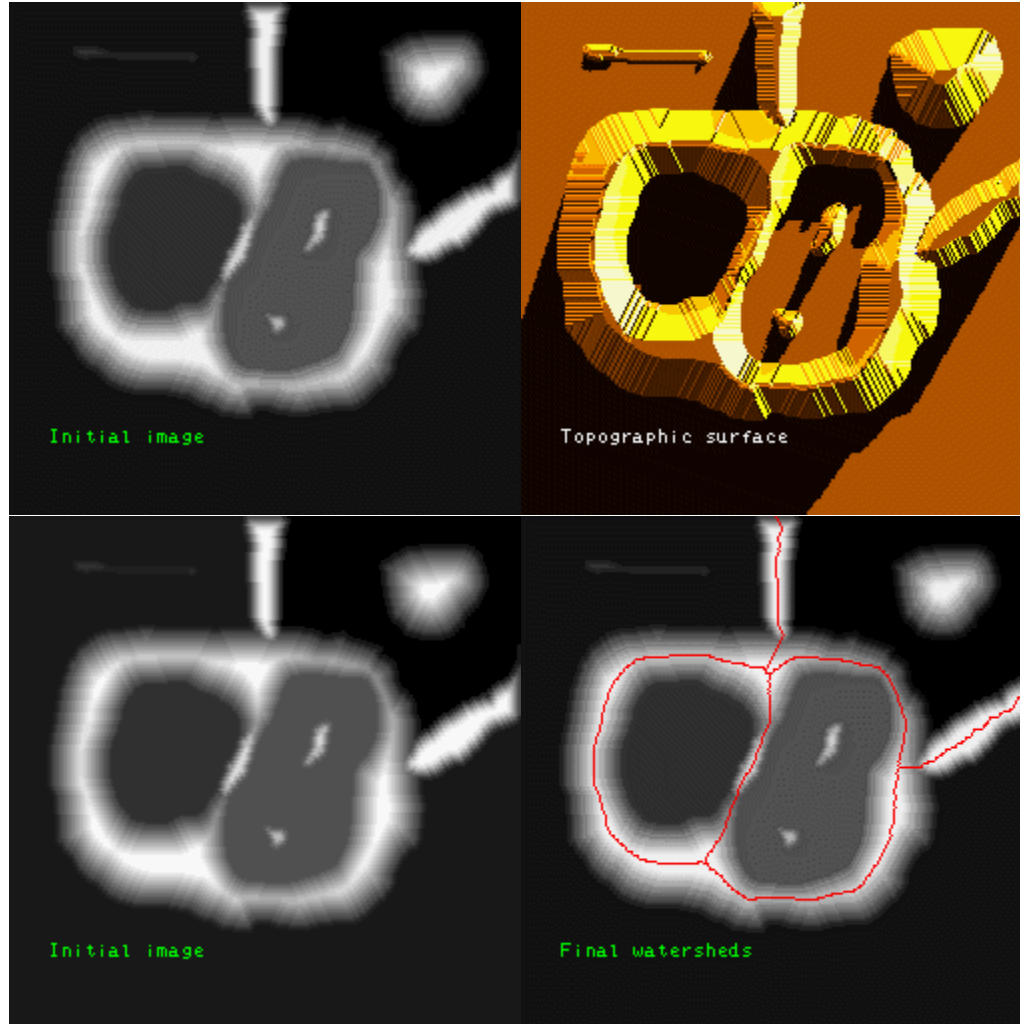
基本原理和步骤

□ 分水岭算法原理

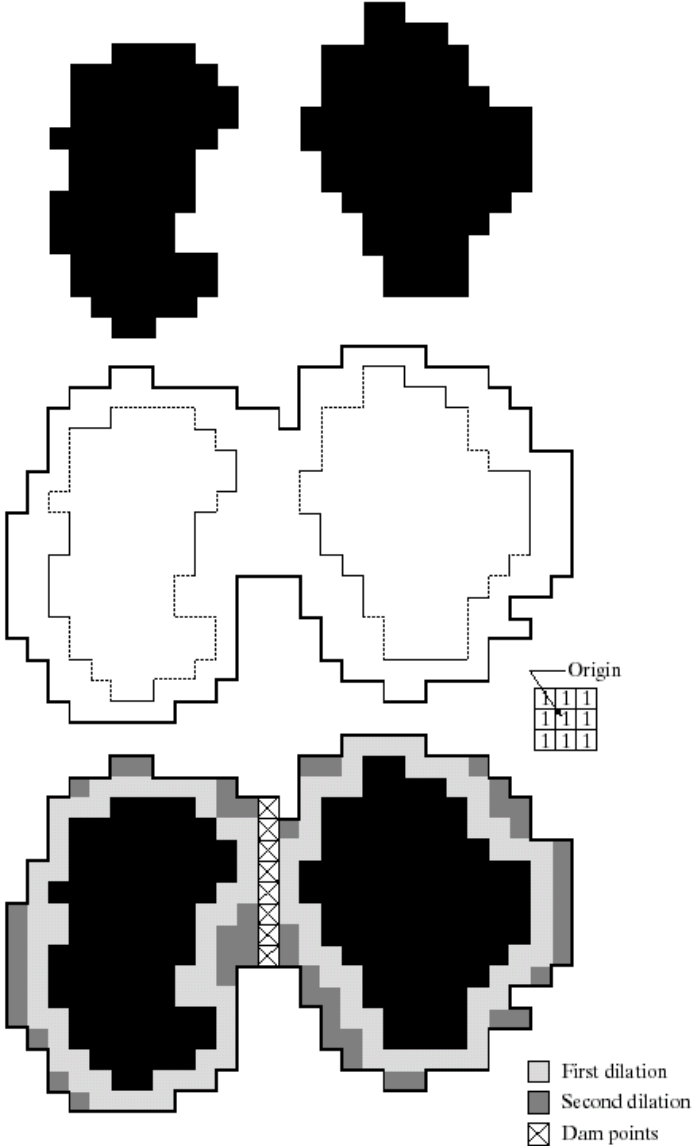
- 下图横轴表示像素坐标，纵轴表示像素灰度的**梯度幅值**
- 建立不同目标间的分水岭



分水岭算法过程演示



分水岭算法-构筑水坝





基本原理和步骤

□ 分水岭计算步骤

1. 待分割图像 $f(x, y)$ ，其**梯度图像**为 $g(x, y)$
2. 用 M_1, M_2, \dots, M_R 表示 $g(x, y)$ 中各**局部极小值**的像素位置， $C(M_i)$ 为与 M_i 对应的区域中的像素坐标集合（**即期望的最终分割结果**）。
3. 用 n 表示当前梯度阈值， $T[n]$ 代表记为 (u, v) 的像素集合，要求 $g(u, v) < n$ ； $T[n]$ 可对应多个联通体：

$$T[n] = \{(u, v) \mid g(u, v) < n\}$$

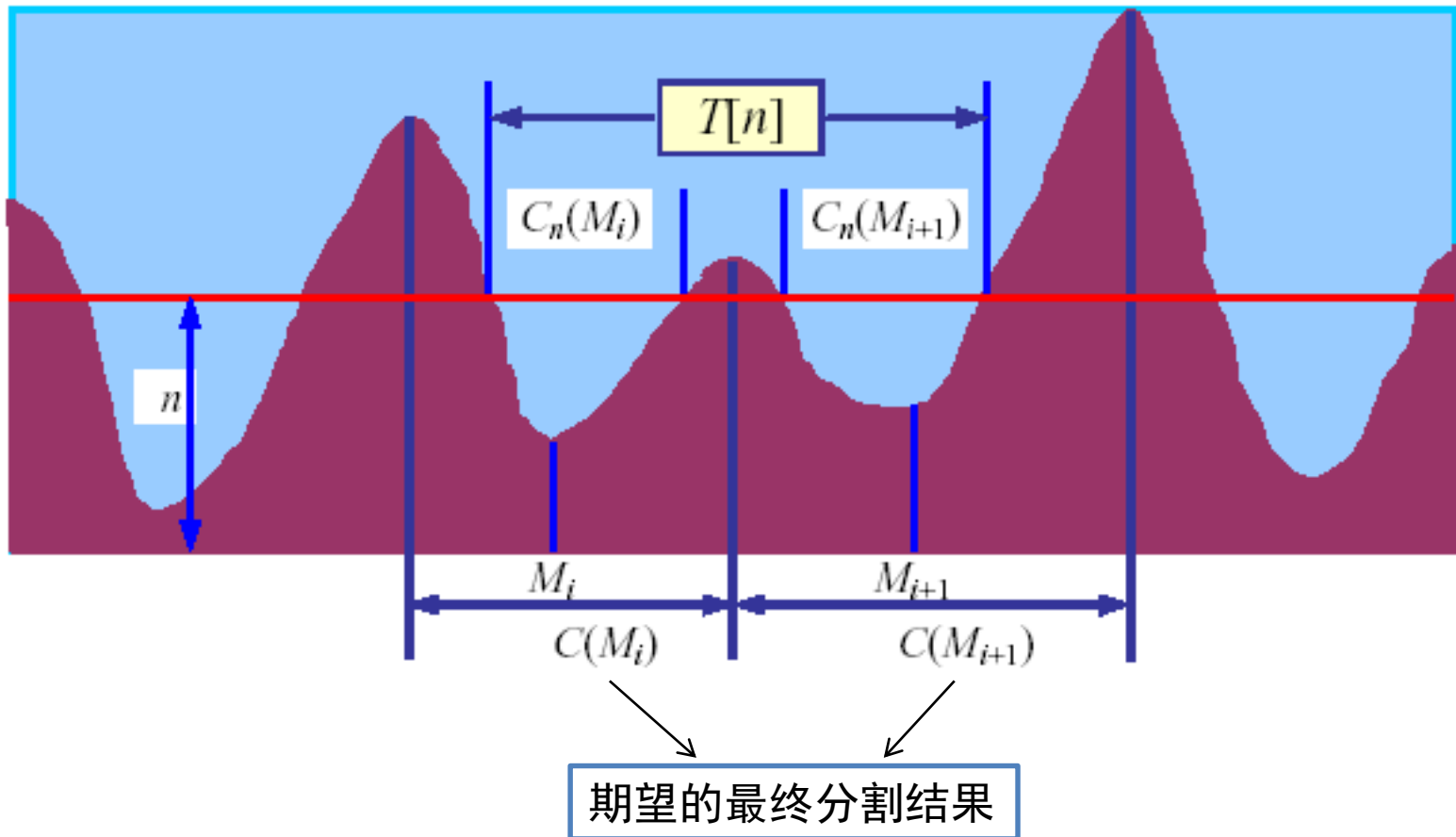
4. 对 M_i 所在的区域，其中满足条件的坐标集合 $C_n(M_i)$ 可看作一幅二值图像：

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

基本原理和步骤

□ 分水岭计算步骤

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





基本原理和步骤

□ 分水岭计算步骤

- 用 $C[n]$ 代表：在梯度阈值为 n 时，图象中所有满足条件的象素

$$C[n] = \bigcup_{i=1}^R C_n(M_i)$$

- $C[\max + 1]$ 将是所有区域的并集

$$C[\max + 1] = \bigcup_{i=1}^R C_{\max+1}(M_i)$$

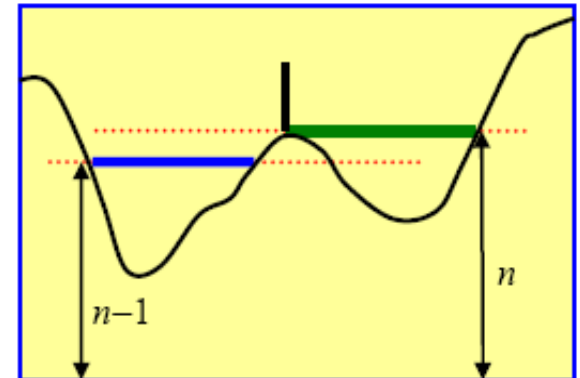
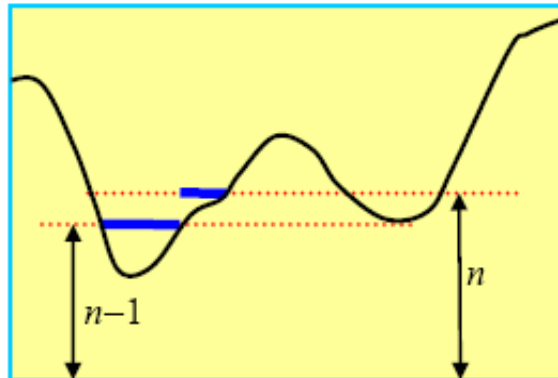
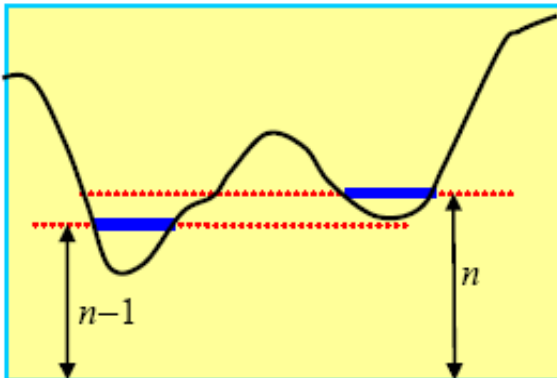
- $C[n - 1]$ 是 $C[n]$ 的子集，也是 $T[n]$ 的子集

基本原理和步骤

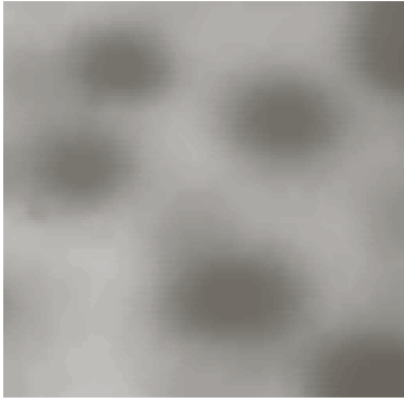
□ 分水岭计算步骤

■ 令 S 代表 $T[n]$ 中的连通组元集合，对每个连通组元 $s \in S[n]$ ，有3种可能性：

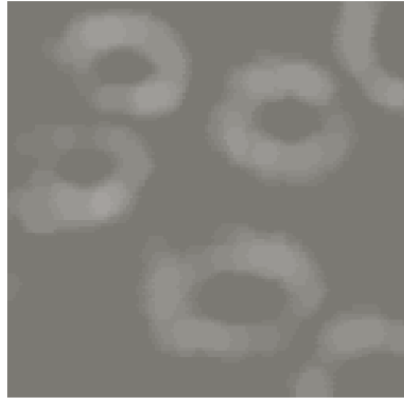
1. $s \cap C[n-1]$ 是1个空集
✓ $C[n]$ 可由把连通组元 s 加到 $C[n-1]$ 中得到
2. $s \cap C[n-1]$ 里包含 $C[n-1]$ 中的一个连通组元
✓ $C[n]$ 可由把连通组元 s 加到 $C[n-1]$ 中得到
3. $s \cap C[n-1]$ 里包含 $C[n-1]$ 中一个以上的连通组元
✓ 需要在 s 中建分水岭



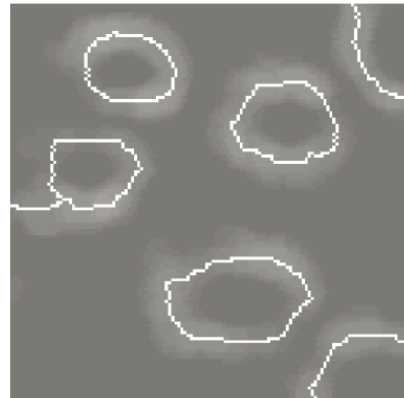
分水岭分割算法实例



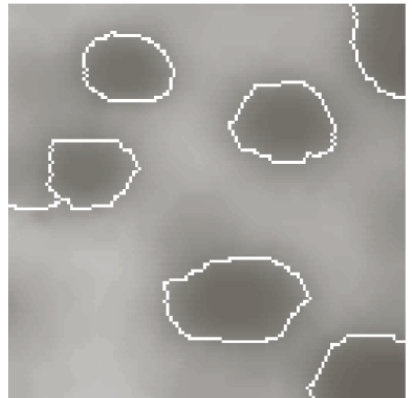
原始图



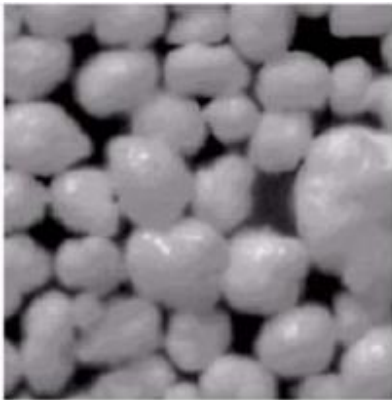
梯度图



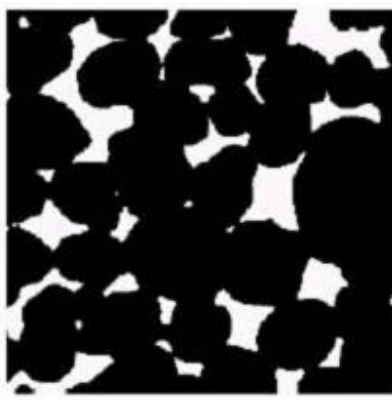
梯度图上的分水岭



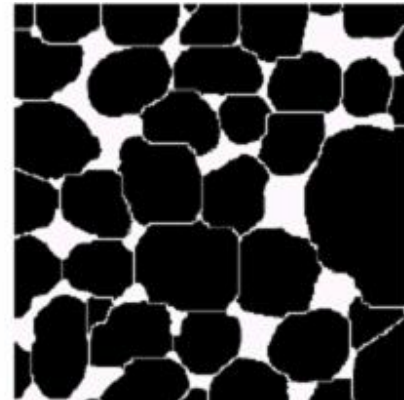
原图上的分水岭



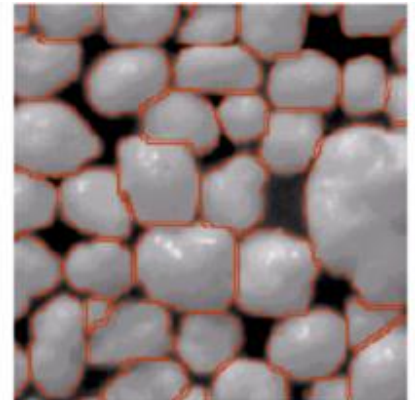
原始图



阈值分割



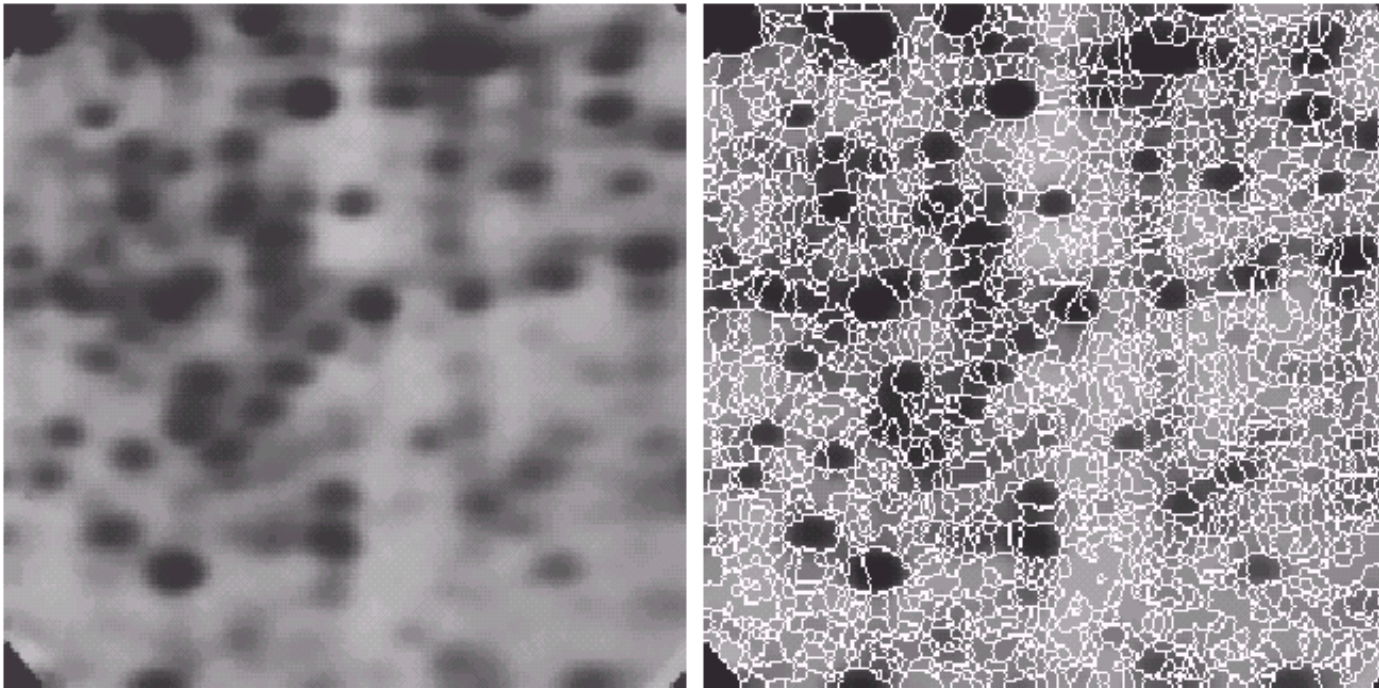
分水岭



叠加轮廓

算法改进

- 过分割 (over-segmentation)
 - 分水岭算法依赖局部极小值的种子点



a b

FIGURE 10.47
(a) Electrophoresis image. (b) Result of applying the watershed segmentation algorithm to the gradient image. Oversegmentation is evident. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)



算法改进

□ 利用标记控制分割

■ 过分割（over-segmentation）

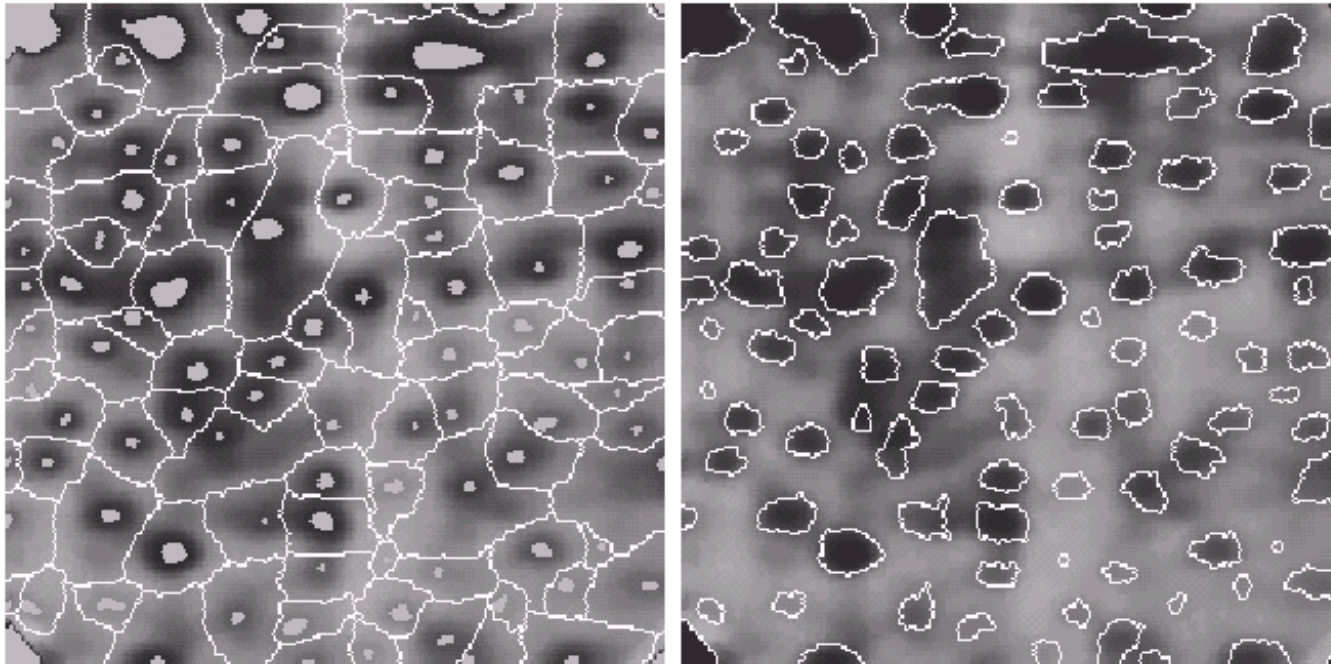
- ✓ 受图像中**噪声**和其它**不规则结构**影响

■ 利用标记（marker）

- ✓ 图象中的一个连通组元（基于灰度和连通性）
 - 内部标记：对应目标
 - 外部标记：对应背景（分水岭）

算法改进

- 利用标记控制分割
 - 限定允许的分割区域数目



a b

FIGURE 10.48
(a) Image showing internal markers (light gray regions) and external markers (watershed lines).
(b) Result of segmentation. Note the improvement over Fig. 10.47(b).
(Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)



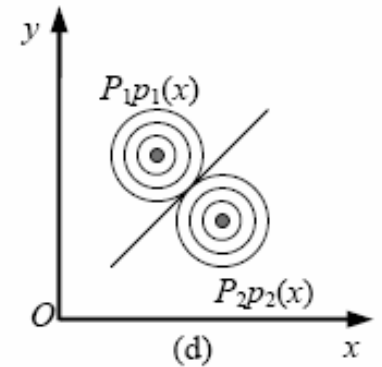
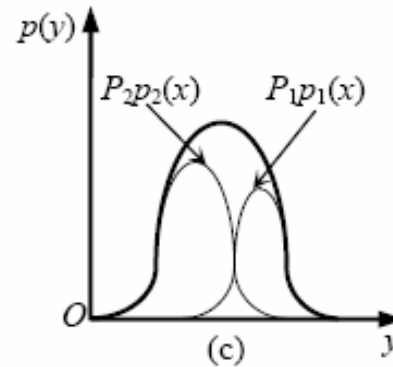
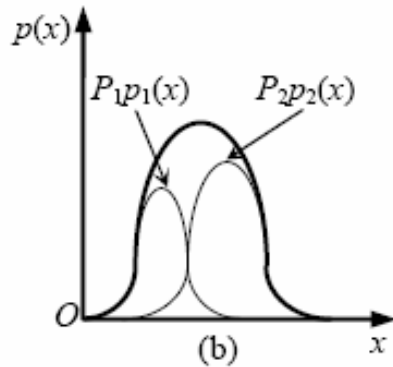
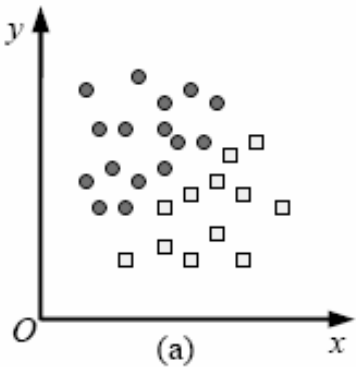
图像分割

- 图像分割定义
- 阈值分割
- 区域生长法
- 分裂合并方法
- 分水岭算法
- **聚类分割算法**
- 主动轮廓分割
- Graph Cut

聚类分割方法

□ 聚类分割：

- 利用图像特征分布特性，通过将像素分簇聚类，实现图像分割
- 视每个像素为一个样本，其表达信息可包括灰度、坐标位置、颜色、纹理，或其他特征
- 灰度聚类
 - ✓ 取阈值，1-D聚类
- 高维特征空间聚类
 - ✓ 区分能力较强



高维空间聚类的优点



聚类分割算法

- K-means聚类算法
- AP聚类算法
- 彩色图像的分割



K-means 聚类算法

基本步骤：

1. 选择 K 个初始分类中心 $\{\mu_1, \dots, \mu_K\}$ ，每个分类中心代表一个类簇；
2. 使用最小距离法将所有样本分类，即若 $\forall j \neq i, Dist(x, \mu_i) < Dist(x, \mu_j)$ ，则将 x 分为第 i 类；
3. 根据第2步的分类结果，重新计算各类中心，并将此作为各类新的中心；
4. 反复进行2、3步，直到各类中心趋于稳定。

问题

1. 如何确定 K 值？
 - ✓ 人为指定？自适应确定？
2. 如何初始化聚类中心？
 - ✓ 随机选样本作为聚类中心
3. 如何定义距离测度？
 - ✓ 欧式距离、汉明距离、……



K-means 聚类形式化分析

- 给定一个样本集 $\{x_1, \dots, x_N\}$, 其中 $x_i \in R^D$
- 目标: 将样本集划分为 K 个簇, 表示为 $\{\mu_1, \dots, \mu_K\}$
 - 划分关系 $r_{n,k} = 1$: 第 n 个样本被划分到第 k 簇(类)

- 优化目标:

$$\arg \min_{\substack{\{\mu_1, \dots, \mu_K\} \\ \{r_{1,1}, \dots, r_{NK}\}}} J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

计算复杂度?

- 方法

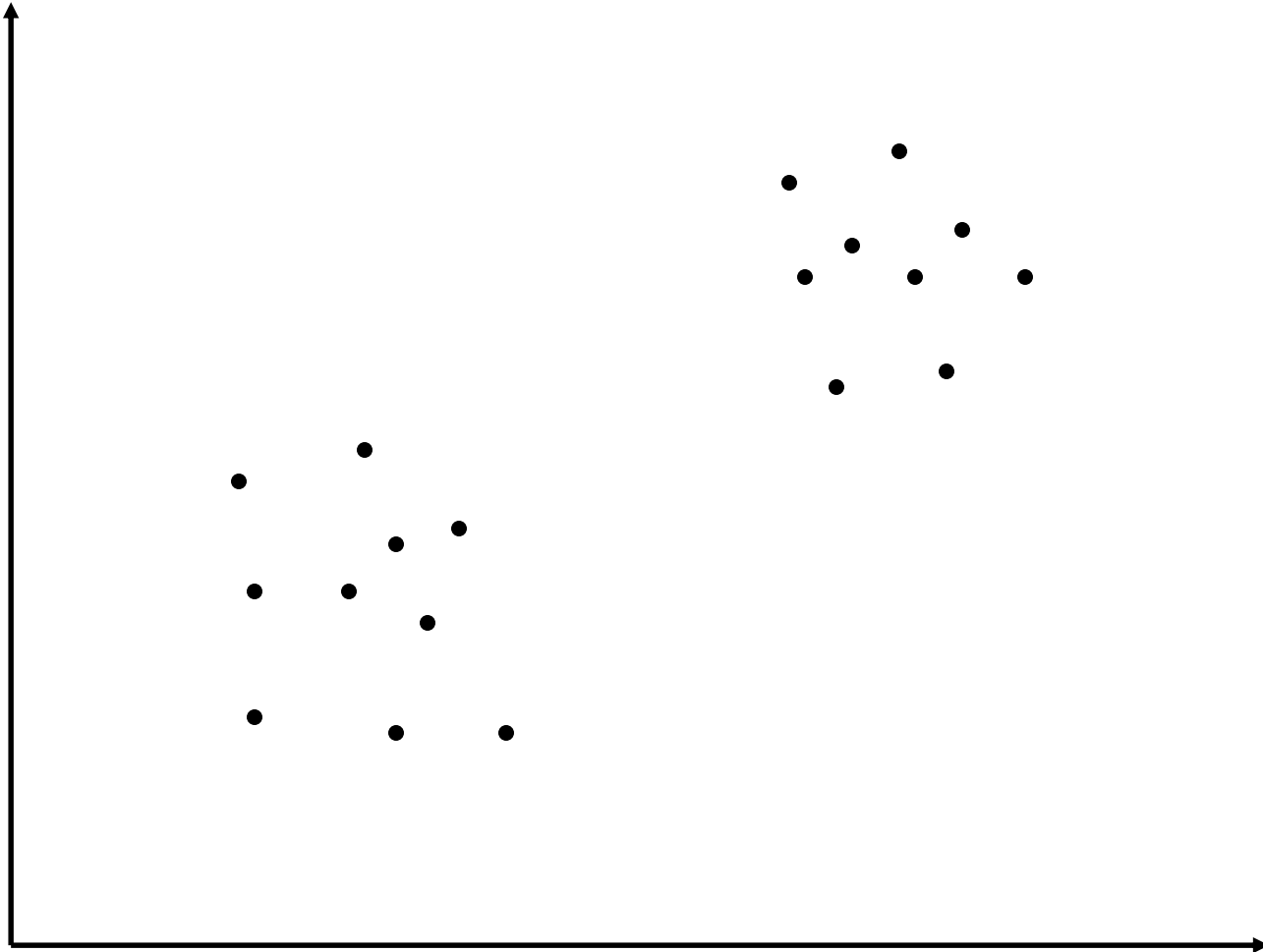
- 固定 $\{\boldsymbol{\mu}_k\}$, 解 $\{r_{nk}\}$:
$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\| \\ 0 & \text{otherwise} \end{cases} \quad \sim O(NKD)$$

- 固定 $\{r_{nk}\}$, 解 $\{\boldsymbol{\mu}_k\}$:
$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}} \quad \sim O(ND)$$

- 迭代进行上面两步, 直至收敛

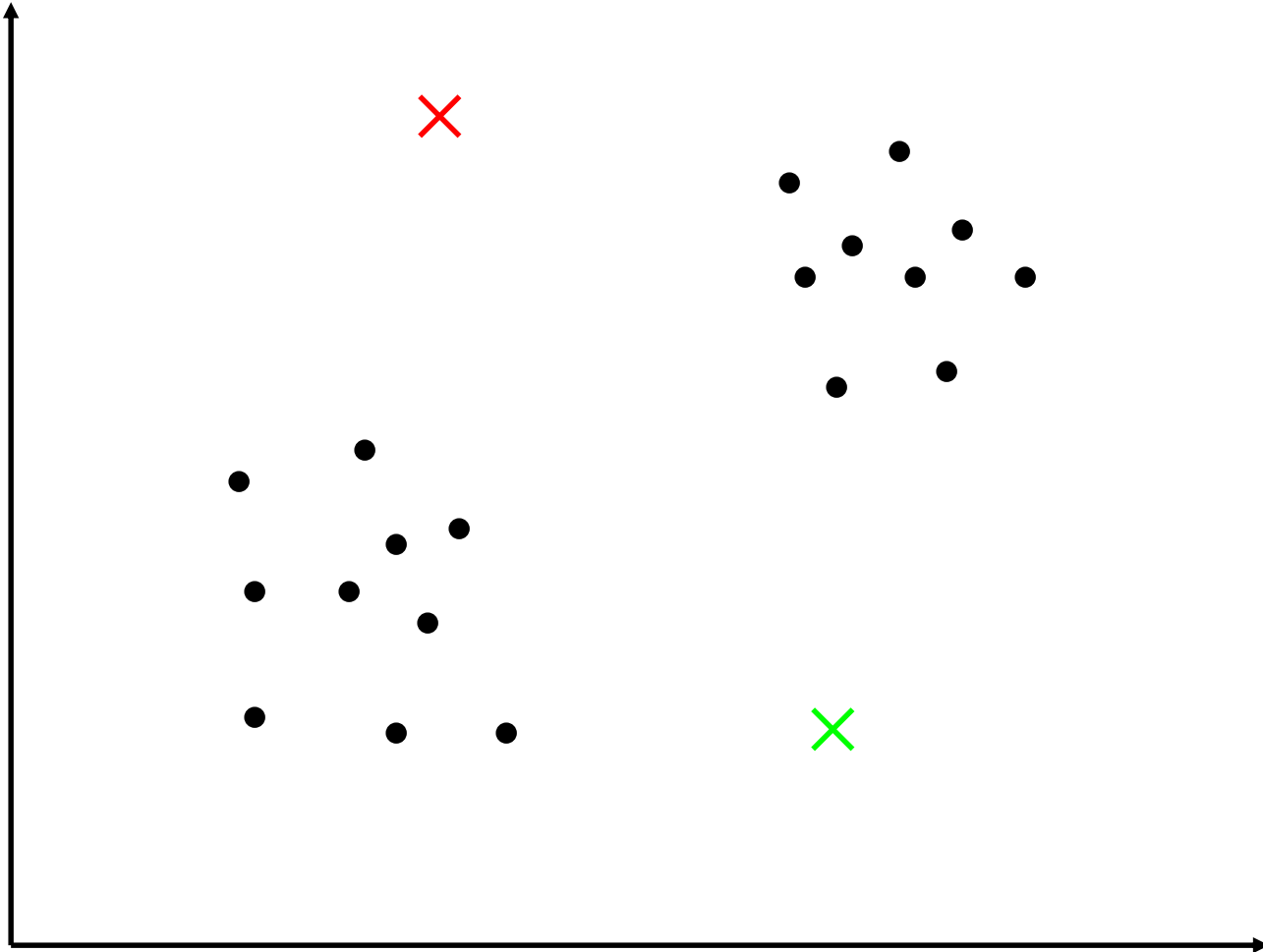


K-means 聚类过程演示 I



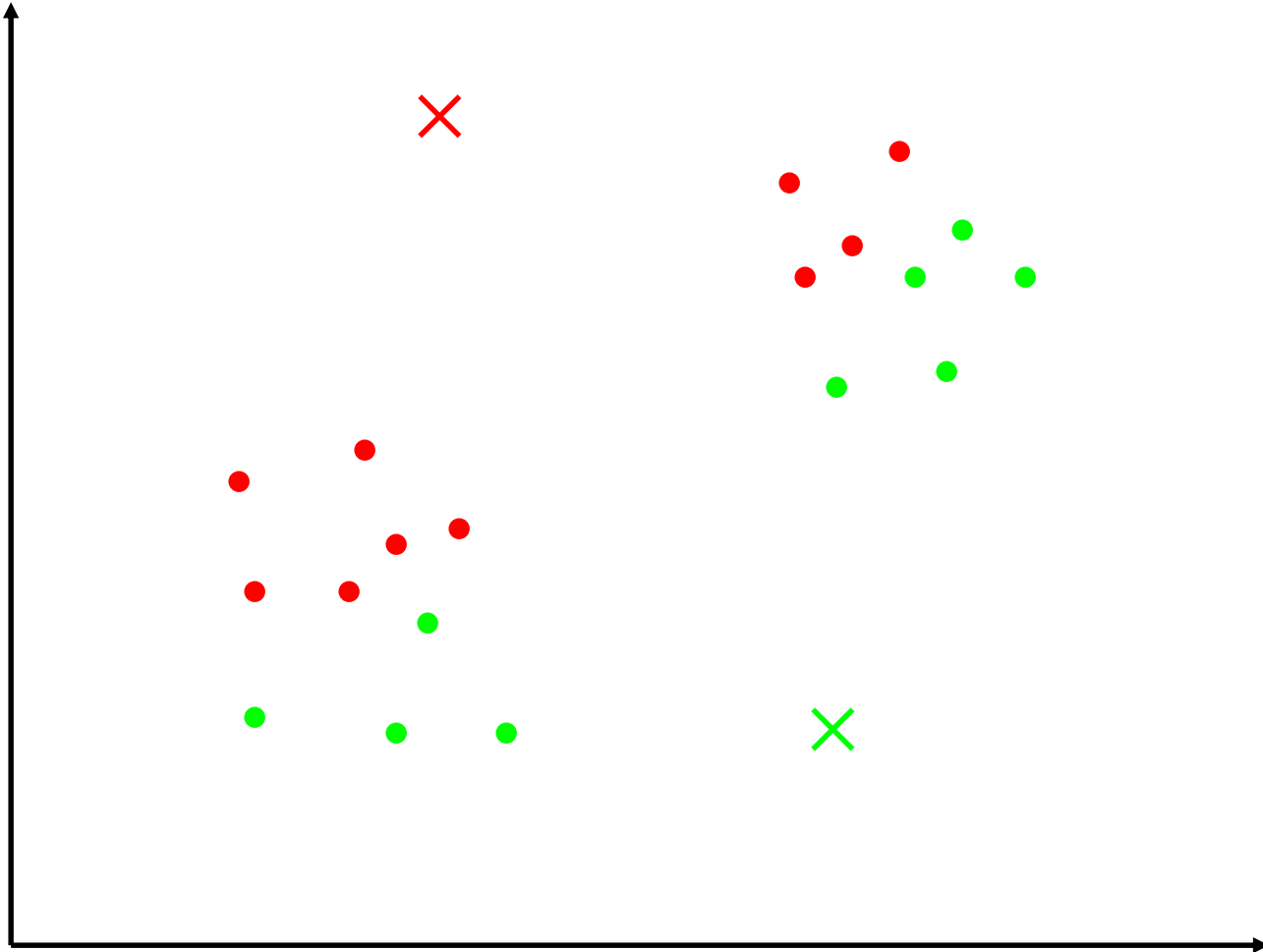


K-means 聚类过程演示 I



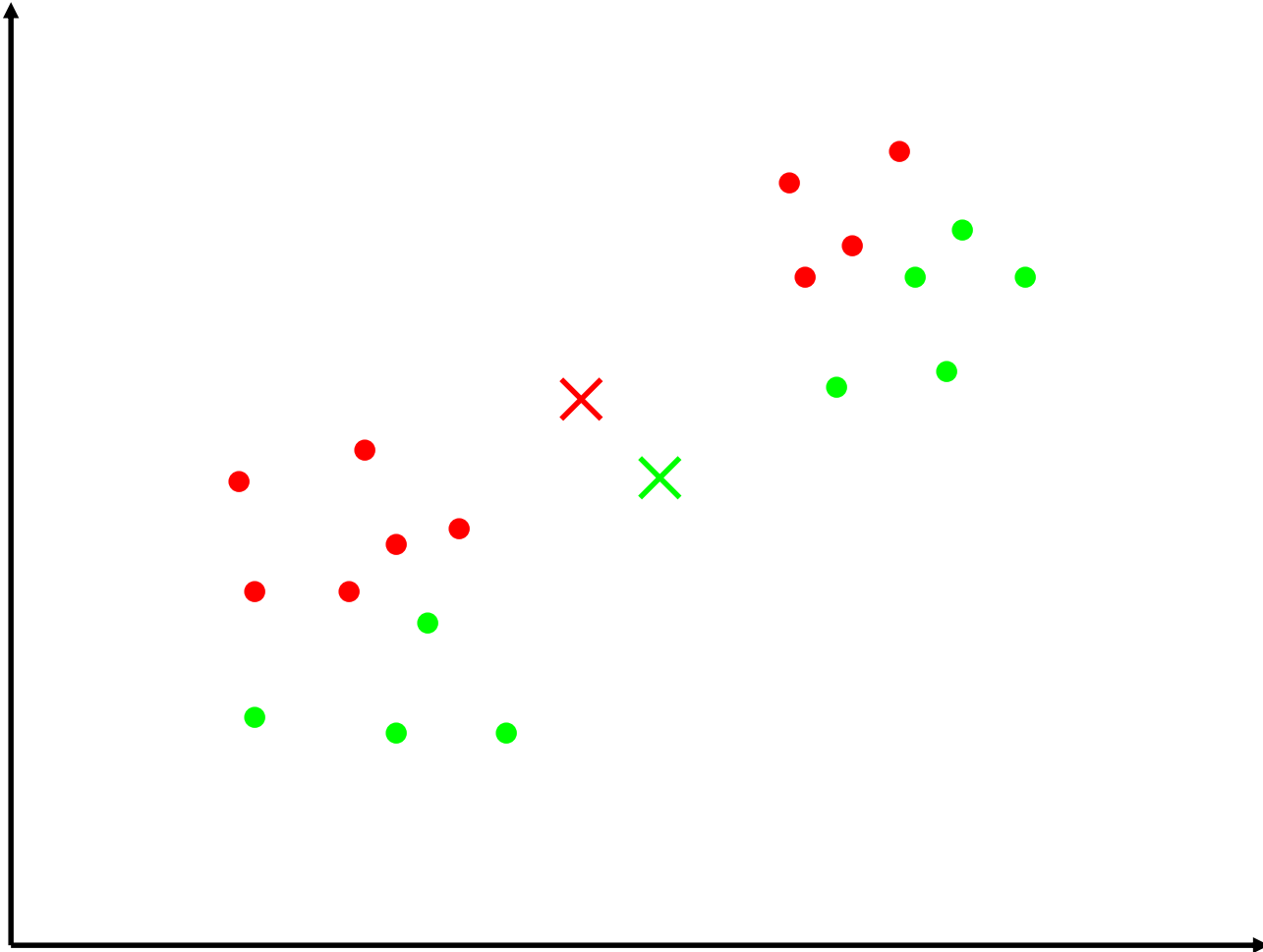


K-means 聚类过程演示 I



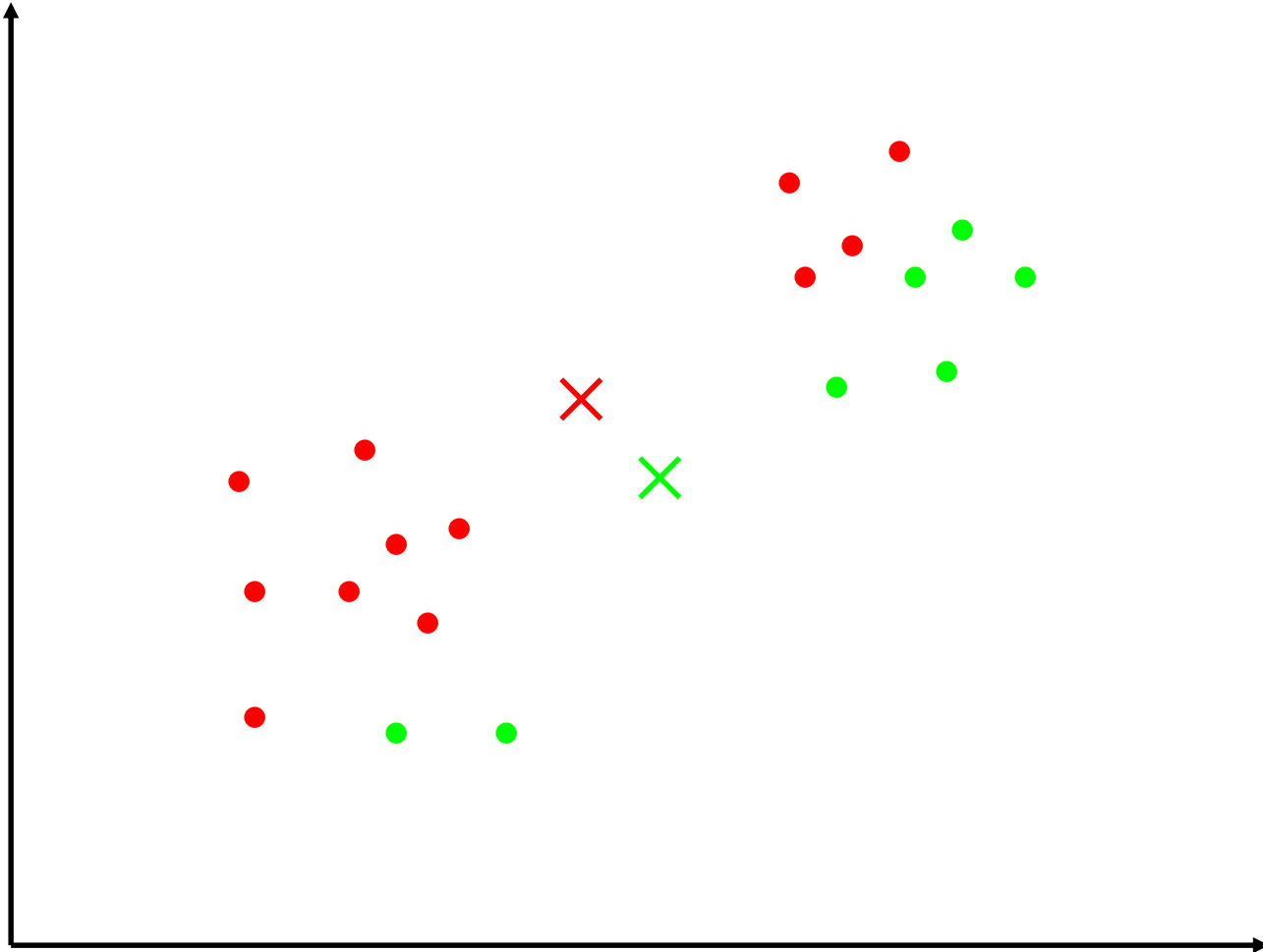


K-means 聚类过程演示 I



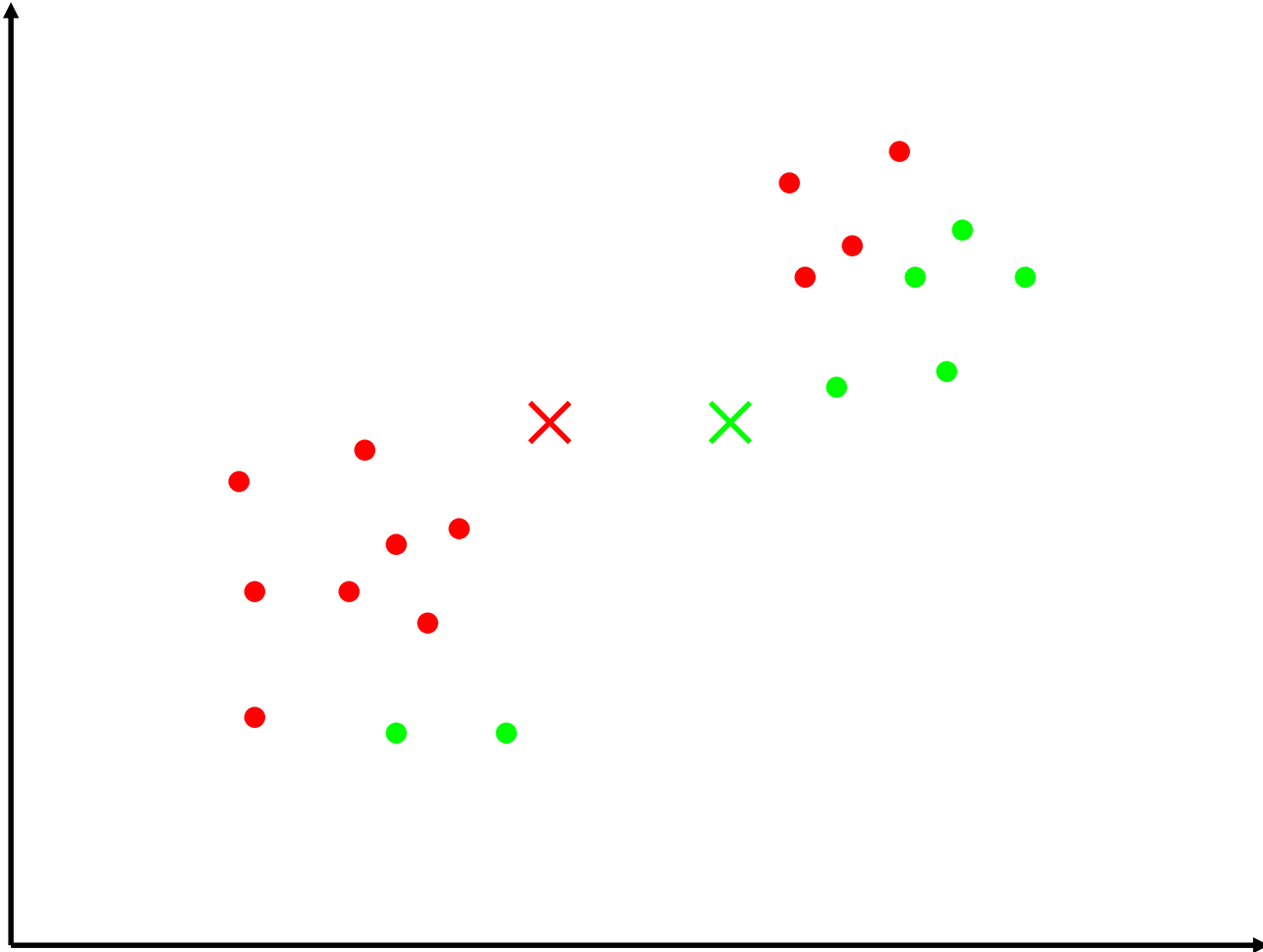


K-means 聚类过程演示 I



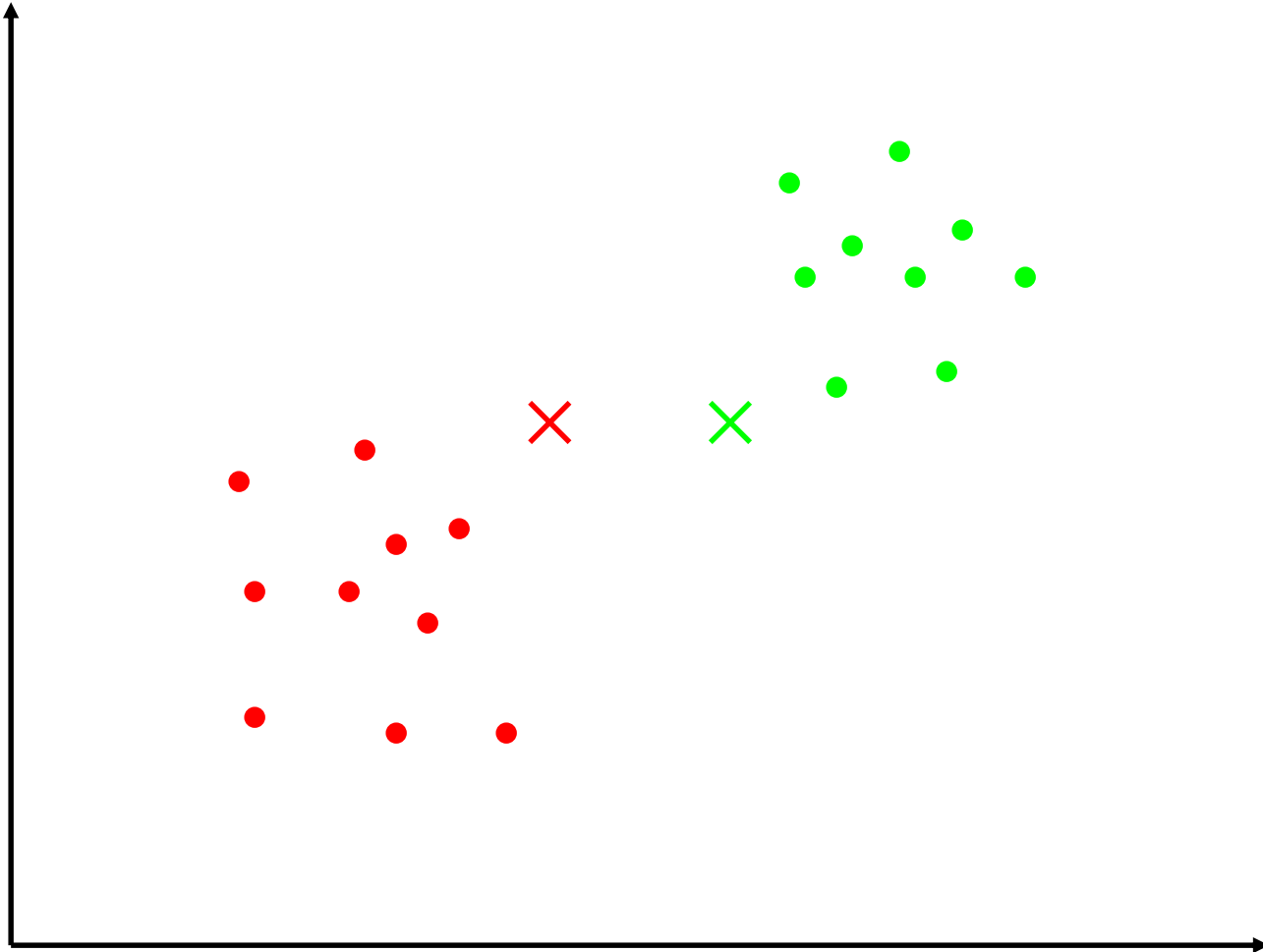


K-means 聚类过程演示 I



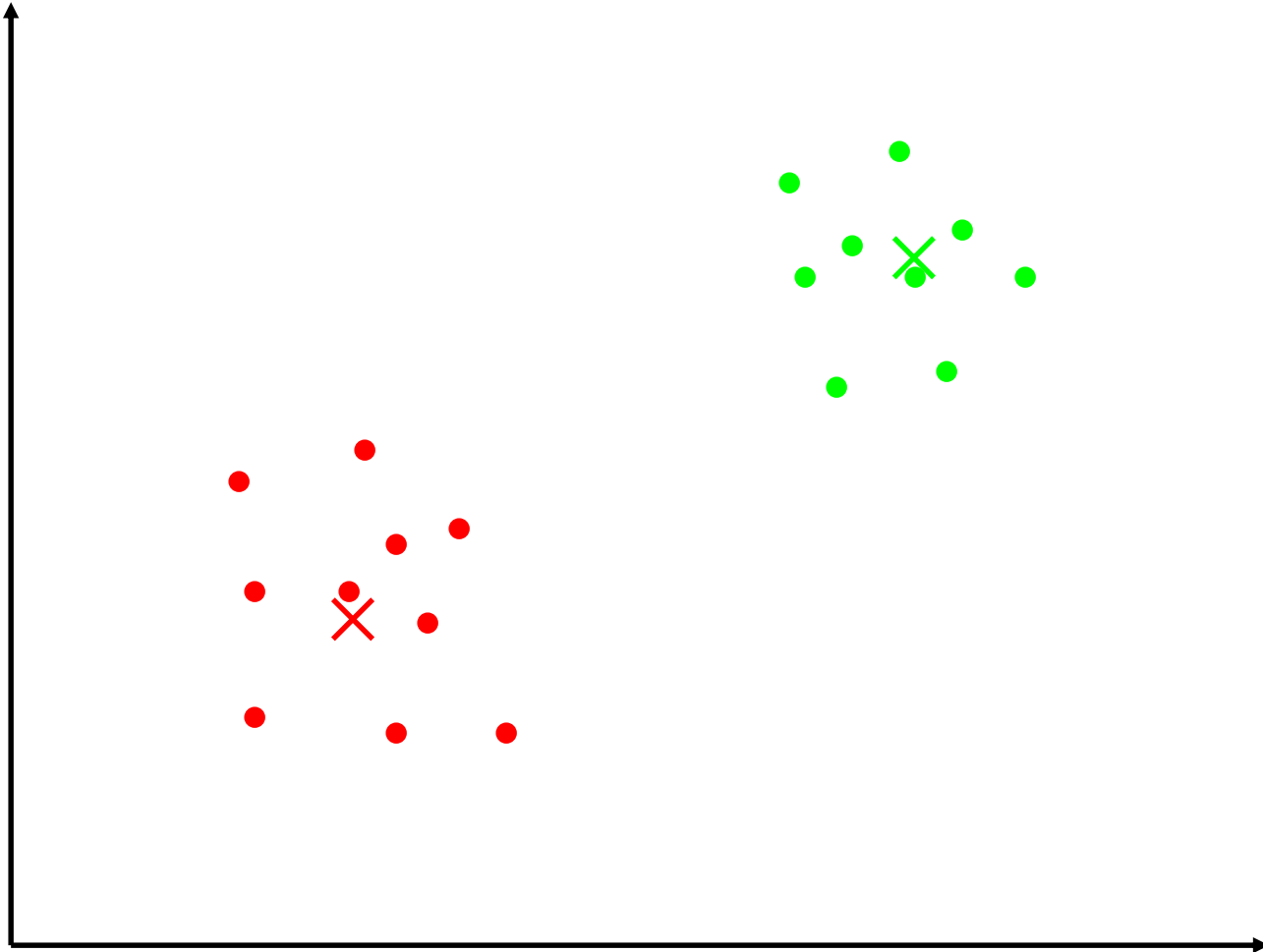


K-means 聚类过程演示 I



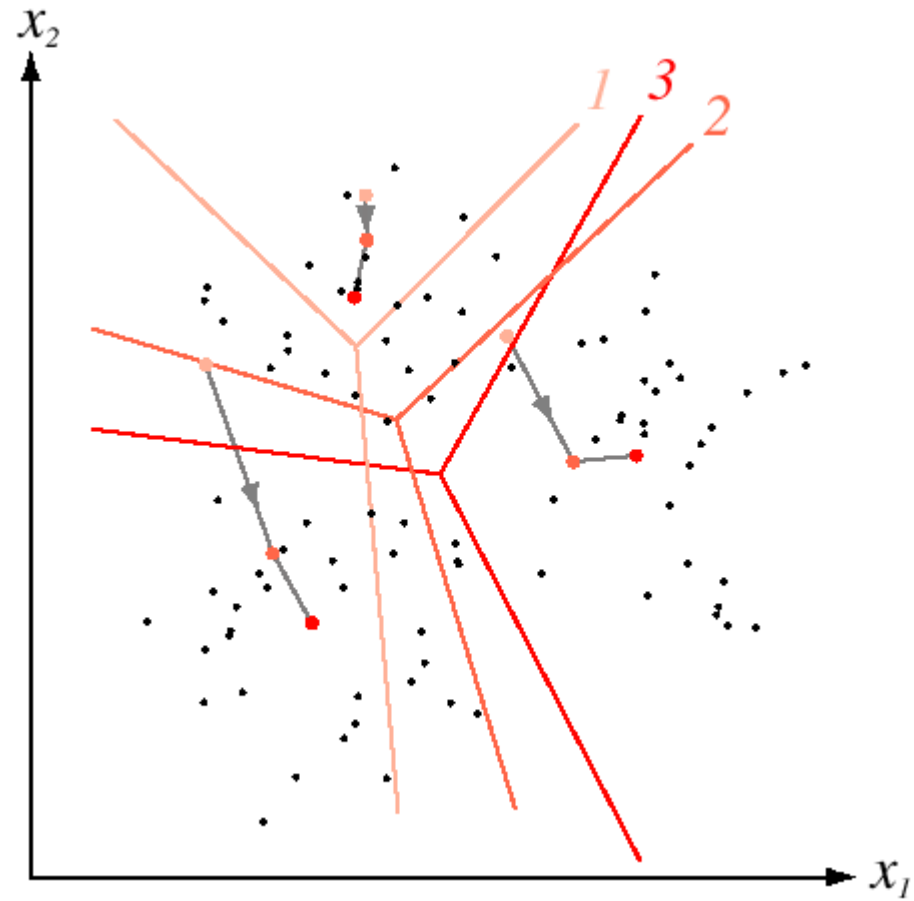


K-means 聚类过程演示 I



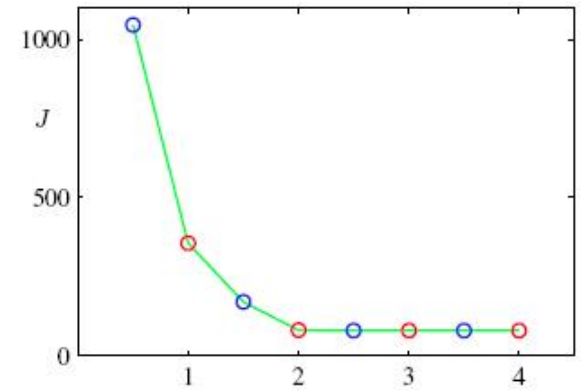
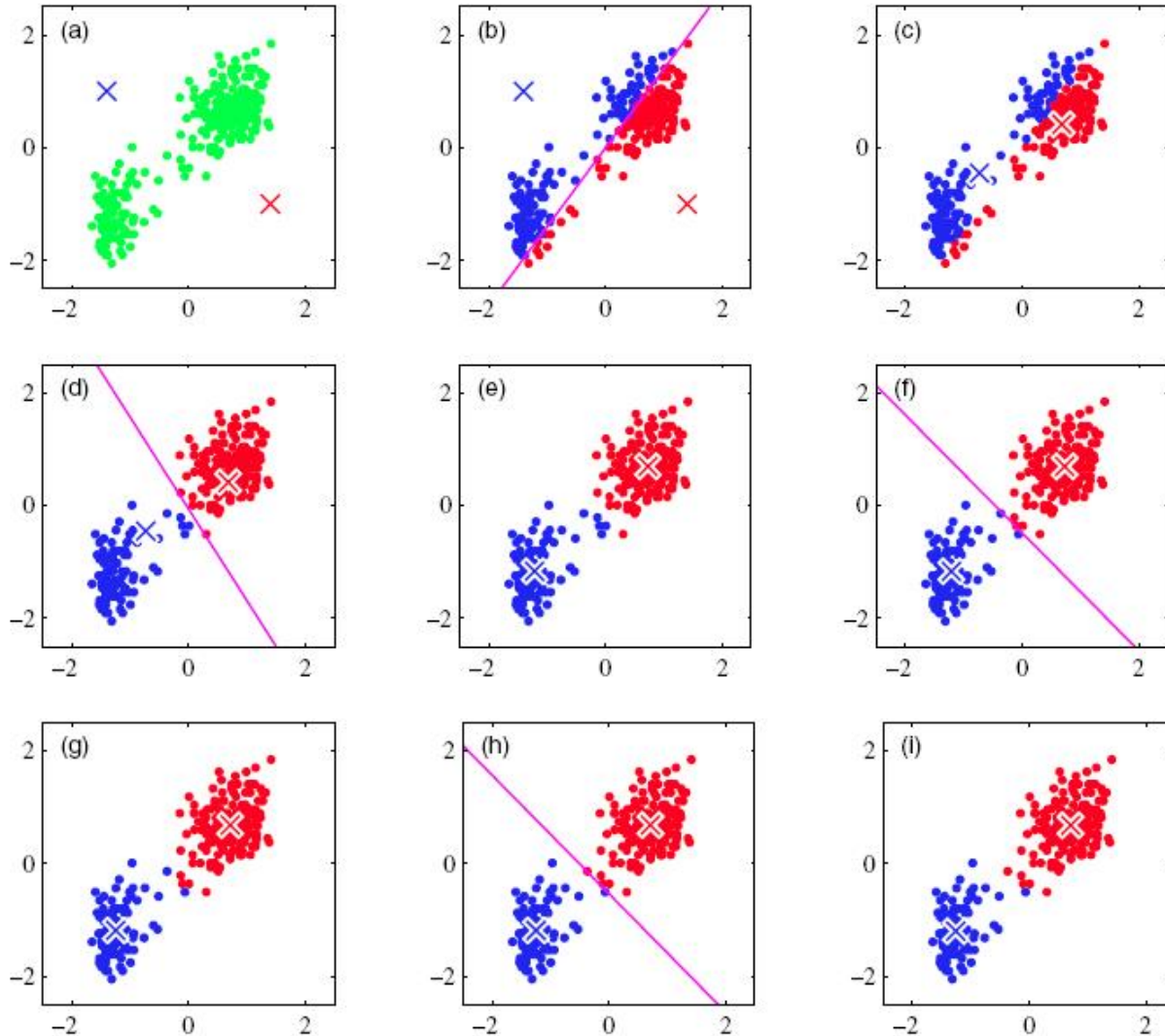


K-means 聚类过程演示 II





K-means algorithm 聚类过程演示

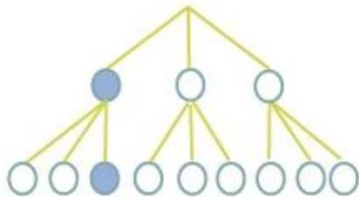


代价的值变化图。经过三次迭代，算法逐渐收敛。

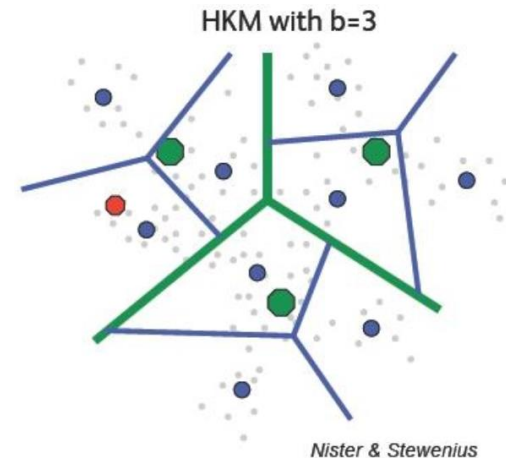
K-means聚类的缺点和应对方法

- 问题1: K-means需要指定聚类数目
 - 采用AP (Affinity Propagation, Science'07) 算法进行聚类
- 问题2: 计算复杂度高
 - 提高数据样本到簇中心的分配计算效率
 - 可采用KD-tree等近似最近邻查找算法, 如Approximate K-means算法
- 替代算法: Hierarchical K-means

▶ K-means tree of height h



- ▶ Branching factor b : $k = b^h$
- ▶ Assignment Complexity: $O(NDhb) = O(NDhk^{\frac{1}{h}})$





聚类分割算法

- K-means聚类算法
- AP聚类算法
- 彩色图像的分割



Affinity Propagation: AP聚类算法

□ 基本思想

- 假设类中心来自某些样本点
- 将全部样本看作一个图（graph）的节点，然后通过图（graph）中各条边的消息传递计算出各样本的聚类中心

□ 聚类过程中，共有两种消息在各节点间传递

- **吸引度** (responsibility)
- **归属度** (availability)。

□ AP算法通过迭代过程不断更新每一个点的吸引度和归属度值，直到产生m个高质量的exemplar（即类中心），同时将其余的数据点分配到相应的类簇中。

AP聚类算法过程

□ 假设 $\{x_1, x_2, \dots, x_n\}$ 为数据样本集，令 S 是一个刻画点之间相似矩阵

1. 算法初始，

(1) $s(i, k) = -\|x_i - x_k\|^2$ ， $s(k, k)$ 为矩阵 S 中元素值的中值或最小值

(2) 将吸引度矩阵 R 和 归属度矩阵 A 初始化为0矩阵

2. 更新吸引信息 (responsibility) 矩阵 R :

- ✓ $r(i, k)$ 刻画了候选中心点 k 对数据对象 i 的所有权的竞争，描述了数据对象 k 适合作为数据对象 i 的聚类中心的程度

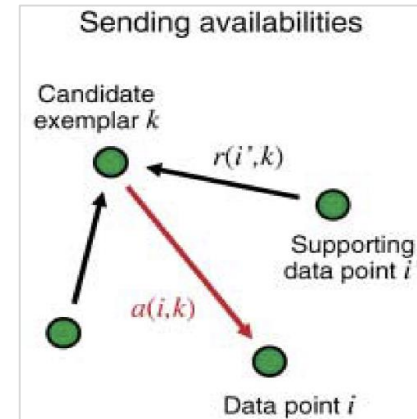
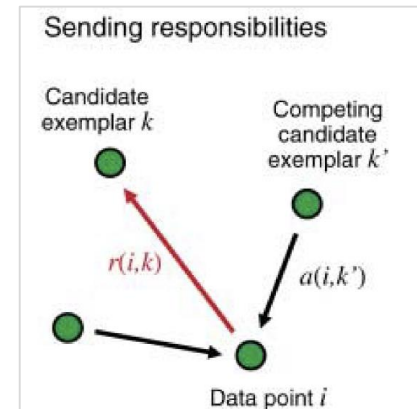
$$r(i, k) \leftarrow s(i, k) - \max_{k' \text{ s.t. } k' \neq k} \{a(i, k') + s(i, k')\}$$

3. 更新归属信息 (availability) 矩阵 A :

- $a(i, k)$ 描述了数据对象 i 收集“证据”，判断选择数据对象 k 作为其聚类中心的适合程度，表示从 k 到 i 的消息

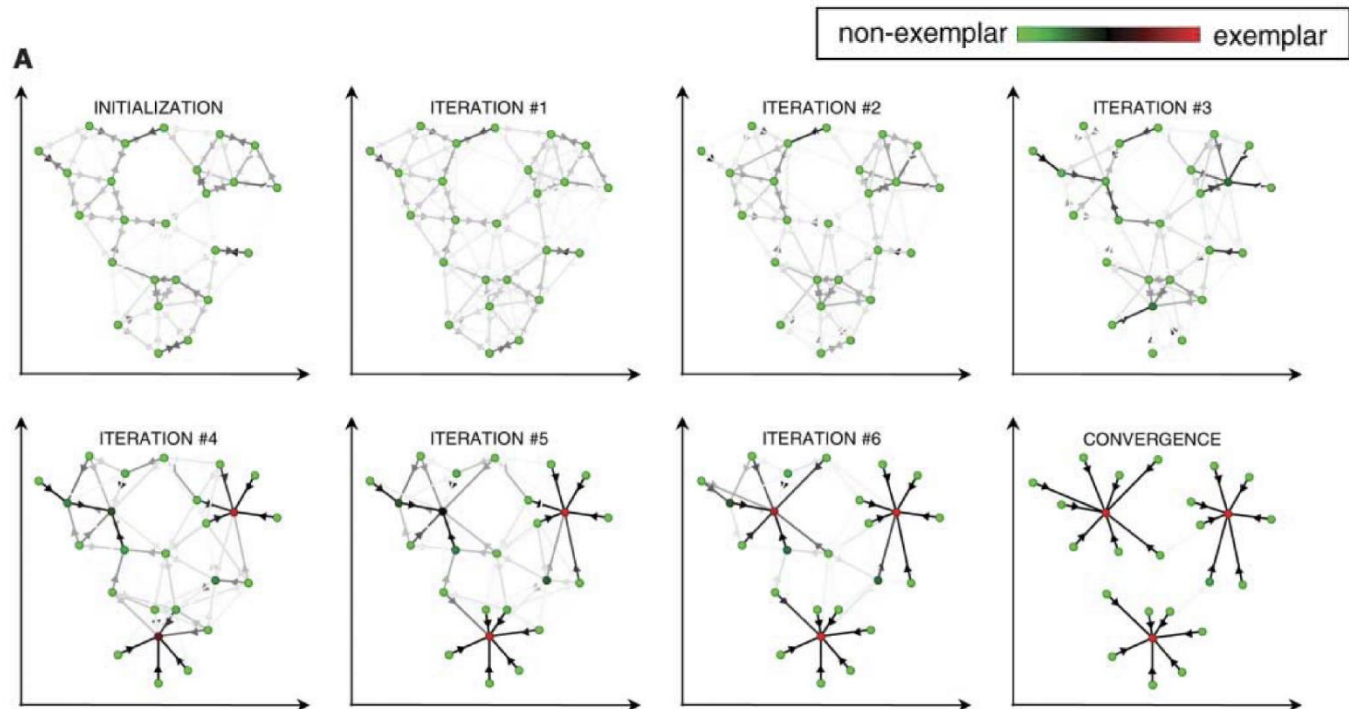
$$a(i, k) \leftarrow \min \left\{ 0, r(k, k) + \sum_{i' \text{ s.t. } i' \in \{i, k\}} \max \{0, r(i', k)\} \right\}$$

$$a(k, k) \leftarrow \sum_{i \text{ s.t. } i \neq k} \max \{0, r(i, k)\}$$



AP聚类算法过程和实例

- 假设 $\{x_1, x_2, \dots, x_n\}$ 为数据样本集，令 S 是一个刻画点之间相似度矩阵
4. 根据衰减系数 λ 对两个变量进行更新
$$r_{t+1}(i, k) = \lambda * r_t(i, k) + (1 - \lambda) * r_{t+1}(i, k)$$
$$a_{t+1}(i, k) = \lambda * a_t(i, k) + (1 - \lambda) * a_{t+1}(i, k)$$
 5. 重复步骤2,3,4直至矩阵稳定或者达到最大迭代次数，算法结束。最终取 $(a + r)$ 的第 i 行最大的 k 作为第 i 个样本的聚类中心。





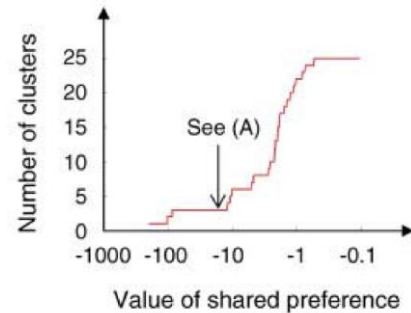
AP聚类算法

□ 优点

- 不需要事先指定聚类簇的个数
- 已有的数据点作为最终的聚类中心，而不是新生成一个簇中心
- 模型对数据的初始值不敏感
- 对初始相似度矩阵数据的对称性没有要求
- 相比与k-centers聚类方法，其结果的平方差误差较小

□ 缺点

- 虽然AP算法不用提前设置聚类中心的个数，但是需要事先设置参考度(preference value)，即 $s(k, k)$ 的值，其大小与聚类中心的个数正相关
- 由于AP算法每次迭代都需要更新每个数据点的吸引度值和归属感值，算法复杂度较高，在大数据量下运行时间较长





聚类分割算法

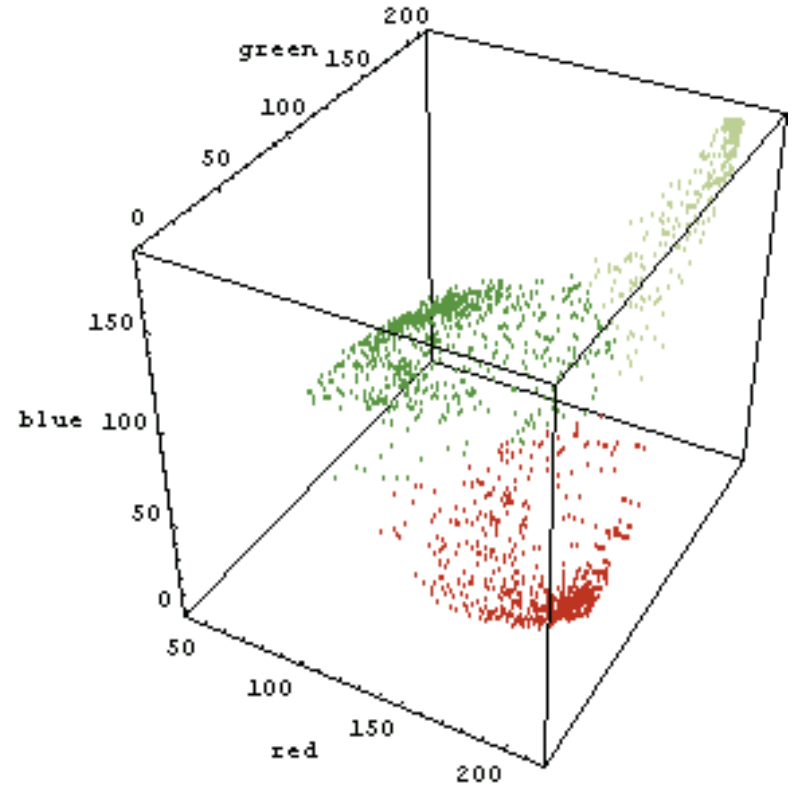
- K-means聚类算法
- AP聚类算法
- 彩色图像的分割

彩色图像的分割

□ 分割策略II—RGB空间K-Means聚类分割

K-means clustering minimizes

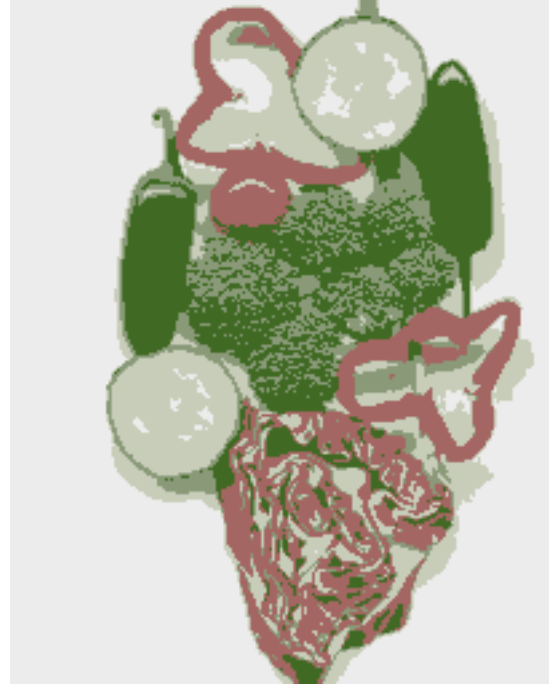
$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of } i\text{'th cluster}} \|x_j - \mu_i\|^2 \right\}$$



RGB空间K-Means聚类分割



Original

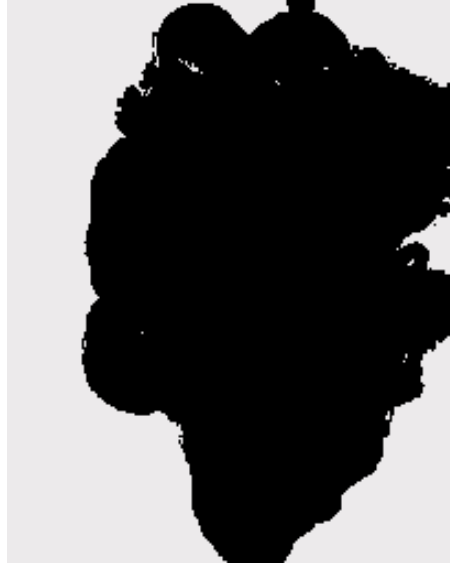


K=5



K=11

RGB空间K-Means聚类分割



只利用**RGB颜色信息**的分割聚类结果(K=20)。右图为其中的4个聚类结果。

- 同一聚类不一定连通。
- 没有纹理相似度量。
- 有些聚类并没有意义。



RGB空间K-Means聚类分割



同时使用色彩和空间位置的
K-means分割($k=20$)。右图为
其中的4个聚类结果。

即以 (R, G, B, x, y) 为特征向量。

- 大区域会分裂成小区域。





彩色图像的分割

□ 彩色空间

- 面向处理分析的颜色模型-HSI (hue, saturation, intensity)
- I 分量与图像的彩色信息无关
- H 和 S 分量与人感受颜色的方式是紧密相连的（颜色和浓淡）
- HSI 三分量之间的相关性比 RGB 三分量之间要小得多

□ 从RGB到HSI的转换关系

$$\left\{ \begin{array}{l} H = \cos^{-1} \left\{ \frac{(R-G) + (R-B)}{2\sqrt{(R-G)^2 + (R-B)(G-B)}} \right\} \quad R \neq G \text{ 或 } R \neq B \\ \quad \text{若 } B > G \quad H = 2\pi - H \\ S = 1 - \frac{3}{R+B+G} [\min(R, G, B)] \\ I = \frac{R+B+G}{3} \end{array} \right.$$

思考： 若一幅图像所有像素的R/G/B三通道的值都经历相同参数 (a, b) 的线性变换： $f(x) = a \cdot x + b$ ，变换前后的图像H通道是否有变化？

彩色图像的分割

□ 彩色空间

- 色度 H 的区分力与饱和度 S 大小成**正比**。
- 当饱和度 S 值为0时，色度 H 无意义，图像变为灰度图像。

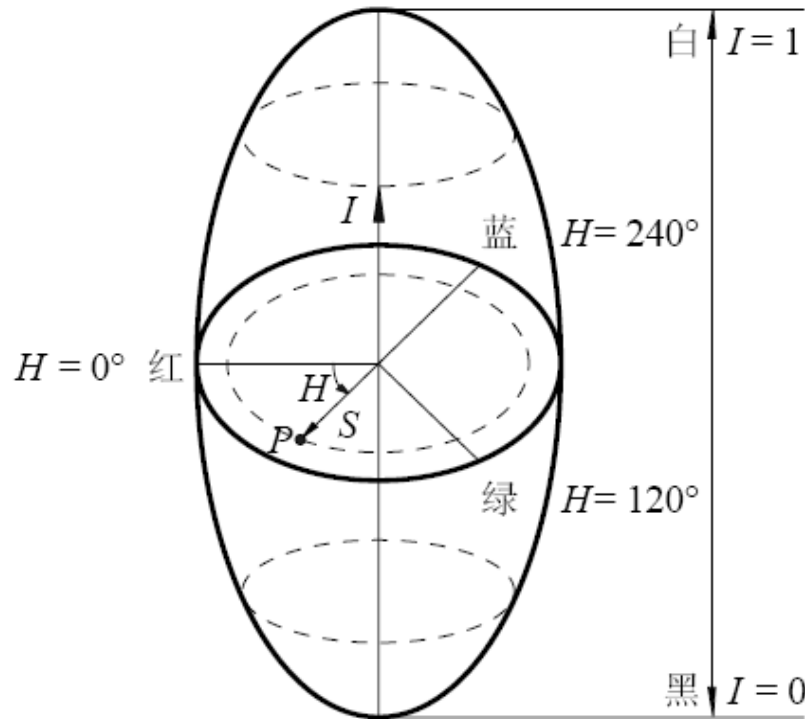
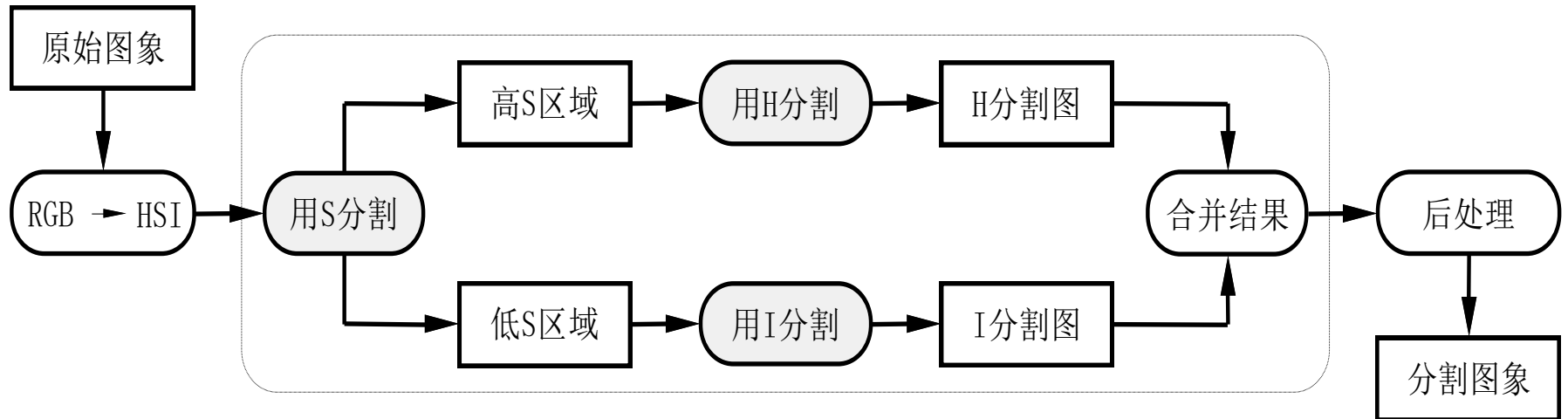


图 7.2.3 HSI 模型的圆柱坐标系统

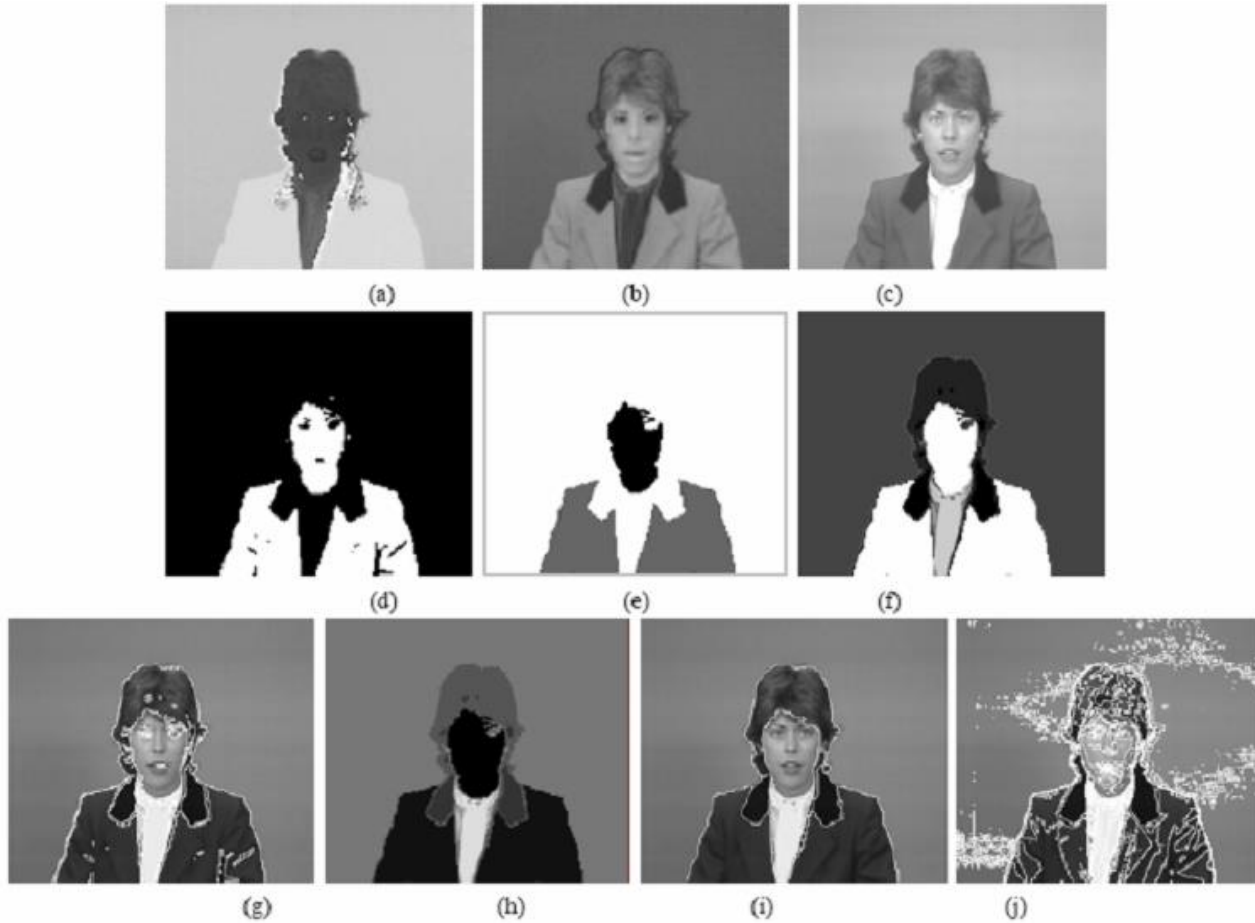
彩色图像的分割

□ 分割策略 I

由于 H, S, I 三个分量是相互独立的，所以有可能将一个 3-D 搜索问题转化为三个 1-D 搜索



彩色图像分割实例



a) H; b) S; c) I; d) S图分割：白色高S区域； e)按H值分割； f)按I值分割； g)综合e和f； h) 后处理； i)加入边界； j)彩色空间分割



图像分割

- 图像分割定义
- 阈值分割
- 区域生长法
- 分裂合并方法
- 分水岭算法
- 聚类分割算法
- 主动轮廓分割
- Graph Cut

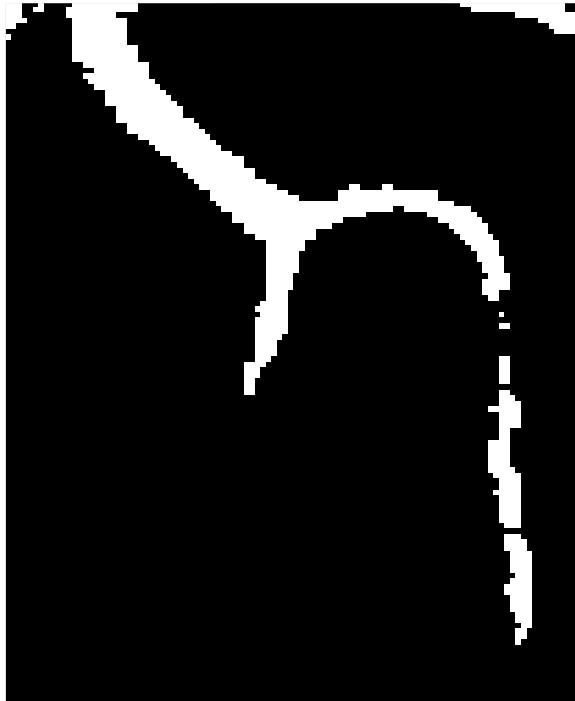
传统的图像分割方法存在的问题

- 基于阈值的分割
 - 难以选择合适的阈值
- 边缘检测
 - 边缘没有闭合，难以得到完整目标区域

血管图像



二值化



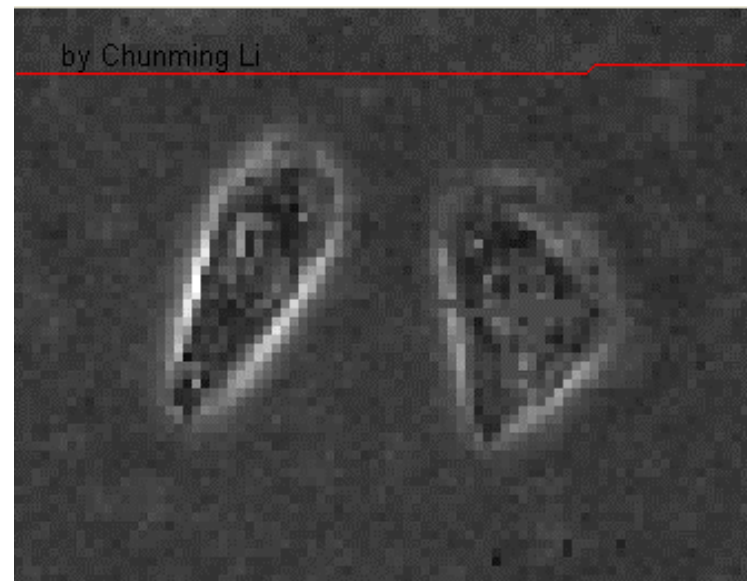
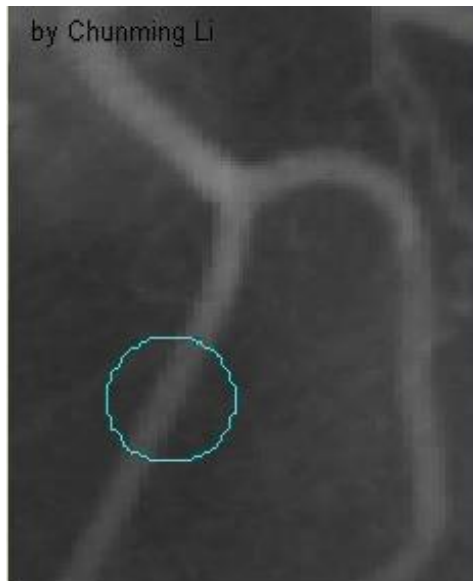
边缘检测



基于主动轮廓模型的图像分割

□ ACM: Active Contour Model

- 基本思想：通过在图像中演化一组**动态曲线**，**自适应地控制曲线的演化速度**，使得其停驻在物体边缘处，从而检测出图像中感兴趣的物体
- 两种类型
 - ✓ 参数化主动轮廓：snake
 - ✓ 几何主动轮廓：水平集方法 (level set)





主动轮廓模型的优势

- 易于对目标进行描述
- 亚像素的精度
- 易于融入各种信息
 - 如形状先验信息、运动信息
- 成熟的数学工具
 - 变分法、偏微分方程(PDE), 微分几何等



基于主动轮廓模型的图像分割

□ 参数化主动轮廓：snake

- 经典工作：“Snakes: Active Contour Models”

Snakes: Active contour models

[M Kass](#), [A Witkin](#), [D Terzopoulos](#) - International journal of computer vision, 1988 - Springer

A snake is an energy-minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. Snakes are active contour models: they lock onto nearby edges, localizing them accurately. Scale-space ...

☆ 被引用次数: 23964 相关文章 所有 47 个版本

□ 几何主动轮廓：水平集方法 (level set)

- From Snake to Level Set
- Classical Level Set Model



参数化主动轮廓 (Kass et al, 1988)

对于一个轮廓曲线 $\mathbf{v}(s) = [x(s), y(s)]$, 定义能量函数:

$$E_{\text{snake}}^* = \int_0^1 E_{\text{snake}}(\mathbf{v}(s)) ds$$
$$= \int_0^1 \left(E_{\text{int}}(\mathbf{v}(s)) + E_{\text{image}}(\mathbf{v}(s)) + E_{\text{con}}(\mathbf{v}(s)) \right) ds$$

曲线内在的弯曲能

图像相关的能量

外在的约束力

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

$$E_{\text{image}} = w_{\text{line}} E_{\text{line}} + w_{\text{edge}} E_{\text{edge}} + w_{\text{term}} E_{\text{term}}$$



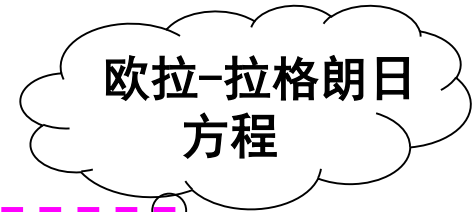
主动轮廓的演化

□ Snakes能量函数

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

$$E_{\text{image}} = w_{\text{line}} E_{\text{line}} + w_{\text{edge}} E_{\text{edge}} + w_{\text{term}} E_{\text{term}}$$

□ 最陡梯度下降算法(变分法):



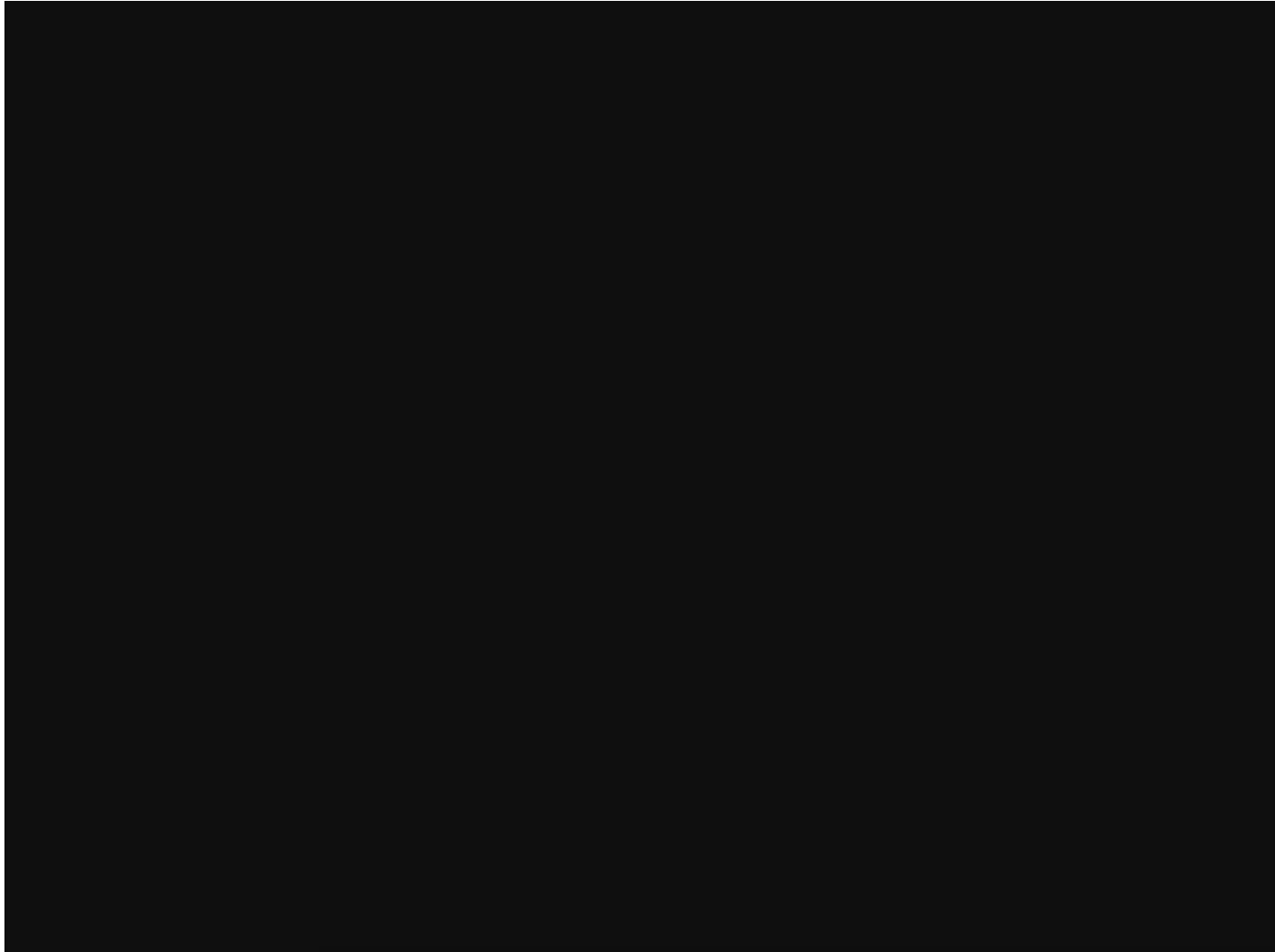
$$\begin{cases} \frac{\partial}{\partial t} C(s, t) = \alpha C''(s, t) - \beta C''''(s, t) - \nabla E_{\text{ext}}, \\ C(s, 0) = C_0(s) \end{cases}$$

内在力: 规则化 (Regularize) 轮廓曲线, 平滑性约束

外力: 推动轮廓向感兴趣的物体边界运动

$$E_{\text{ext}} = E_{\text{image}} + E_{\text{con}}$$

Demo for Snakes





曲线微分几何

□ 平面曲线可以表达为

$$\mathcal{C} : [0, 1] \rightarrow \mathbb{R}^2 \quad \mathcal{C}(p) = (x(p), y(p))$$

□ 其中

■ 切向量:

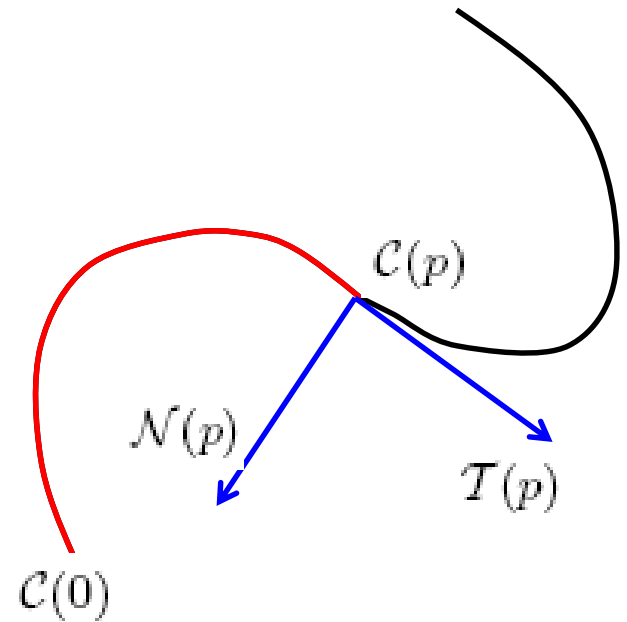
$$\mathcal{T}(p) = \mathcal{C}'(p) = (x'(p), y'(p))$$

■ 法向量

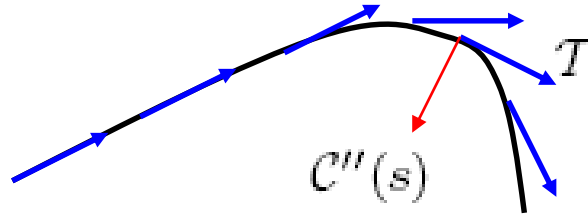
$$\mathcal{N}(p) = \left(\frac{-y'(p)}{\sqrt{x'(p)^2 + y'(p)^2}}, \frac{x'(p)}{\sqrt{x'(p)^2 + y'(p)^2}} \right)$$

■ 弧线长度: 从 $\mathcal{C}(0)$ 到 $\mathcal{C}(p)$

$$s(p) = \int_0^p |\mathcal{C}'(s)| ds$$



曲率 (curvature)



□ 采用弧长 s 作为参数

■ $ds = |C'(p)|dp = \sqrt{(x'(p))^2 + (y'(p))^2} dp$

■ 切向矢量 $T(s) = C'(s)$ 是一个单位向量

$$T(s) = C'(s) = \left(\frac{dx}{ds}, \frac{dy}{ds} \right) = \left(\frac{dx/dp}{ds/dp}, \frac{dy/dp}{ds/dp} \right) = \left(\frac{x'(p)}{\sqrt{(x'(p))^2 + (y'(p))^2}}, \frac{y'(p)}{\sqrt{(x'(p))^2 + (y'(p))^2}} \right)$$

单位切向量 T 定义了曲线的方向

□ 曲率描述了曲线方向改变的速度

□ 再次求导: $C''(s) = T'(s)$

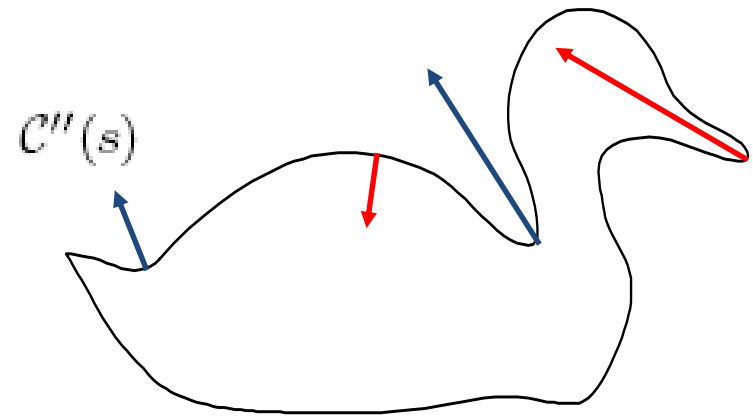
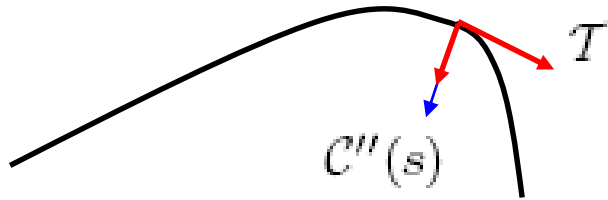
□ 注意 $C'(s) \cdot C'(s) \equiv 1 \implies C''(s) \cdot C'(s) = 0$

曲率 (curvature)

□ $C''(s)$ 和 \mathcal{N} 均正交于 T

⇒ $C''(s)$ 平行于 \mathcal{N}

⇒ $C''(s) = \kappa(s)\mathcal{N}(s)$



□ 曲线的曲率：曲线上某个点的切线方向角对弧长的转动率

□ 对于一般的参数化形式 $C(p) = (x(p), y(p))$ ，曲率可以表示为：

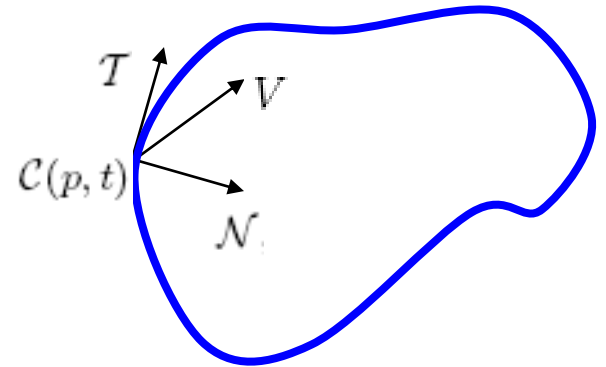
$$\kappa(p) = \frac{x'(p)y''(p) - y'(p)x''(p)}{(x'(p)^2 + y'(p)^2)^{3/2}}$$

动态曲线

□ **动态**曲线随时间变化: $C(p, t) = (x(p, t), y(p, t))$

□ 曲线运动由曲线演化方程确定:

$$\frac{\partial C(p, t)}{\partial t} = V(p, t)$$



□ 切向量 T 和法向量 N 构成 \mathbb{R}^2 空间的正交基

⇒
$$\frac{\partial C(p, t)}{\partial t} = \alpha T + \beta N$$

问题: 当 $\alpha \neq 0, \beta = 0$ 时, 动态曲线随时间是否有变化?

几何曲线演化

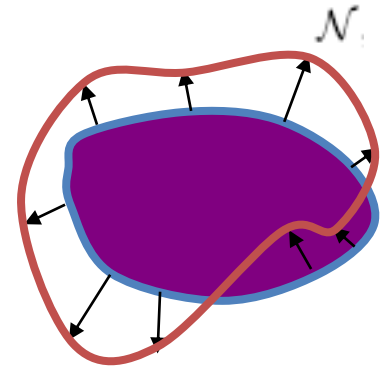
- **定理:** 设 β 与参数表示方式无关. 若 $C(p, t)$ 按照以下方程进行演化

$$\frac{\partial C(p, t)}{\partial t} = \alpha T + \beta N$$

则存在 $C(p, t)$ 的另一个参数化形式 $\bar{C}(q, t)$, 使得 $\bar{C}(q, t)$ 为以下方程的解:

$$\frac{\partial \bar{C}(q, t)}{\partial t} = \bar{\beta} N$$

其中 $\bar{\beta} = \beta$ 对于同一个点: $\bar{C}(q) = C(p)$



- 参数演化的一般方程:

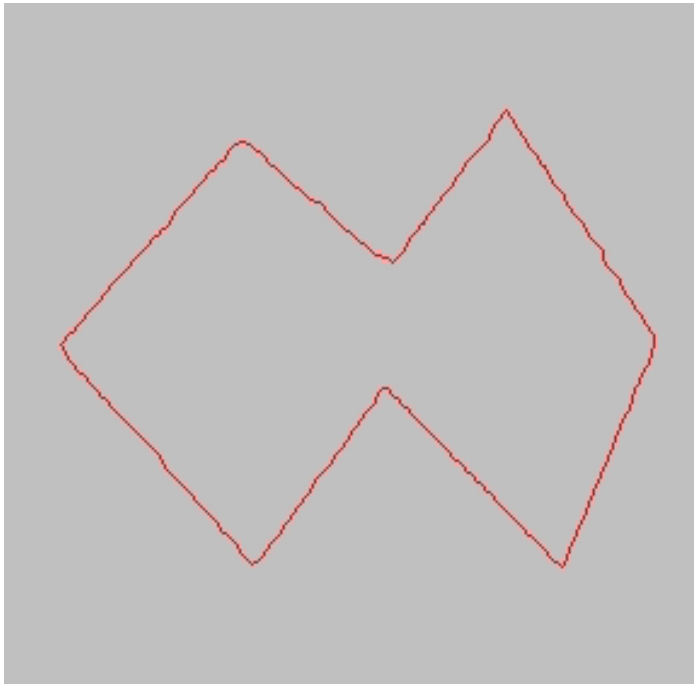
$$\frac{\partial C}{\partial t} = FN$$

常速运动和平均曲率运动

常速运动

(Area decreasing/increasing)

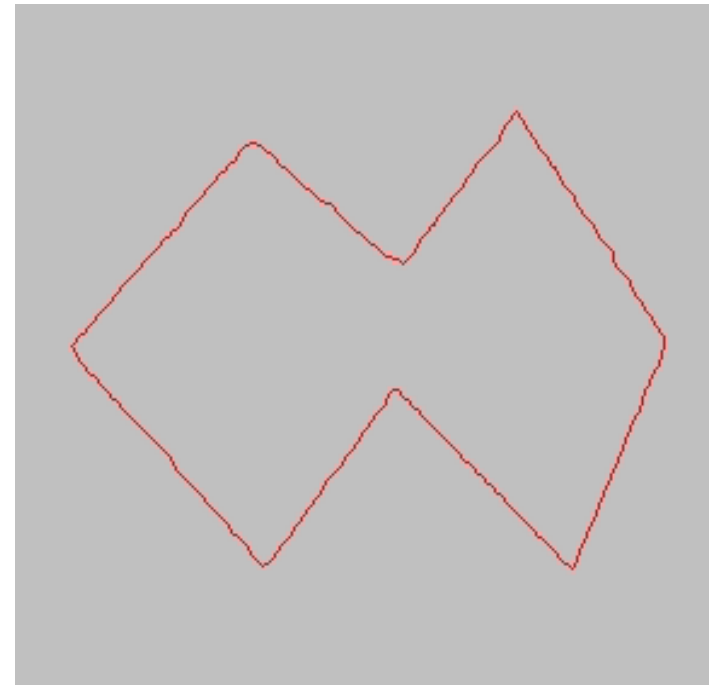
$$\frac{\partial C}{\partial t} = cN$$



平均曲率运动

(Length shortening flow)

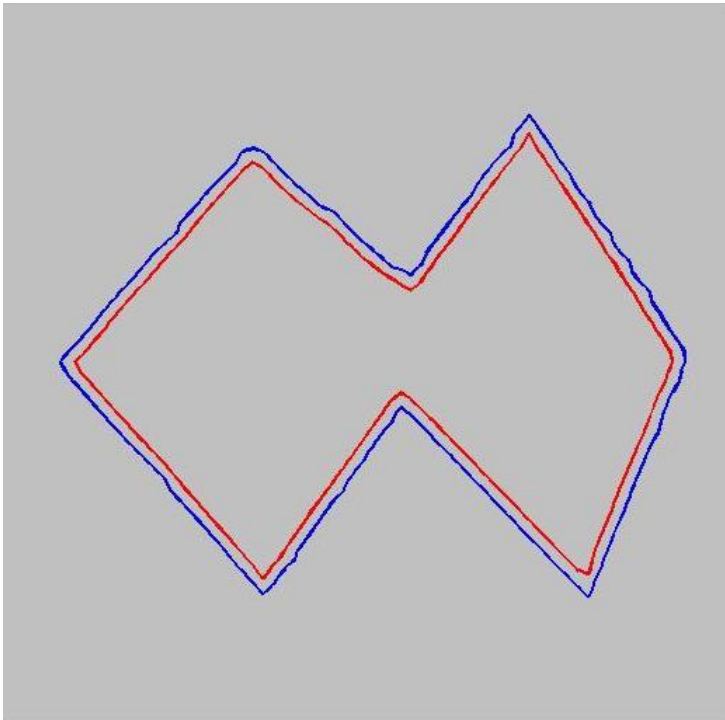
$$\frac{\partial C}{\partial t} = \kappa N$$



演化中间结果

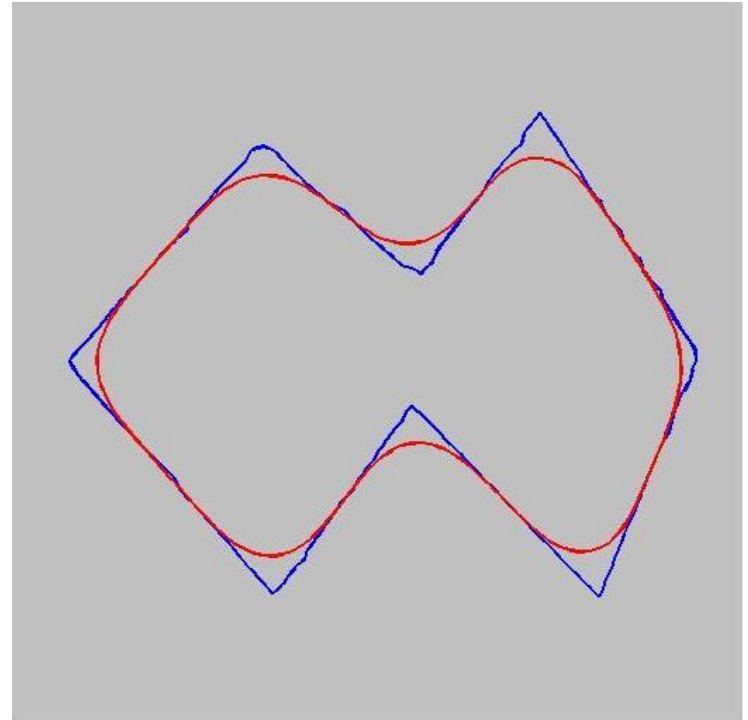
常速运动
(Area decreasing/increasing)

$$\frac{\partial C}{\partial t} = c\mathcal{N}$$



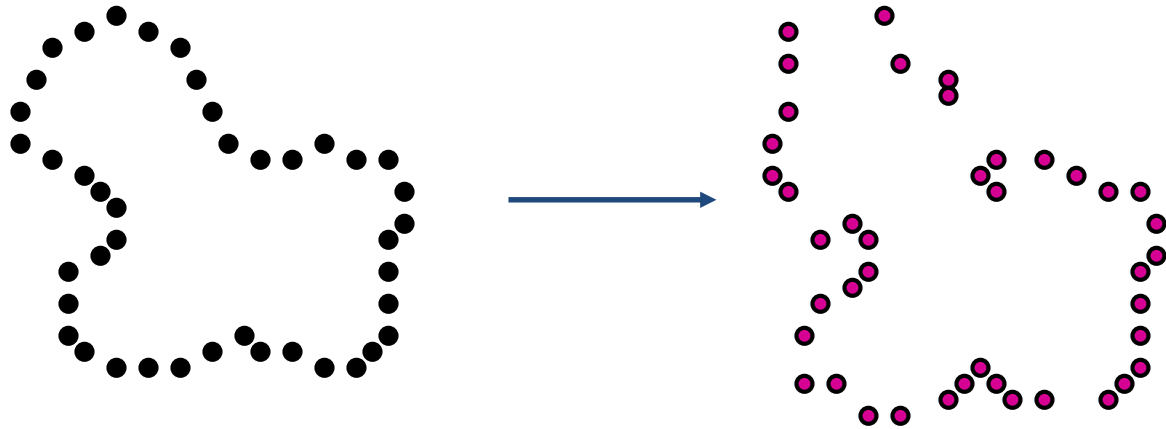
平均曲率运动
(Length shortening flow)

$$\frac{\partial C}{\partial t} = \kappa\mathcal{N}$$

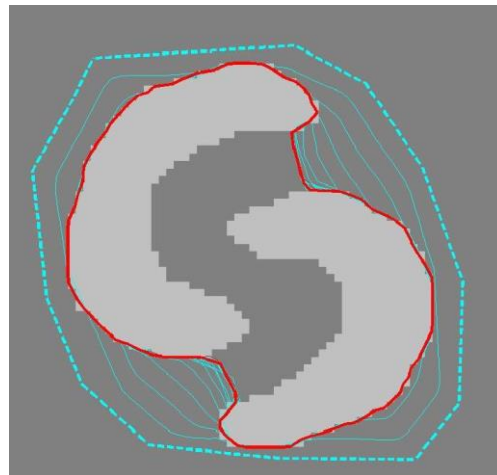


参数化主动轮廓的缺陷

- 演化过程中重新参数化: 对于3D 曲面尤为困难



- 不能处理拓扑变化





解决途径

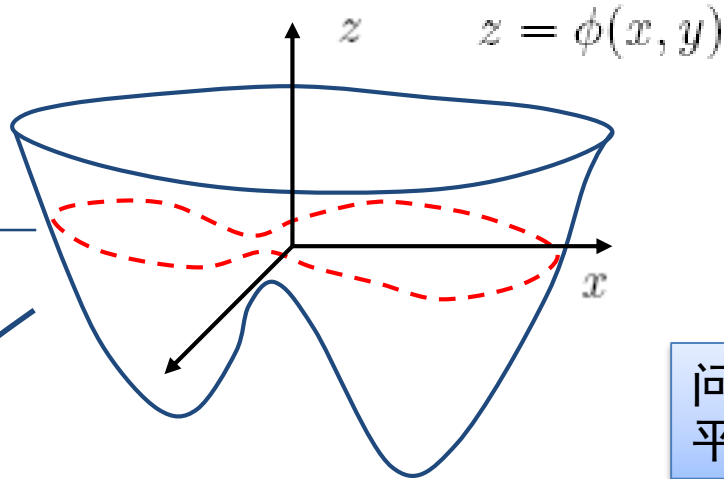
- 基于参数化主动轮廓框架下的方法:
 - T-Snakes: topologically adaptable snakes
 - ✓ McInerney and Terzopoulos, 1995
 - ✓ Ray and Acton, 2003
 - ✓ Li, Liu, and Fox, 2005

- 更好的选择:
 - 水平集方法 (Level set methods)

曲线的水平集描述

- 水平集：曲面上，同一等高线上的点的集合
- 升维：2D \rightarrow 3D

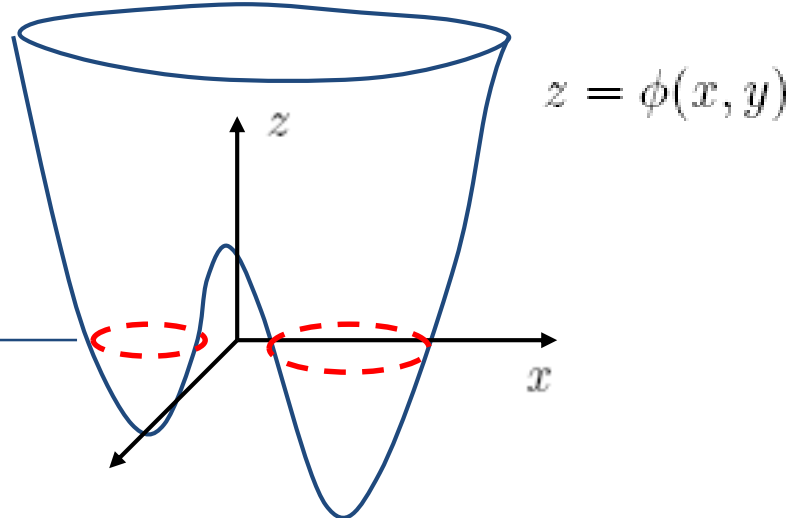
零水平集



问题：如何选择水平集函数 $\phi(x, y)$ ？

- 降维：3D \rightarrow 2D

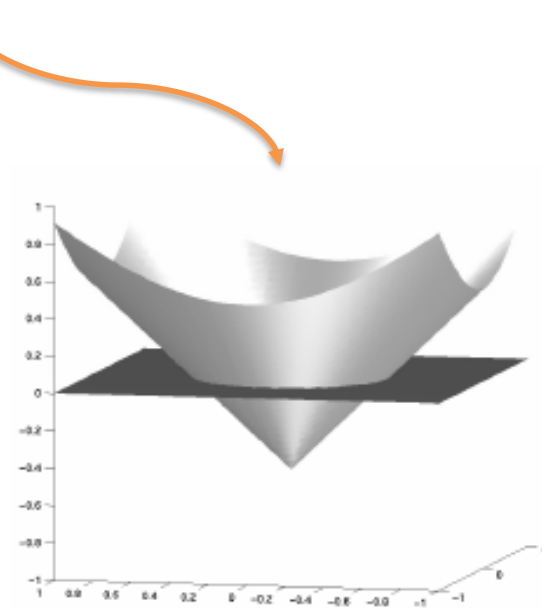
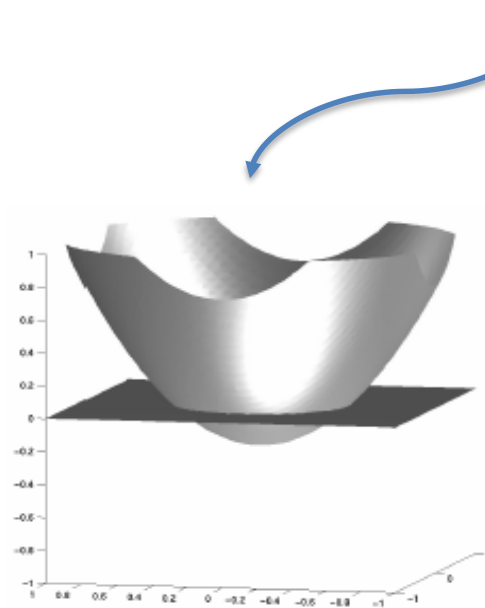
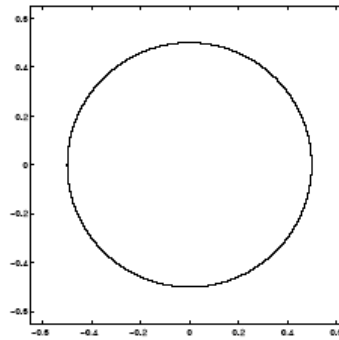
零水平集



水平集函数 (level set function)

- 如何选择水平集函数 $\phi(x, y)$?
 - 曲面梯度方向与水平面夹角为45度

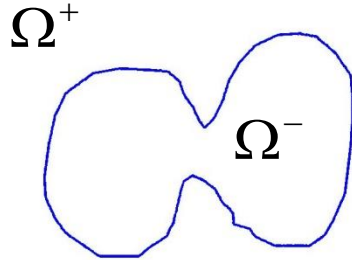
轮廓线



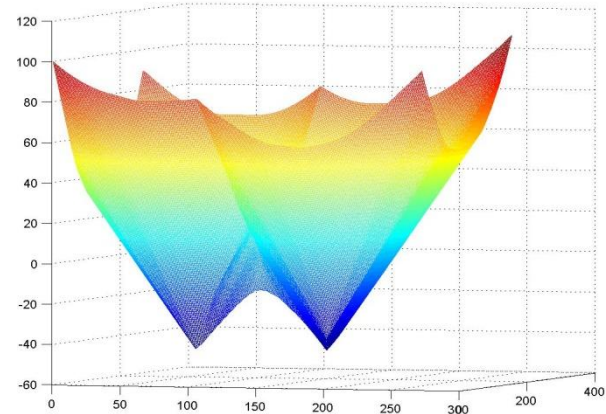
零水平集相同的
水平集函数

符号距离函数 (Signed distance function)

轮廓 C



符号距离函数



符号距离函数定义: $\phi(\mathbf{x}) = \begin{cases} \text{dist}(\mathbf{x}, C) & \text{if } \mathbf{x} \text{ is outside } C \\ 0 & \mathbf{x} \in C \\ -\text{dist}(\mathbf{x}, C) & \text{if } \mathbf{x} \text{ is inside } C \end{cases}$

可以证明: $|\nabla\phi(\mathbf{x})| = 1$



从曲线演化到水平集演化

□ 曲线演化方程: $\frac{\partial C}{\partial t} = FN$

其中 F 为速度函数, N 为曲线 C 的法向量

□ 把动态闭合曲线 $C(\mathbf{p}, t)$ 作为零水平集, 嵌入到一个随时间变化的函数 $\phi(\mathbf{x}, t)$ 中, 使得 $\phi(C(\mathbf{p}, t), t) = 0$

□ 对 $\phi(C(\mathbf{p}, t), t)$ 关于时间 t 求微分 $\Rightarrow \frac{\partial \phi}{\partial t} + \nabla \phi \cdot \frac{\partial C}{\partial t} = 0$


$$\frac{\partial \phi}{\partial t} = -\nabla \phi \cdot \frac{\partial C}{\partial t} = -\nabla \phi \cdot FN$$


$$N = -\frac{\nabla \phi}{|\nabla \phi|}$$

□ 水平集演化方程: $\frac{\partial \phi}{\partial t} = F|\nabla \phi|$

一些特殊的演化方程

□ 平均曲率运动 $\frac{\partial \mathcal{C}}{\partial t} = \kappa \mathcal{N}$  $\frac{\partial \phi}{\partial t} = \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) |\nabla \phi|$

□ 常速率运动 $\frac{\partial \mathcal{C}}{\partial t} = c \mathcal{N}$  $\frac{\partial \phi}{\partial t} = c |\nabla \phi|$

□ 对流运动 $\frac{\partial \mathcal{C}}{\partial t} = \langle \bar{V}, \mathcal{N} \rangle \mathcal{N}$  $\frac{\partial \phi}{\partial t} = -\langle \bar{V}, \nabla \phi \rangle$

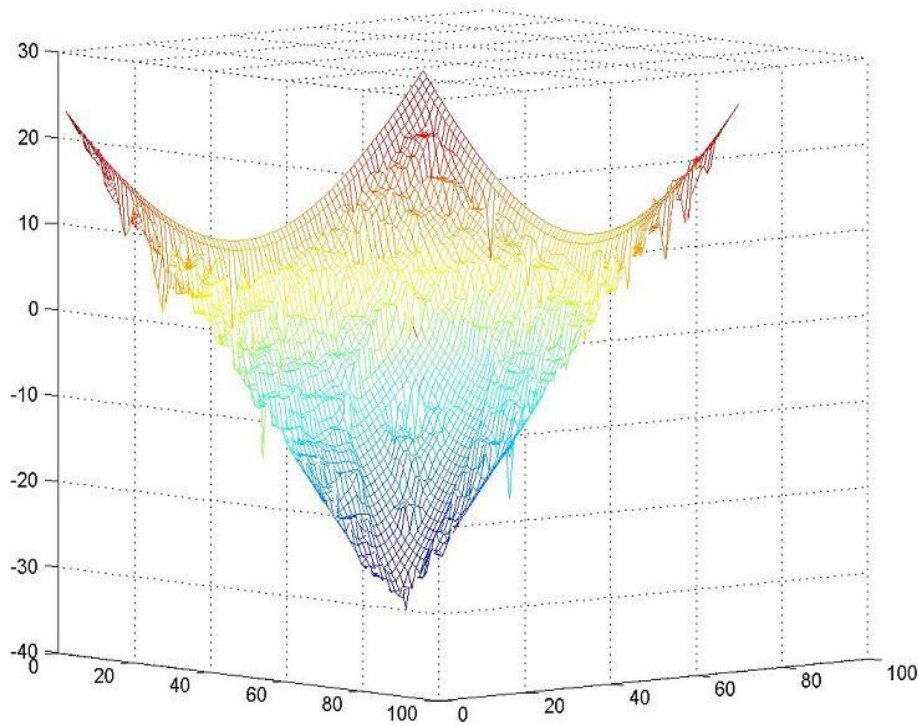
□ 测地线主动轮廓 $\frac{\partial \mathcal{C}}{\partial t} = (c + g) \kappa \mathcal{N} - \langle \nabla g, \mathcal{N} \rangle \mathcal{N}$



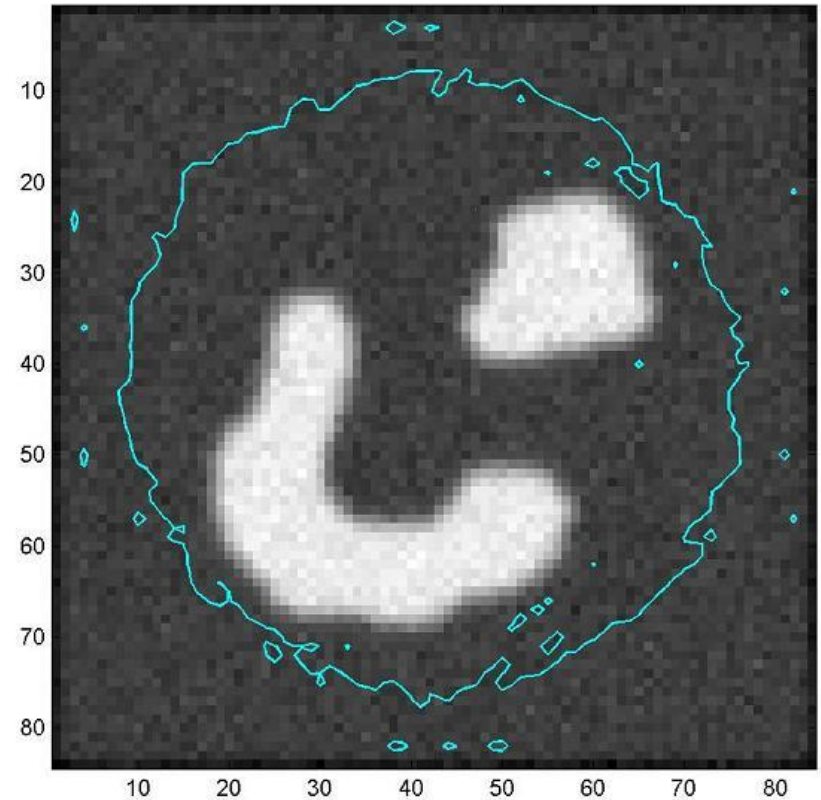
$$\frac{\partial \phi}{\partial t} = \left(c + \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right) |\nabla \phi| + \langle \nabla g, \nabla \phi \rangle$$

演化不稳定 重新初始化

降质(Degraded)的水平集函数, 50步迭代后, 时间步长为0.1

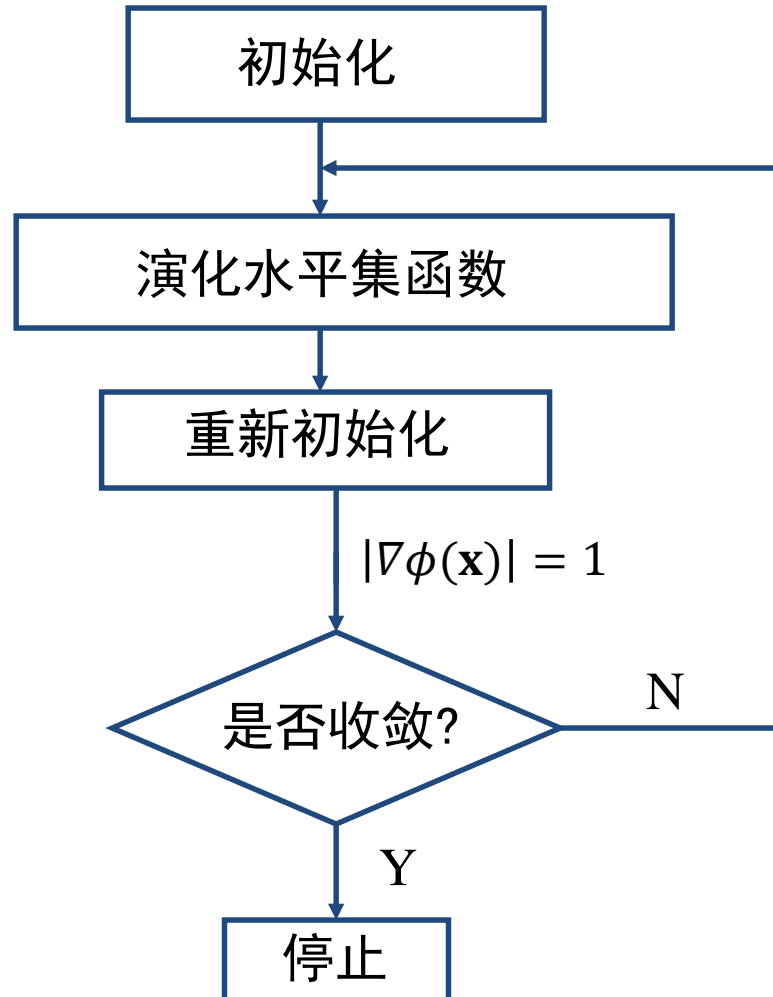


降质水平集函数的零水平集





水平集方法的一般步骤





Level Set without Re-initialization

□ 将re-initialization形式化为能量泛函: $\mathcal{P}(\phi) = \int_{\Omega} \frac{1}{2} (|\nabla\phi| - 1)^2 dx dy$

□ 定义变分能量函数 $\mathcal{E}(\phi) = \mu\mathcal{P}(\phi) + \mathcal{E}_{g,\lambda,\nu}(\phi)$

其中 $= \mu\mathcal{P}(\phi) + \lambda\mathcal{L}_g(\phi) + \nu\mathcal{A}_g(\phi)$

$$\mathcal{L}_g(\phi) = \int_{\Omega} g\delta(\phi)|\nabla\phi| dx dy \quad \mathcal{A}_g(\phi) = \int_{\Omega} gH(-\phi) dx dy, \quad g = \frac{1}{1 + |\nabla G_{\sigma} * I|^2}$$

□ 对能量函数求梯度

$$\frac{\partial \mathcal{E}}{\partial \phi} = -\mu[\Delta \phi - \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)] - \lambda\delta(\phi)\operatorname{div}\left(g\frac{\nabla\phi}{|\nabla\phi|}\right) - \nu g\delta(\phi)$$

□ 基于梯度下降算法, 得到演化方程:

$$\frac{\partial \phi}{\partial t} = \mu[\Delta \phi - \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)] + \lambda\delta(\phi)\operatorname{div}\left(g\frac{\nabla\phi}{|\nabla\phi|}\right) + \nu g\delta(\phi)$$



变分法推导示例

An instance for calculus of variations:

$$\begin{aligned} \text{Define a functional : } E(\phi) &= \int_{\Omega} \frac{1}{2} (|\nabla \phi| - 1)^2 dx dy \\ &= \int_{\Omega} \frac{1}{2} (|\nabla \phi|^2 - 2|\nabla \phi| + 1) dx dy \end{aligned}$$

$$\begin{aligned} \text{Let } F(\phi) &= \frac{1}{2} (|\nabla \phi|^2 - 2|\nabla \phi| + 1) \\ &= \frac{1}{2} \left((\phi_x^2 + \phi_y^2) - 2\sqrt{\phi_x^2 + \phi_y^2} + 1 \right) \end{aligned}$$

in terms of a small variable δ and an arbitrary function h which satisfies : $h|_{\partial\Omega} = 0$, we can get:

$$F(\phi + \delta h) = \frac{1}{2} \left[((\phi + \delta h)_x^2 + (\phi + \delta h)_y^2) - 2\sqrt{(\phi + \delta h)_x^2 + (\phi + \delta h)_y^2} + 1 \right]$$



变分法推导示例

then:

$$\begin{aligned}\frac{\partial F(\phi + \delta h)}{\partial \delta} &= h_x(\phi + \delta h)_x + h_y(\phi + \delta h)_y - \frac{h_x(\phi + \delta h)_x + h_y(\phi + \delta h)_y}{\sqrt{(\phi + \delta h)_x^2 + (\phi + \delta h)_y^2}} \\ &= \nabla h \cdot \nabla(\phi + \delta h) - \frac{\nabla h \cdot \nabla(\phi + \delta h)}{\sqrt{(\phi_x^2 + \phi_y^2) + \delta^2(h_x^2 + h_y^2) + 2\delta \nabla h \cdot \nabla \phi}}\end{aligned}$$

$$\left. \frac{\partial F(\phi + \delta h)}{\partial \delta} \right|_{\delta \rightarrow 0} = \nabla h \cdot \nabla \phi - \frac{\nabla h \cdot \nabla \phi}{\sqrt{(\phi_x^2 + \phi_y^2)}}$$

$$\begin{aligned}\text{Then } \left. \frac{\partial E(\phi + \delta h)}{\partial \delta} \right|_{\delta \rightarrow 0} &= \int_{\Omega} \left(\nabla h \cdot \nabla \phi - \frac{\nabla h \cdot \nabla \phi}{\sqrt{(\phi_x^2 + \phi_y^2)}} \right) dx dy \\ &= \int_{\Omega} (h_x \phi_x + h_y \phi_y) dx dy - \int_{\Omega} \frac{(h_x \phi_x + h_y \phi_y)}{\sqrt{(\phi_x^2 + \phi_y^2)}} dx dy\end{aligned}$$



变分法推导示例

$$\because \frac{\partial}{\partial x} [\phi_x h] = h_x \phi_x + h \phi_{xx},$$

$$\frac{\partial}{\partial y} [\phi_y h] = h_y \phi_y + h \phi_{yy},$$

$$\begin{aligned} \therefore \frac{\partial E(\phi + \delta h)}{\partial \delta} \Big|_{\delta \rightarrow 0} &= \int_{\Omega} \frac{\partial}{\partial x} [\phi_x h] + \frac{\partial}{\partial y} [\phi_y h] dx dy - \int_{\Omega} h \phi_{xx} + h \phi_{yy} dx dy \\ &\quad - \int_{\Omega} \frac{\partial}{\partial x} \left[\frac{\phi_x}{|\nabla \phi|} h \right] + \frac{\partial}{\partial y} \left[\frac{\phi_y}{|\nabla \phi|} h \right] dx dy + \int_{\Omega} \frac{\partial}{\partial x} \left[\frac{\phi_x}{|\nabla \phi|} \right] h + h \frac{\partial}{\partial y} \left[\frac{\phi_y}{|\nabla \phi|} \right] dx dy \end{aligned}$$

According to Green Equation: $\iint_D \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dx dy = \oint_L P dx + Q dy$, and $h|_{\partial \Omega} = 0$,

we can get

$$\int_{\Omega} \frac{\partial}{\partial x} [\phi_x h] + \frac{\partial}{\partial y} [\phi_y h] dx dy = \oint_{\partial \Omega} h (\phi_x dy - \phi_y dx) = 0$$



变分法推导示例

$$\int_{\Omega} \frac{\partial}{\partial x} \left[\frac{\phi_x}{|\nabla \phi|} h \right] + \frac{\partial}{\partial y} \left[\frac{\phi_y}{|\nabla \phi|} h \right] dx dy = \oint_{\partial \Omega} h \left(\frac{\phi_x}{|\phi|} dy - \frac{\phi_y}{|\phi|} dx \right) = 0$$

Then

$$\begin{aligned} \frac{\partial E(\phi + \delta h)}{\partial \delta} \Big|_{\delta \rightarrow 0} &= - \int_{\Omega} h \phi_{xx} + h \phi_{yy} dx dy + \int_{\Omega} \left(\frac{\partial}{\partial x} \left[\frac{\phi_x}{|\nabla \phi|} \right] h + h \frac{\partial}{\partial y} \left[\frac{\phi_y}{|\nabla \phi|} \right] \right) dx dy \\ &= - \int_{\Omega} h \Delta \phi dx dy + \int_{\Omega} h \nabla \cdot \left[\frac{\nabla \phi}{|\nabla \phi|} \right] dx dy \end{aligned}$$

When $E(\phi)$ reach the minimal,

$$\frac{\partial E(\phi + \delta h)}{\partial \delta} \Big|_{\delta \rightarrow 0} = - \int_{\Omega} h \left(\Delta \phi - \nabla \cdot \left[\frac{\nabla \phi}{|\nabla \phi|} \right] \right) dx dy = 0,$$



变分法推导示例

Since function h is arbitrary, we obtain:

$$\Delta\phi - \nabla \cdot \left[\frac{\nabla\phi}{|\nabla\phi|} \right] = 0$$

The above equation is the Euler-Lagrange Equation.

Generally, the gradient of functional $E(\phi)$ is denoted as $\left(\Delta\phi - \nabla \cdot \left[\frac{\nabla\phi}{|\nabla\phi|} \right] \right)$,

algorithm, we get the LEVLE SET evolution equation:

$$\phi_t = \Delta\phi - \nabla \cdot \left[\frac{\nabla\phi}{|\nabla\phi|} \right]$$



Reference

- M. Kass, et al, “Snakes: active contour models”, *IJCV*, 1988
- Guillermo Sapiro, “Geometric partial differential equations and image analysis”, *Cambridge University Press*, 2001
- Stanley Osher, Ronald Fedkiw, “Level set methods and dynamic implicit surfaces”, *Springer*, 2003
- Li C, Xu C, Gui C, et al. Level set evolution without re-initialization: a new variational formulation[C]//Computer Vision and Pattern Recognition, *CVPR 2005*.
- A Matlab toolbox for implementing Level Set Methods:
http://barissumengen.com/level_set_methods/



图像分割

- 图像分割定义
- 阈值分割
- 区域生长法
- 分裂合并方法
- 分水岭算法
- 聚类分割算法
- 主动轮廓分割
- **Graph Cut**

Graph Cut 图割法

□ Graph cut: 一种能量函数的优化方法

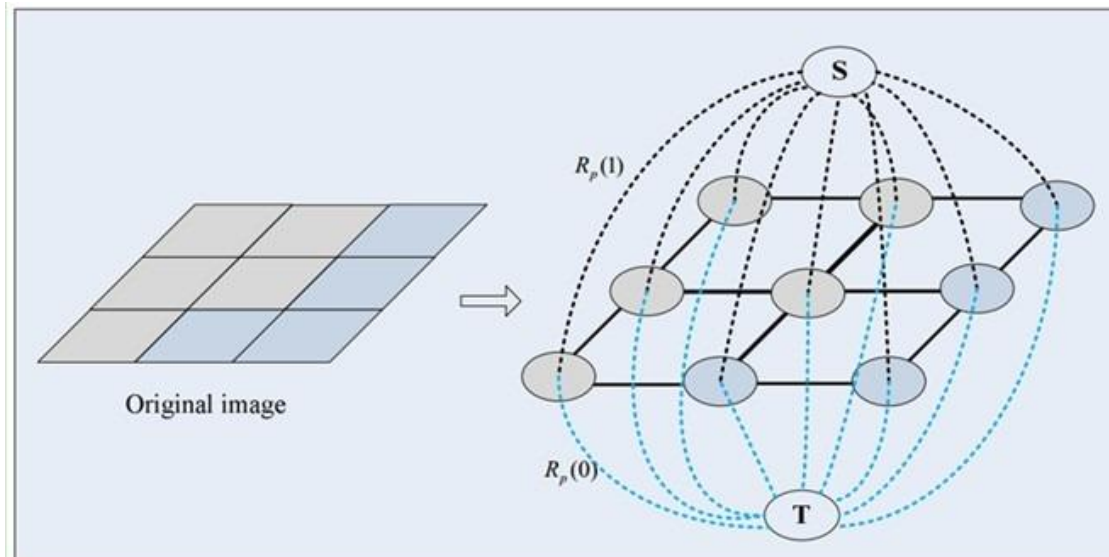
■ 基于图像定义graph（包括节/顶点和边）

■ 第一种顶点和边

- ✓ 普通顶点，对应于图像中的每个像素。每两个**四邻域**顶点的连接就是一条边（**图中实线**），这种边也叫n-links。

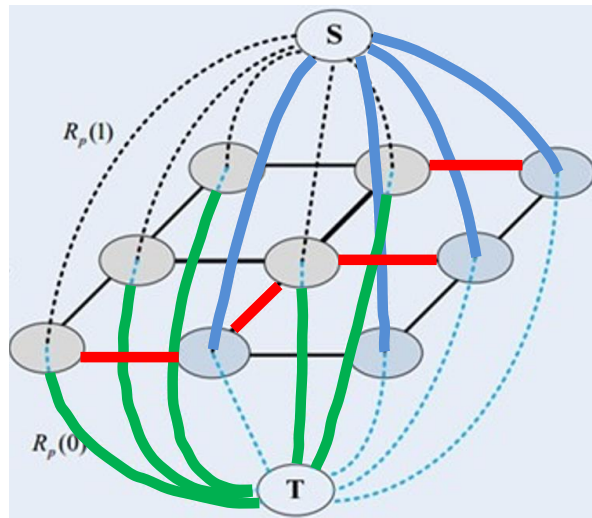
■ 第二种顶点和边

- ✓ 两个终端顶点，分别称作S（source：源点）和T（sink：汇点）。
- ✓ 每个普通顶点和这2个终端顶点之间都有连接，组成第二种边（**图中虚线**），这种边也叫t-links。



Graph Cut

- Cut: 一个cut（割）就是图中边集合 E 的一个子集 C
 - 在图中去掉割(cut)后源点和汇点之间再没有通路，而加入割中的任一条边后都会产生一条通路。
- 图中每条边都有一个非负的权值 w_e ，也可以理解为cost（代价或者费用）
 - 割的cost（表示为 $|C|$ ）就是边子集 C 的所有边的权值的总和。
- 最小割（min cut）：权值之和最小的割
 - 这个最小割把图的顶点划分为两个不相交的子集 S 和 T ，分别对应于图像的前景像素集和背景像素集。



图像分割问题转化为找最小割问题

图中边的权值就决定了最后的分割结果。那么给定一张图像，图中边的权值怎么确定呢？



能量函数

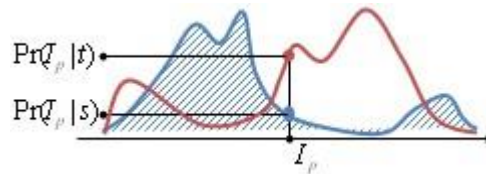
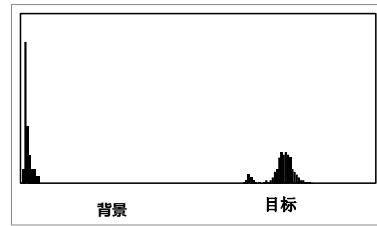
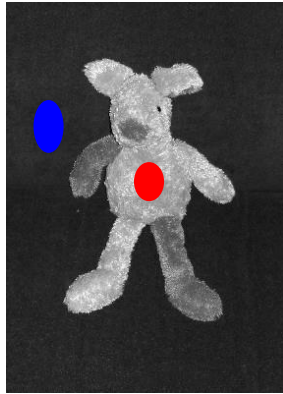
□ 假设图像的分割为 x 时，图像的能量可以表示为：

$$E_x = E_{smooth}(x) + E_{data}(x)$$

- E_x ：表示关于 x 的损失函数，也叫能量函数。图割的目标就是优化能量函数使其值达到最小。
- Data项：普通顶点与终端顶点之间的边的权值
- Smooth项：普通顶点之间的边的权值

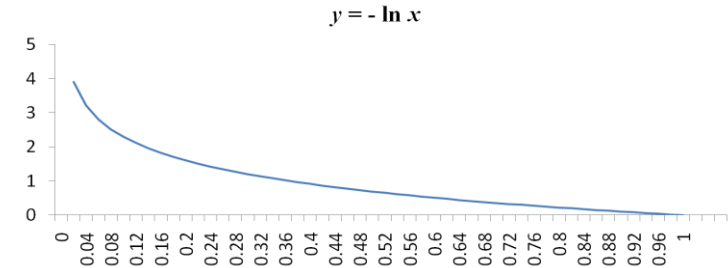
构造能量函数

□ Data项



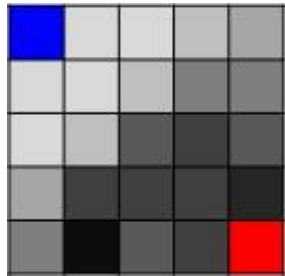
$$E^1(x_i) \propto -\ln \Pr(I_i | A_i)$$

$A_i \in \{s, t\}$ 表示像素 x_i 为前景或者背景



条件概率越高，能量越低

□ Smooth项：针对相邻像素



$$E^2(x_i, x_j) = \exp\left(-\frac{(I_i - I_j)^2}{2\sigma^2}\right) \cdot \delta(A_i, A_j)$$

$$\delta(A_i, A_j) = \begin{cases} 1 & \text{if } A_i \neq A_j \\ 0 & \text{if } A_i = A_j \end{cases}$$



x_i 与 x_j 相邻

优化（最小化）能量函数方法：穷举、模拟退火、graph cut.....



构造能量函数

□ Data项

- 仅有data项，分割结果相当于用颜色的先验概率对像素进行分类
- 分类的结果没有考虑相邻像素之间的空间关系，造成分割不连续

□ Smooth项

- 在data项提供先验信息后，再加入相邻像素间的约束关系，得到平滑的分割结果

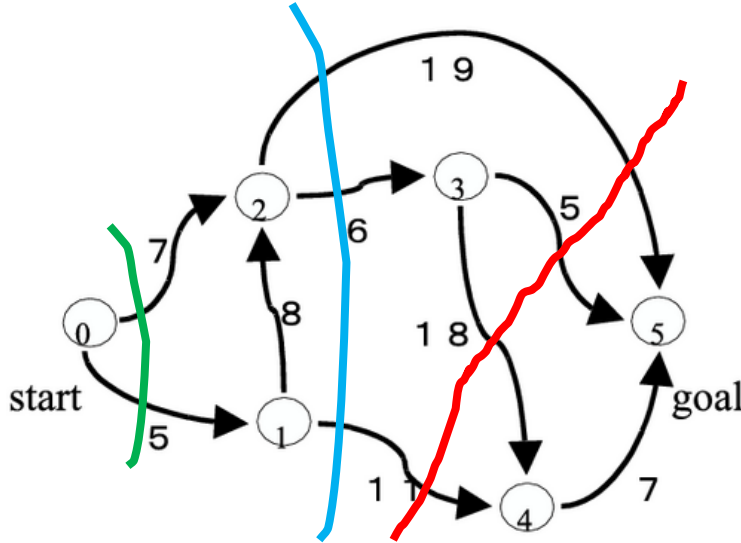
$$E_{\mathbf{x}} = E_{data}(\mathbf{x}) + \lambda E_{smooth}(\mathbf{x})$$



Graph cut过程

1. 给定一个图像，我们需要构建一个图，图有两类顶点，两类边和两类权值。
2. 普通顶点由图像每个像素组成，然后每两个邻域像素之间存在一条边，它的权值由Smooth项来决定。
3. 两个终端顶点s（目标）和t（背景），每个普通顶点和终端顶点都存在连接，这种边的权值由Data项来决定。
4. 这样可以确定所有边的权值，即完成图的构建。
5. 找到最小的割，这个min cut就是权值和最小的边的集合，这些边的断开恰好可以使目标和背景被分割开。

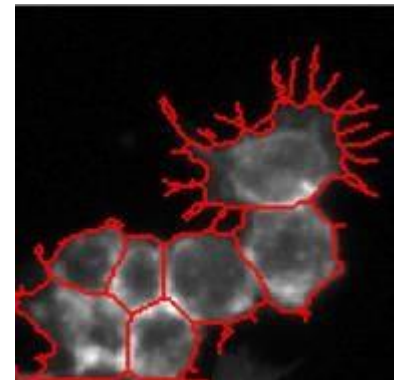
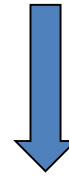
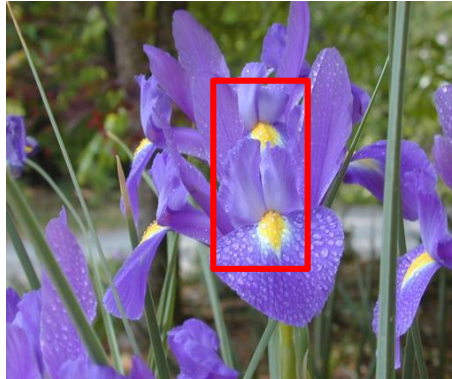
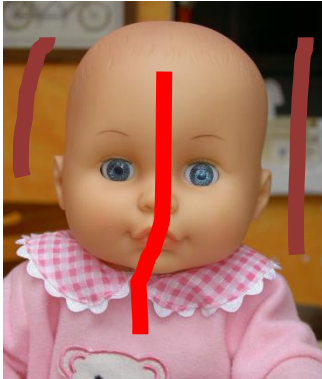
求最小割



最大流最小割定理：在一个网络流中，能够从源点到达汇点的最大流量等于如果从网络中移除就能够导致网络流中断的边的集合的最小容量和。

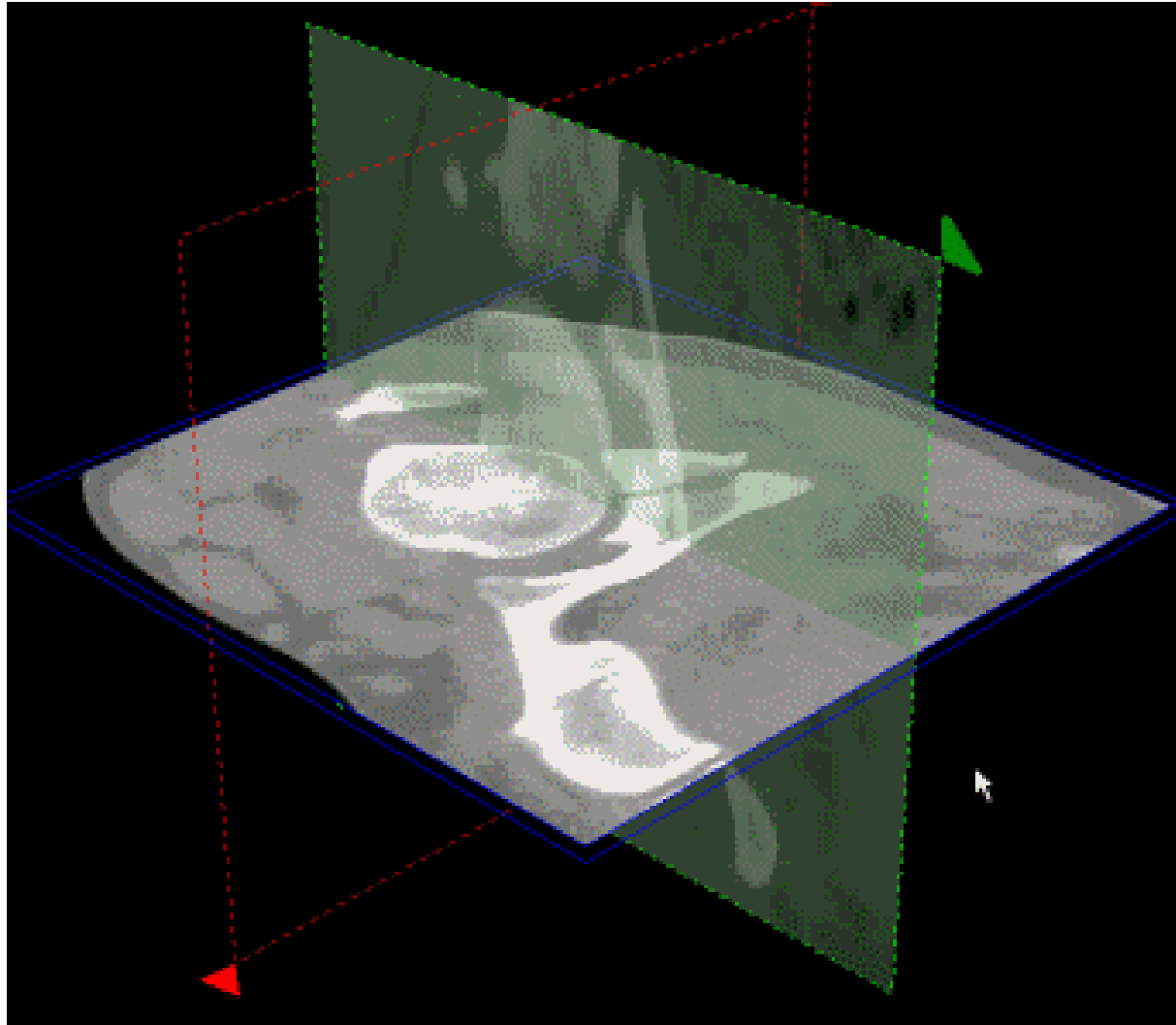
- **最小割(min cut) -> 最大流(max flow)**
 - 最大流最小割定理是网络流理论的重要定理
- 对于一个有向图 (graph)，权值当做弧的容量 (最大流量)
- 这时从一点(start)到另一点(goal)存在最大流 (max flow)
- 当流量达到最大，所有**满流弧**构成一个最小割 (min cut)
- 通过不断增加网络的流量，即可达到最大流，从而找到最小割
- 最小代价的cut可以以多项式时间的复杂度计算得到

采用Graph cut分割的结果





基于 Graph cut的3D 医学图像分割：bone





Graph相关资源

□ Ramin Zabih

<http://www.cs.cornell.edu/~rdz/>

□ Yuri Boykov

<http://www.cs.cornell.edu/home/yura/>

□ Vladimir Kolmogorov:

<http://www.adastral.ucl.ac.uk/~vladkolm/>