



第五章 图象分割 (Image Segmentation)

5.1 阈值分割

5.2 区域生长法

5.3 分裂合并方法

5.4 分水岭算法

5.5 聚类分割算法

5.6 主动轮廓分割

5.7 Graph Cut



图象分割的定义

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 $i \neq j$, 有 $R_i \cap R_j = \emptyset$;

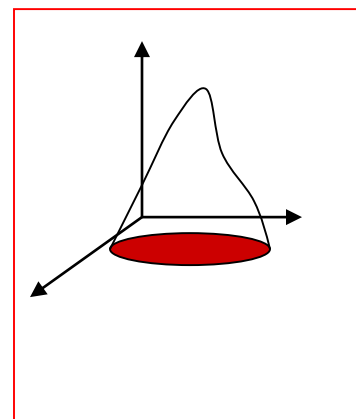
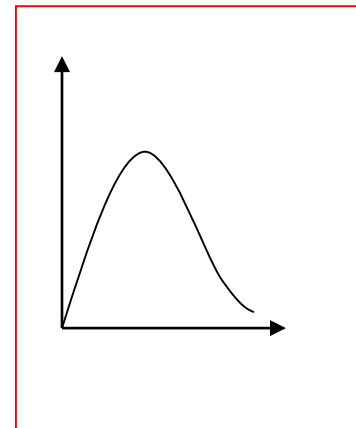
3、每个子区域 R_1, R_2, \dots, R_n 是联通的;

4、对于各个子区域，有均匀性测度度量 P 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。即：

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

5.1 阈值分割 (Thresholding)

- 取阈值法是以图象直方图为依据，选定阈值，再逐个对象素作判决。
- 图象直方图可以是：
 - 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，也可是其他值。



原理和分类

□ 图象模型：

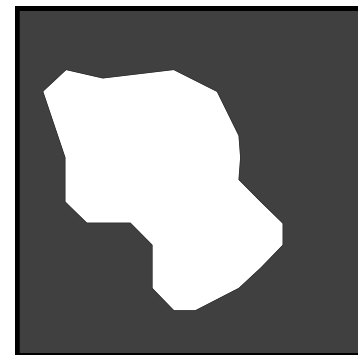
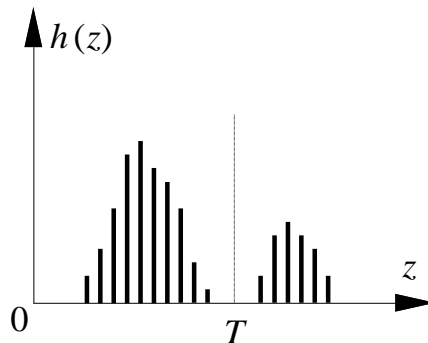
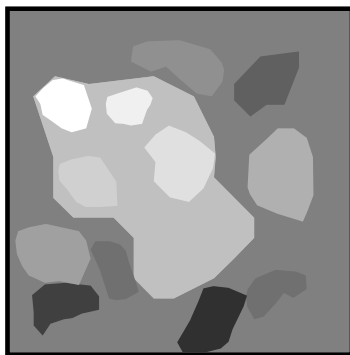
- 图象由具有单峰测度(灰度, 颜色, 纹理分布的目标和背景组成
- 目标或背景内部的相邻像素间的测度值高度相关
- 目标和背景交界处两边的像素测度值差异大

原理和分类

单阈值分割图象

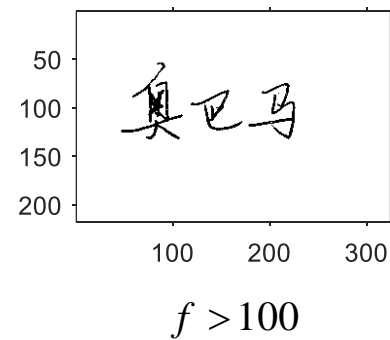
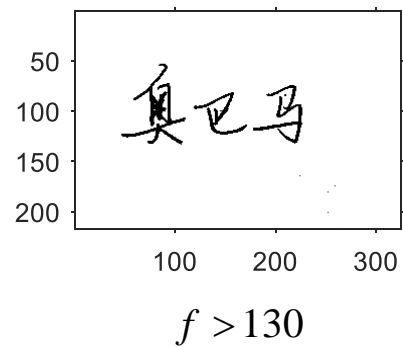
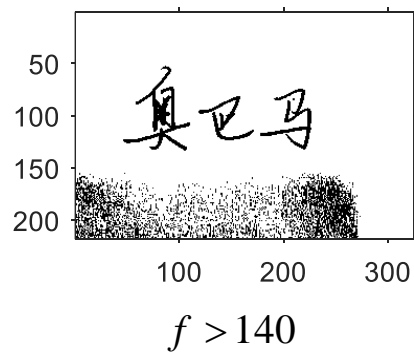
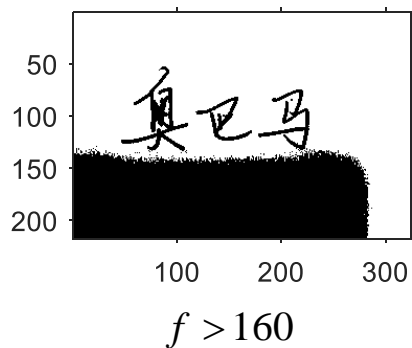
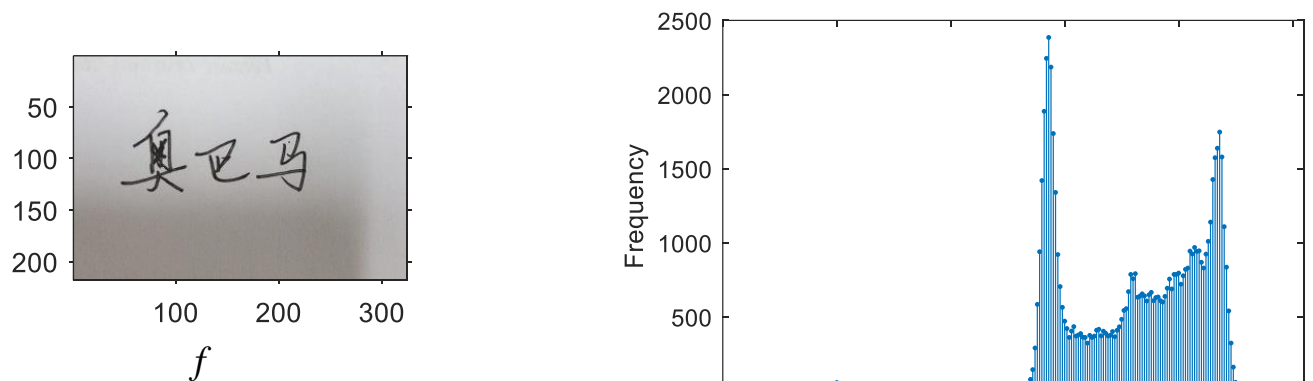
对灰度图（取值在 g_{\min} 和 g_{\max} 之间）确定一个灰度阈值 T （ $g_{\min} < T < g_{\max}$ ）

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



原理和分类

□ 实例：生成电子签名

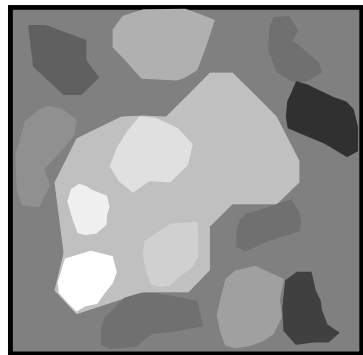


原理和分类

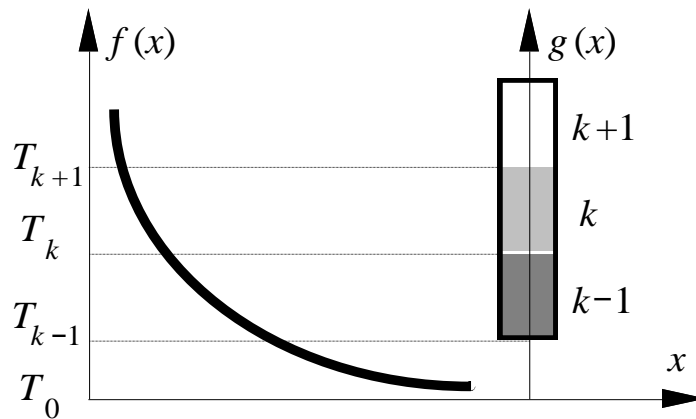
□ 多閾值分割图象

■ 确定一系列分割閾值

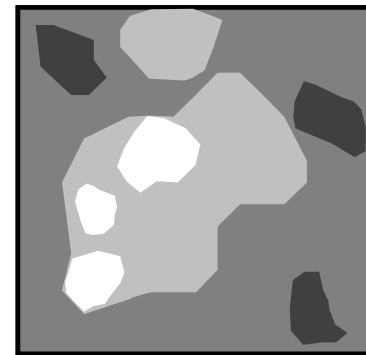
$g(x, y) = k$ 如 $T_k < f(x, y) \leq T_{k+1}$ $k = 0, 1, 2, \dots, K$



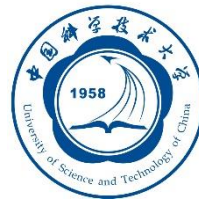
(a)



(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

(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

将前两种阈值也称为固定阈值

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，把图象分成目标和背景两部分

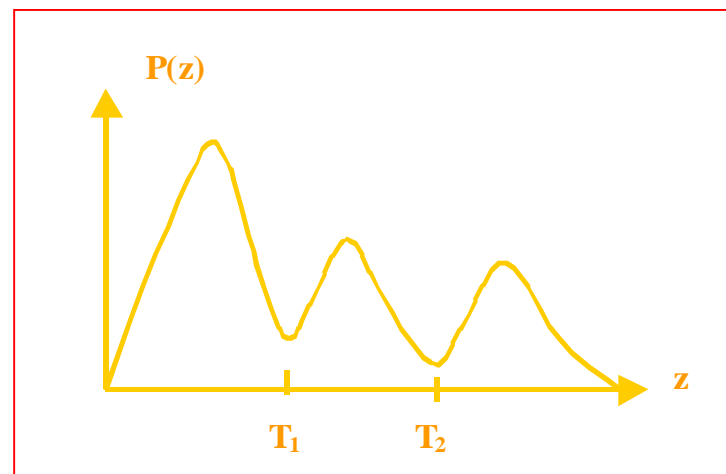
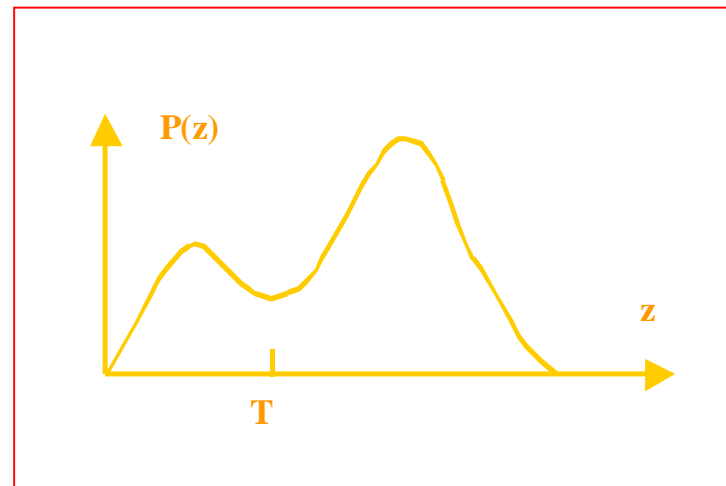
□ 多个阈值

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：

$$\frac{\partial h(z)}{\partial z} = 0 \text{ 和 } \frac{\partial^2 h(z)}{\partial z^2} > 0$$



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。

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

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

$$E_1(T) = \int_{-\infty}^T p_1(z) dz$$

$$E_2(T) = \int_T^{\infty} p_2(z) dz$$

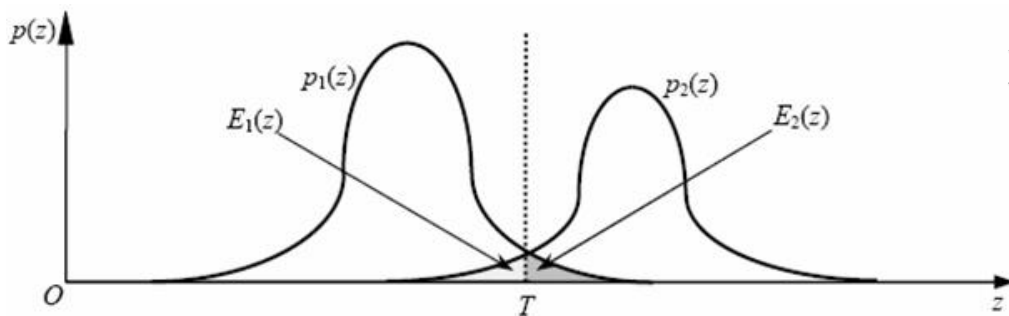
$$E(T) = P_1 E_1(T) + P_2 E_2(T)$$

(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

□ 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 ，计算其直方图 $P(i)$
- 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'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： $\tau = \arg \min_t \sigma_w^2(t)$

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

依赖象素的阈值选取

由直方图凹凸性确定的阈值

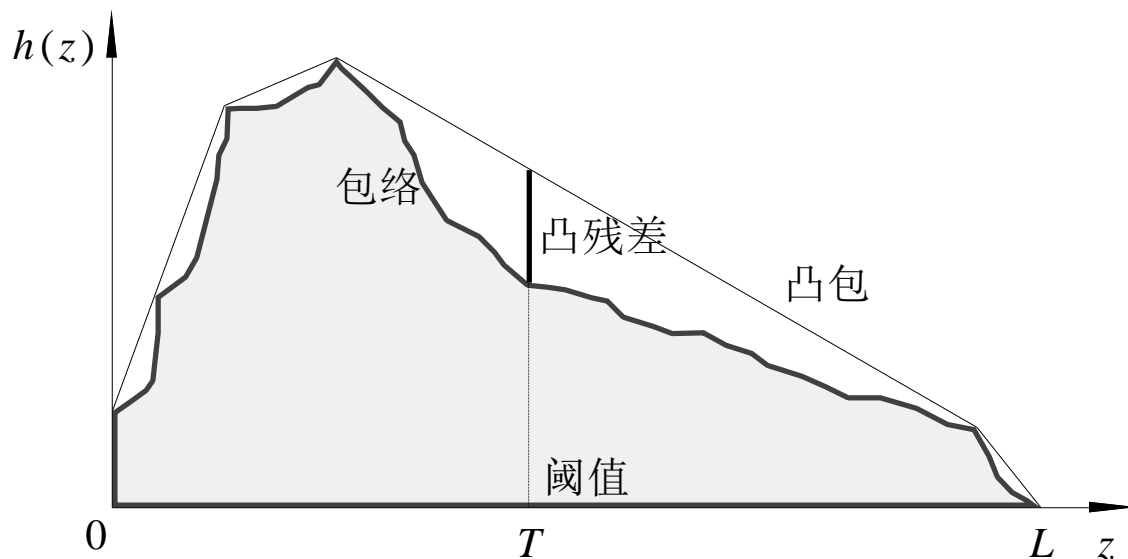
直方图的一个峰淹没在另一个峰旁的缓坡里

直方图的包络

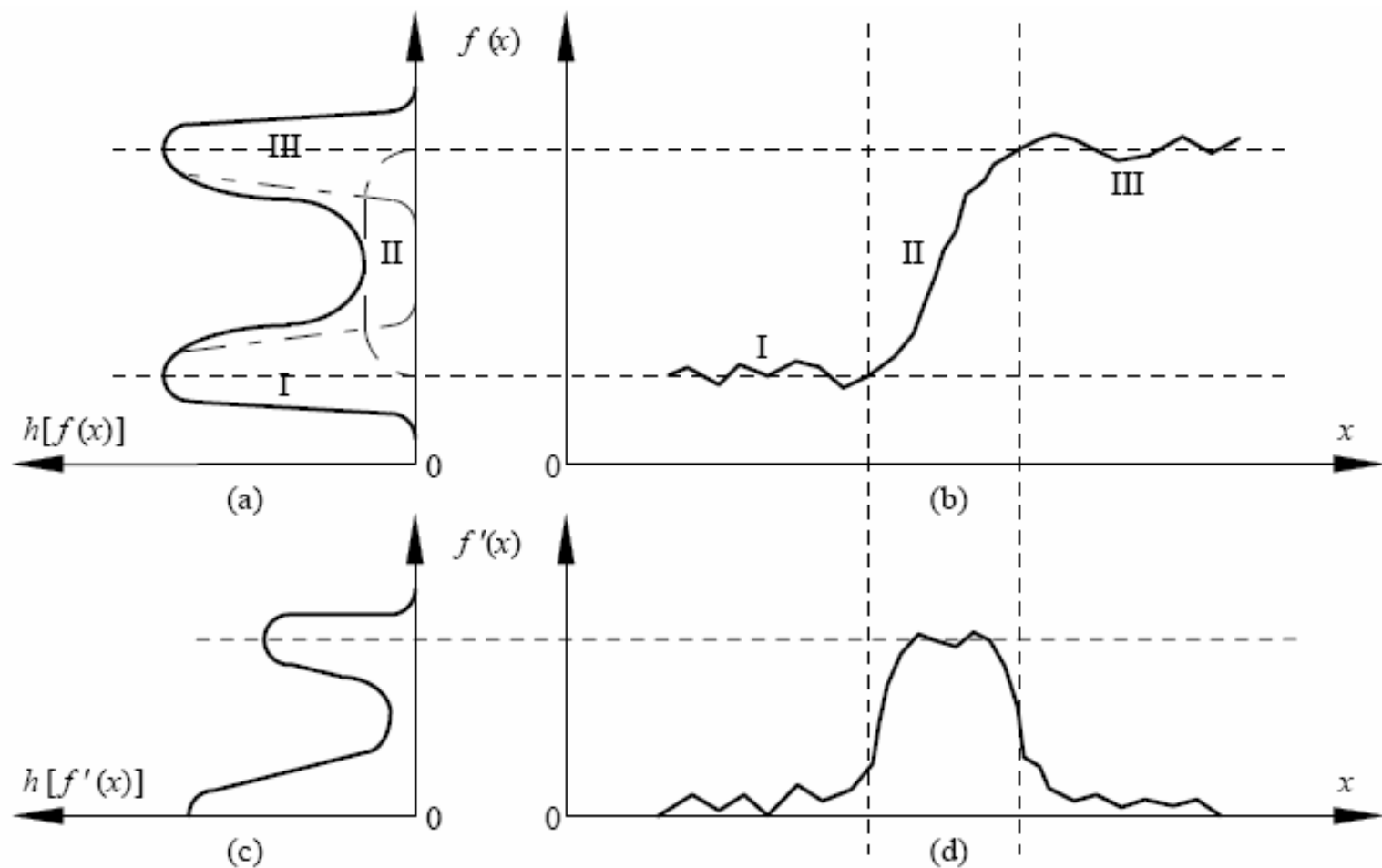
→ 区域凸包

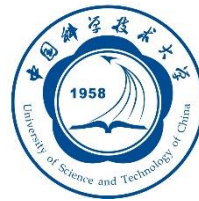
→ 最大凸残差

→ 分割阈值



依赖区域的阈值选取





依赖区域的阈值选取

1. 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

借助邻域性质变换原来的直方图

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

依赖区域的阈值选取

1. 直方图变换

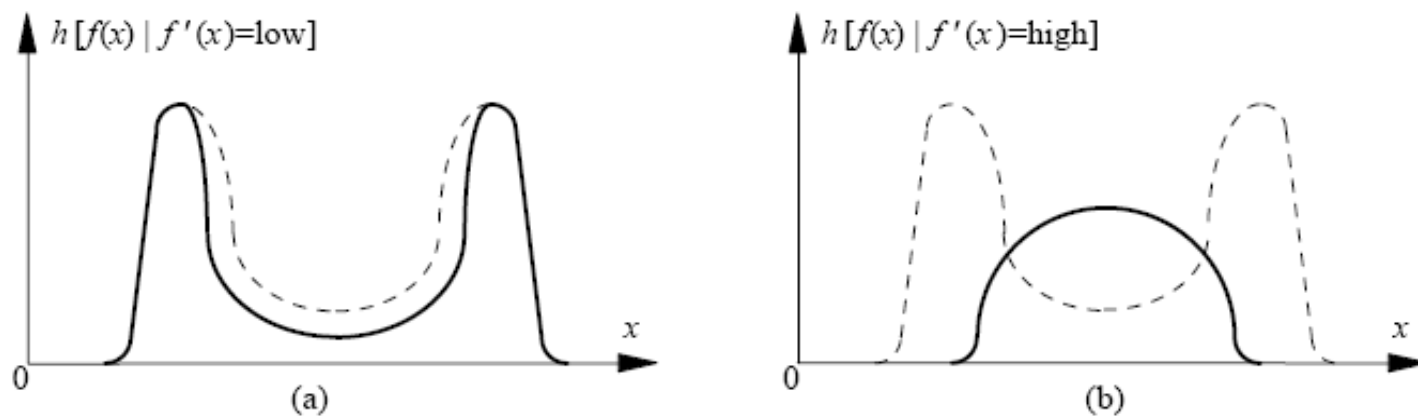
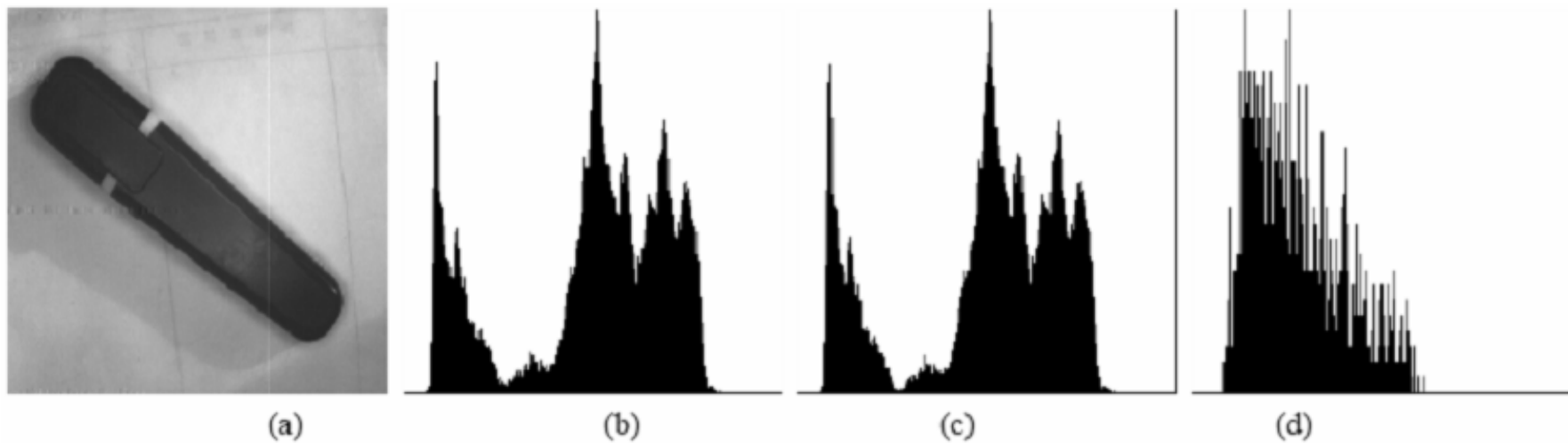


图 4.4.6 变换直方图示例

变换直方图实例

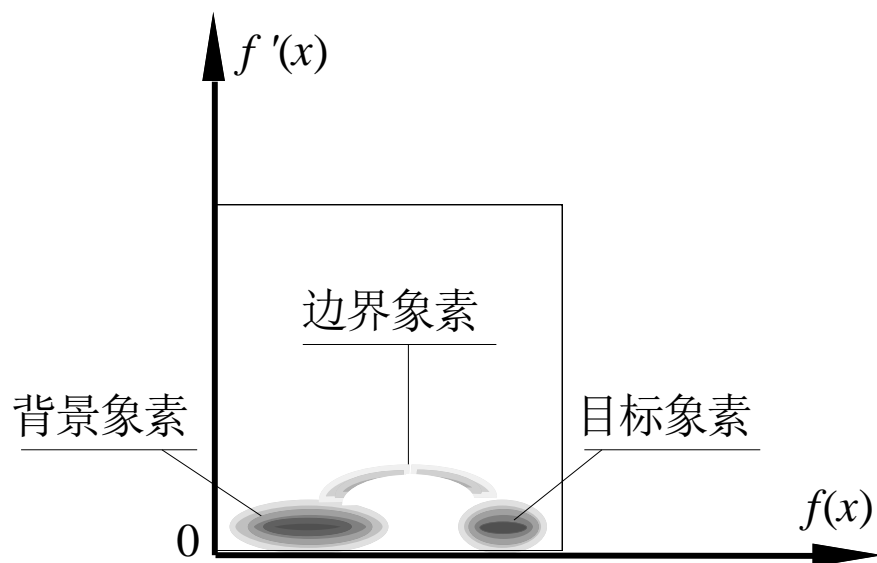


(a) 原图像；(b) 为其直方图；(c) 具有低梯度像素的直方图；(d) 具有高梯度像素的直方图

依赖区域的阈值选取

2、灰度-梯度散射图

2-D直方图



2-D图象

1个轴是灰度值轴

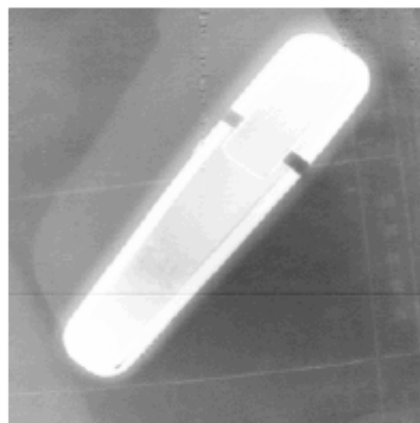
1个轴是梯度值轴

示例

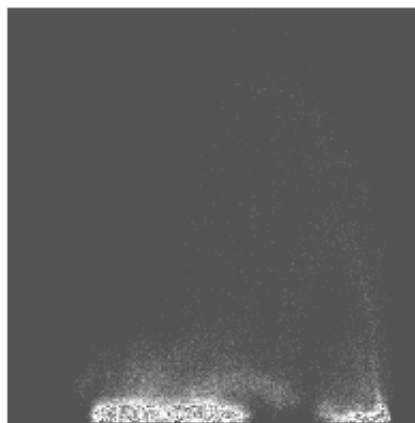
目标，背景聚类

依赖区域的阈值选取

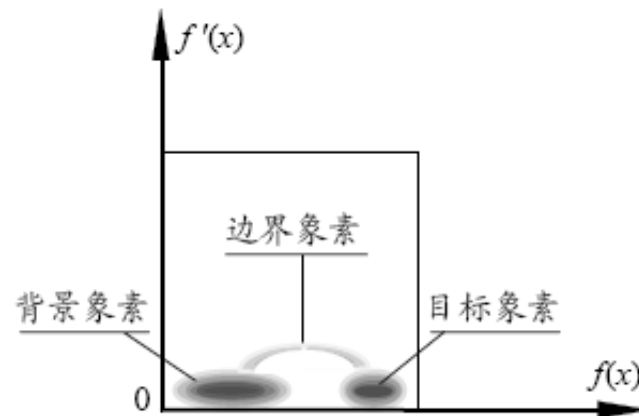
2、灰度-梯度散射图



(a)



(b)



(c)

图 4.4.8 灰度和梯度散射图



依赖坐标的阈值选取

- 全局阈值不能兼顾图象各处的情况
- 用与坐标相关的一系列阈值来对图象分割

基本思想/思路：

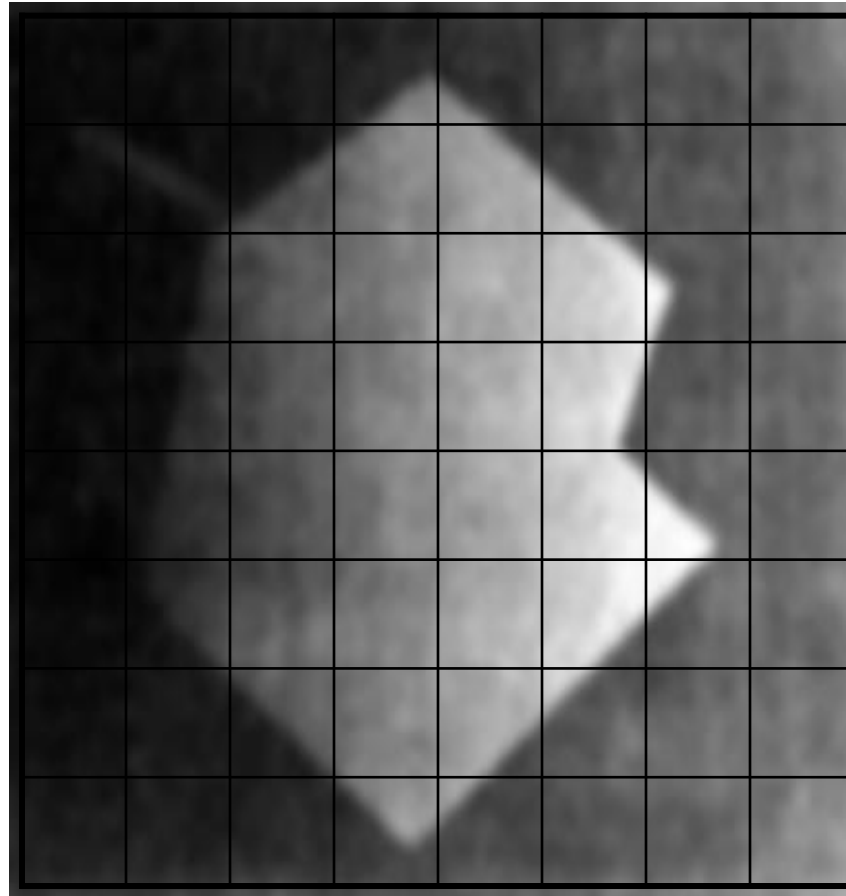
将图象分解成一系列子图象

对每个子图象计算一个阈值

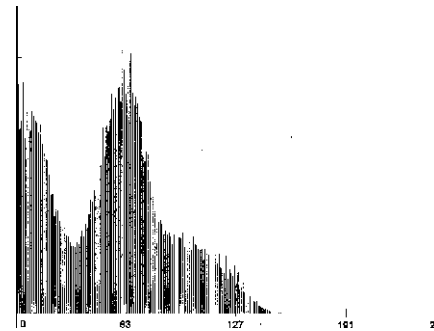
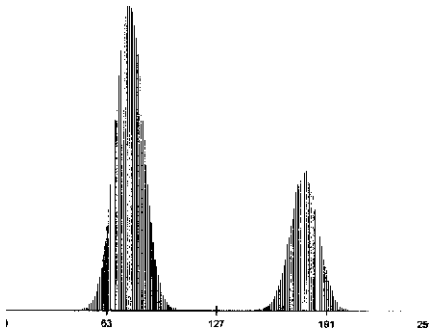
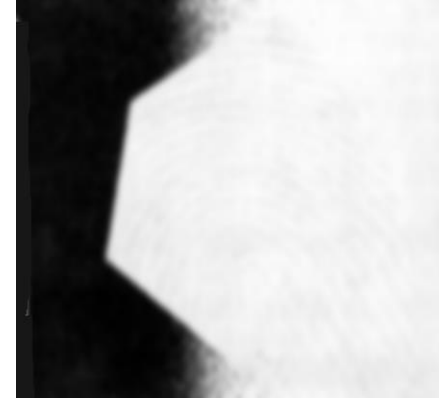
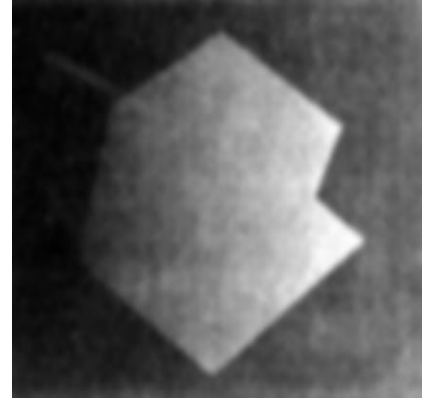
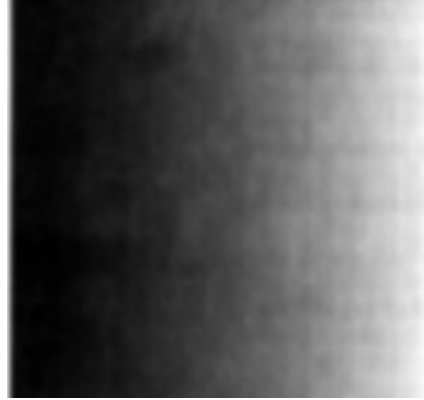
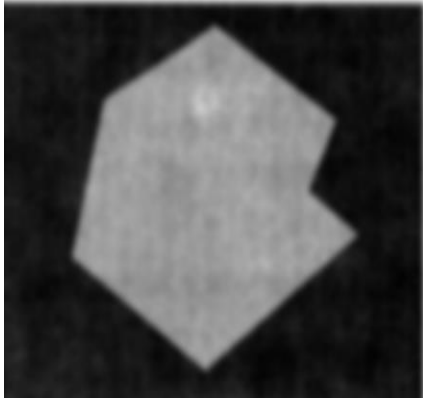
对这些子图象阈值进行插值

用插值结果（阈值曲面）进行分割

依赖坐标的阈值选取



例：光照不均匀对分割的影响



变化阈值法

- 将整幅图像分成一系列互相之间有50%重叠的子图像
- 做出每个子图像的直方图
- 检测各个子图像的直方图是否为双峰的，如是采用最优阈值法确定一个阈值，否则不进行处理
- 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
- 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

分块取阈值一例



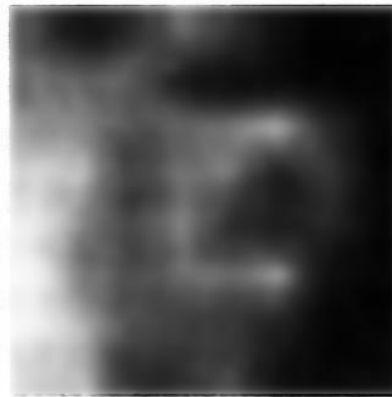
(a)



(b)



(c)



(d)



(e)



特征直方图

□ 图象的特征直方图可以是：

■ 灰度直方图

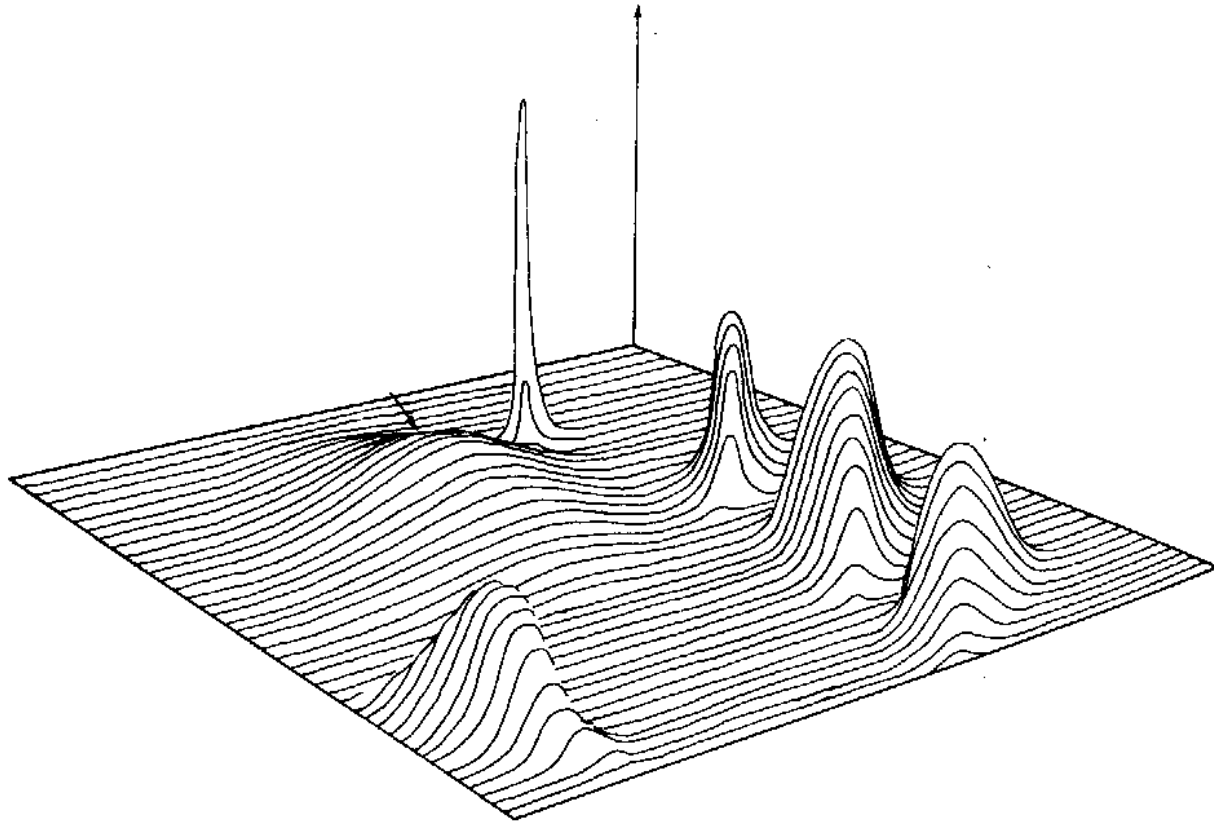
■ 梯度直方图

■ 纹理直方图

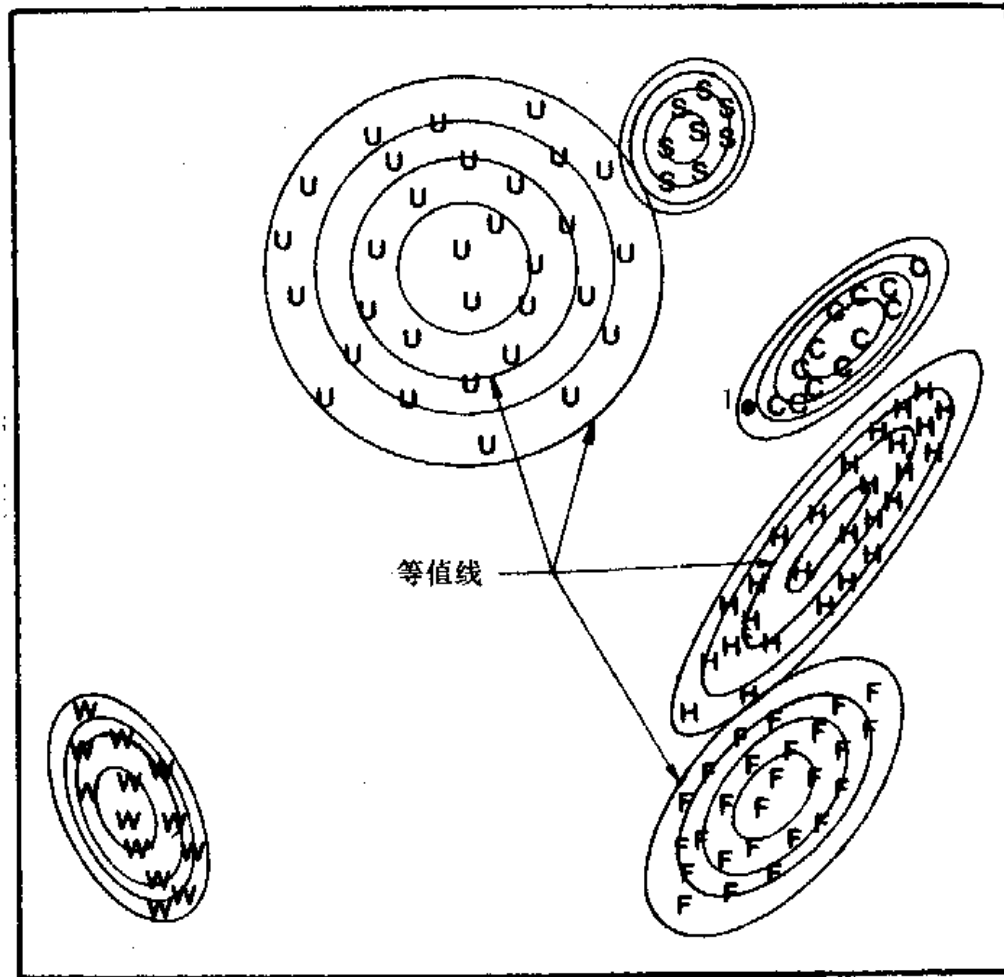
.....

□ 多个特征组成多维直方图

二维直方图示例



基于二维直方图的分割





5.2 区域生长法

基本思想：

将相似像素结合起来构成区域

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
- (2) 生长准则依赖应用

5.2 区域生长法

生长示例

- (1) 根据直方图选取聚类中心的像素为种子
- (2) 根据与种子像素灰度差 ($< T$) 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

5.3 分裂合并

主要步骤

- ◆ 先把图象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）

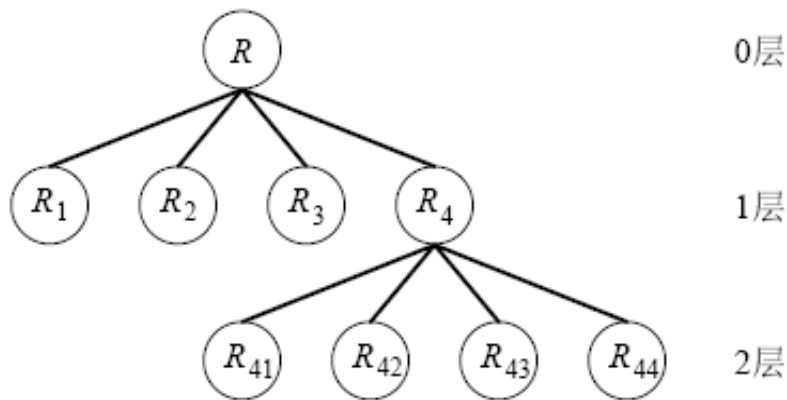
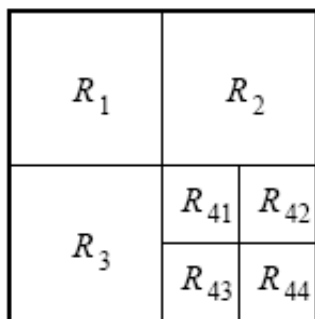


图 4.5.2 图像的四叉树表达法

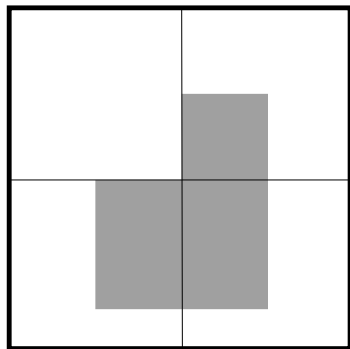


5.3 分裂合并

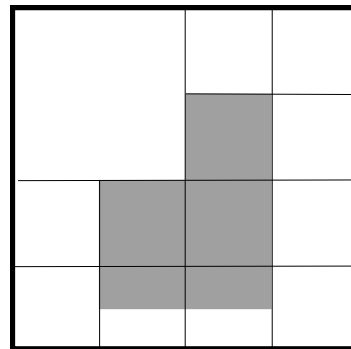
- ◆ 令 R 代表整个图象区域， P 代表逻辑谓词
- ◆ 把 R 连续地分裂成越来越小的 $1/4$ 的正方形子区域 R_i ，并且始终使 $P(R_i) = \text{TRUE}$
- 1. 对任一个区域 R_i ，如果 $P(R_i) = \text{FALSE}$ ，就将其分裂成不重叠的四等分
- 2. 对相邻的两个区域 R_i 和 R_j ，如果 $P(R_i \cup R_j) = \text{TRUE}$ ，就将它们合并起来
- 3. 如果进一步的分裂或合并都不可能了，则结束

5.3 分裂合并

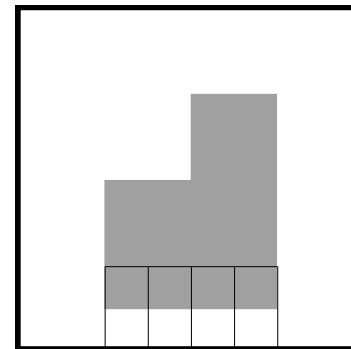
示例（四叉树）：分裂



(a)



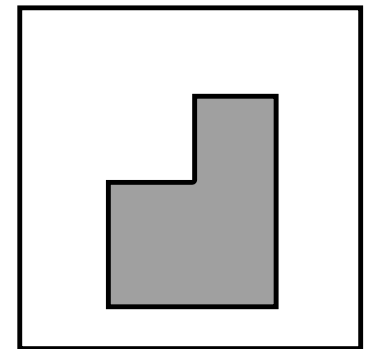
(b)



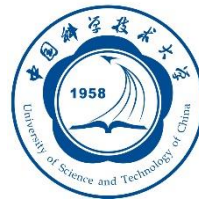
(c)

分裂

合并



(d)



分一合图象分割方法

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，如果它们之和的均匀性测度为真，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真的一对区域。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。

5.4 分水岭分割算法

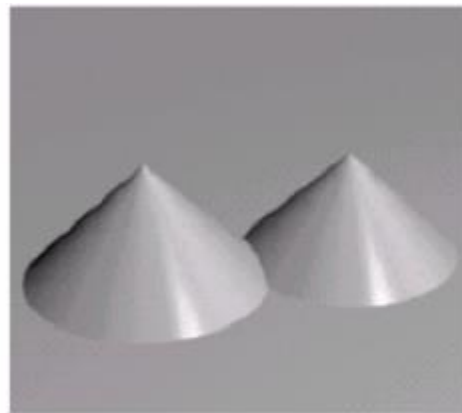
- 分水岭 (watershed, 也称分水线/水线)
- 把图象看成3-D地形的表示, 即2-D的地基 (对应图象空间) 加上第3维的高度 (对应图象灰度)



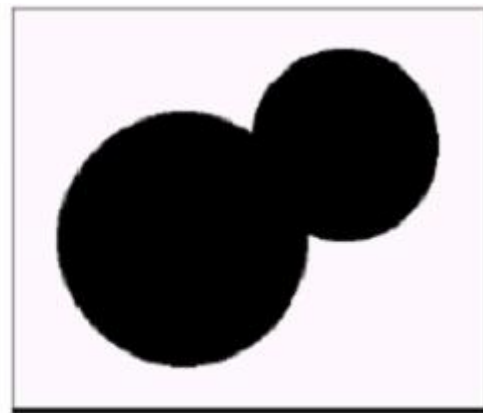
(a)



(b)



(c)



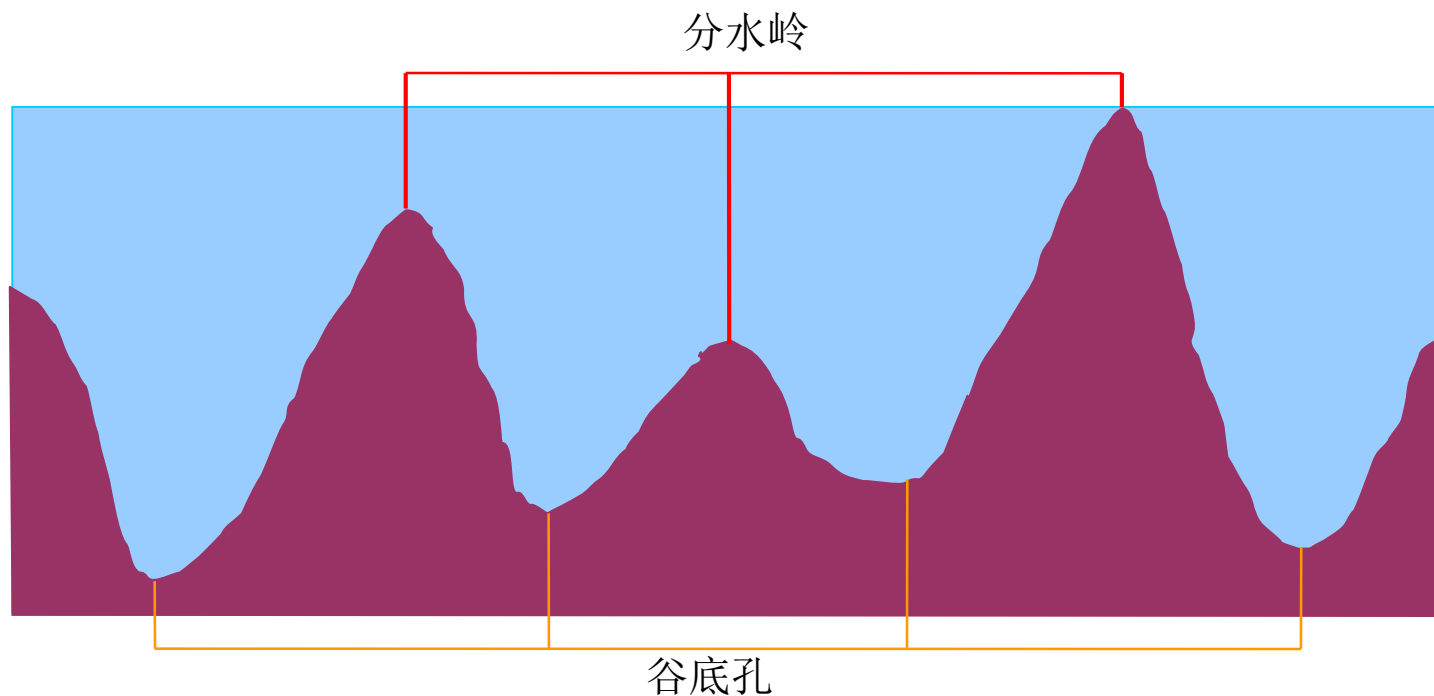
(d)

将图像看成地形图

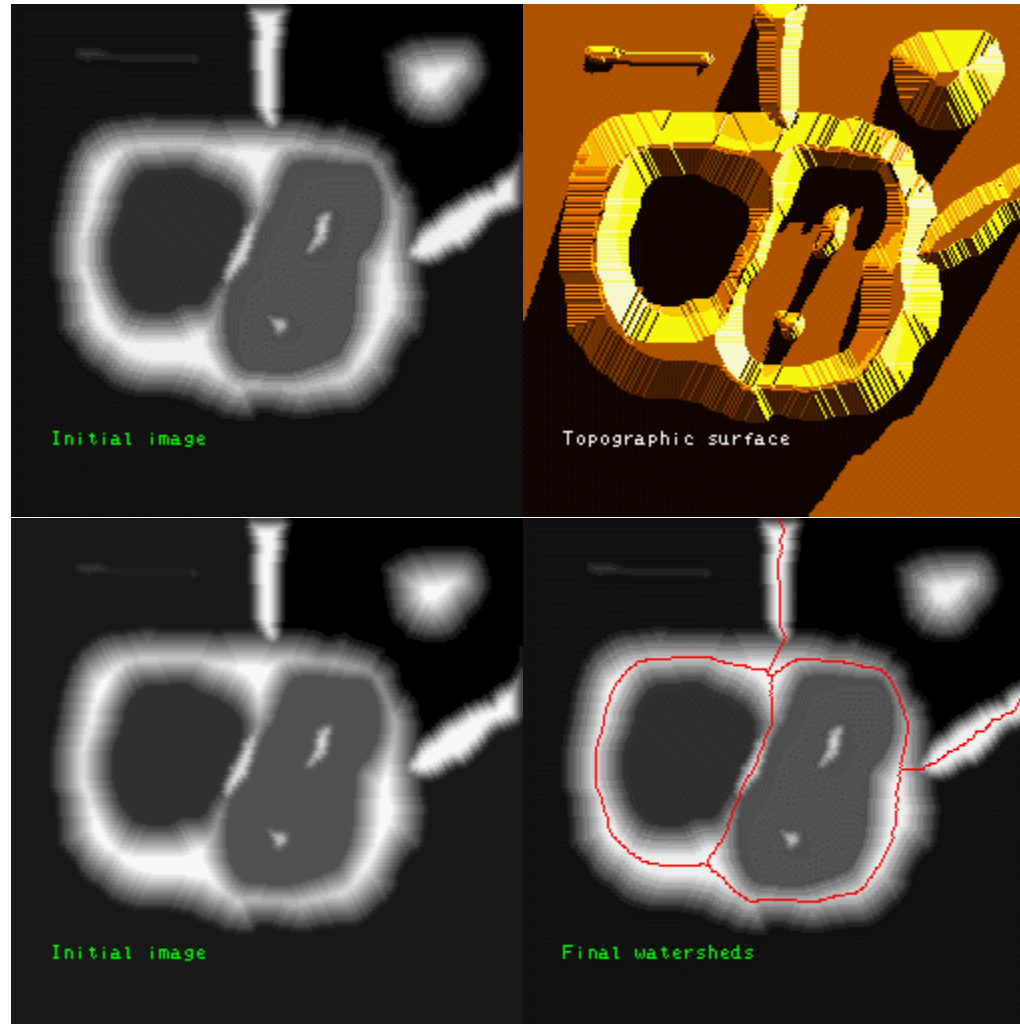
5.4.1 基本原理和步骤

1. 分水岭算法原理

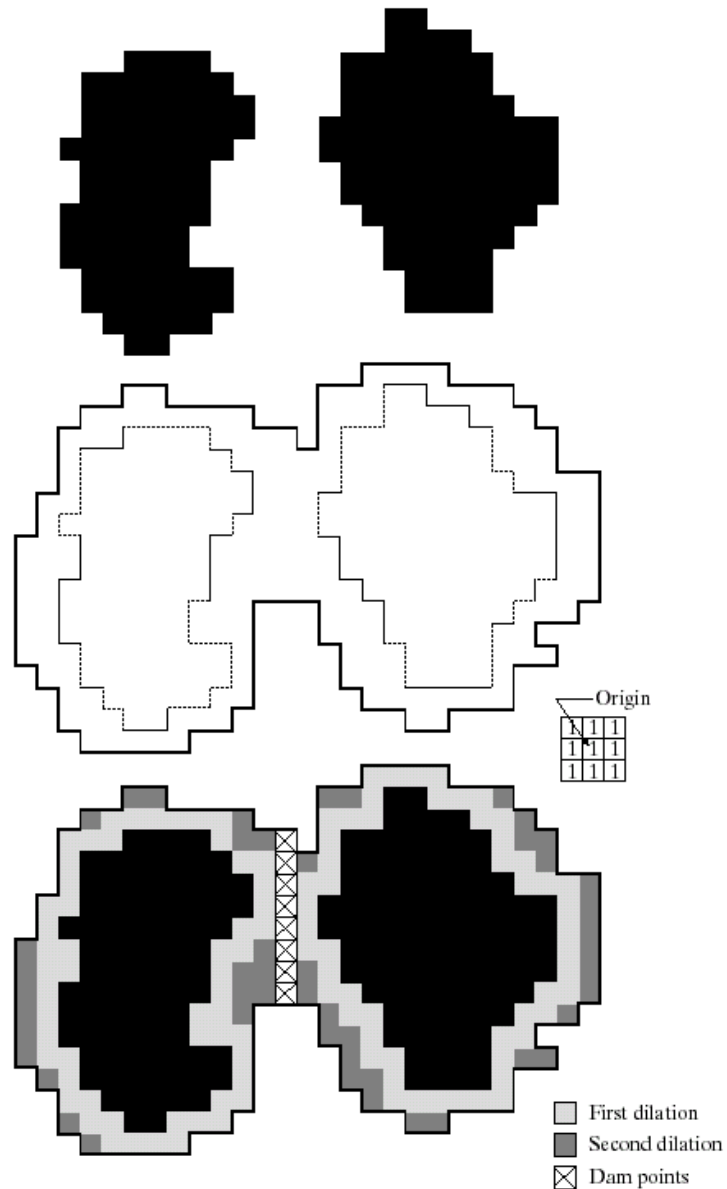
建立不同目标间的分水岭



分水岭算法过程演示



分水岭算法-构筑水坝





5.4.1 基本原理和步骤

2. 分水岭计算步骤

待分割图象 $f(x, y)$ ，其梯度图象为 $g(x, y)$

用 M_1, M_2, \dots, M_R 表示 $g(x, y)$ 中各局部极小值的象素位置， $C(M_i)$ 为与 M_i 对应的区域中的象素坐标集合。

用 n 表示当前灰度阈值， $T[n]$ 代表记为 (u, v) 的象素集合， $g(u, v) < n$ ，

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

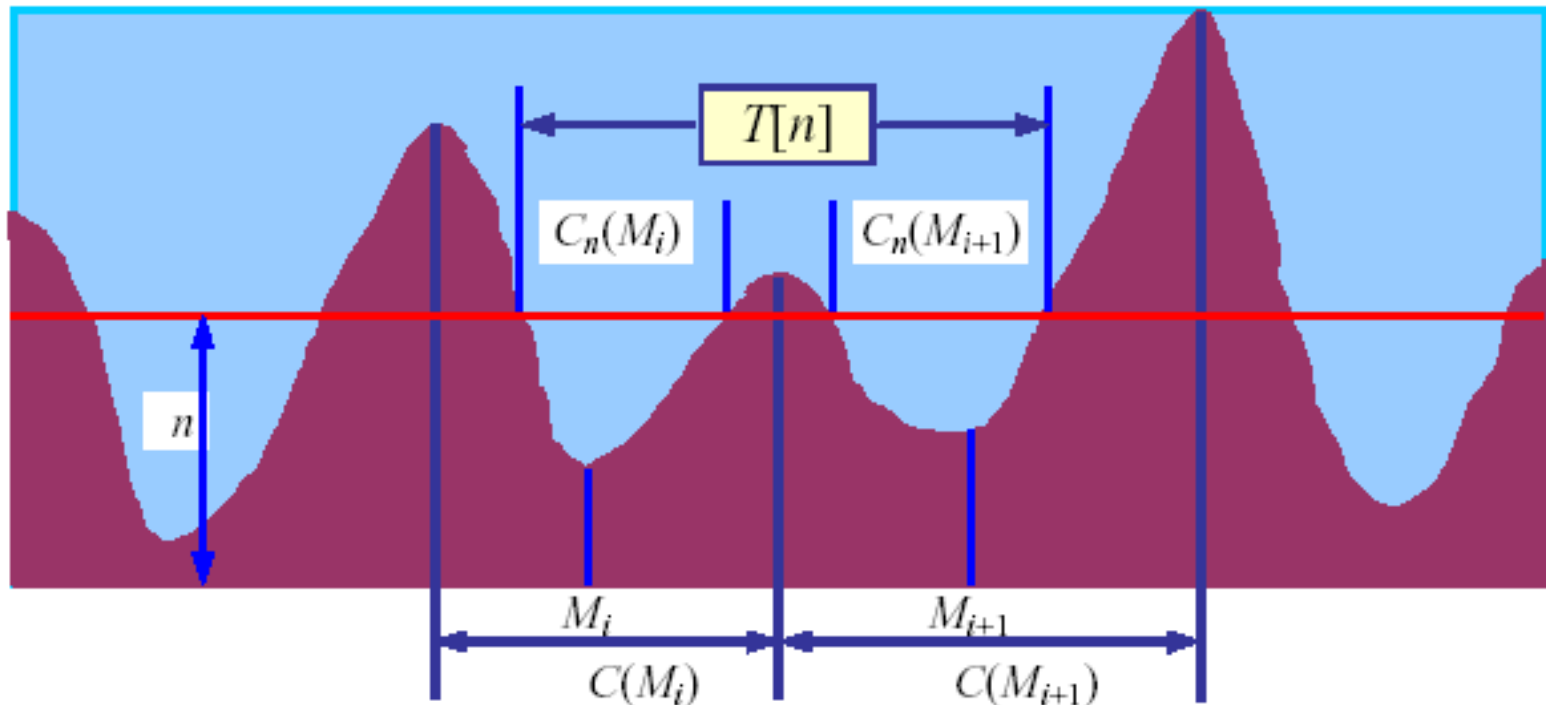
对 M_i 所在的区域，其中满足条件的坐标集合 $C_n(M_i)$ 可看作一幅二值图象

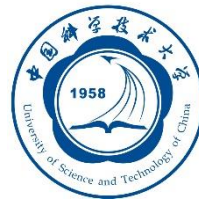
$$C_n(M_i) = C(M_i) \cap T[n]$$

5.4.1 基本原理和步骤

2. 分水岭计算步骤

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





5.4.1 基本原理和步骤

2. 分水岭计算步骤

用 $C[n]$ 代表在灰度阈值为 n 时图象中所有满足条件的像素

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

$C[\max + 1]$ 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, $C[n]$ 是 $T[n]$ 的子集, 所以 $C[n - 1]$ 又是 $T[n]$ 的子集

5.4.1 基本原理和步骤

2. 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
- (2) $s \cap C[n-1]$ 里包含 $C[n-1]$ 中的一个连通组元
- (3) $s \cap C[n-1]$ 里包含 $C[n-1]$ 中一个以上的连通组元

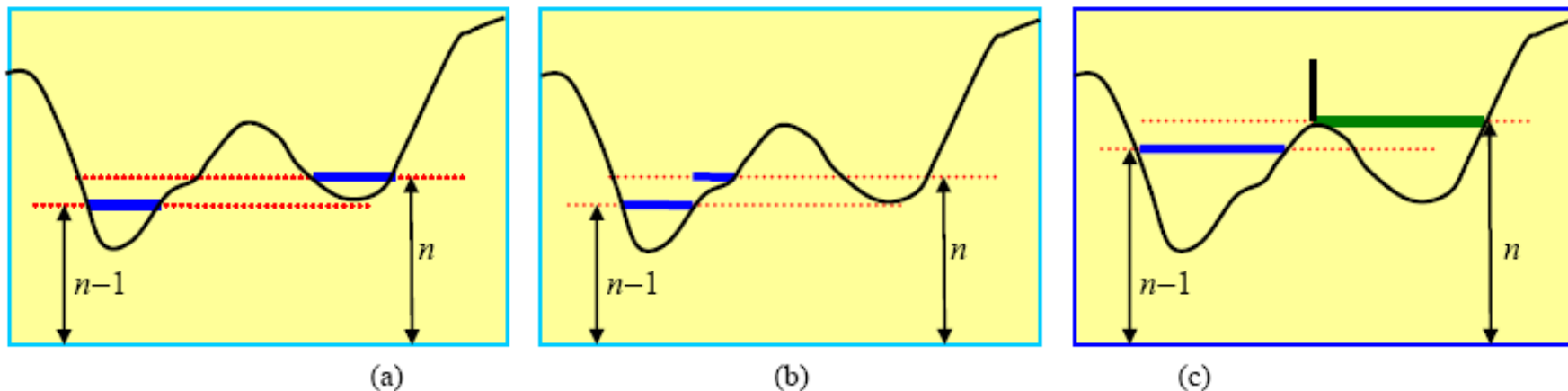


图 5.4.5 计算 $C_n(M_i)$ 的示意图

5.4.1 基本原理和步骤

2. 分水岭计算步骤

分别处理：

- (1) $C[n]$ 可由把连通组元 s 加到 $C[n-1]$ 中得到
- (2) $C[n]$ 可由把连通组元 s 加到 $C[n-1]$ 中得到
- (3) 需要在 s 中建分水岭

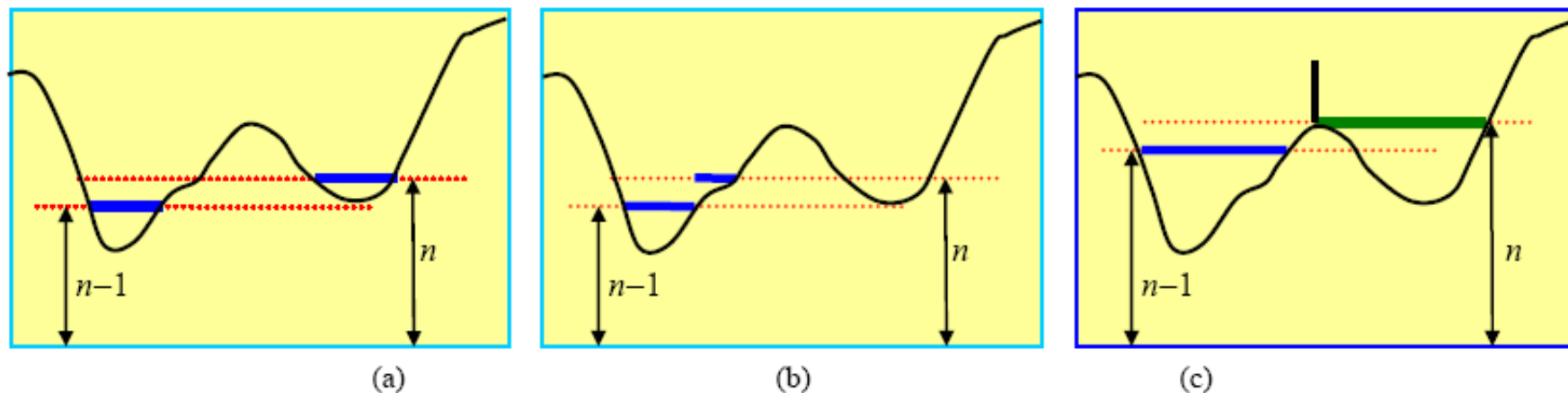
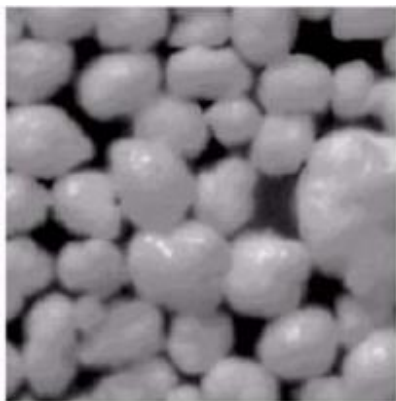


图 5.4.5 计算 $C_n(M_i)$ 的示意图

5.4.1 基本原理和步骤

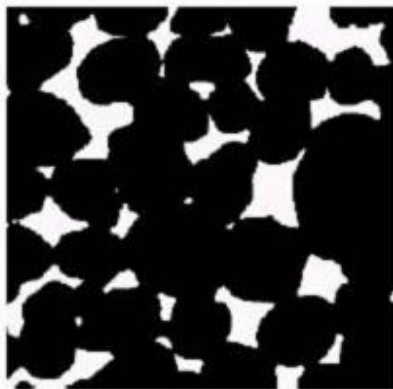
2. 分水岭计算步骤

原始图



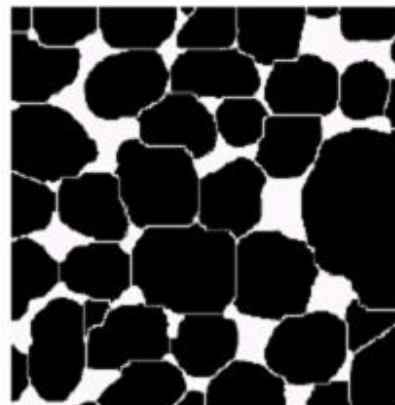
(a)

阈值分割



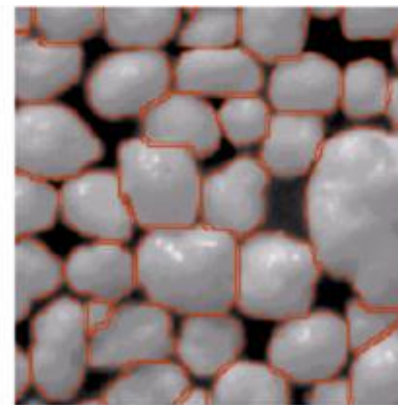
(b)

分水岭



(c)

叠加轮廓

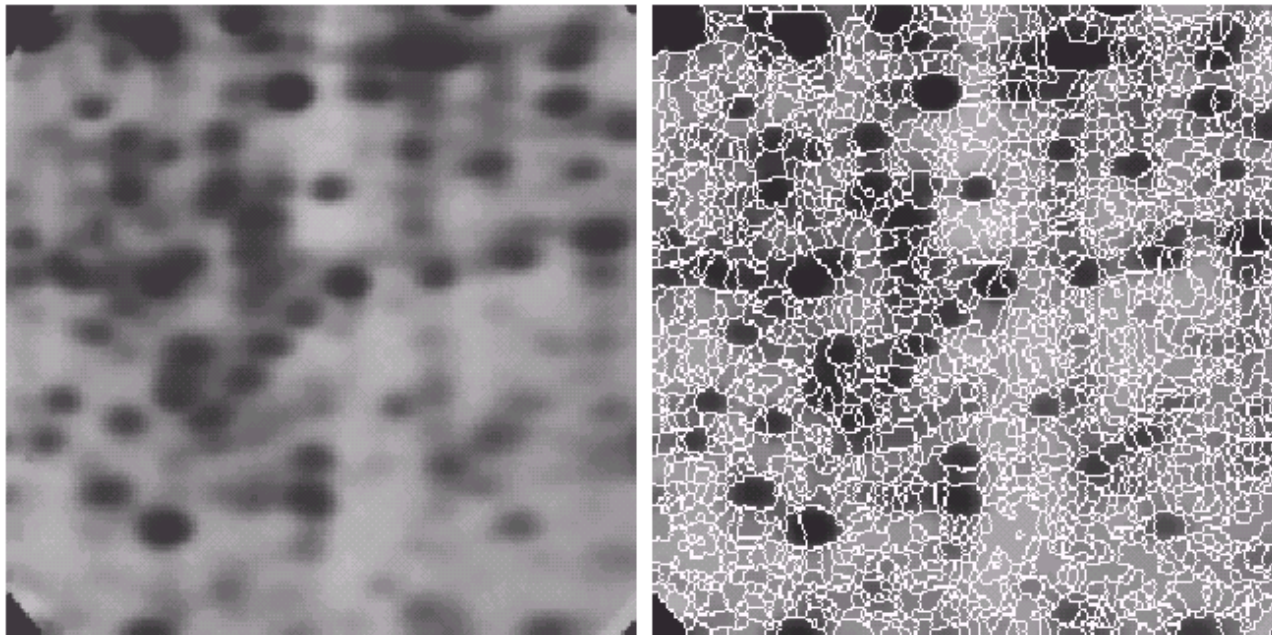


(d)

分水岭算法分割实例

5.4.2 算法改进

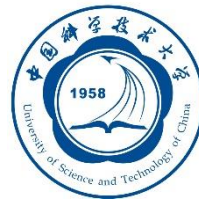
过分割 (over-segmentation)



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.)



5.4.2 算法改进

1. 利用标记控制分割

- 过分割 (over-segmentation)

受图象中噪声和其它不规则结构影响

- 利用标记 (marker)

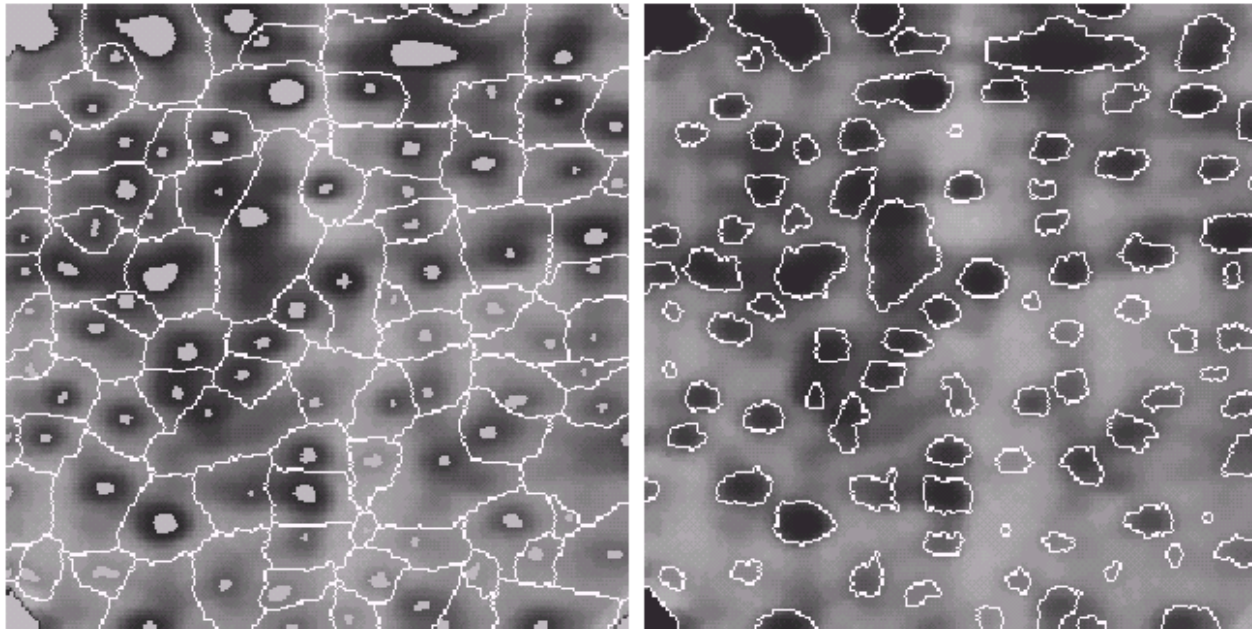
图象中的一个连通组元

内部标记：对应目标

外部标记：对应背景（分水岭）

5.4.2 算法改进

1. 利用标记控制分割



a b

FIGURE 10.48

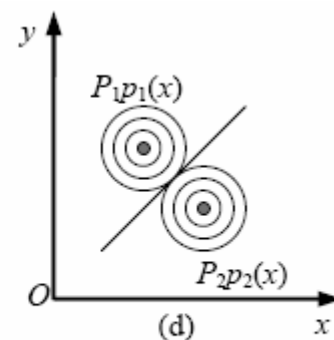
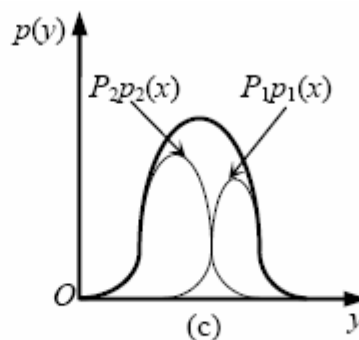
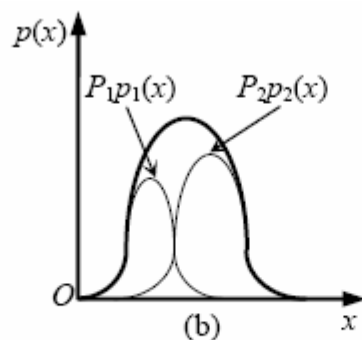
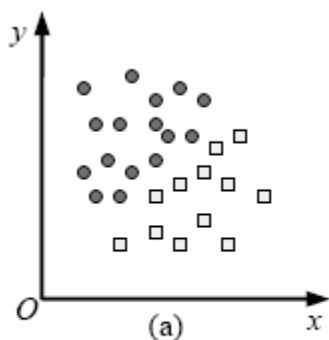
(a) Image showing internal markers (light gray regions) and external markers (watershed lines). (b) Result of segmentation. Note the improvement over Fig. 10.47(b). (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

5.5 聚类分割方法

分割：像素分类

特征空间聚类（取阈值是1-D聚类）

高维空间聚类（分类能力较强）



高维空间聚类的优点



K-means 聚类算法

1) 选择K个初始分类中心 $\{\mu_1, \dots, \mu_K\}$;

2) 使用最小距离法将所有样本分类:

若 $\forall j \neq i, \text{Dist}(\mathbf{x} - \mu_i) < \text{Dist}(\mathbf{x} - \mu_j)$, 则将 \mathbf{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?
2. 如何初始化聚类中心;
3. 如何定义距离测度?



K-means 聚类形式化分析

□ Given a dataset $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ $\mathbf{x}_i \in \mathbb{R}^D$

□ **Goal**: Partition the dataset into K clusters denoted by $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\}$

□ **Formulation**:

$$\arg \min_{\substack{\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\} \\ \{r_{11}, \dots, r_{NK}\}}} J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

r_{nk} : assignment of a data to a cluster

□ **Solution**:

计算复杂度?

■ Fix $\{\boldsymbol{\mu}_k\}$, and solve $\{r_{nk}\}$: $r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\| \\ 0 & \text{otherwise} \end{cases}$

$\sim O(NKD)$

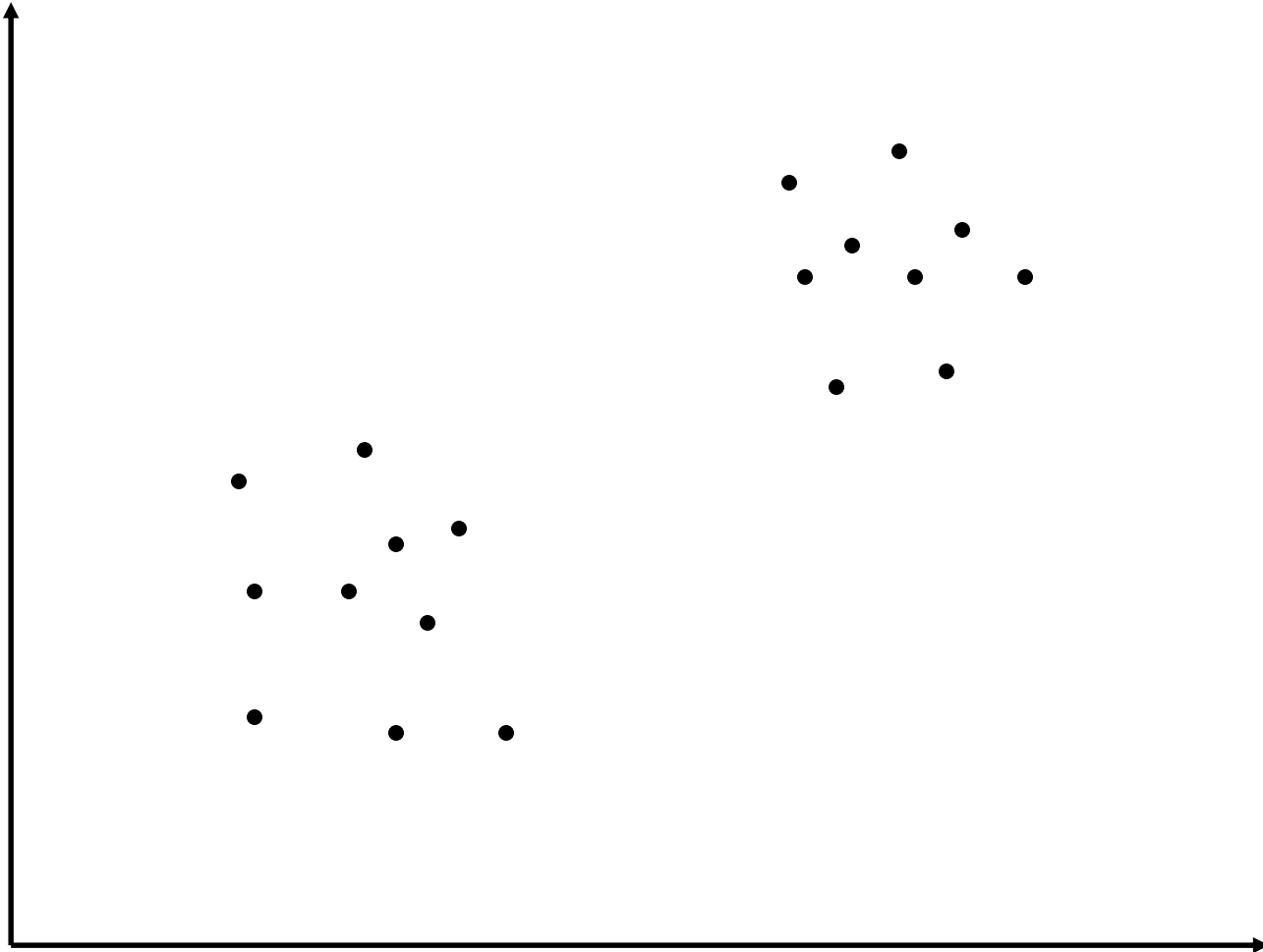
■ Fix $\{r_{nk}\}$, and solve $\{\boldsymbol{\mu}_k\}$: $\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$

$\sim O(ND)$

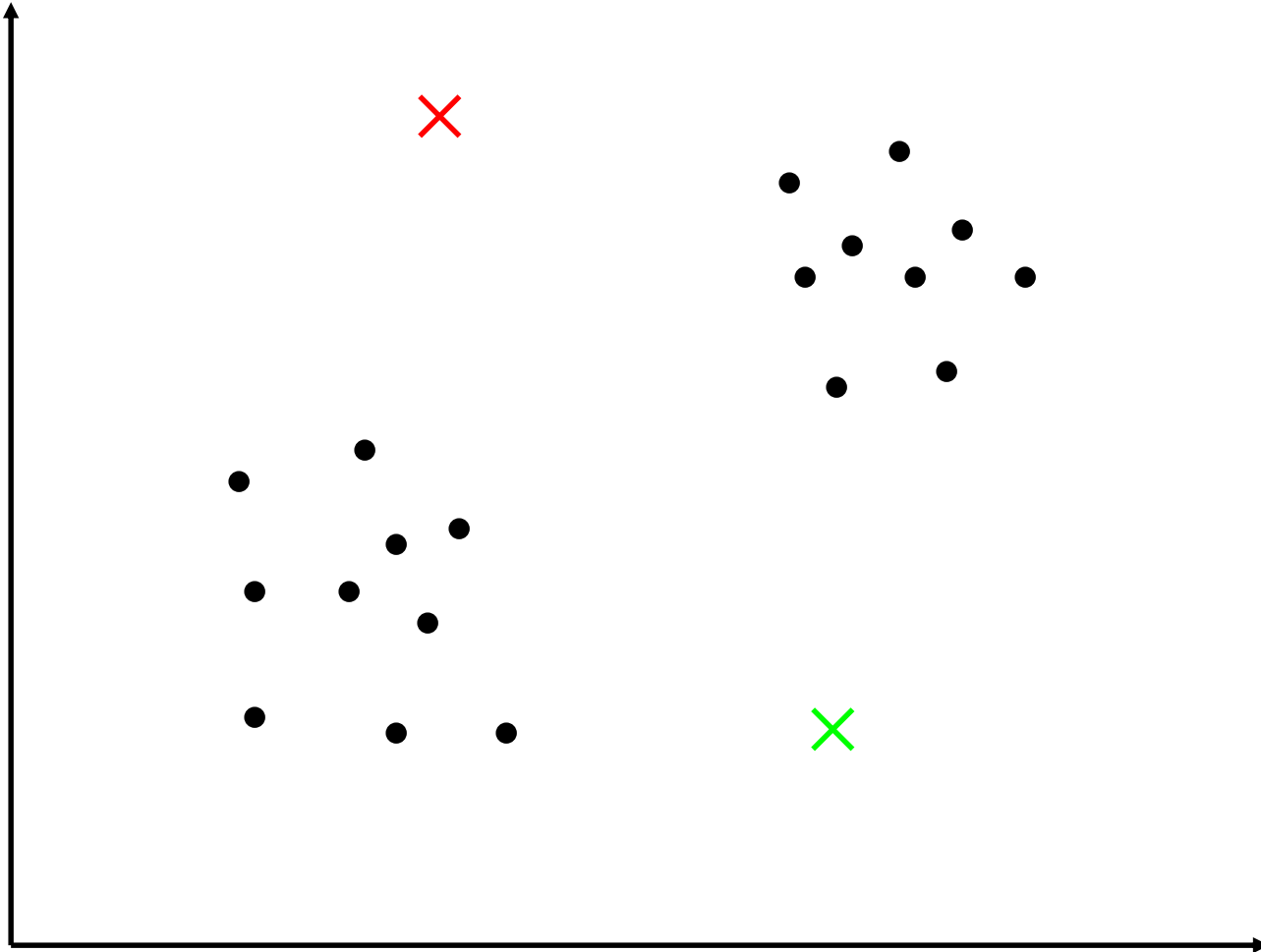
■ Iterate above two steps until convergence



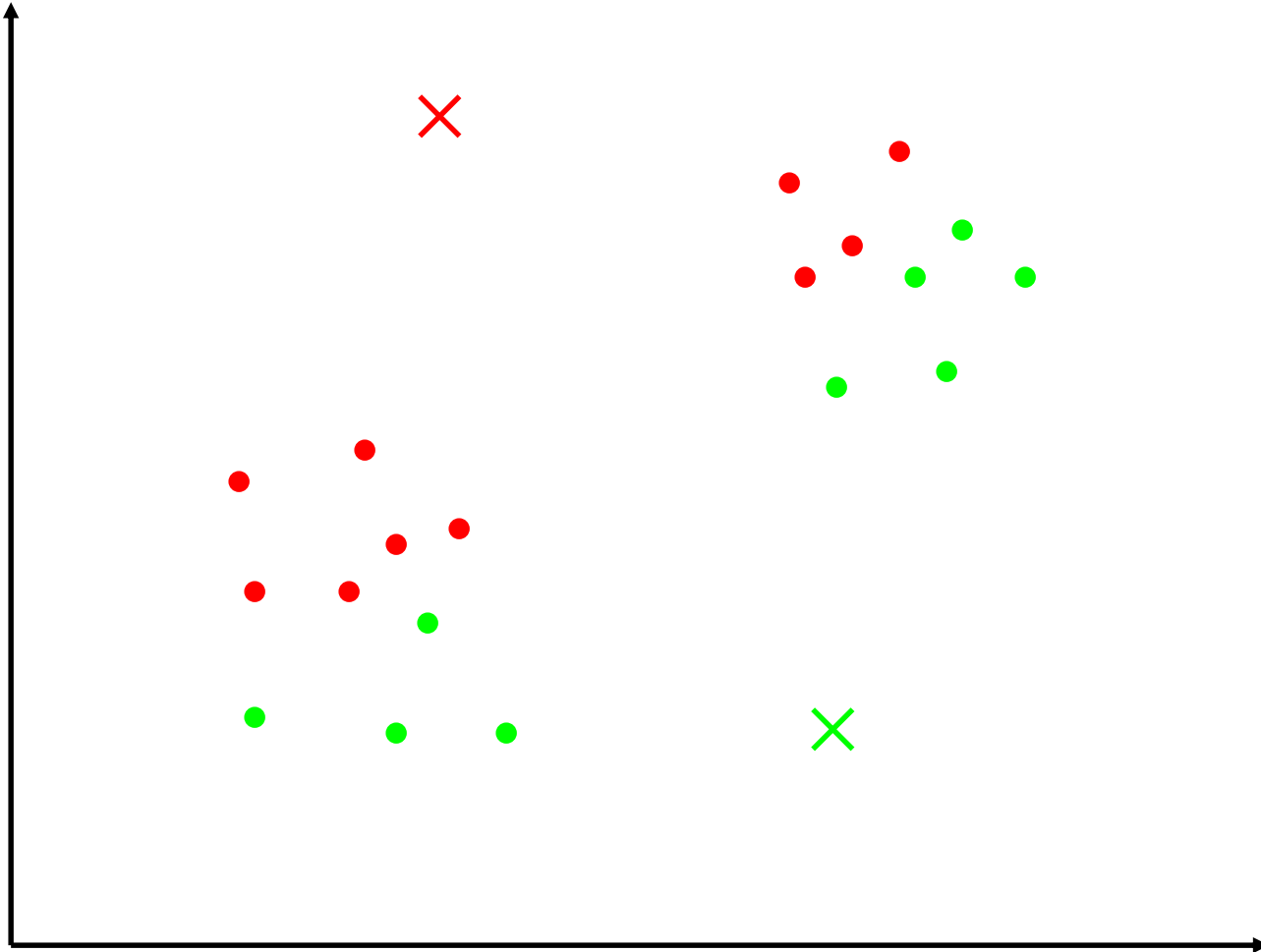
K-Means 聚类过程演示 I



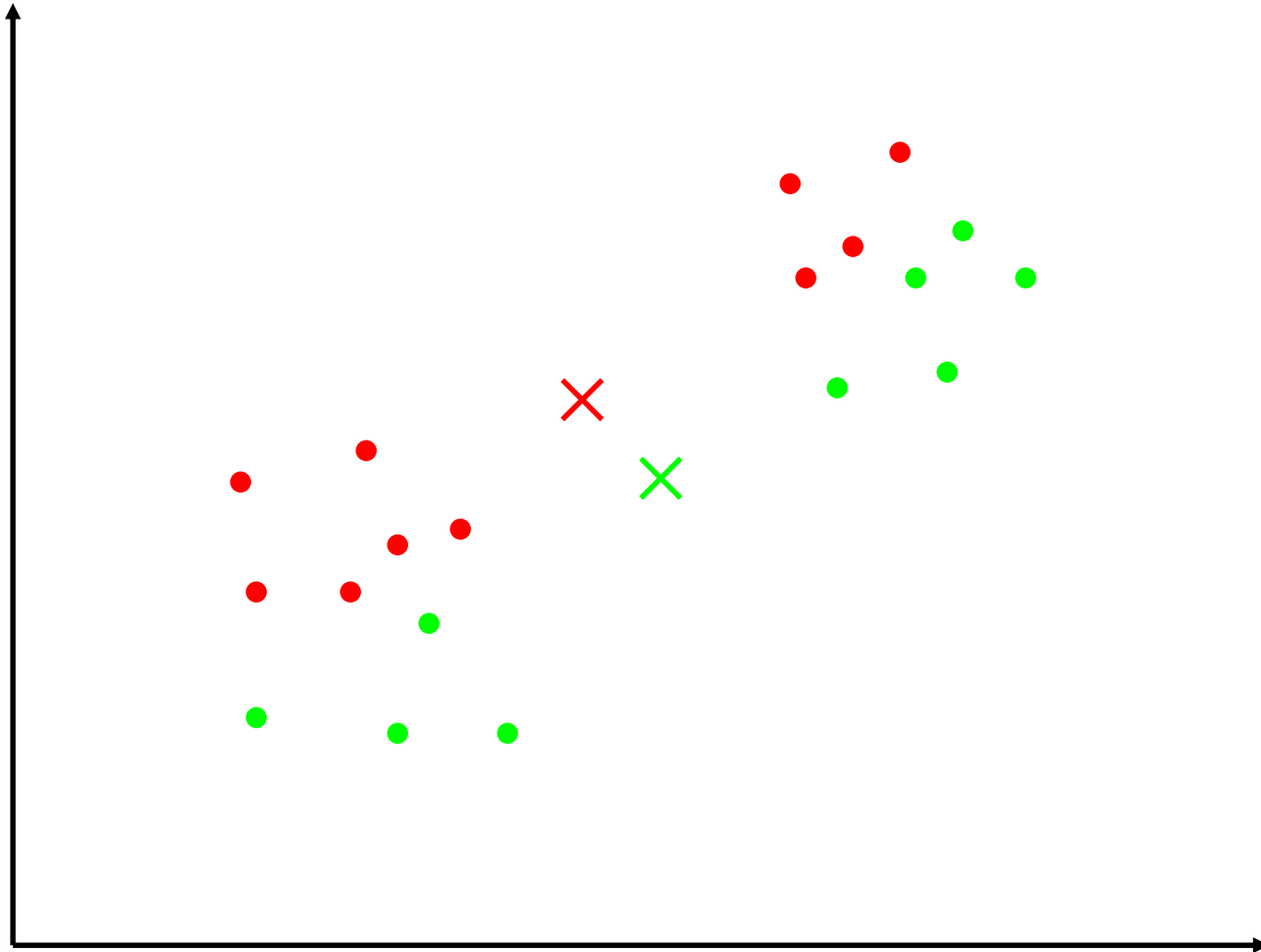
K-Means 聚类过程演示 I



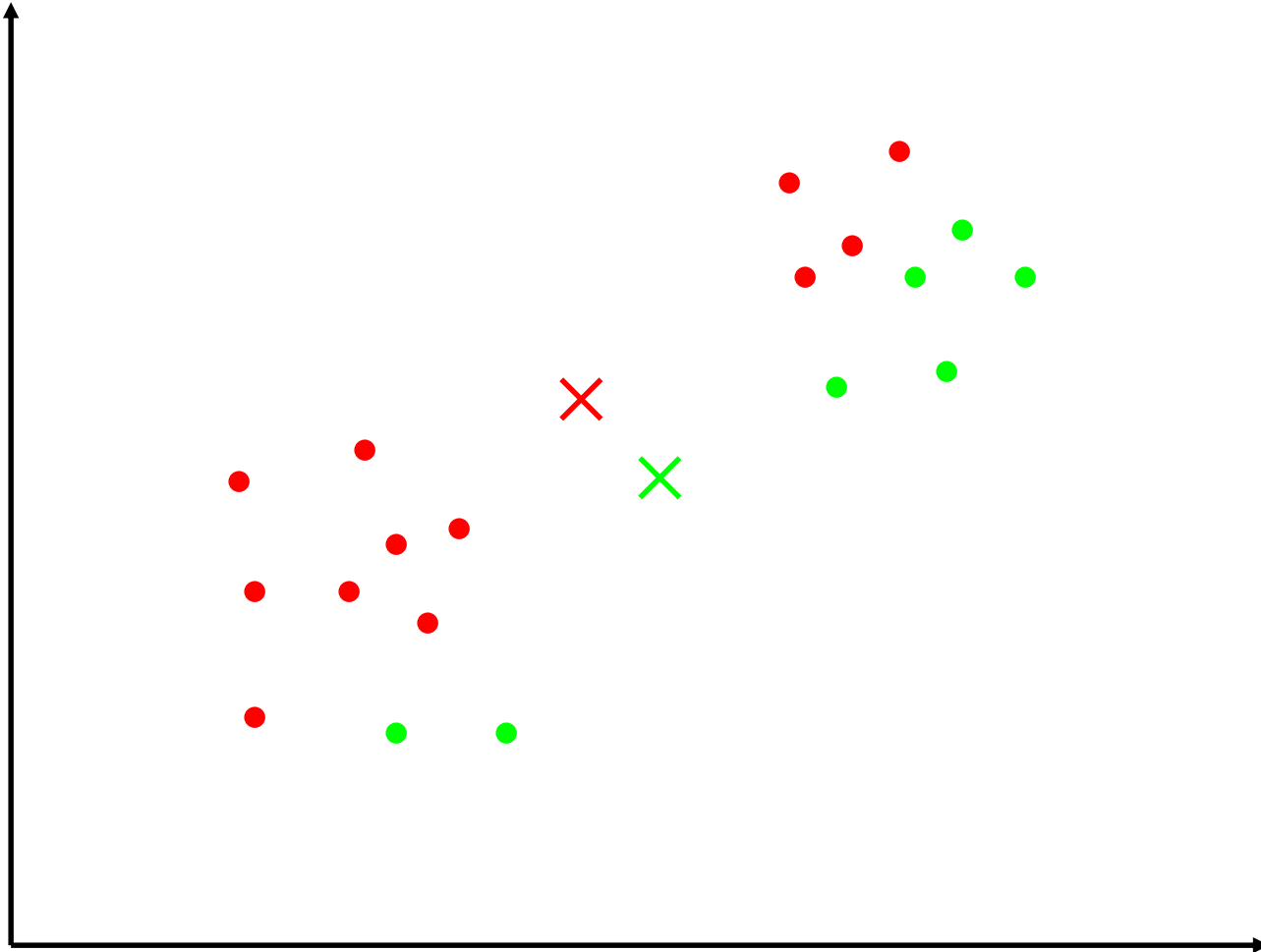
K-Means 聚类过程演示 I



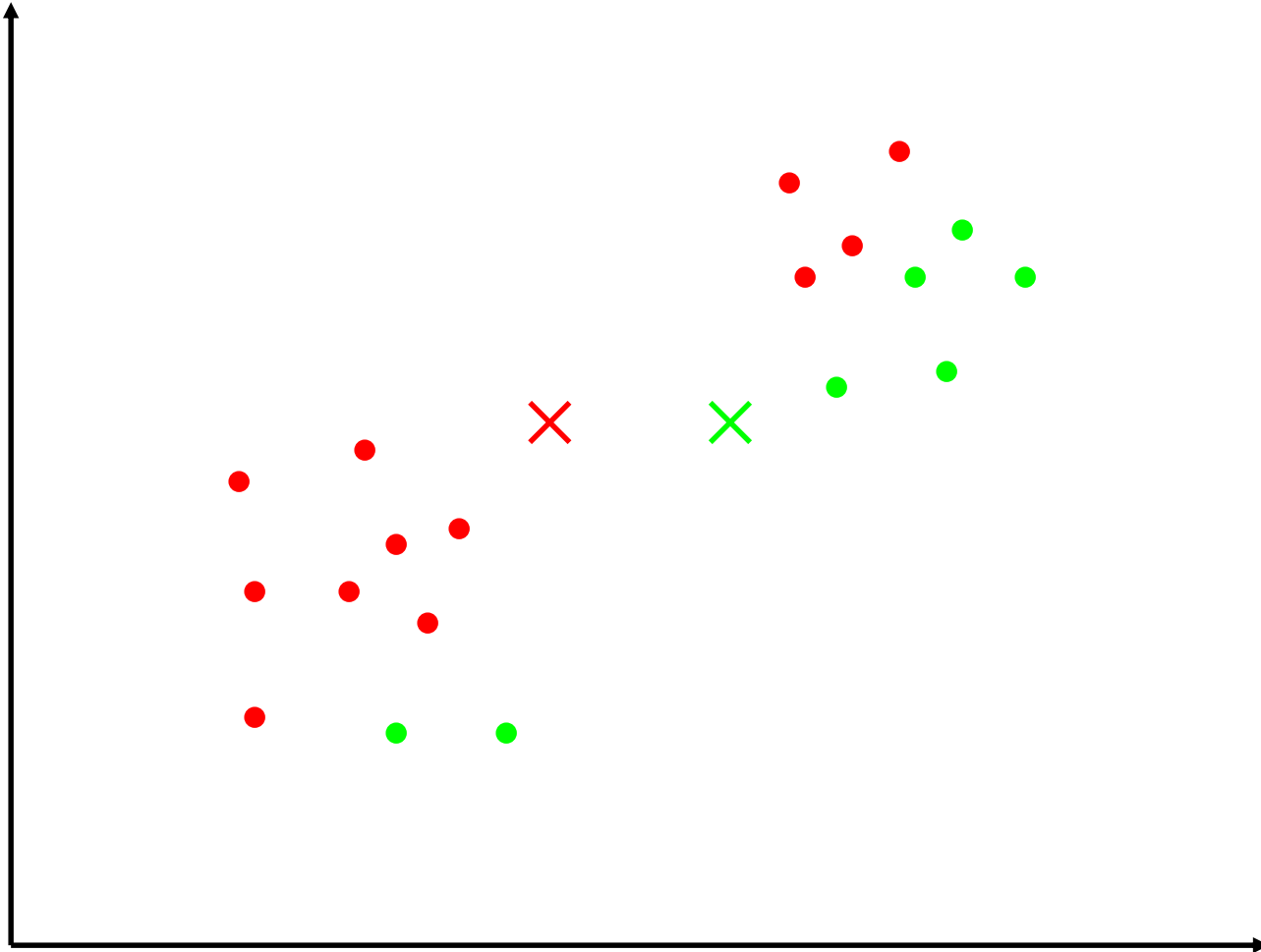
K-Means 聚类过程演示 I



K-Means 聚类过程演示 I

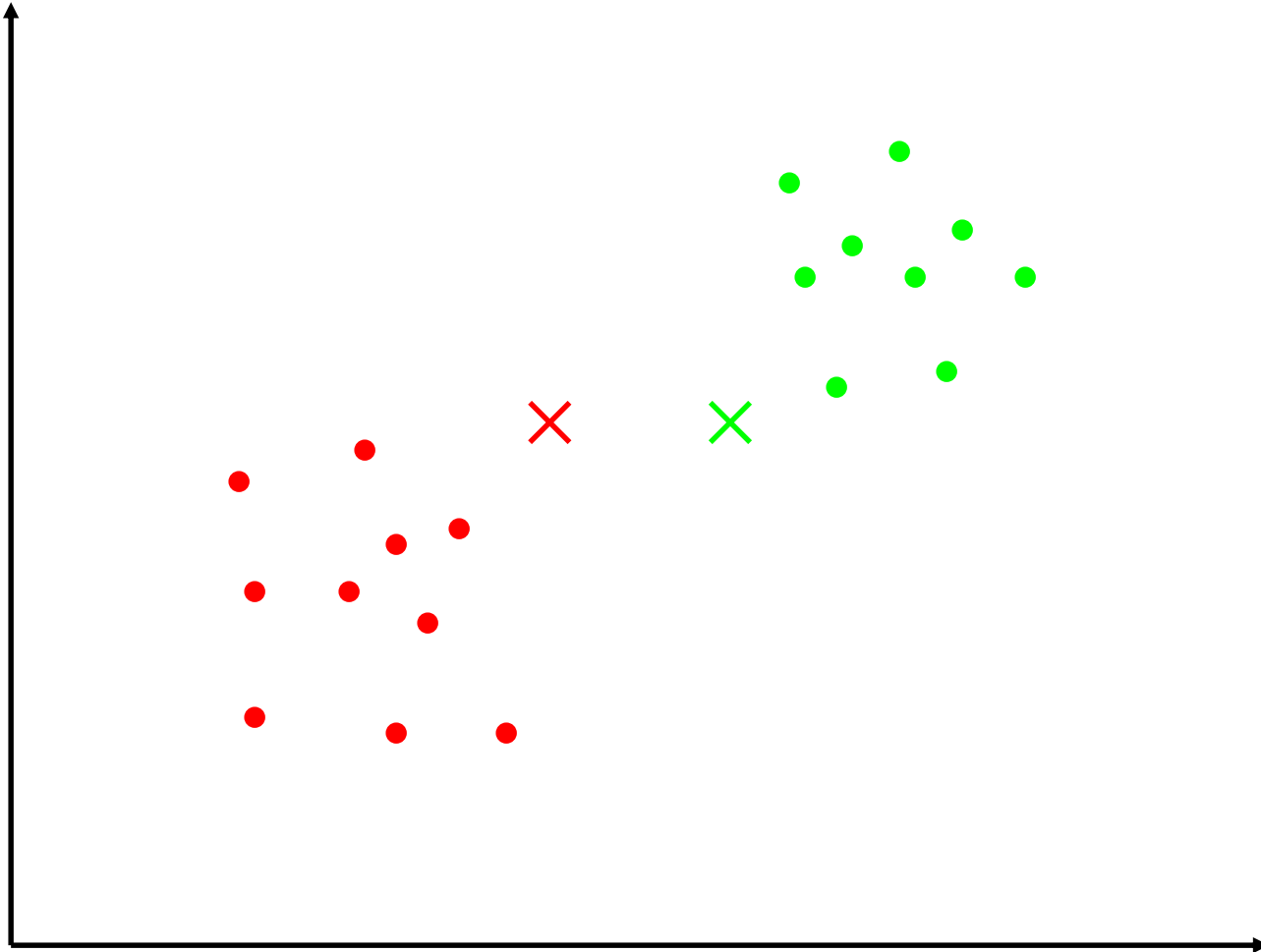


K-Means 聚类过程演示 I

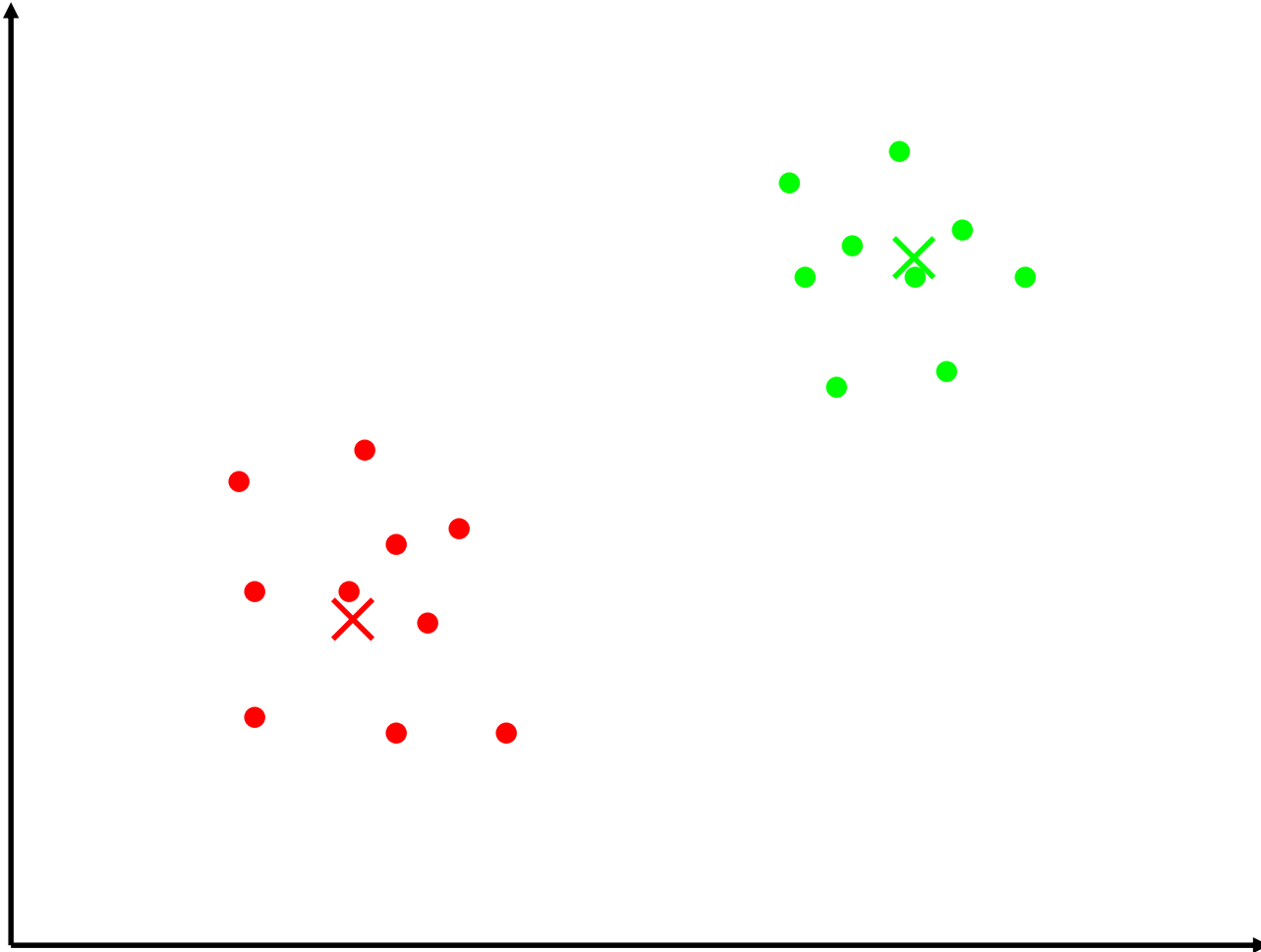




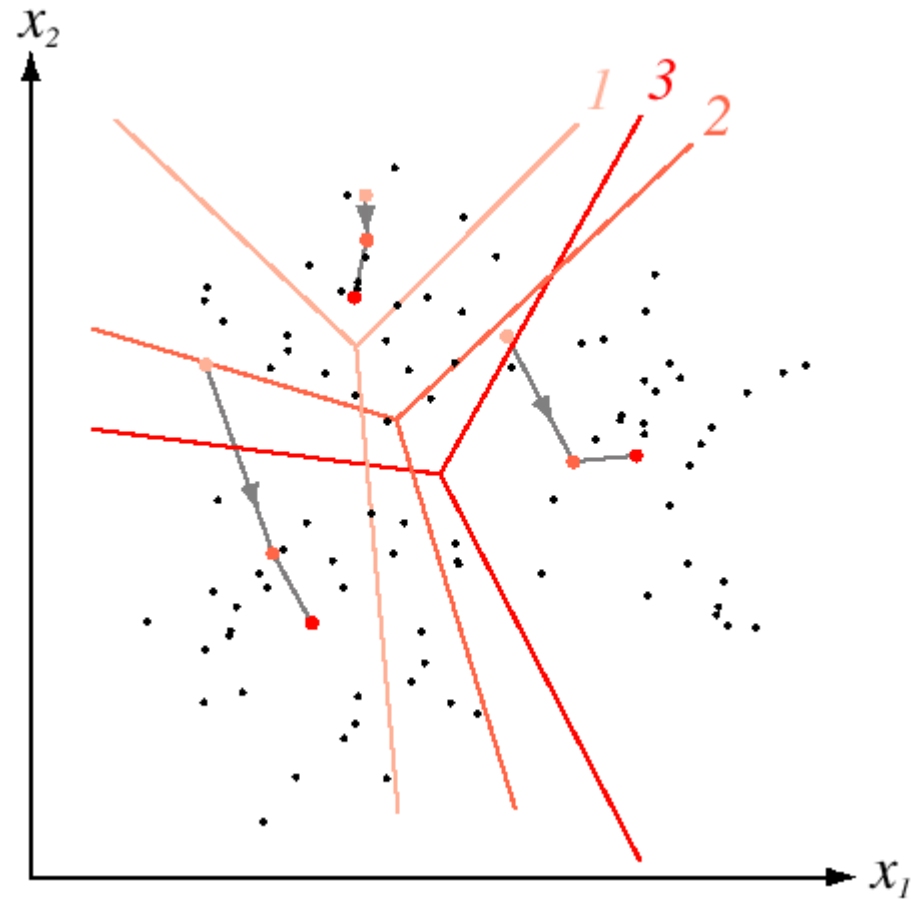
K-Means 聚类过程演示 I



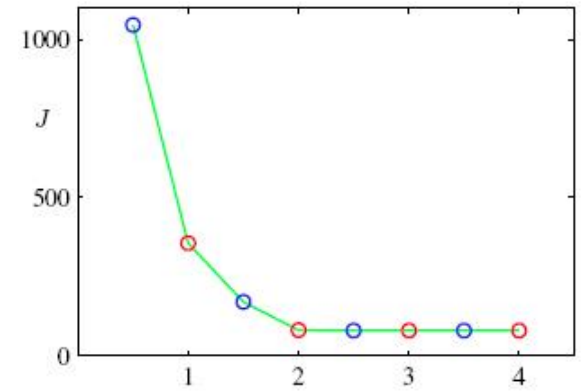
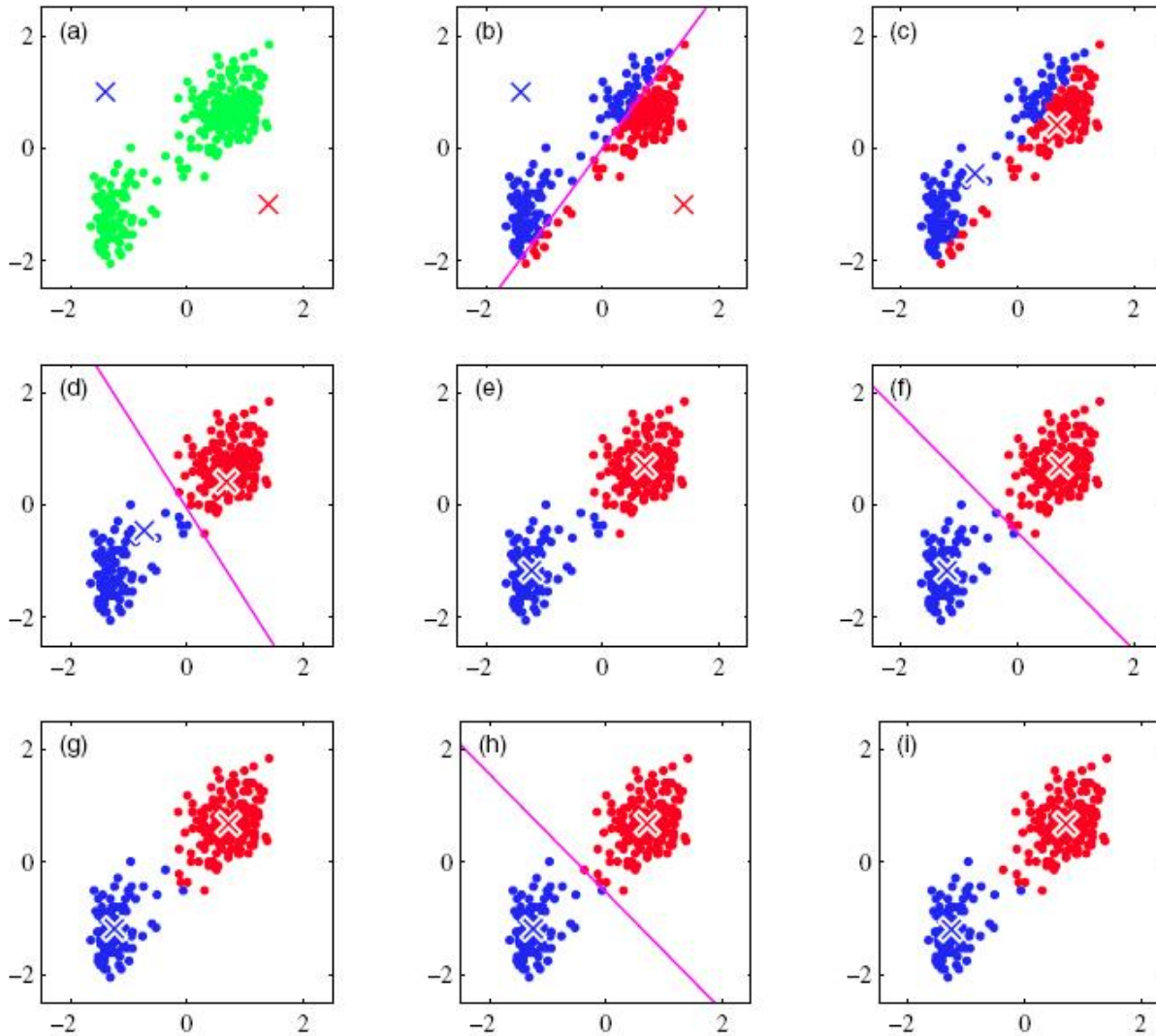
K-Means 聚类过程演示 I



K-Means 聚类过程演示 II



K-means algorithm 聚类过程演示



Plot of the cost function. The algorithm has converged after the third iterations



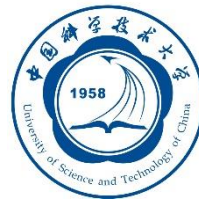
彩色图象的分割

1、彩色空间

面向处理分析的颜色模型-HSI

- (1) I 分量与图象的彩色信息无关
- (2) 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



彩色图象的分割

1、彩色空间

从RGB到HSI的转换关系

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

彩色图象的分割

1、彩色空间

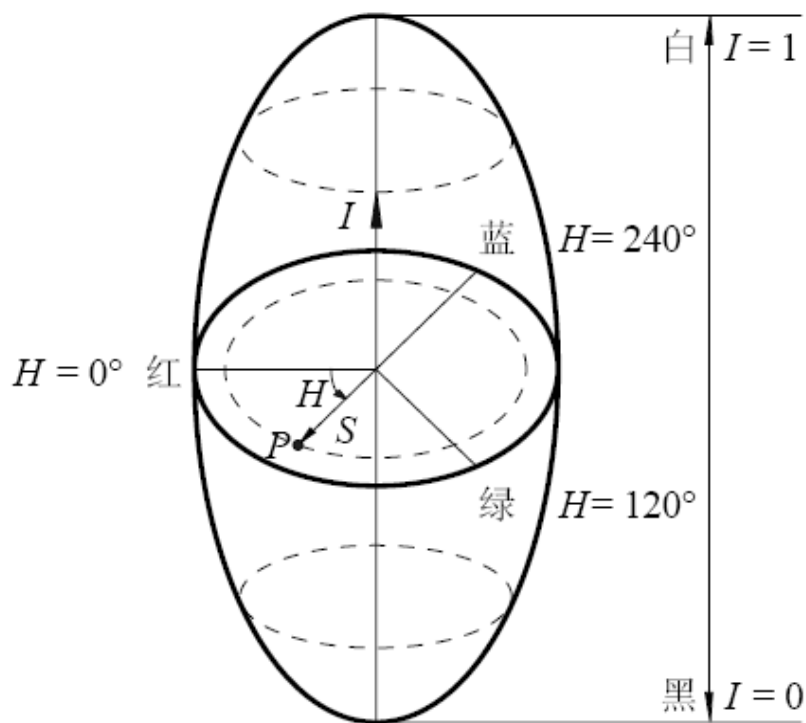
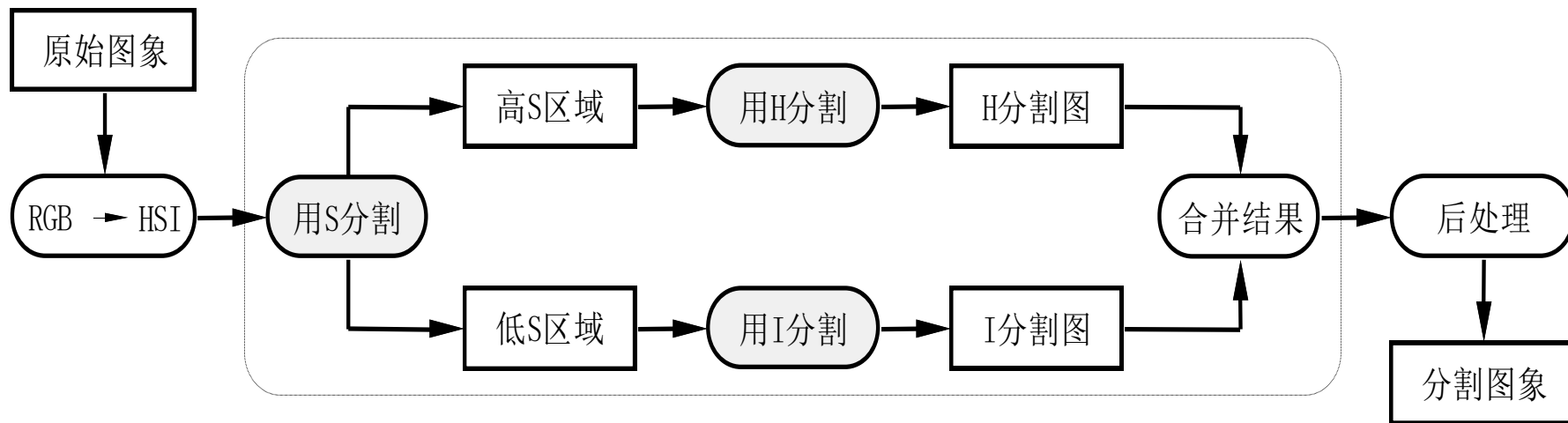


图 7.2.3 HSI 模型的圆柱坐标系统

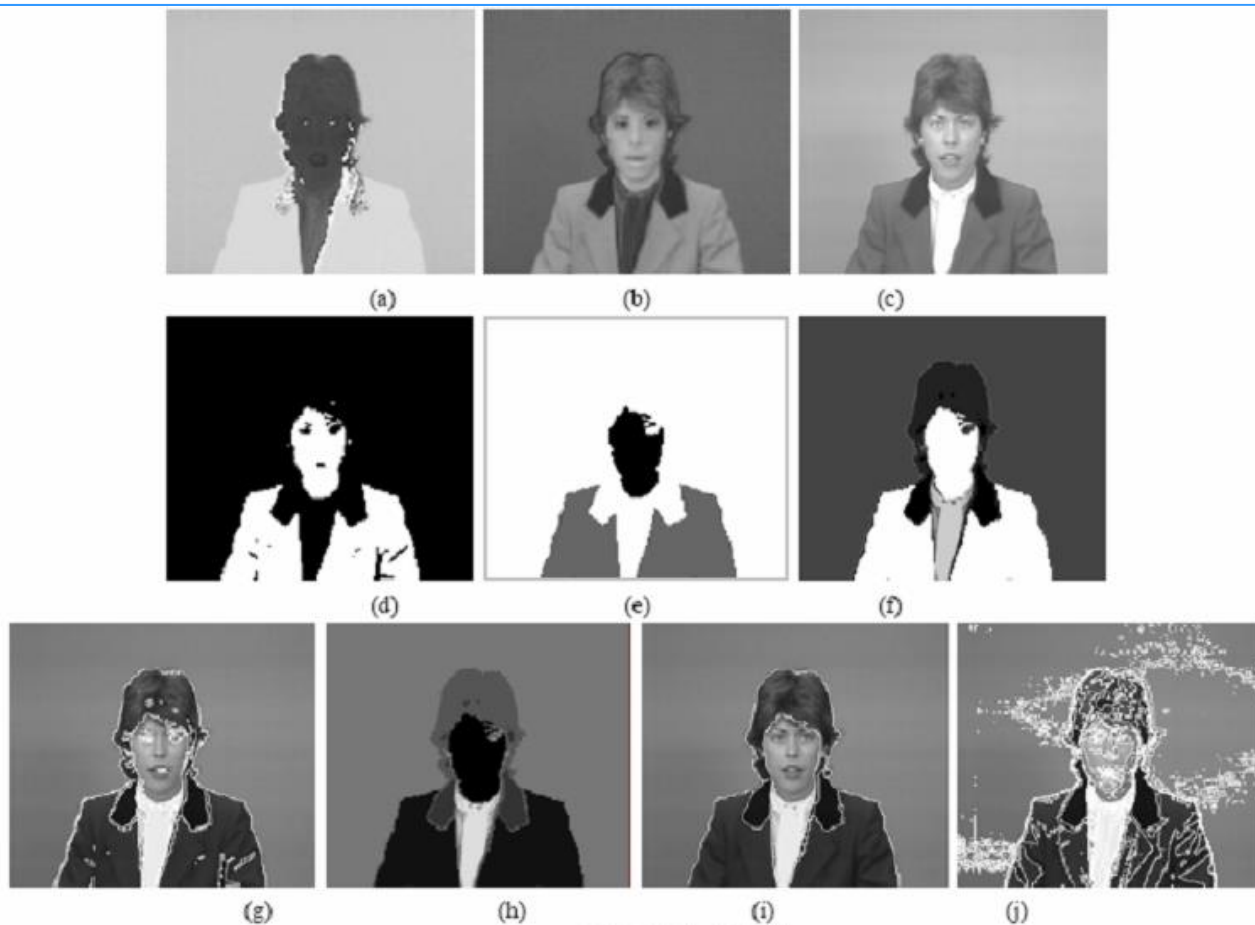
彩色图象的分割

2、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



彩色图象分割实例



彩色图像分割实例

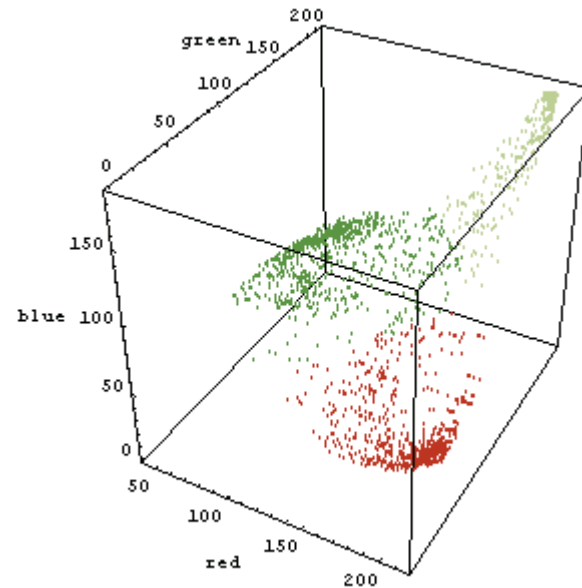
a) H; b) S; c) I; d) S图分割：白色高S区域； e)按H值分割； f)按I值分割； g)综合e和f；
h) 后处理； i)加入边界； j)彩色空间分割

彩色图象的分割

● 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



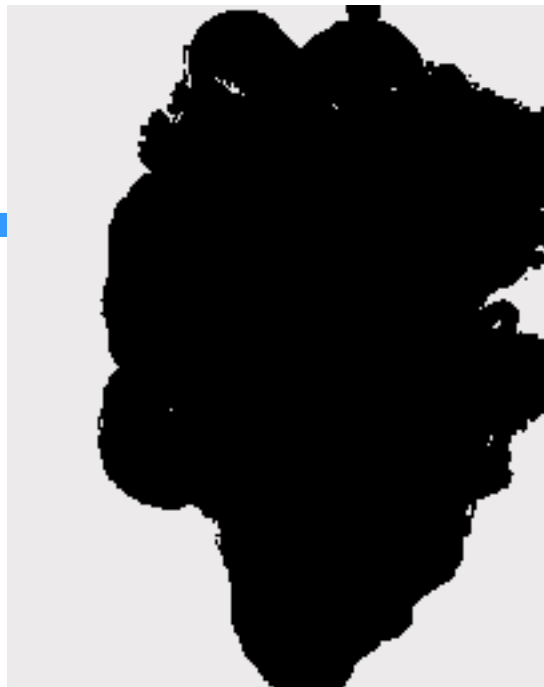
Original



K=5

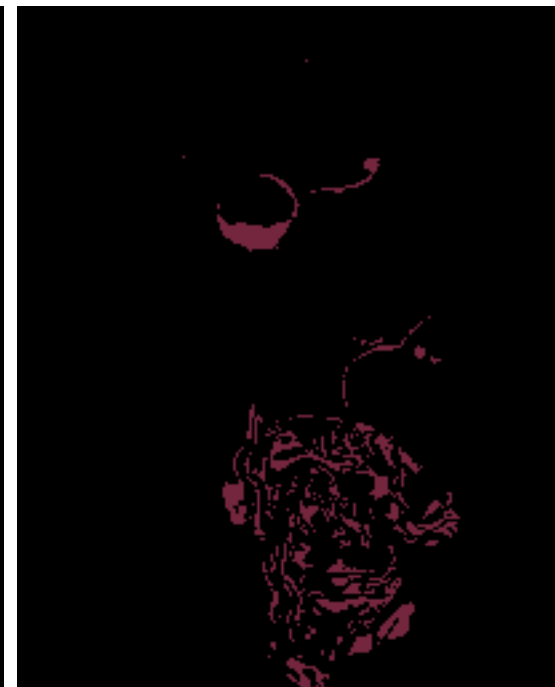


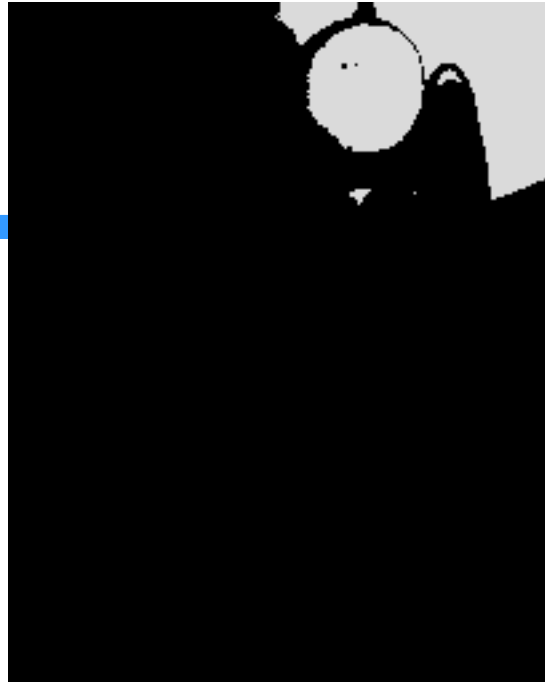
K=11



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。





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。





5.6 基于主动轮廓模型的图像分割

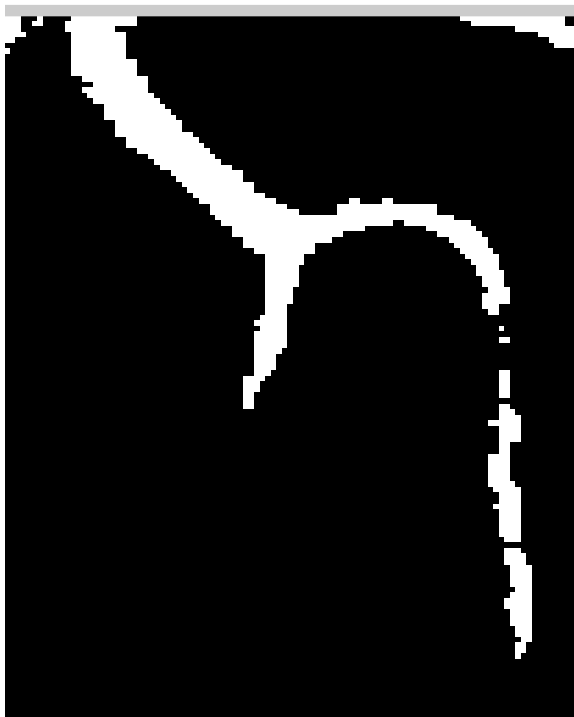
- 参数化主动轮廓：snake
- 几何主动轮廓：水平集方法 (level set)
 - From Snake to Level Set

传统的图像分割方法

血管图像



二值化

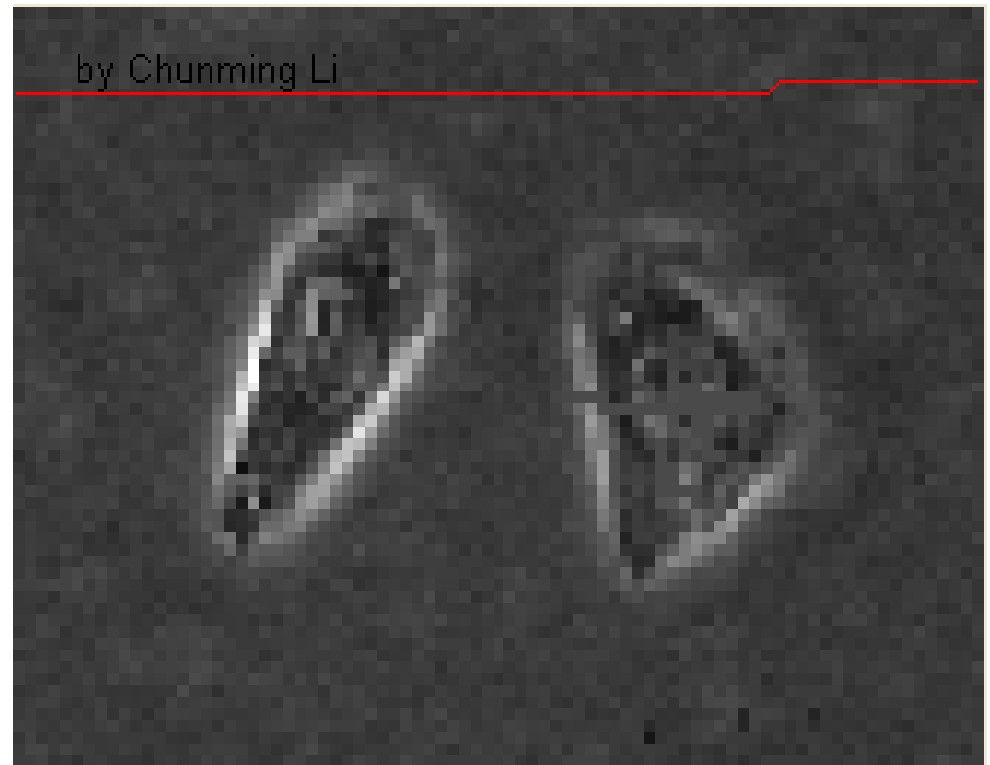
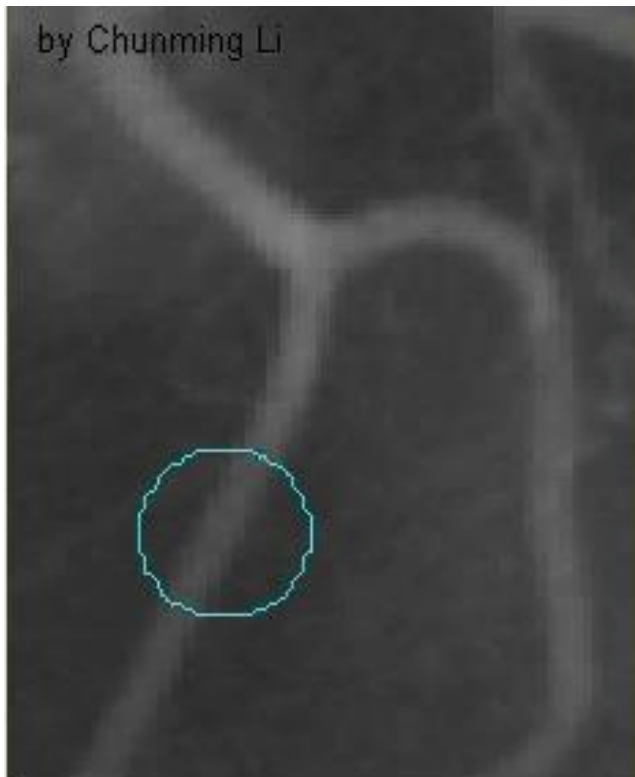


边缘检测



An Advanced Method: Active Contour Model

通过在图像中演化一组动态曲线，检测出图像中感兴趣的物体





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
- 几何主动轮廓：水平集方法 (level set)
 - From Snake to Level Set
 - Classical Level Set Model

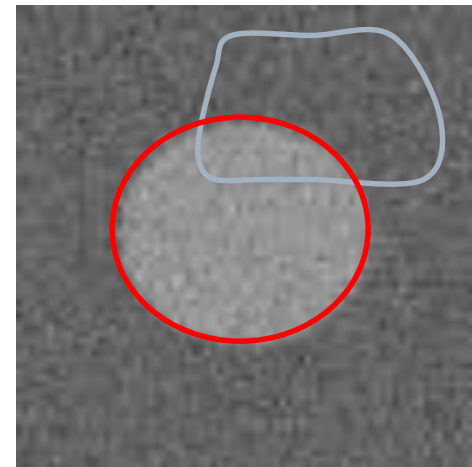
参数化主动轮廓 (Kass et al, 1988)

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

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

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

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



主动轮廓的演化

最陡梯度下降算法:

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

欧拉-拉格朗日
方程

内在力: 规则化 (Regularize) 轮廓曲线

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:

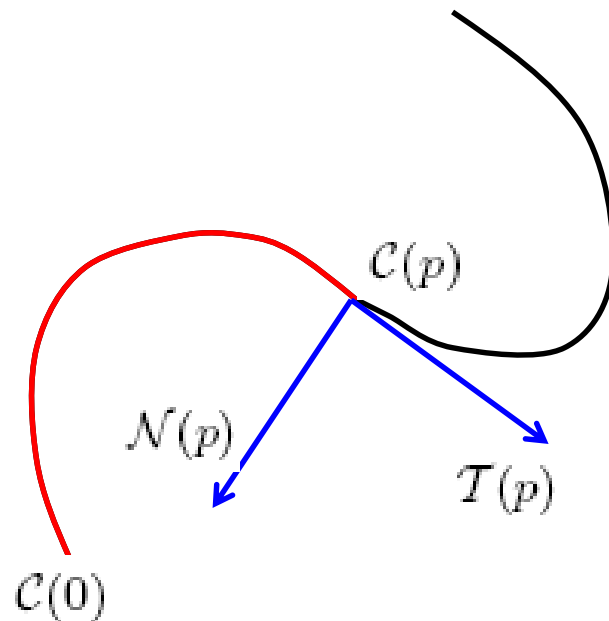
$$T(p) = \mathcal{C}'(p) = (x'(p), y'(p))$$

■ 法向量

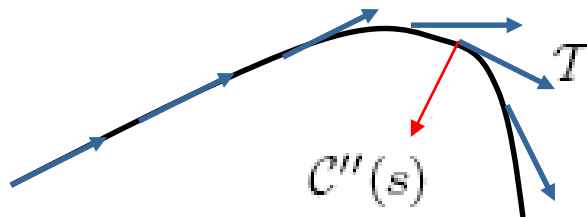
$$\mathcal{N}(p) = \left(\frac{-y'(p)}{\sqrt{x'(p)^2 + y'(p)^2}}, \frac{x'(p)}{\sqrt{x'(p)^2 + y'(p)^2}} \right)$$

■ 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

⇒ 切向矢量 $T(s) = C'(s)$ 是一个单位向量

- 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: $C''(s) = T'(s)$

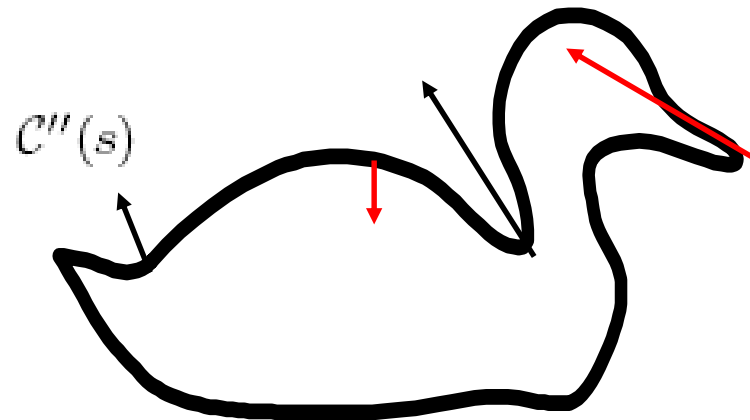
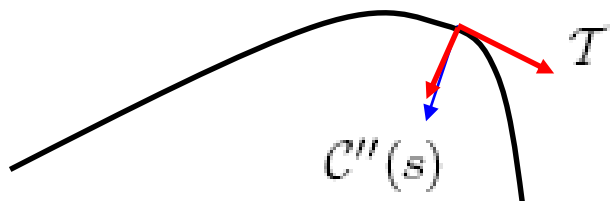
- 注意 $C'(s) \cdot C'(s) \equiv 1 \Rightarrow C''(s) \cdot C'(s) = 0$

曲率 (curvature)

■ $\mathcal{C}''(s)$ 和 \mathcal{N} 均正交于 T

⇒ $\mathcal{C}''(s)$ 平行于 \mathcal{N}

⇒ $\mathcal{C}''(s) = \kappa(s)\mathcal{N}(s)$



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

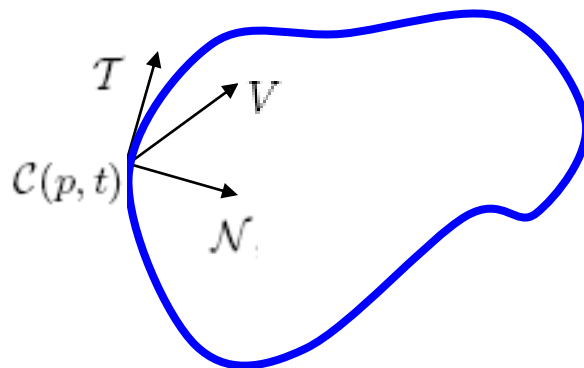
$$\kappa(p) = \frac{x'(p)y''(p) - y'(p)x''(p)}{(x'(p)^2 + y'(p)^2)^{3/2}}$$

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

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

几何曲线演化

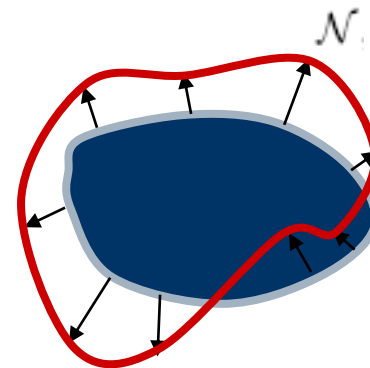
■ 定理：设 β 与参数表示方式无关. 若 $\mathcal{C}(p, t)$ 按照以下方程进行演化

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

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

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

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



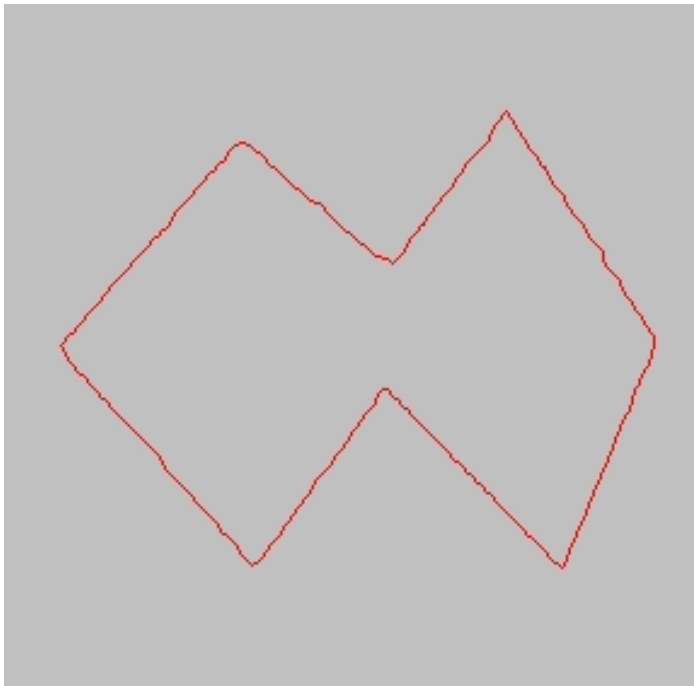
■ 参数演化的一般方程: $\frac{\partial \mathcal{C}}{\partial t} = F \mathcal{N}$

常速运动和平均曲率运动

常速运动

(Area decreasing/increasing)

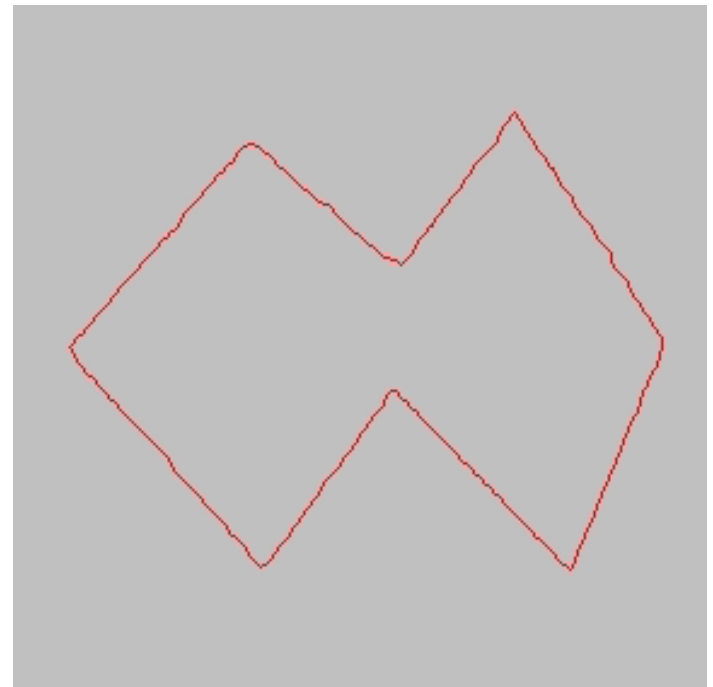
$$\frac{\partial \mathcal{C}}{\partial t} = c\mathcal{N}$$



平均曲率运动

(Length shortening flow)

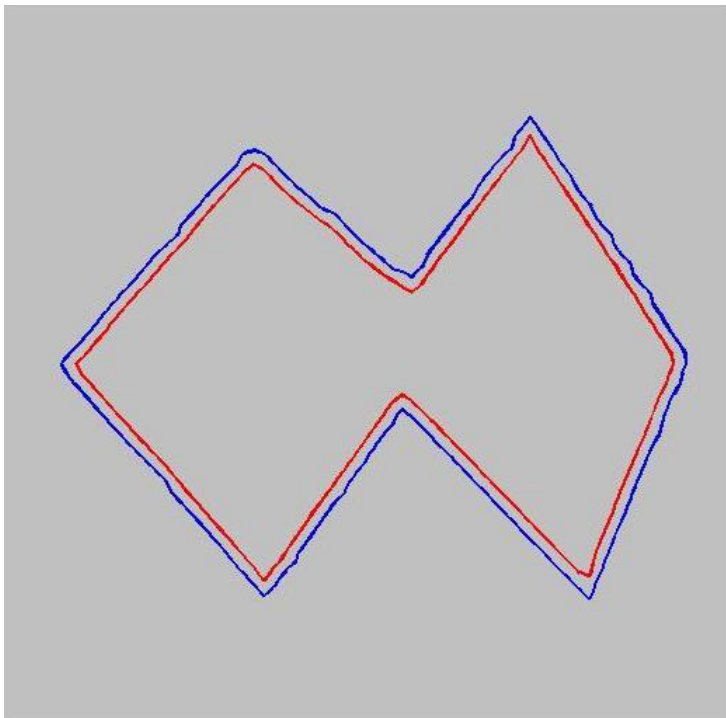
$$\frac{\partial \mathcal{C}}{\partial t} = \kappa\mathcal{N}$$



演化中间结果

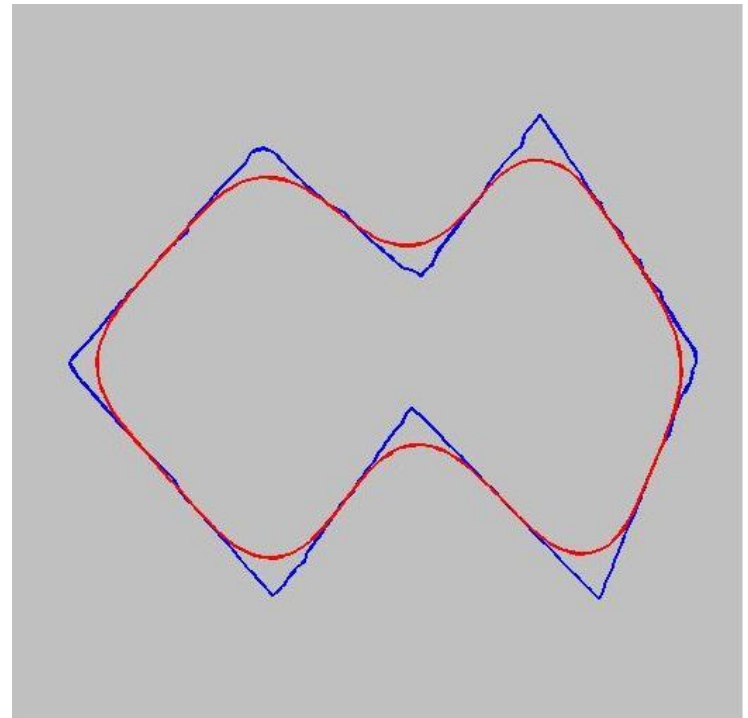
- 常速运动 (Area decreasing flow)

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



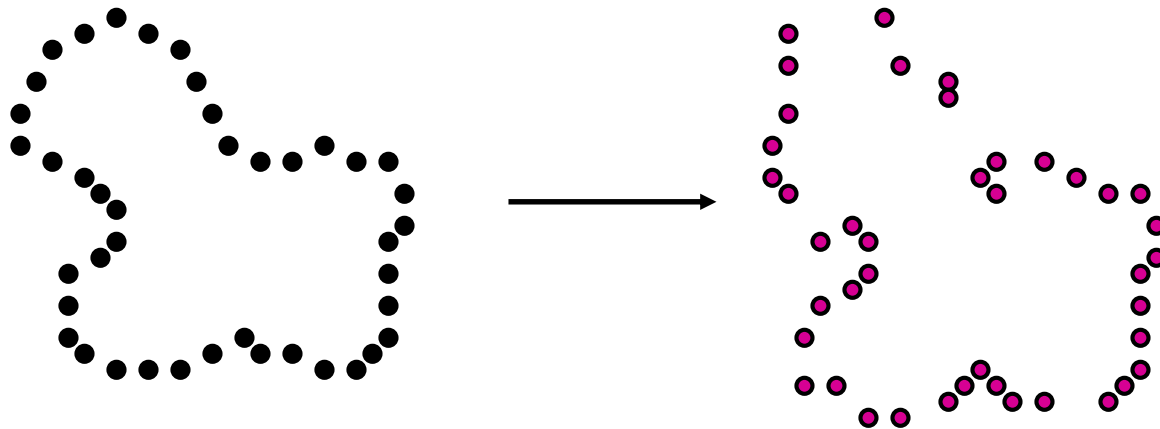
- 平均曲率运行 (Length shortening flow)

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

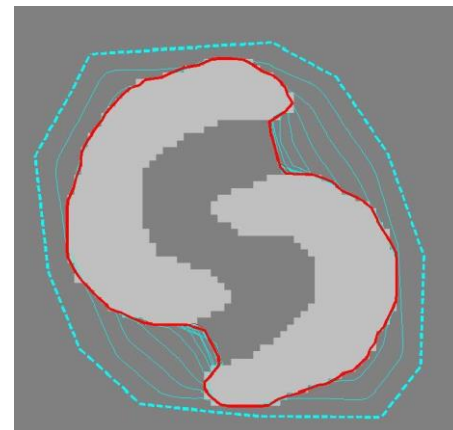


参数化主动轮廓的缺陷

- 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:

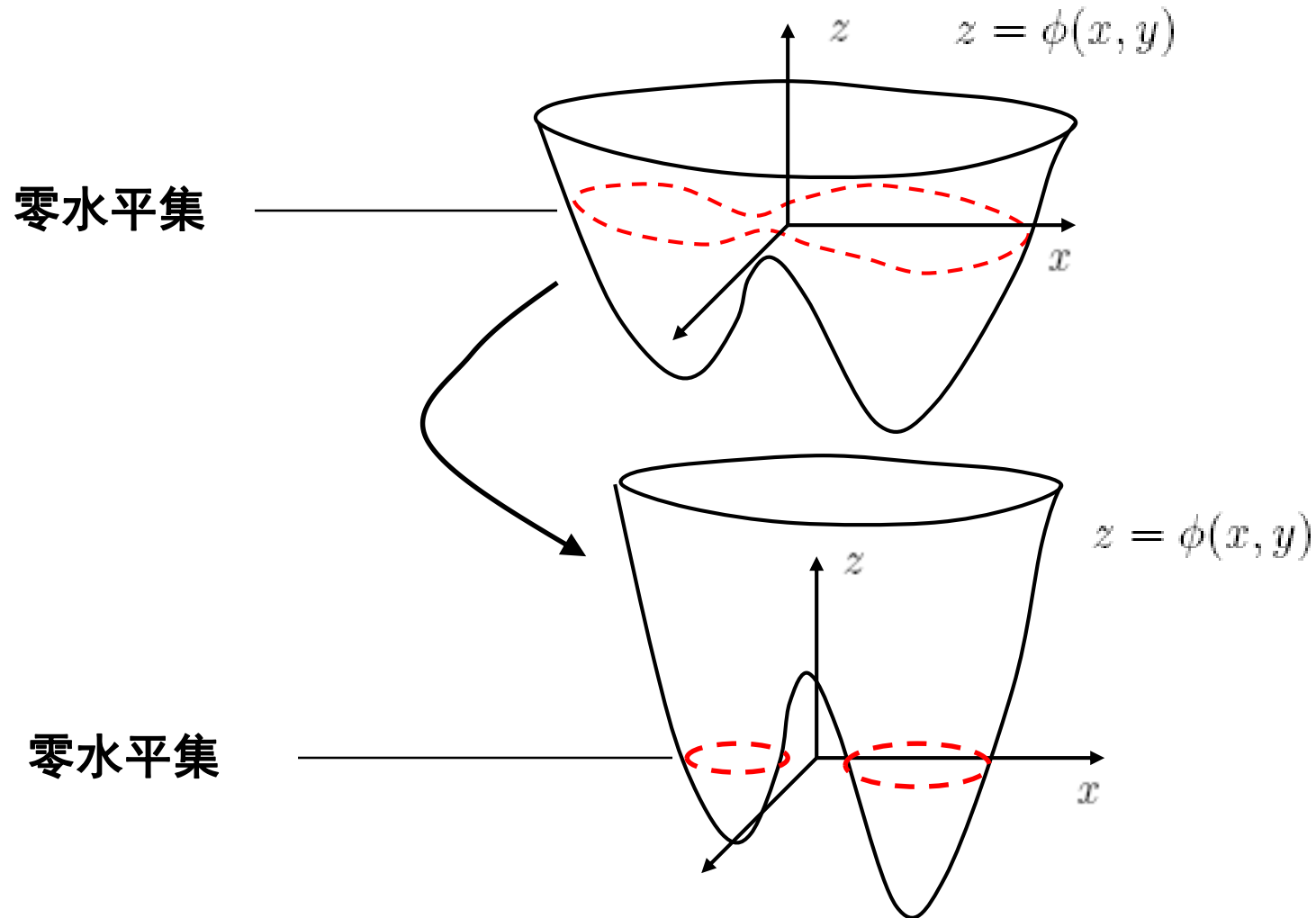
- **水平集方法 (Level set methods)**



基于主动轮廓模型的图像分割

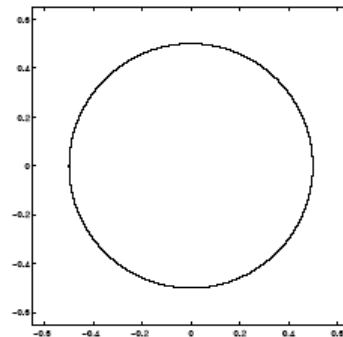
- 参数化主动轮廓：snakes
- 几何主动轮廓：水平集方法 (level set)
 - From Snake to Level Set
 - Classical Level Set Model

曲线的水平集描述

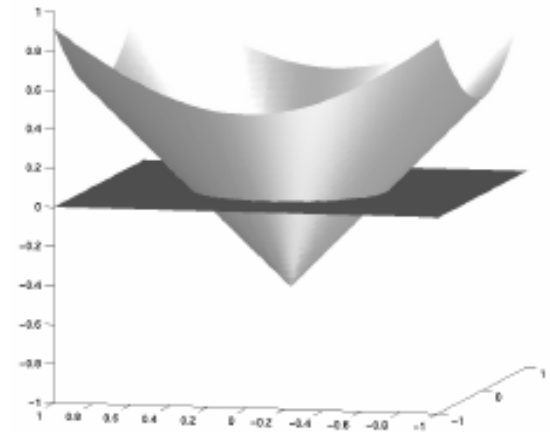
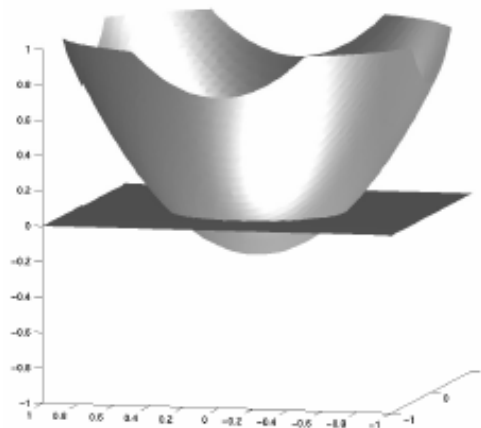


水平集函数 (level set function)

轮廓线

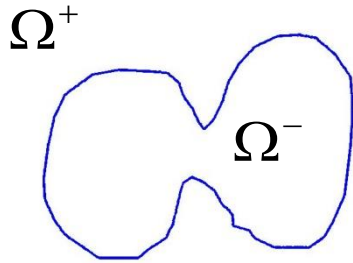


零水平集相同的
水平集函数

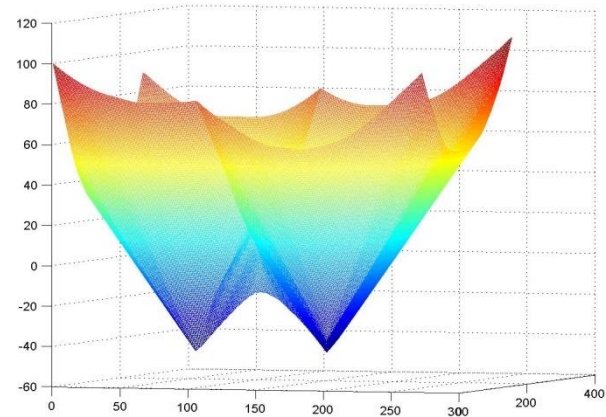


符号距离函数 (Signed distance function)

轮廓 C



符号距离函数



符号距离函数定义如下：

$$\phi(\mathbf{x}) = \begin{cases} \text{dist}(\mathbf{x}, C) & \text{if } \mathbf{x} \text{ is outside } C \\ 0 & \mathbf{x} \in C \\ -\text{dist}(\mathbf{x}, C) & \text{if } \mathbf{x} \text{ is inside } C \end{cases}$$

从曲线演化到水平集演化

- 曲线演化方程: $\frac{\partial C}{\partial t} = F\mathcal{N}$

其中 F 为速度函数, \mathcal{N} 为曲线 C 的法向量

- 把动态曲线 $C(p, t)$ 作为零水平集, 嵌入到一个随时间变化的函数 $\phi(x, t)$ 中, 使得 $\phi(C(p, t), t) = 0$


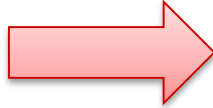


- 关于时间 t 求微分 $\Rightarrow \frac{\partial \phi}{\partial t} + \nabla \phi \cdot \frac{\partial C}{\partial t} = 0$

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

$$\frac{\partial \phi}{\partial t} = -\nabla \phi \cdot \frac{\partial C}{\partial t} = -\nabla \phi \cdot F\mathcal{N}$$

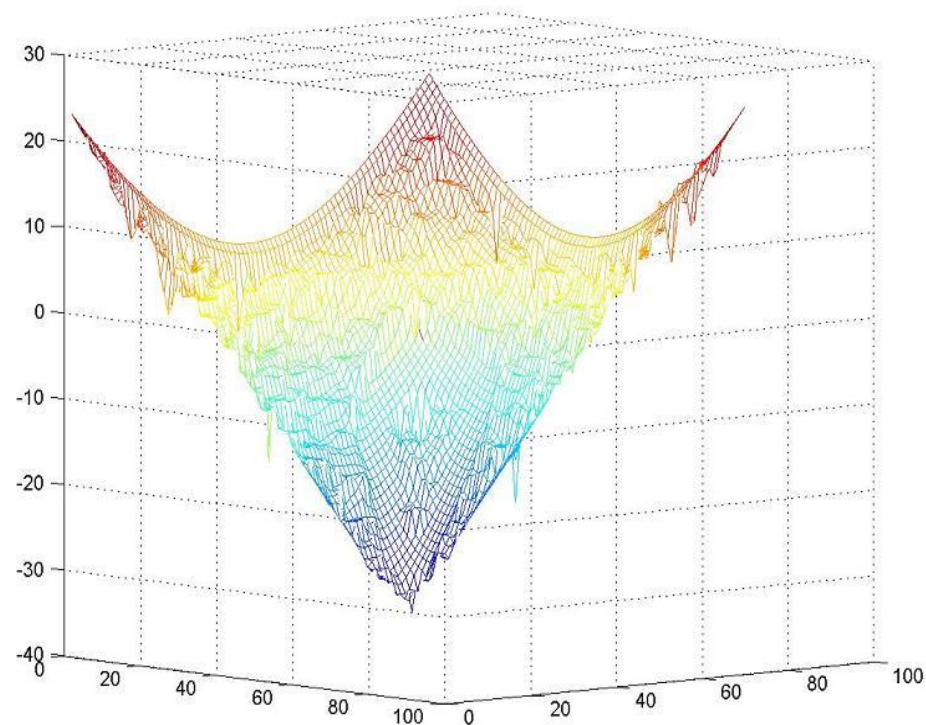
- 水平集演化方程: $\frac{\partial \phi}{\partial t} = F|\nabla \phi|$

一些特殊的演化方程

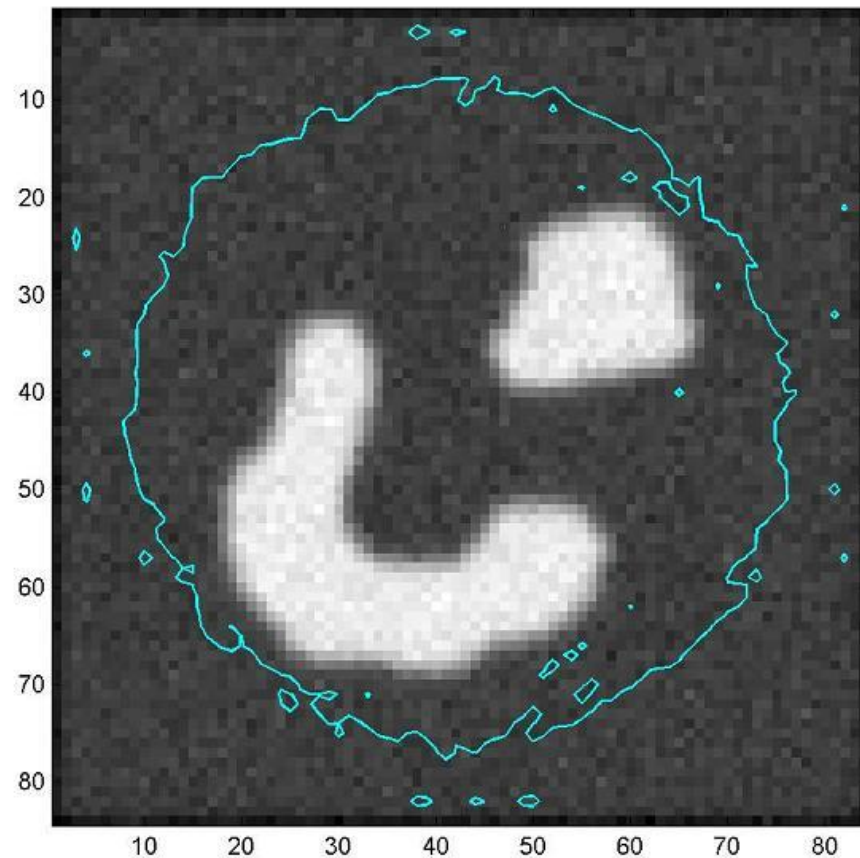
- 平均曲率运动 $\frac{\partial \mathcal{C}}{\partial t} = \kappa \mathcal{N}$  $\frac{\partial \phi}{\partial t} = \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) |\nabla \phi|$
- 常速率运动 $\frac{\partial \mathcal{C}}{\partial t} = c \mathcal{N}$  $\frac{\partial \phi}{\partial t} = c |\nabla \phi|$
- 对流运动 $\frac{\partial \mathcal{C}}{\partial t} = \langle \bar{V}, N \rangle \mathcal{N}$  $\frac{\partial \phi}{\partial t} = -\langle \bar{V}, \nabla \phi \rangle$
- 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

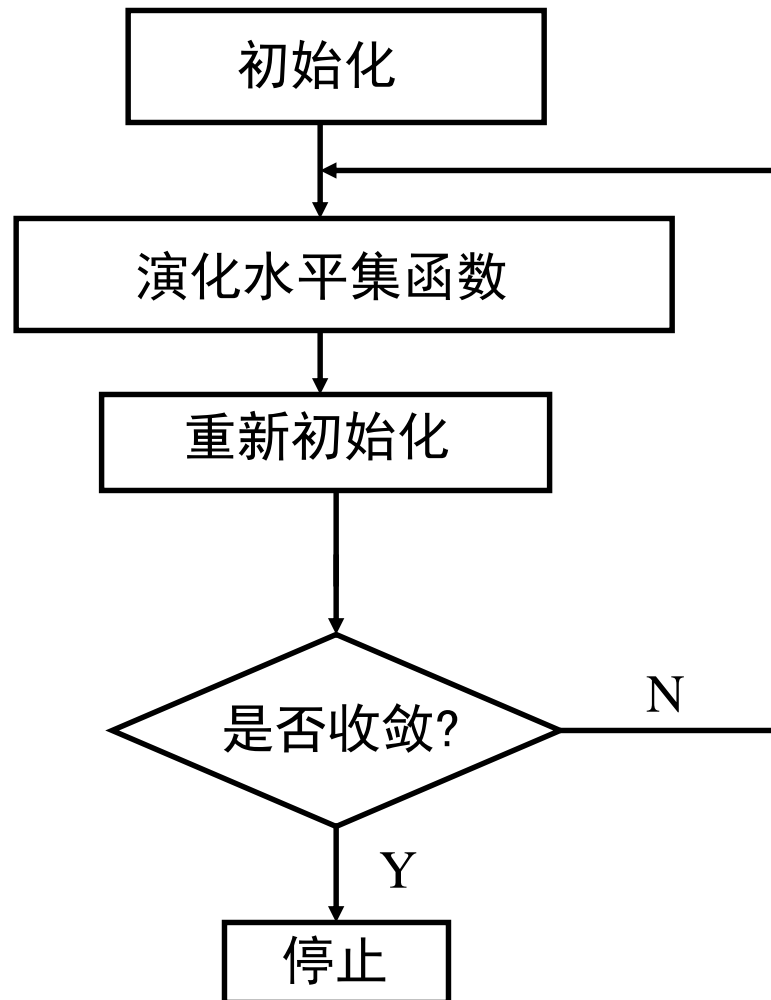
降质 (Degraded) 的水平集函数,
50 步迭代后, 时间步长为0.1



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 (CVPR'05)

- 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 $\mathcal{E}_{g,\lambda,\nu}(\phi) = \lambda \mathcal{L}_g(\phi) + \nu \mathcal{A}_g(\phi)$

$$\mathcal{L}_g(\phi) = \int_{\Omega} g \delta(\phi) |\nabla \phi| dx dy \quad \mathcal{A}_g(\phi) = \int_{\Omega} g H(-\phi) dx dy,$$

- 对能量函数求梯度
$$\begin{aligned} \frac{\partial \mathcal{E}}{\partial \phi} = & -\mu [\Delta \phi - \operatorname{div}(\frac{\nabla \phi}{|\nabla \phi|})] \\ & -\lambda \delta(\phi) \operatorname{div}(g \frac{\nabla \phi}{|\nabla \phi|}) - \nu g \delta(\phi) \end{aligned}$$

- 基于梯度下降算法, 得到演化方程:

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



变分法推导

An instance for calculus of variations:

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

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

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

then:

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

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

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



变分法推导

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

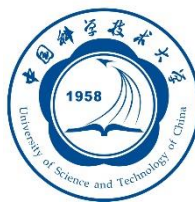
$$\frac{\partial}{\partial y} [\phi_y h] = h_y \phi_y + h \phi_{yy},$$

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

According to Green Equation: $\oint_{\partial \Omega} R dx + S dy = \iint_{\Omega} \left(\frac{dS}{dx} - \frac{dR}{dy} \right) dx dy$, and $h|_{\partial \Omega} = 0$,

we can get

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



变分法推导

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

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

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 $-\left(\Delta\phi - \nabla \cdot \left[\frac{\nabla\phi}{|\nabla\phi|} \right] \right)$,

with the gradient descent algorithm, we get the LEVLE SET evolution equation:

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



Reference

- [1] M. Kass, et al, “Snakes: active contour models”, *IJCV*, 1988
- [2] Guillermo Sapiro, “Geometric partial differential equations and image analysis”, *Cambridge University Press*, 2001
- [3] Stanley Osher, Ronald Fedkiw, “Level set methods and dynamic implicit surfaces”, *Springer*, 2003
- [4] 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(II)

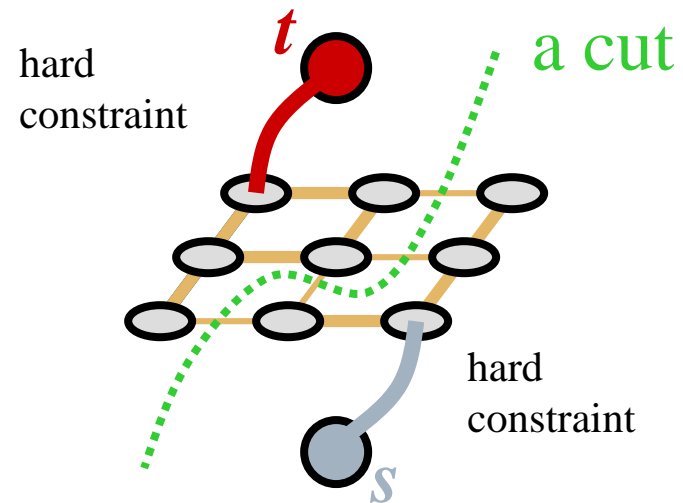
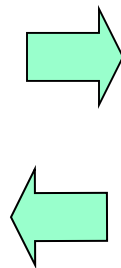
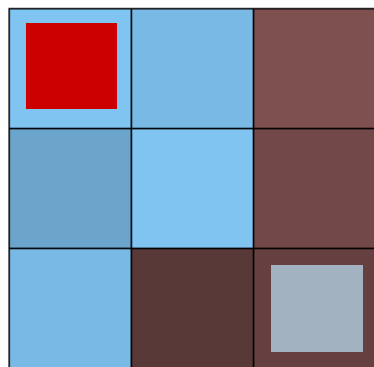
- ☐ Mean-shift
- ☐ Active Contour Model
- ☐ Graph cut



5.7 Graph cut 图割法

- Graph cut: 一种能量函数的优化方法
 - 基于图像定义某种能量函数
 - 能量函数的最小值, 对应着一种期望的分割

Graph cut 示例



最小代价的 cut 可以以多项式时间的复杂度计算得到
(max-flow/min-cut algorithms)

1. 能量函数

根据图像的颜色分布，设计能量函数。即某种分割和图像像素颜色分布的不一致性的惩罚函数，分割和图像的颜色分布、边缘符合的越好，惩罚就越小，能量函数就越

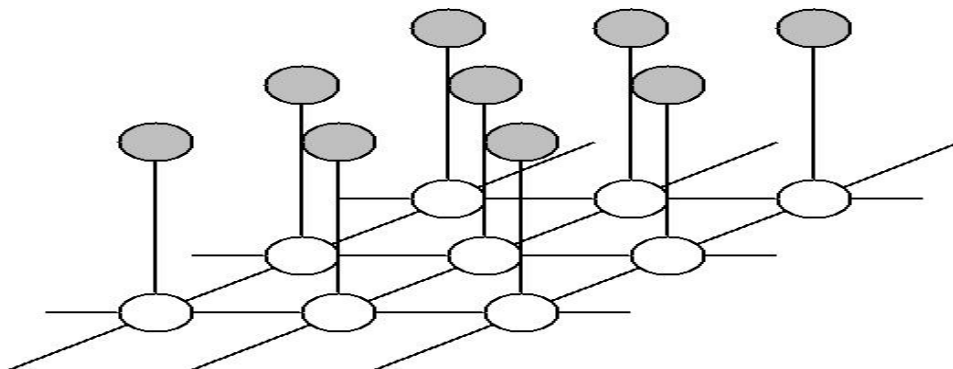
0	0	0	0	0
1	0	0	0	0
0	1	1	0	0
0	1	1	1	0
0	1	1	1	1

X表示分割结果，**1**代表前景，**0**代表背景。则能量是**X**的函数：

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

能量函数一般形式

在下图中，白色圆圈代表分割 X ，灰色圆圈代表像素值，用线段相连的两个圆构成一个clique。每个clique贡献一个能量值。



白色与灰色圆圈组成的一对的能量如下，每一项是一个 x_i 的单变量函数：

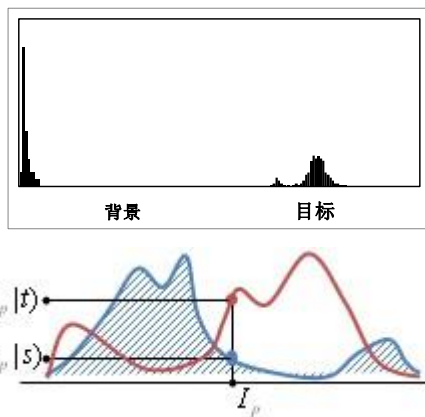
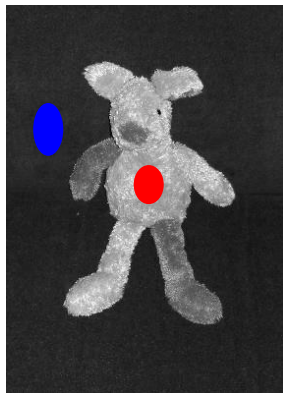
$$E_{data}(\mathbf{x}) = \sum_i E^1(x_i)$$

相邻的白色圆圈构成的一对贡献的能量如下，每一项是一个 x_i 、 x_j 的双变量函数：

$$E_{smooth}(\mathbf{x}) = \sum_{i < j} E^2(x_i, x_j)$$

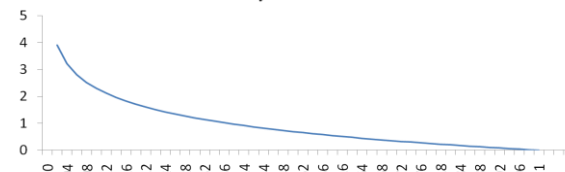
构造能量函数

Data项



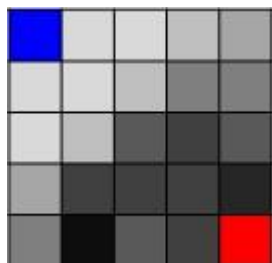
$$E^1(x_i) \propto -\ln \Pr(I_i | x_i)$$

$$y = -\ln x$$



条件概率越高当然能量越低

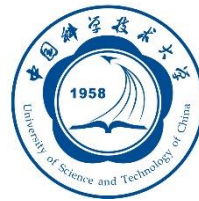
Smooth项



$$E^2(x_i, x_j) = \begin{cases} \exp\left(-\frac{(I_i - I_j)^2}{2\sigma^2}\right) & , x_i \neq x_j \\ 0 & , x_i = x_j \end{cases}$$

0	0	0	0	0
0	0	0	1	1
0	0	1	1	1
0	1	1	1	1
0	1	1	1	1

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

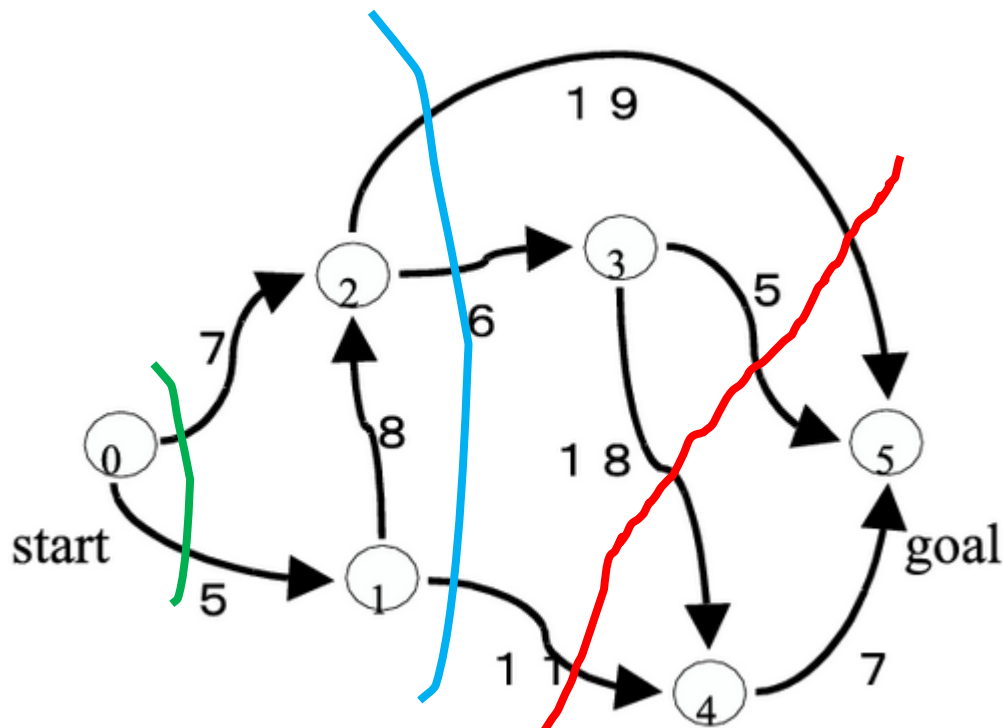
仅有**data**项，分割结果相当于用颜色的先验概率对像素进行分类。分类的结果没有考虑相邻像素之间的关系，造成分割不连续。

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

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

Graph 和 Cut

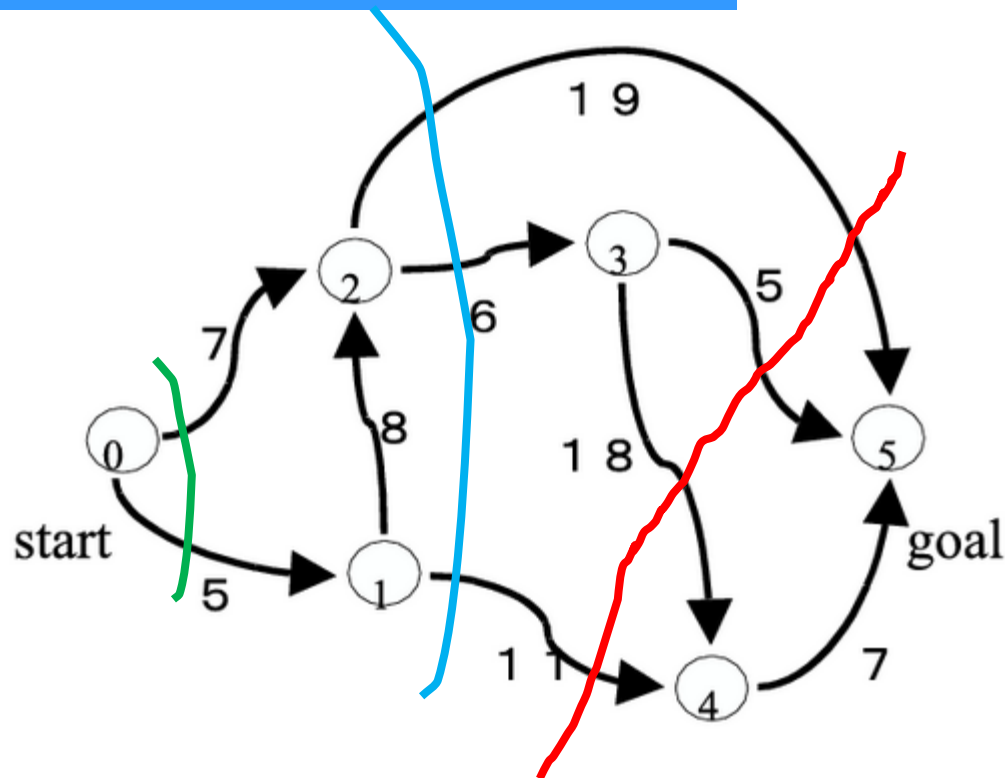


我们把这样一个边集叫做一个**割** (cut) :

如果在图中去掉割(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。

最小割 (min cut) : 权值之和最小的割。(图中绿色的线)

Graph 和 Cut



最小割(min cut) -> 最大流(max flow)

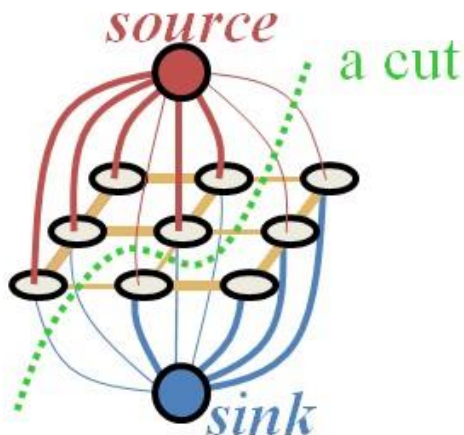
对于一个有向图 (graph)，权值当做弧的流量。

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。

Graph Cut

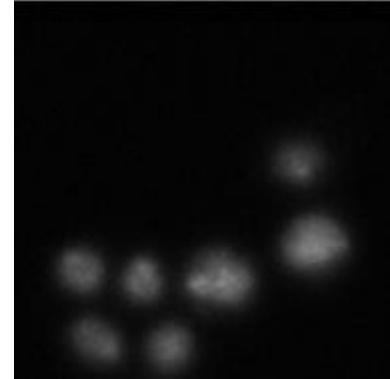
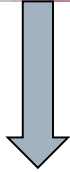
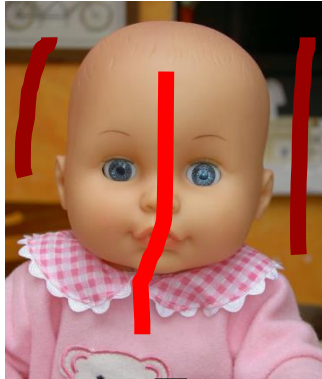


优化能量函数=Graph Cut问题

把能量函数通过简单的变换映射为一个图模型，此时每一个割(cut)对应了一种图像分割结果。

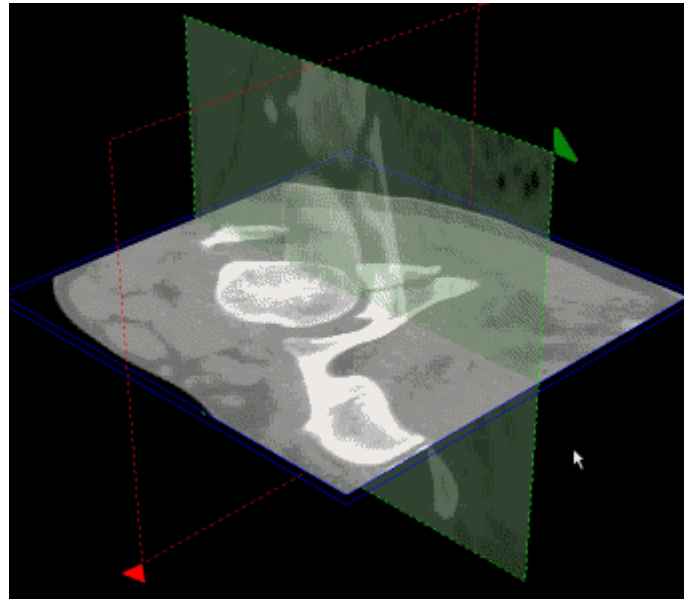
找到一个最小割相当于找到了一个使能量函数最小的解。（权值不是简单的能量函数项的对应，所以cut的值不等于能量函数，而是差了一个常数）

采用Graph cut分割的结果





基于 Graph cut的3D 医学图像分割：bone





References

□ 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/>