



图像分割

董秋雷



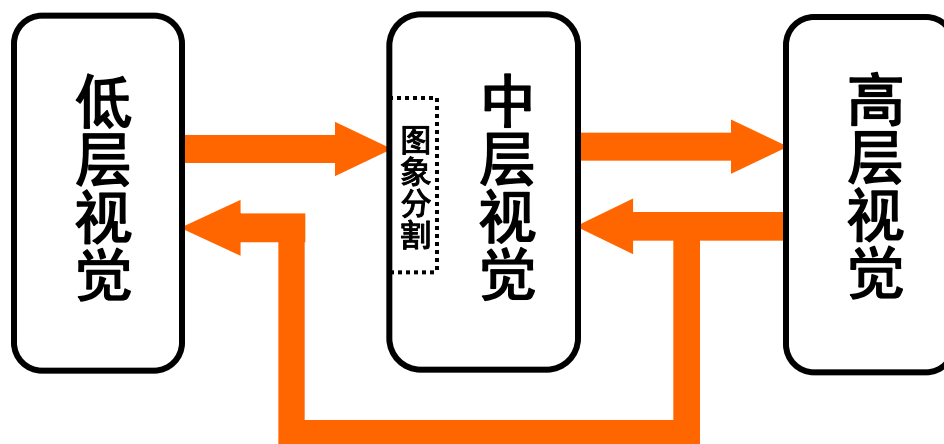
什么是图像分割

- 图像分割就是指把图像分成互不重叠的区域并提取出感兴趣目标的技术和过程。



为什么要图像分割

- 图像分割是由图像处理进到图像分析的关键步骤。它是目标表达的基础，使得更高层的图像分析和理解成为可能。



图像分割的应用领域

- 医学图像处理
- 遥感图像处理
- 目标跟踪
- 生物特征识别
- 等等

图像分割

- 图像分割是中层视觉中的最基本问题，也是计算视觉和图像理解中的最基本问题之一。它还是该领域国际学术界公认的将会长期存在的最困难的问题之一。
- 图像分割之所以困难的一个重要原因是其并不完全属于图象特征提取问题，它还涉及到各种图像特征的知觉组织。

图像分割

从一般意义上来说，
只有对图象内容的彻底理
解，才能产生完美的分割。

通过限制图象的类型，
可以降低图象分割的难度。



图像分割的基本依据

- 基本依据
 1. 区域内的一致性
 2. 区域间的不一致性

1. 基于区域的方法
2. 基于边缘的方法
3. 综合考虑边缘和区域信息的混合分割方法
4. 基于特定理论的方法

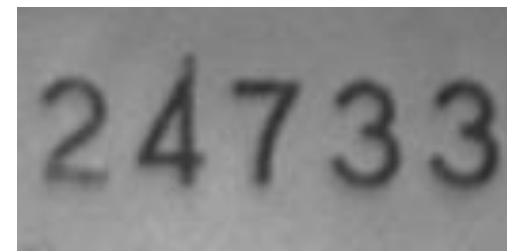
基于区域的方法

1. 基于区域的方法
 - I. 阈值法
 - II. 区域生长法
 - III. 分裂合并法

阈值法

- 基本原理是：通过设定不同的特征阈值，把图像像素点分为若干类。常用的特征包括：灰度、彩色特征、由原始灰度或彩色值变换得到的特征。

$$g(i, j) = \begin{cases} 1, & \text{if } f(i, j) \geq T \\ 0, & \text{if } f(i, j) < T \end{cases}$$



24733

阈值法

- 阈值分割法的关键是如何选取合适的阈值！
- 如果阈值选取过高，则过多的目标点被错误的归为背景；阈值选得过低，则会出现相反的情况。

$$g(i, j) = \begin{cases} 1, & \text{if } f(i, j) \geq T \\ 0, & \text{if } f(i, j) < T \end{cases}$$

阈值法

- 1.1.1 全局阈值方法

p-分位数法：

1962年Doyle提出p-分位数法：使目标或背景的像素比例等于其先验概率来设定阈值。

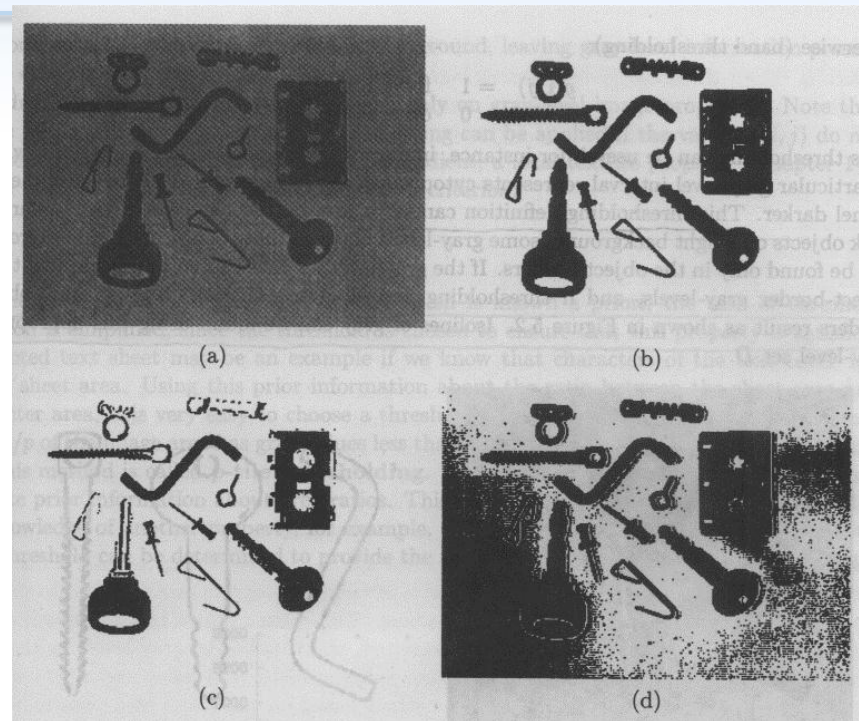
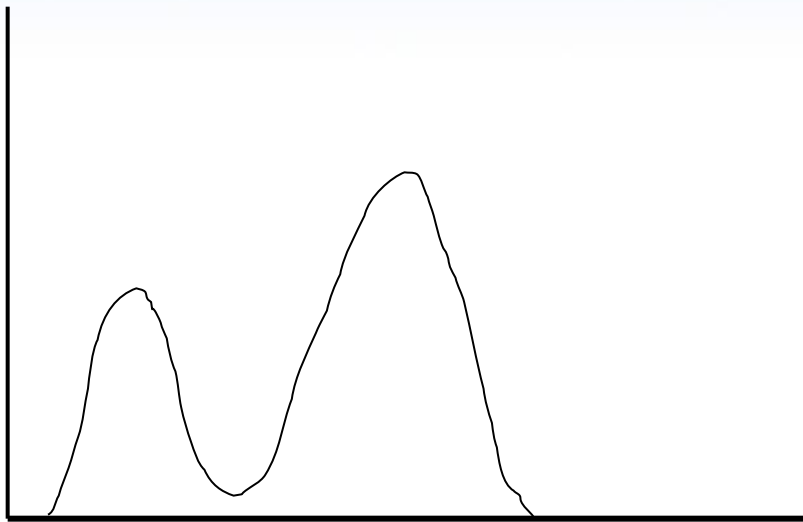
p-分位数法：



- 不足：对于先验概率难于估计的图像却无能为力。

阈值法

- 众数法 (J.M.S. Prewitt, et al., 1966, *Ann. New York Acad. Sci.*)
- 已知图象仅包含具有明显灰度差别的目标和背景。此时灰度直方图通常为双峰单谷型。取谷底点为阈值即可完成分割。



- 阈值方法并不要求直方图必须同时包含峰和谷

- 迭代方法

- 基本原理:

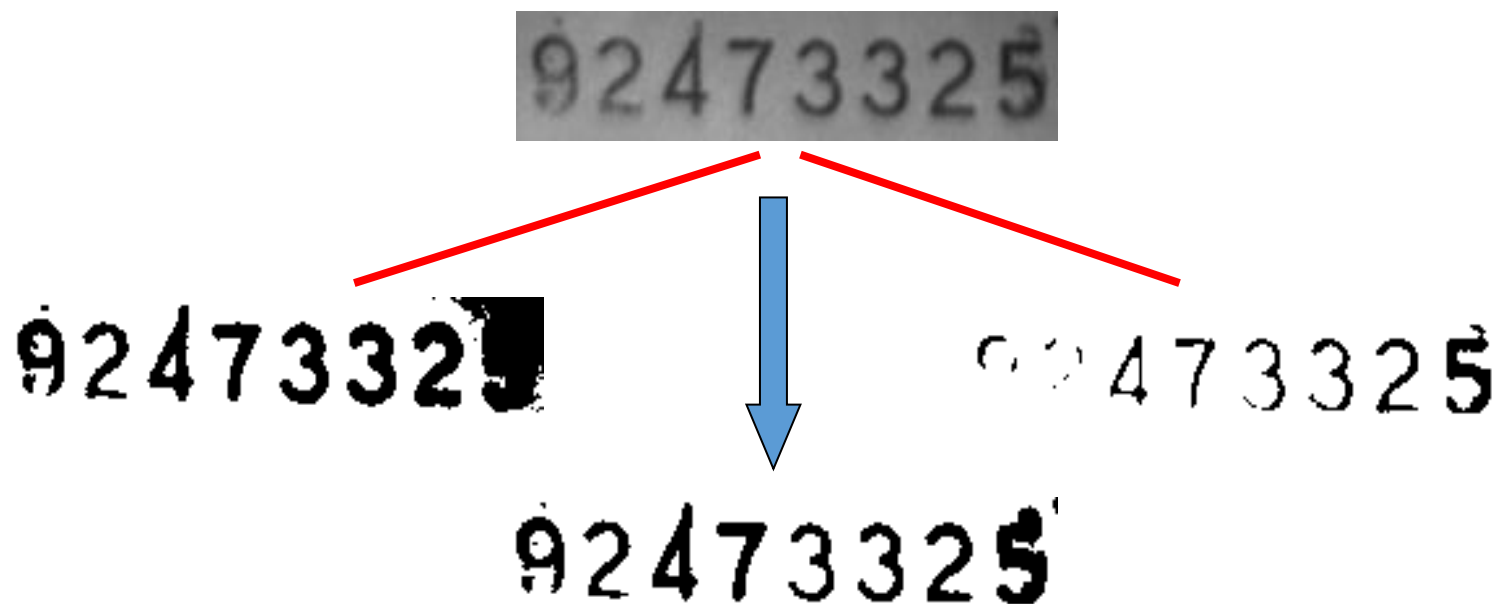
- 初始阈值选取为图像的平均灰度 T_0 ，然后用 T_0 将图像的像素点分作两部分，计算两部分各自的平均灰度，小于 T_0 的部分为 T_A ，大于 T_0 的部分为 T_B ；
- 计算 $T_1 = \frac{T_A + T_B}{2}$ ，将 T_1 作为新的全局阈值代替 T_0 ，重复以上过程，如此迭代，直至 T_k 收敛，即 $T_{k+1} = T_k$ 。

2	1	1	2	1	1
1	5	4	5	5	1
2	5	8	9	6	2
1	6	9	8	6	1
1	6	8	9	8	1
1	2	1	1	2	1

- 优点：算法时间复杂度较低，易于实现，适合应用于在线实时图像处理系统。
- 不足：忽略了图像的空间位置信息

• 1.1.2. 局部阈值方法

- 将图象分块，分别用全局阈值方法分割，最后再综合。

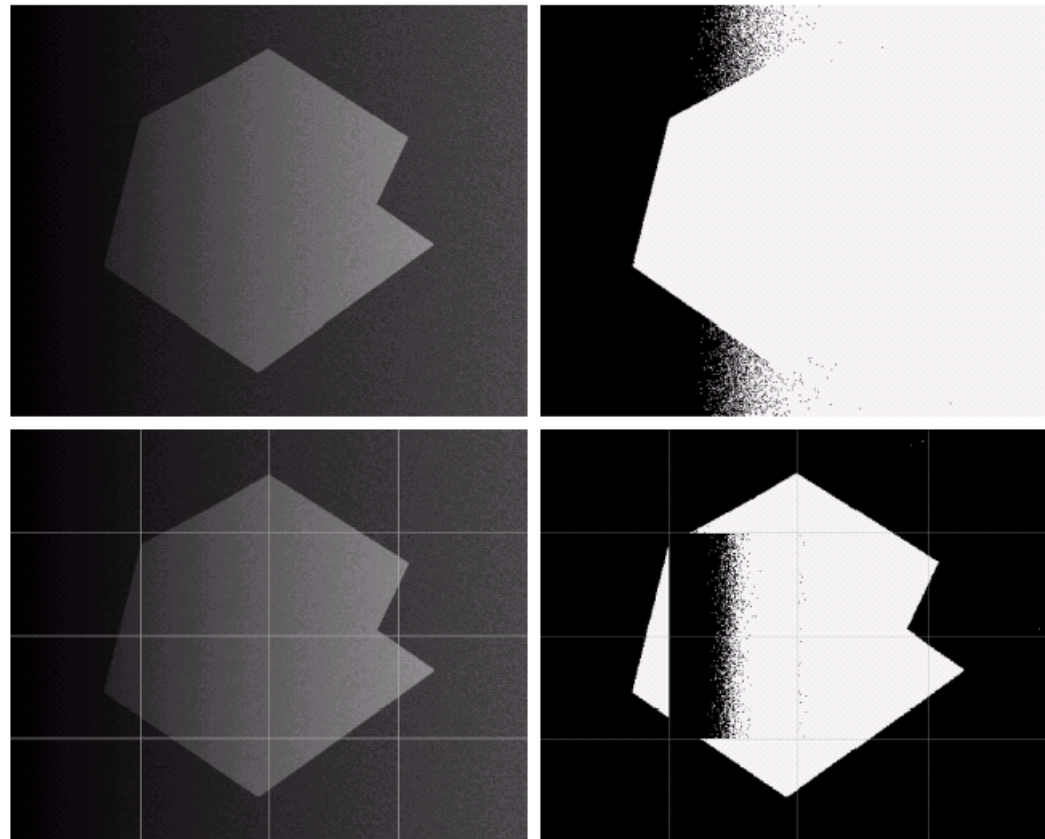


• 1.1.2. 局部阈值方法

a b
c d

FIGURE 10.30

(a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.

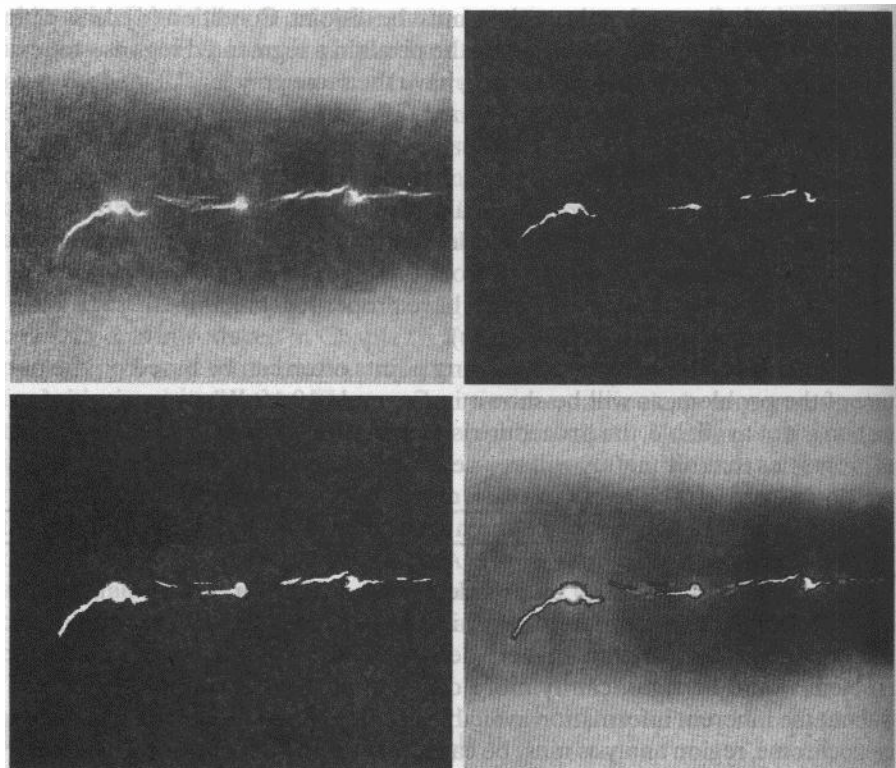


1.1.2.多阈值法

- 如果图像中含有占据不同灰度级区域的几个目标，则需要使用多个阈值才能将它们分开。
- 多域值分割，可以看作单阈值分割的推广

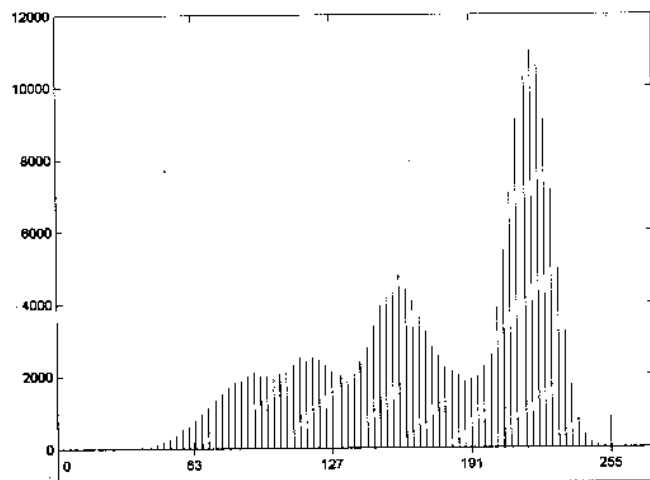


1.2 区域生长法



$$\begin{array}{c|c} a & b \\ \hline c & d \end{array}$$

- a) 原始X线探伤图象
- b) 种子区域
- c) 生长结果
- d) 缺陷区边缘



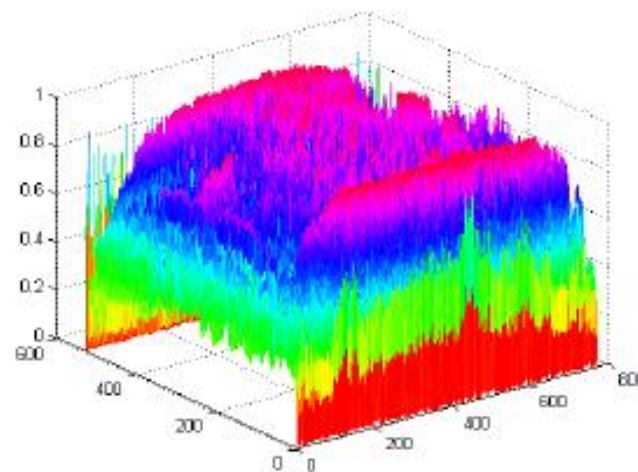
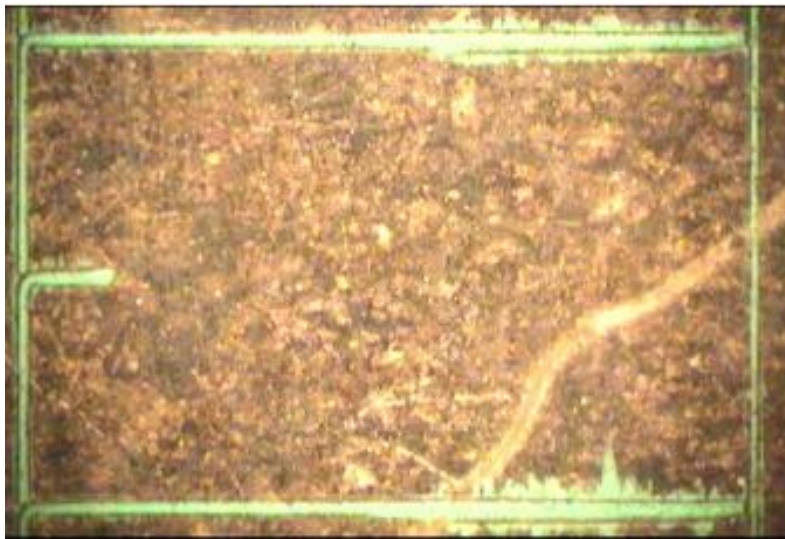
上页图a的直方图。种子区域由灰度为255的像素构成。

- 从种子区域开始以8-邻域方式向外生长，只要一个像素的灰度 g 与种子点的灰度差小于65 (即 $g \geq 191$)，即将该像素归入目标区

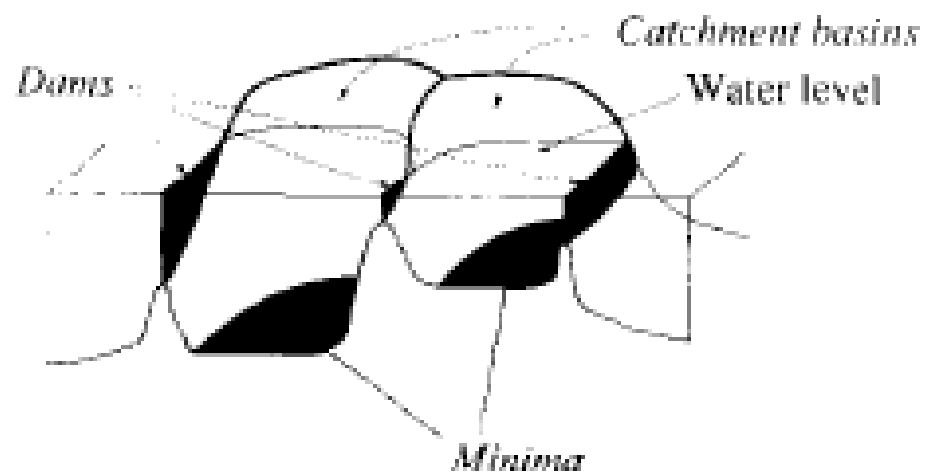
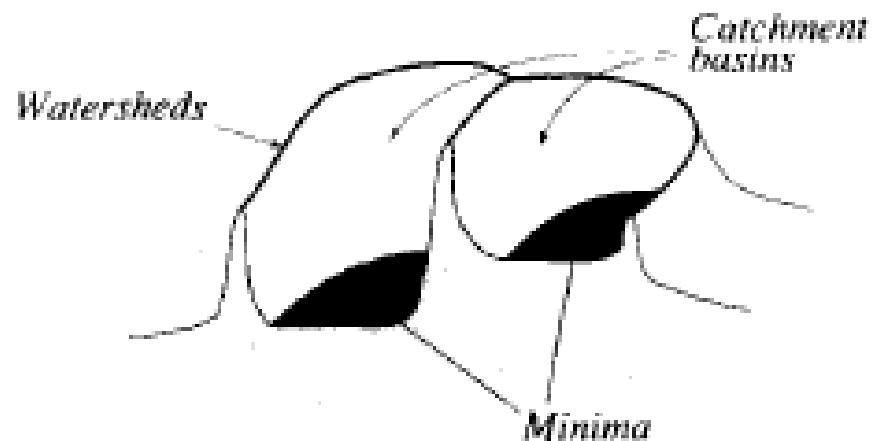
分水岭分割法 (watershed)

- 标准步骤：
 - 1) 将图象看作地形图；
 - 2) 在每一个极小点处“打一个孔”；
 - 3) 以一致的速率从小孔向外“喷水”，并始终保持地形中所有的水位一致；
 - 4) 不同盆地的水相遇时则筑坝，并且随着水位的不断升高，坝也升高；
 - 5) 当水位达到地形的最高点时算法终止

分水岭分割法 (watershed)



分水岭分割法 (watershed)

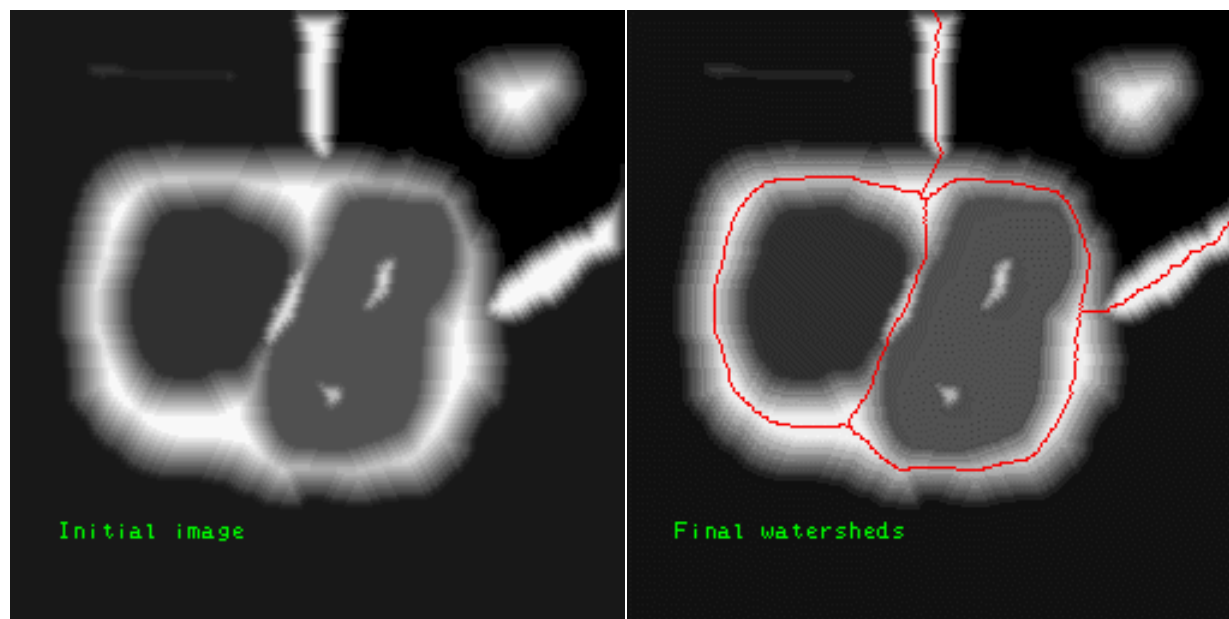


分水岭分割法 (watershed)

- 带标记的分水岭算法 — “泉眼”的位置是(自动或手工) 指定的， 而不是由极小点确定的。 这样可以避免“过度分割”问题。

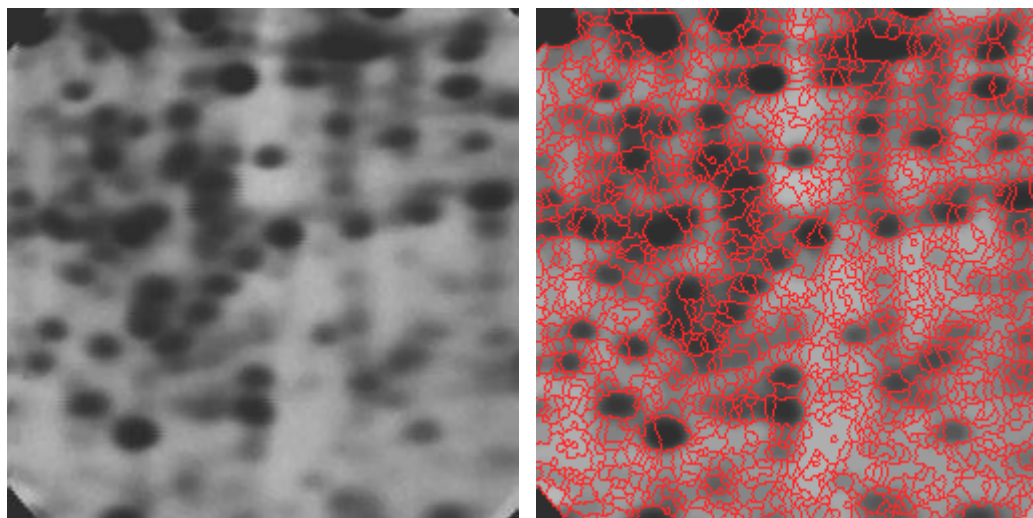


分水岭分割法 (watershed)



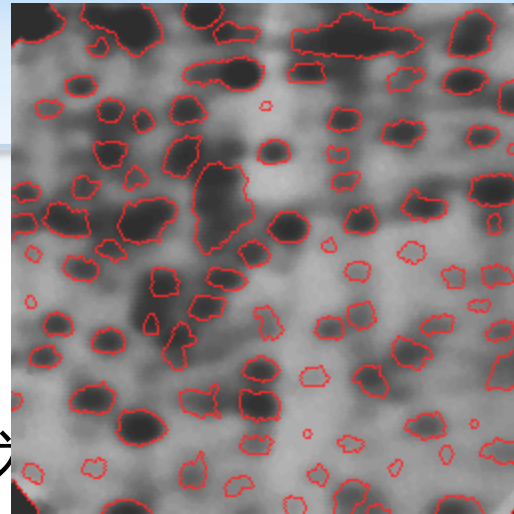
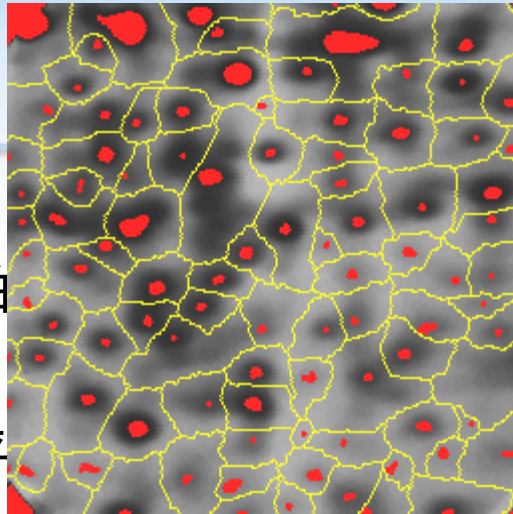
分水岭分割法 (watershed)

- 缺陷：可能出现“过度分割”问题。



图a 为原始电泳图象，图b为标准分水岭分割法分割图a 的结果。

- 电泳图象的分割
- 1) 滤波;
- 2) 选择平面; 为
- 3) 调用分水岭算法确定分水线(黄线);
- 4) 在每个分水线分出的小区内完成分割。



基本方法

1. 基于区域的方法
2. 基于边缘的方法
3. 综合考虑边缘和区域信息的混合分割方法
4. 基于特定理论的方法

2. 基于边缘的方法

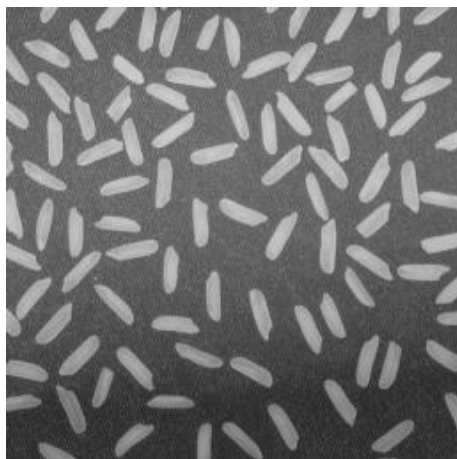
- 2.1 边缘检测算子
- 边缘的特性是沿边缘走向的像素变化平缓，而垂直于边缘方向的像素变化剧烈。
- 边缘检测：
 - 首先检出图像局部特性的不连续性；
 - 然后再将这些不连续的边缘像素连接成完整的边界。边缘的特性是沿边缘走向的像素变化平缓，而垂直于边缘方向的像素变化剧烈。

2.1 边缘检测算子

几种常用的边缘检测算子：

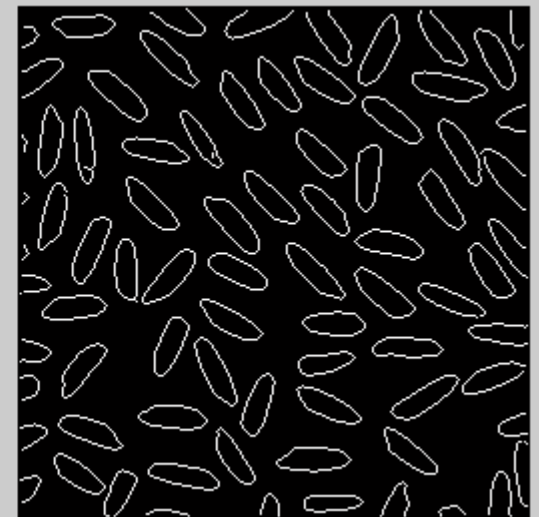
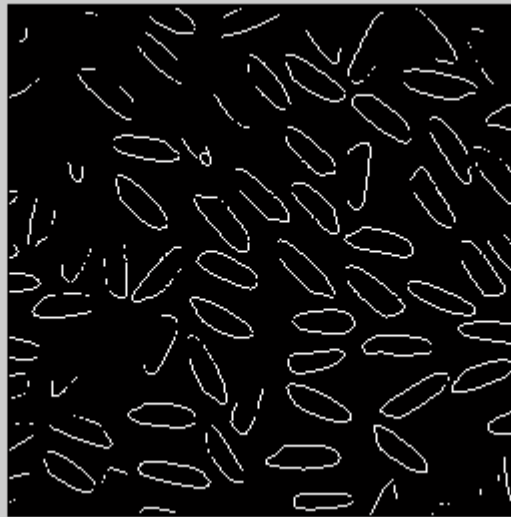
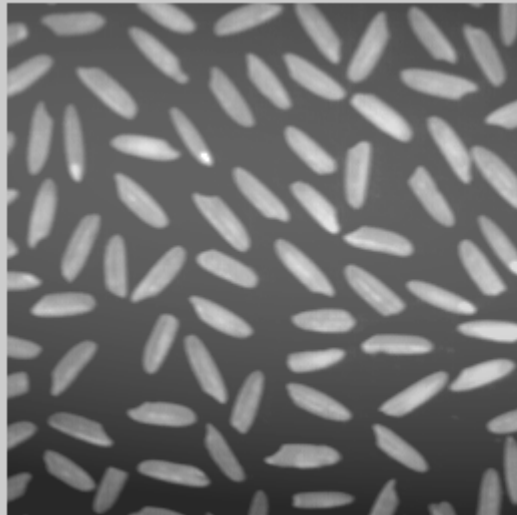
- 梯度算子
- Roberts算子
- Prewitt算子
- Sobel算子
- Kirsch算子
- Laplacian算子
- Marr算子

- 用SOBEL算子处理, 最后根据梯度模是否大于一个阈值来决定边缘。



-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

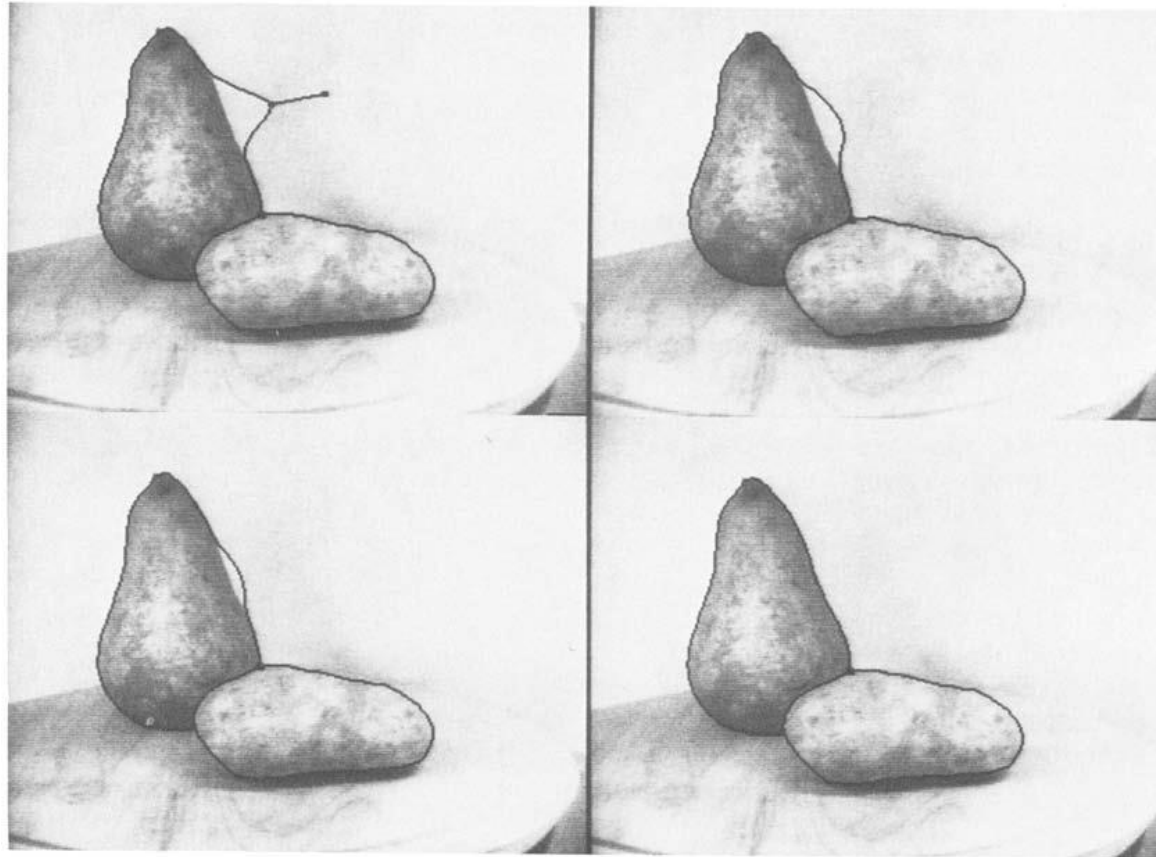


2. 基于边缘的方法

- 2.2 主动边缘模型

经典主动边缘模型 (M. Kass, et al, 1988)

$$E(C) = \int_0^1 (\alpha |C'|^2 + \beta |C''|^2) dq - \lambda \int_0^1 (|\nabla I(C)|) dq$$



基本方法

1. 基于区域的方法
2. 基于边缘的方法
3. 综合考虑边缘和区域信息的混合分割方法
4. 基于特定理论的方法

3. 混合分割方法

- 在实际应用中，为发挥各种方法的优势，克服它们的缺陷以获得更好的分割效果，经常把各种方法结合起来使用。
- 例如，基于区域的分割方法往往会造成图像的过度分割，而单纯的基于边缘检测方法有时不能提供较好的区域结构。为此可将基于区域的方法和边缘检测的方法结合起来解决这个问题。

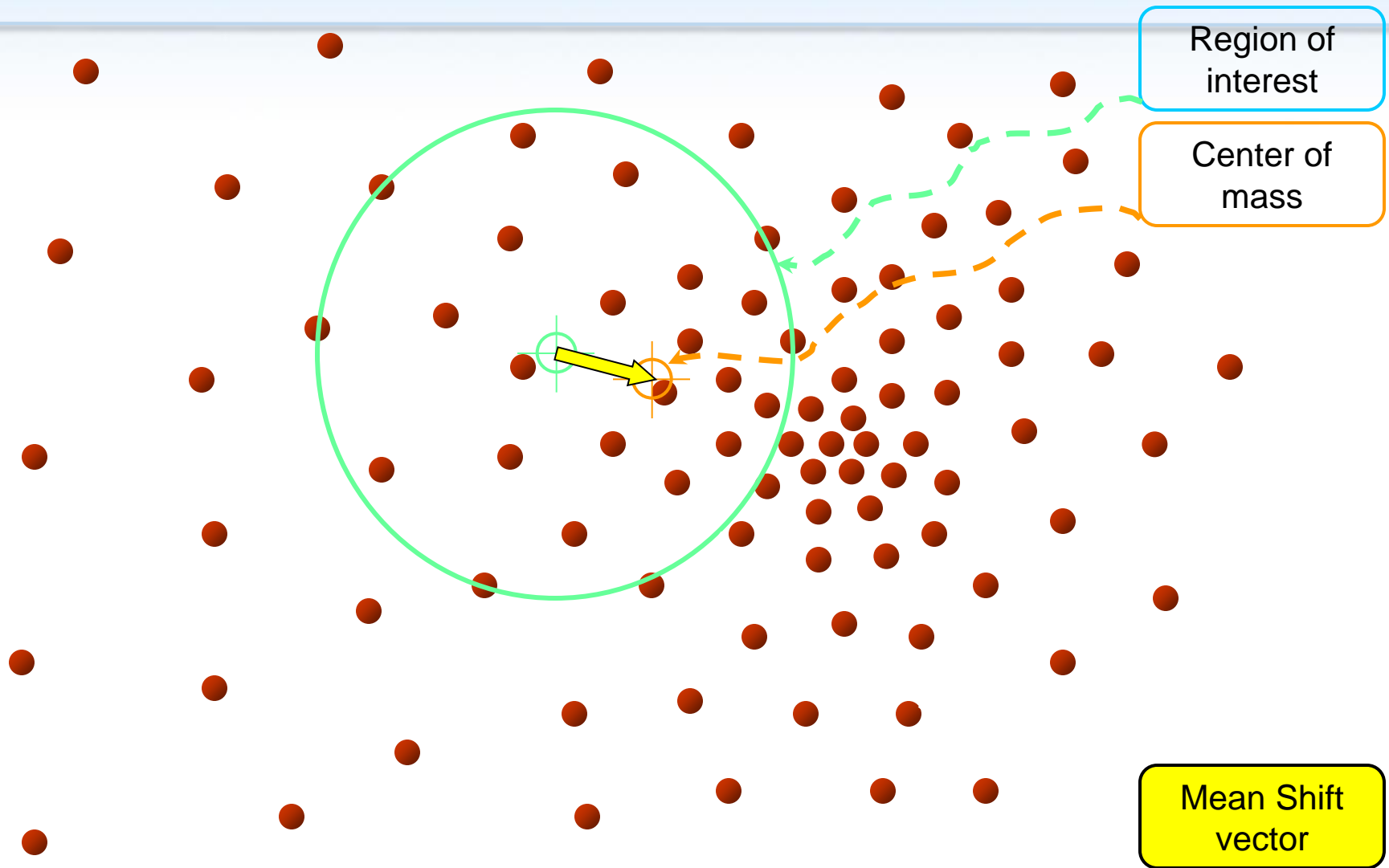
基本方法

1. 基于区域的方法
2. 基于边缘的方法
3. 综合考虑边缘和区域信息的混合分割方法
4. 基于特定理论的方法

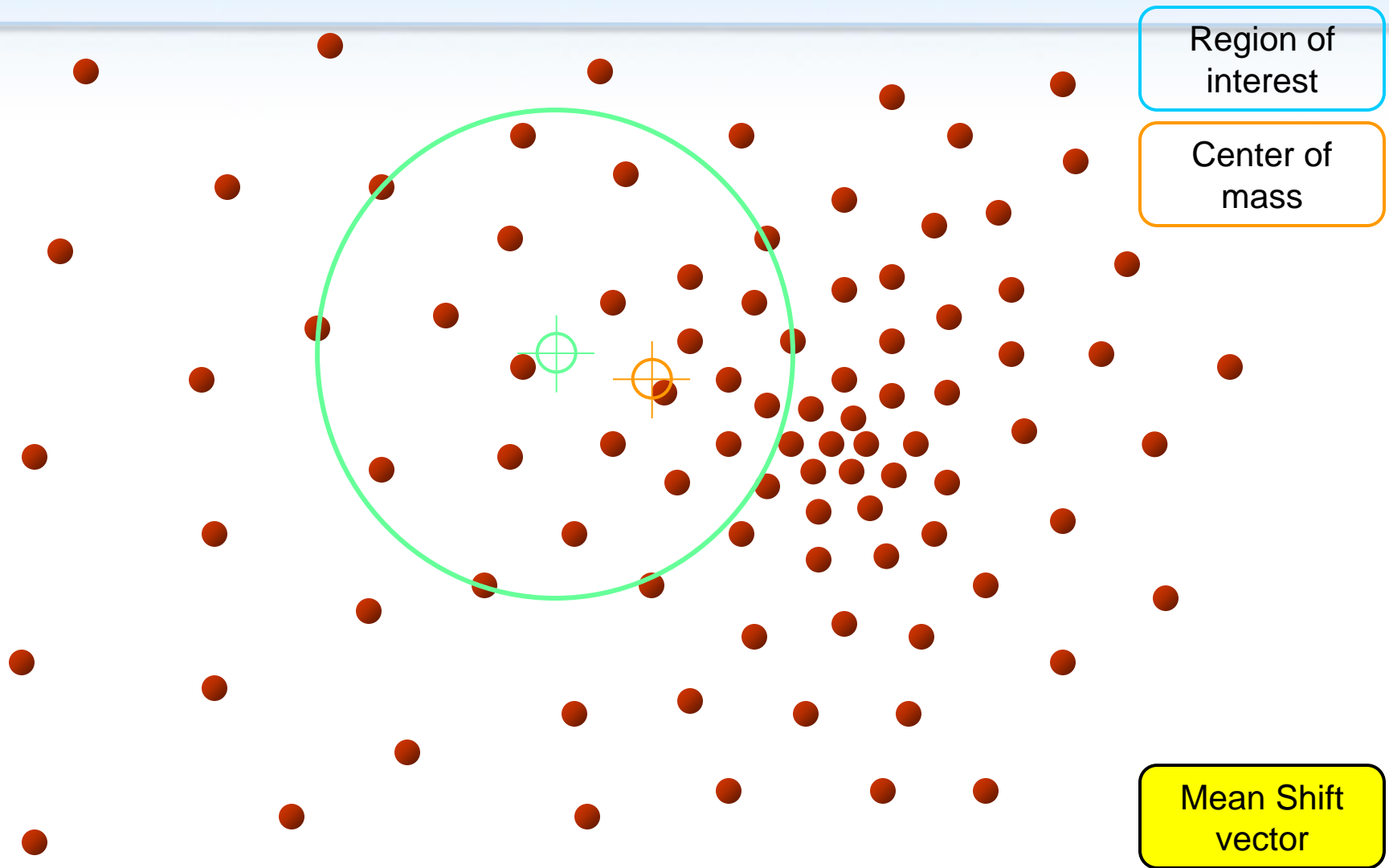
4.1 基于“均值移动”的图象分割方法

- 1975年，Fukunaga和Hostetle提出了一种基于一般核函数的非参数密度梯度的估计算法，并给出了保证估计值与真实值之间渐近无偏、一致和均匀连续时核函数应满足的条件。
- 2001年，Comaniciu (ICCV)，将均值移动应用于图像分析。
- 核心思想：找到概率密度梯度为零的采样点，并以此作为特征空间聚类的模式点。

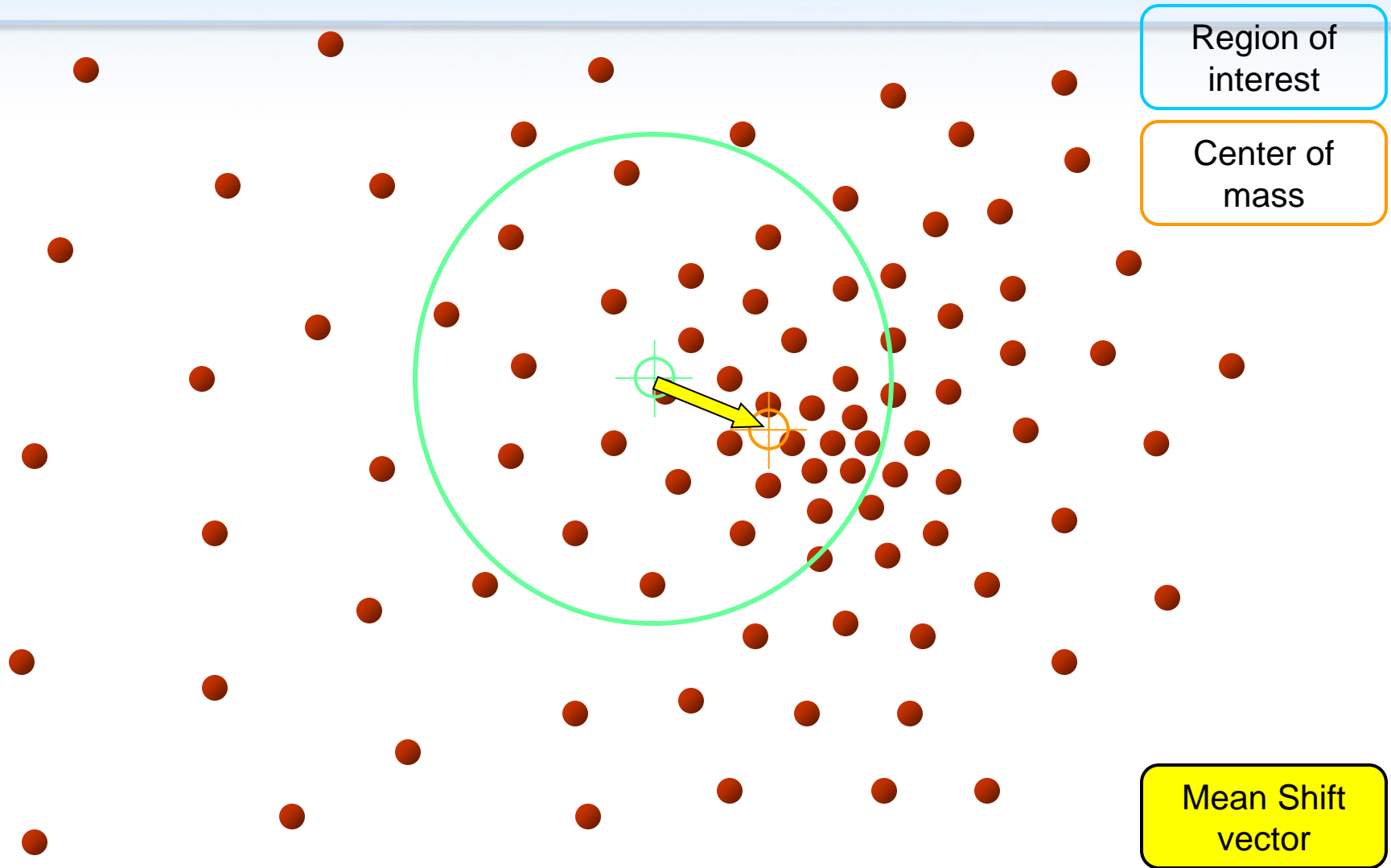
Intuitive Description



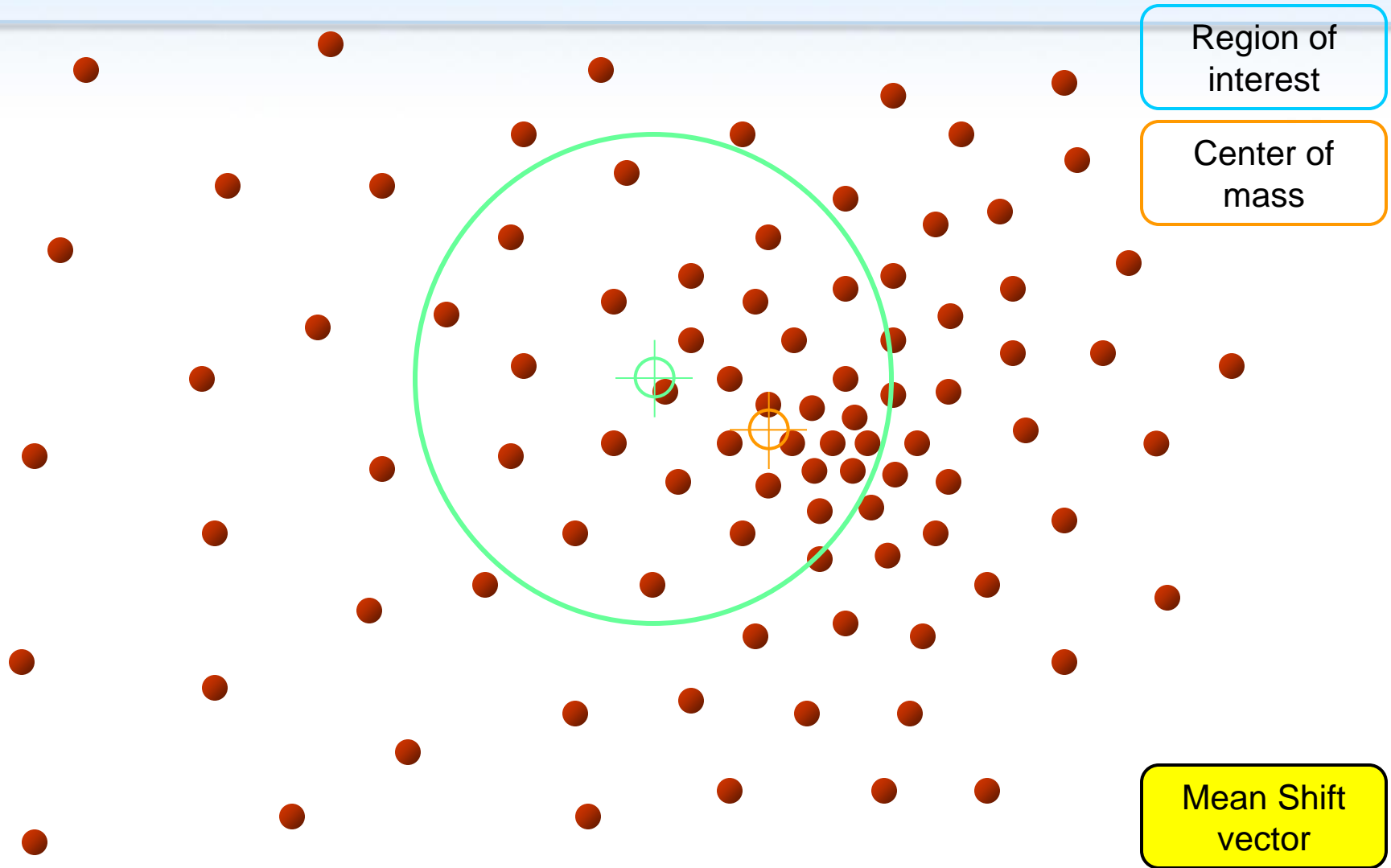
Intuitive Description



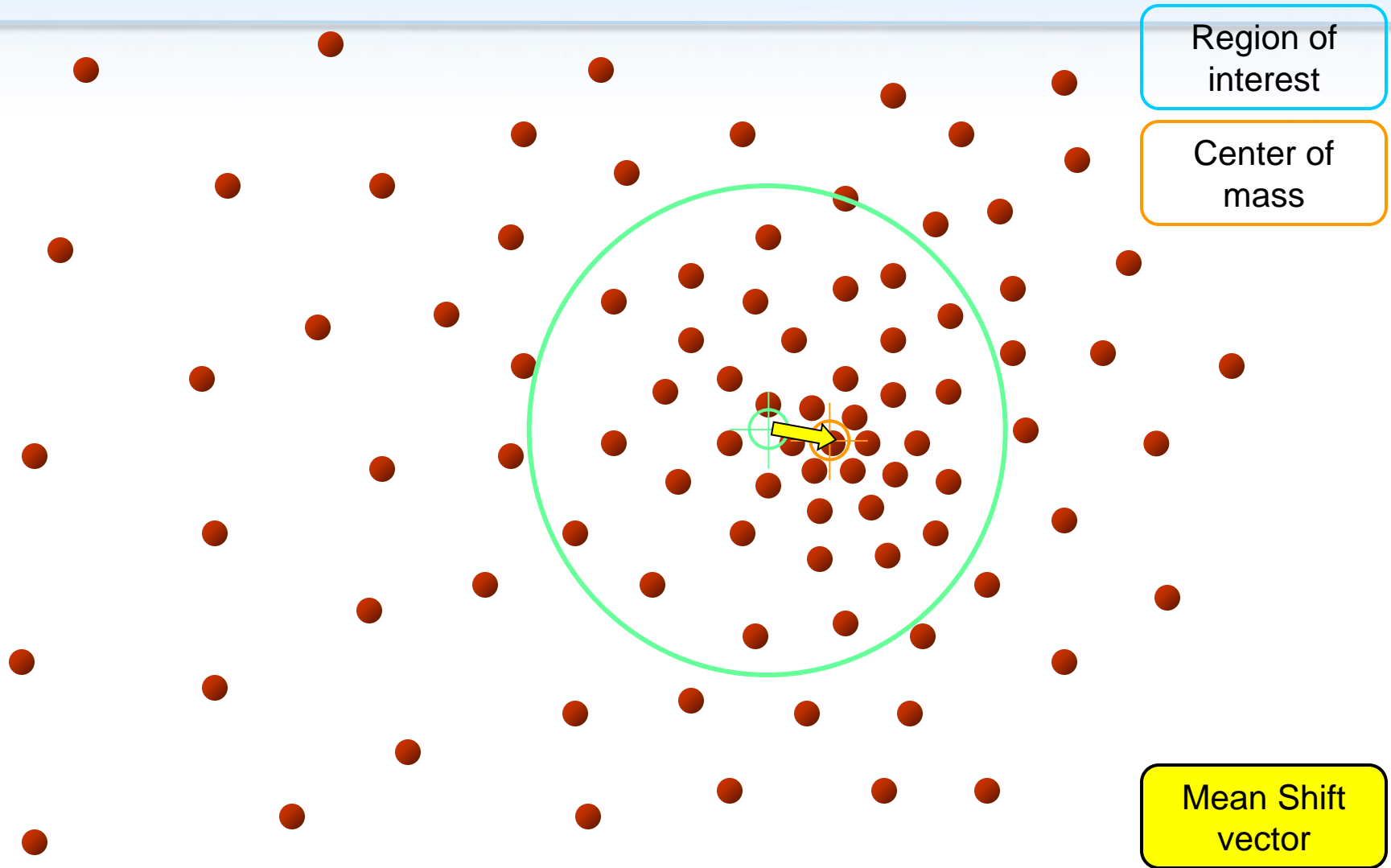
Intuitive Description



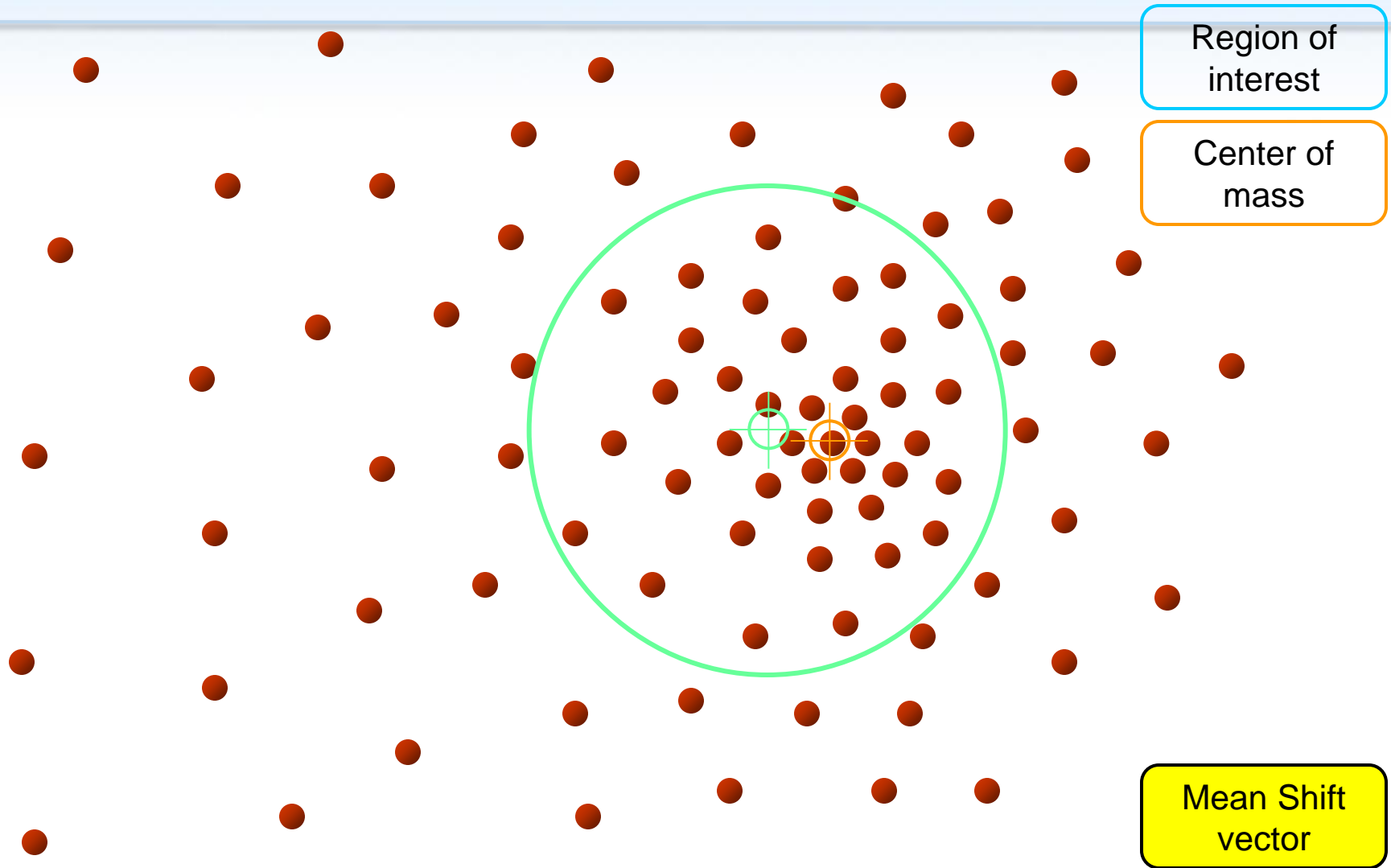
Intuitive Description



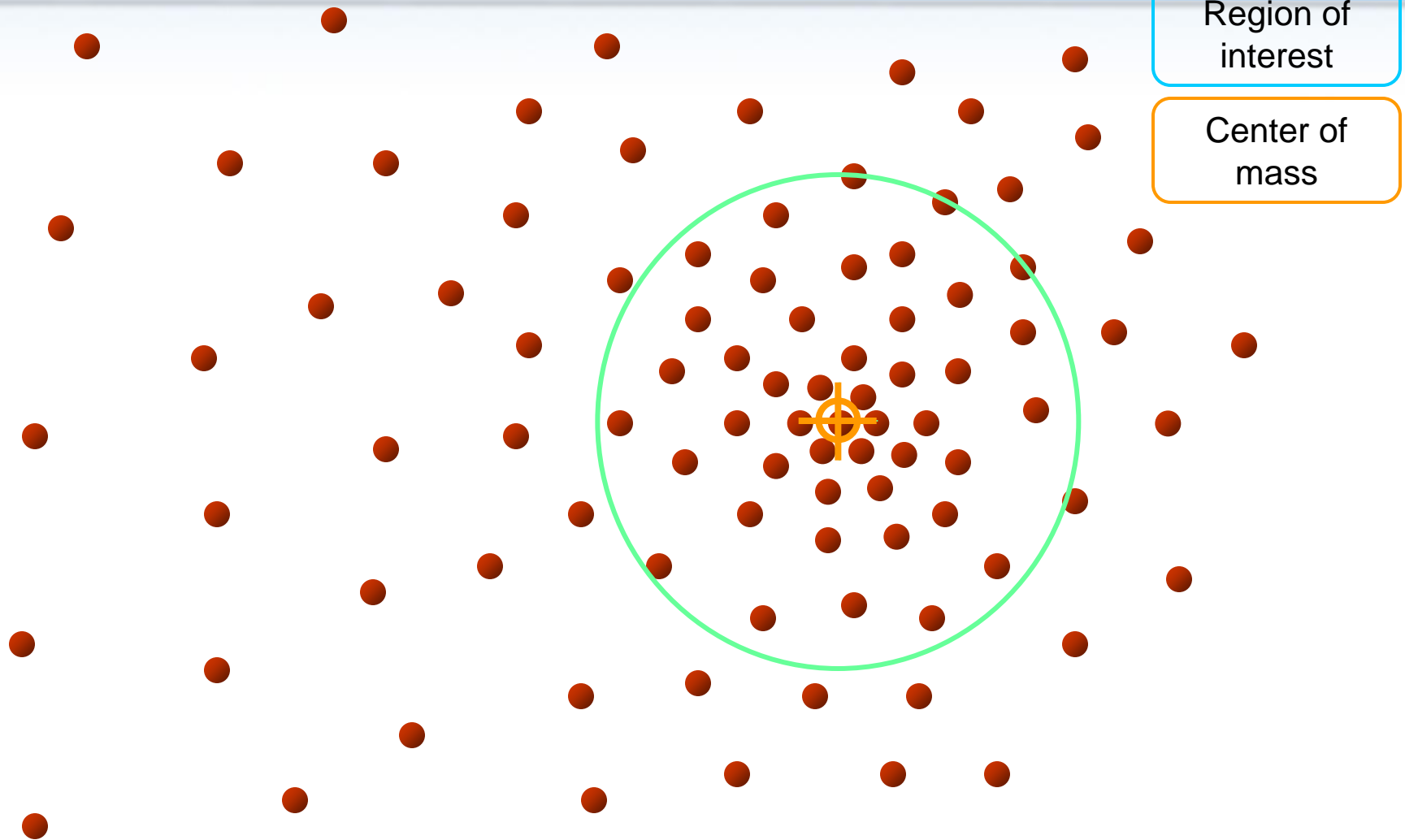
Intuitive Description



Intuitive Description



Intuitive Description

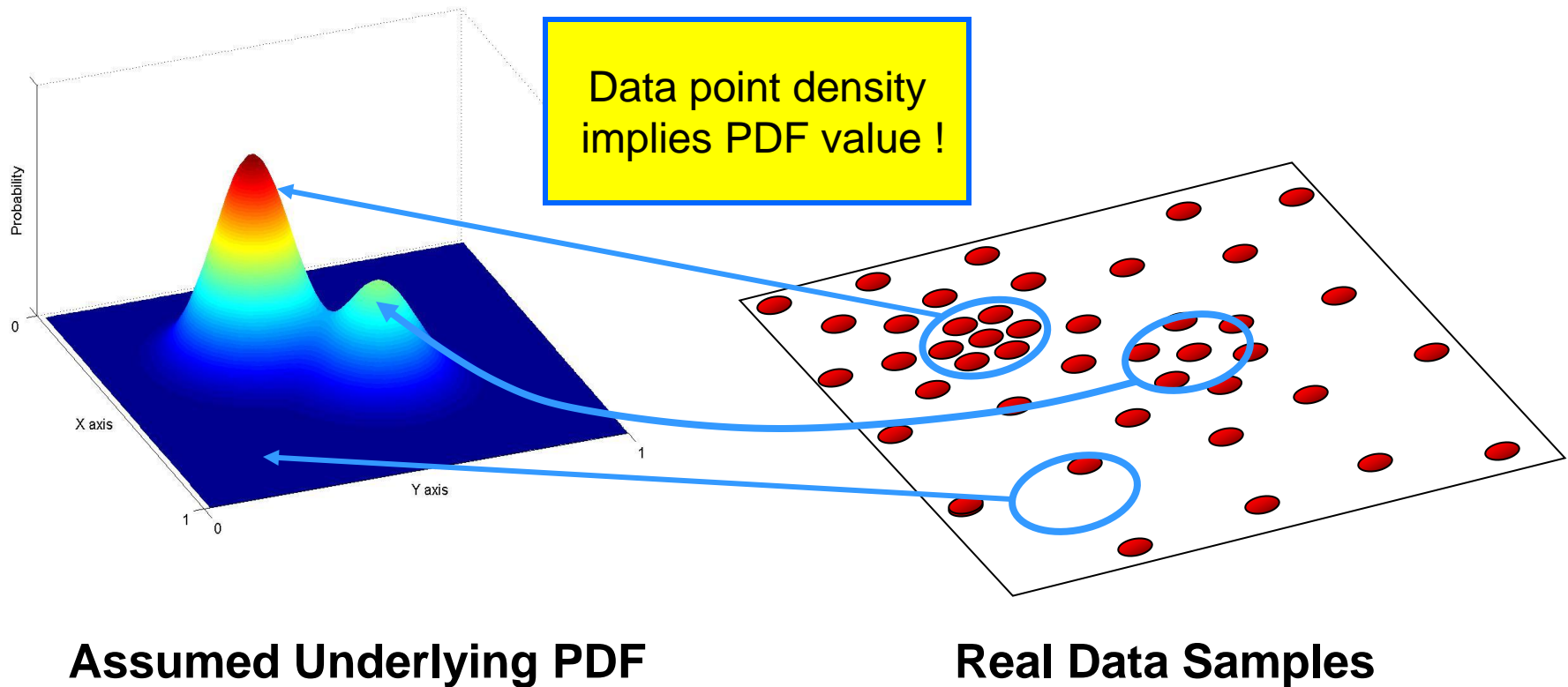


均值移动分割

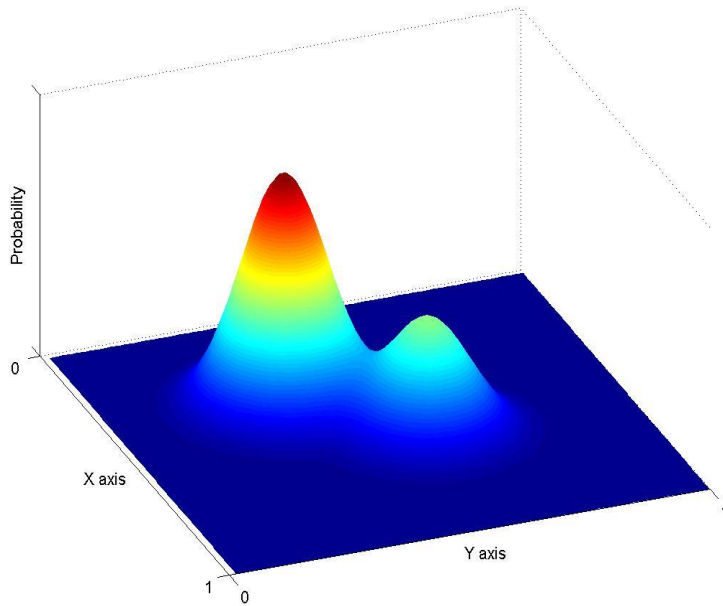
- 核心思想：找到概率密度梯度为零的采样点，并以此作为特征空间聚类的模式点。

Non-Parametric Density Estimation

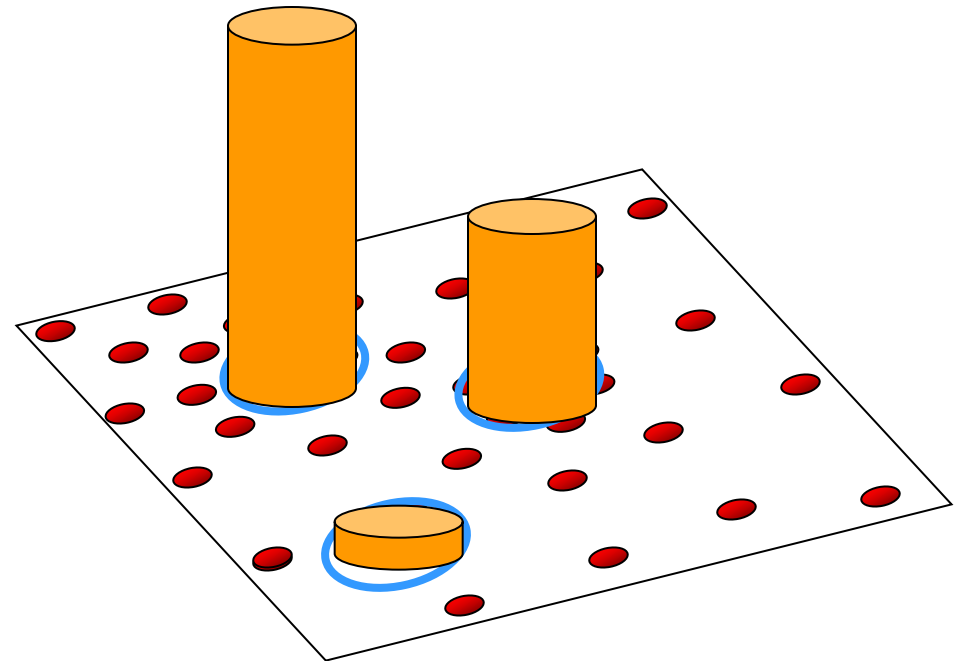
Assumption : The data points are sampled from an underlying PDF



Non-Parametric Density Estimation

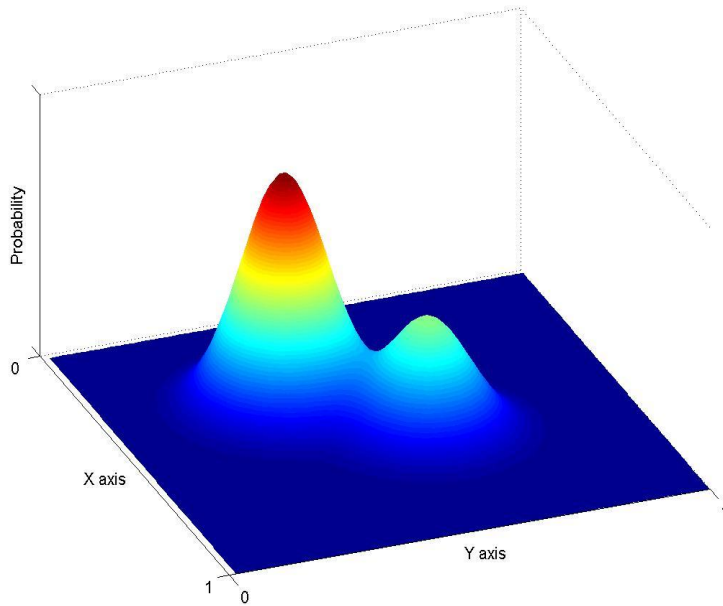


Assumed Underlying PDF

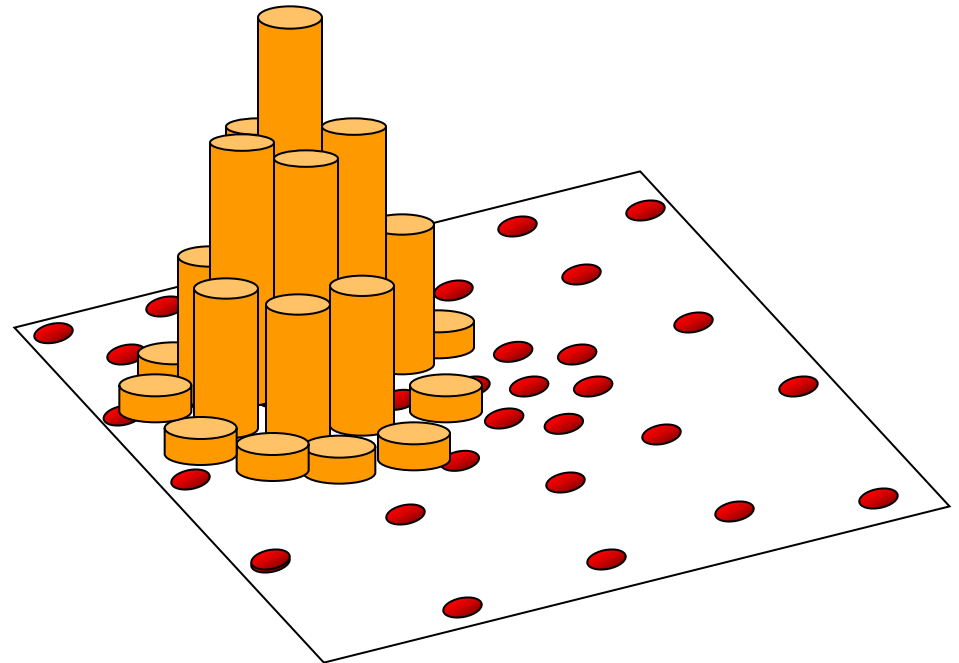


Real Data Samples

Non-Parametric Density Estimation

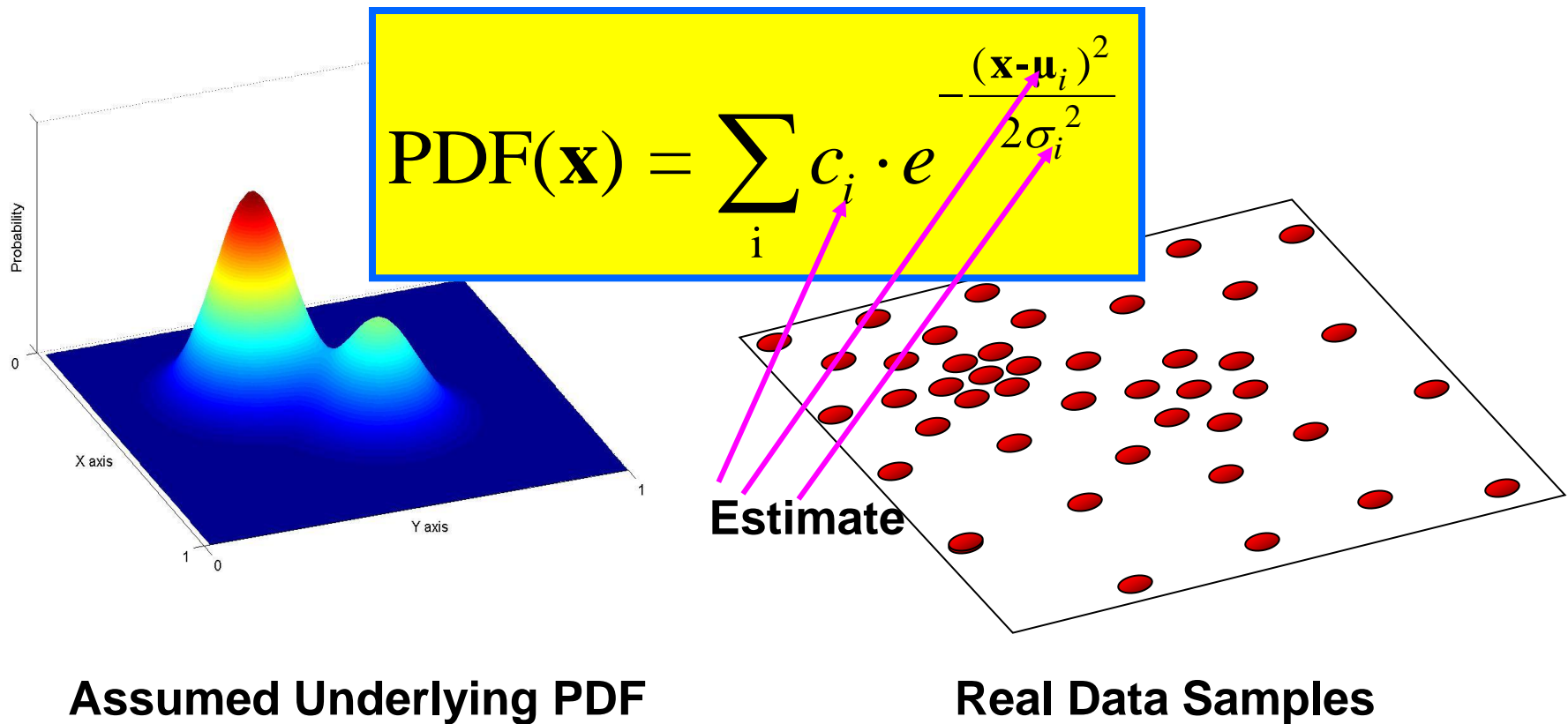


Assumed Underlying PDF



Real Data Samples

Parametric Density Estimation

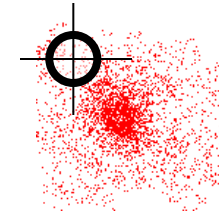


Kernel Density Estimation

Parzen Windows - General Framework

$$P(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n K(\mathbf{x} - \mathbf{x}_i)$$

A function of some finite number of data points
 $\mathbf{x}_1 \dots \mathbf{x}_n$



Data

Kernel Properties:

- Normalized

$$\int_{R^d} K(\mathbf{x}) d\mathbf{x} = 1$$

- Symmetric

$$\int_{R^d} \mathbf{x} K(\mathbf{x}) d\mathbf{x} = 0$$

- Exponential weight decay

$$\lim_{\|\mathbf{x}\| \rightarrow \infty} \|\mathbf{x}\|^d K(\mathbf{x}) = 0$$

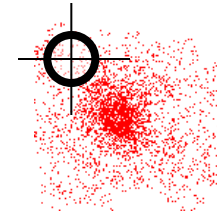
$$\int_{R^d} \mathbf{x} \mathbf{x}^T K(\mathbf{x}) d\mathbf{x} = c \mathbf{I}$$

Kernel Density Estimation

Parzen Windows - Function Forms

$$P(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n K(\mathbf{x} - \mathbf{x}_i)$$

A function of some finite number of data points
 $\mathbf{x}_1 \dots \mathbf{x}_n$



Data

In practice one uses the forms:

$$K(\mathbf{x}) = c \prod_{i=1}^d k(x_i) \quad \text{or} \quad K(\mathbf{x}) = ck(\|\mathbf{x}\|)$$

Same function on each dimension

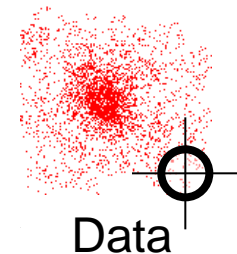
Function of vector length only

Kernel Density Estimation

Various Kernels

$$P(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n K(\mathbf{x} - \mathbf{x}_i)$$

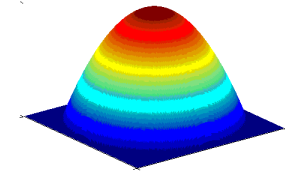
A function of some finite number of data points
 $\mathbf{x}_1, \dots, \mathbf{x}_n$



Examples:

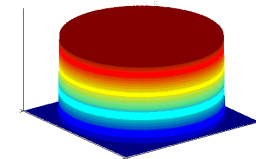
- Epanechnikov Kernel

$$K_E(\mathbf{x}) = \begin{cases} c(1 - \|\mathbf{x}\|^2) & \|\mathbf{x}\| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$



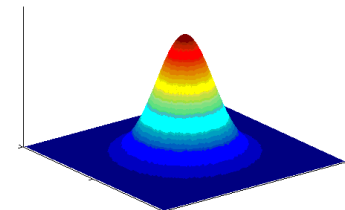
- Uniform Kernel

$$K_U(\mathbf{x}) = \begin{cases} c & \|\mathbf{x}\| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$



- Normal Kernel

$$K_N(\mathbf{x}) = c \cdot \exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)$$



Kernel Density Estimation

Gradient

$$\nabla P(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \nabla K(\mathbf{x} - \mathbf{x}_i)$$

Give up estimating the PDF !
Estimate ONLY the gradient

Using the
Kernel form:

$$K(\mathbf{x} - \mathbf{x}_i) = ck \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)$$

We get :

Size of window

$$\nabla P(\mathbf{x}) = \frac{c}{n} \sum_{i=1}^n \nabla k_i = \frac{c}{n} \left[\sum_{i=1}^n g_i \right] \cdot \left[\frac{\sum_{i=1}^n \mathbf{x}_i g_i}{\sum_{i=1}^n g_i} - \mathbf{x} \right]$$

$$g(\mathbf{x}) = -k'(\mathbf{x})$$

Kernel Density Estimation

Computing The Mean Shift

Gradient

$$\nabla P(\mathbf{x}) = \frac{c}{n} \sum_{i=1}^n \nabla k_i = \frac{c}{n} \left[\sum_{i=1}^n g_i \right] \cdot \left[\frac{\sum_{i=1}^n \mathbf{x}_i g_i}{\sum_{i=1}^n g_i} - \mathbf{x} \right]$$

$$g(\mathbf{x}) = -k'(\mathbf{x})$$

Computing The Mean Shift

$$\nabla P(\mathbf{x}) = \frac{c}{n} \sum_{i=1}^n \nabla k_i = \frac{c}{n} \left[\sum_{i=1}^n g_i \right] \cdot \left[\frac{\sum_{i=1}^n \mathbf{x}_i g_i}{\sum_{i=1}^n g_i} - \mathbf{x} \right]$$

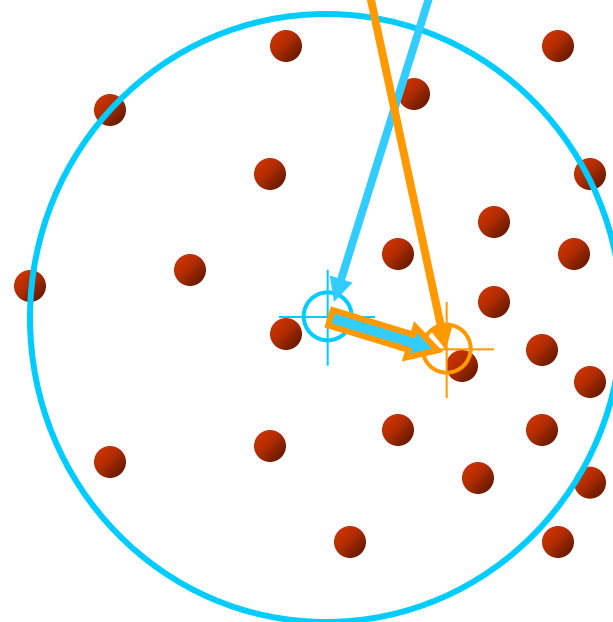
Yet another Kernel density estimation !

Simple Mean Shift procedure:

- Compute mean shift vector

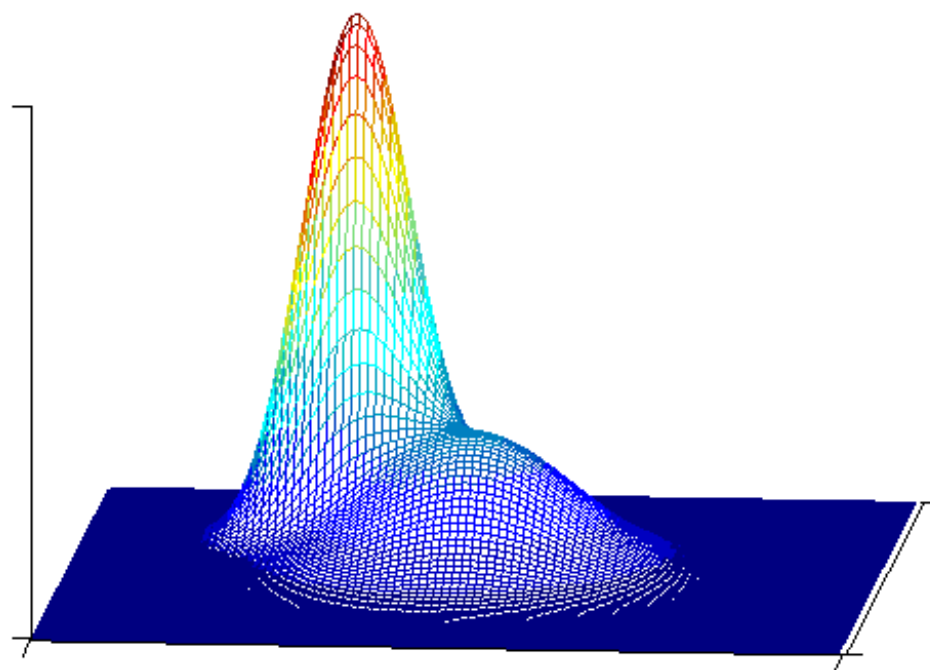
$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^n \mathbf{x}_i g \left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h} \right)}{\sum_{i=1}^n g \left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h} \right)} - \mathbf{x}$$

- Translate the Kernel window by $\mathbf{m}(\mathbf{x})$

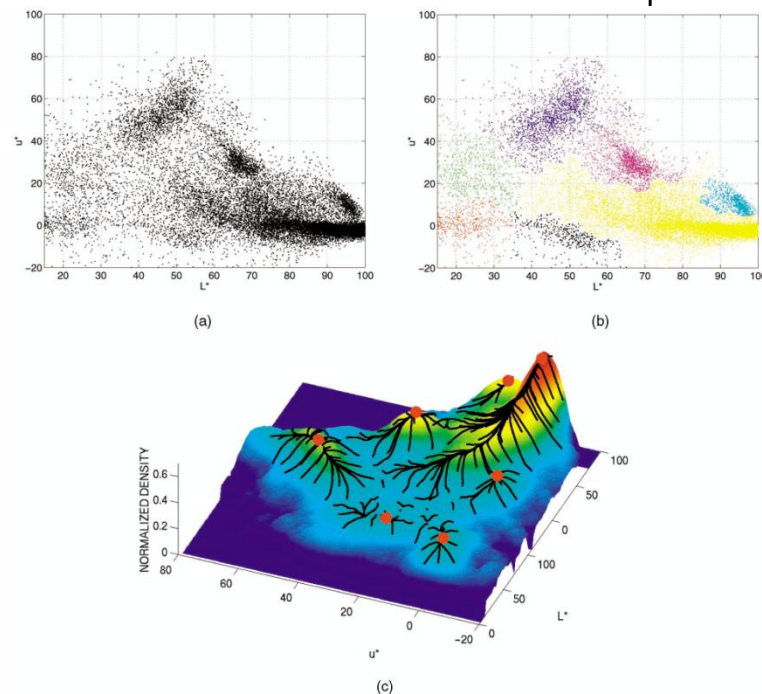


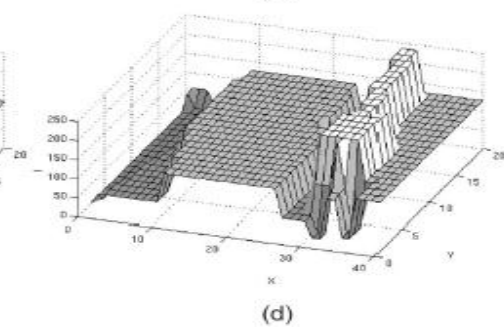
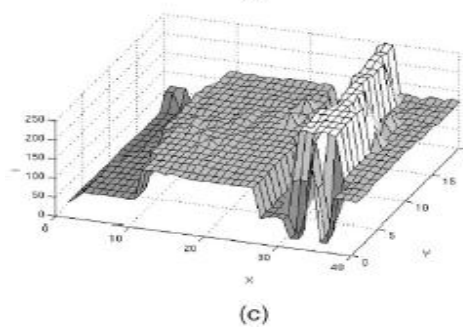
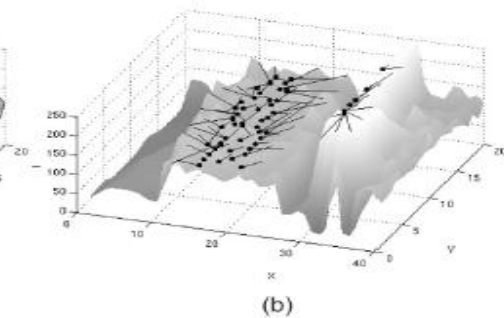
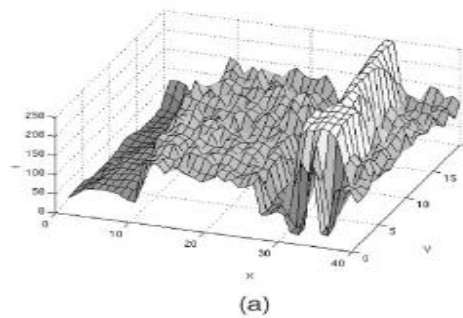
$$g(\mathbf{x}) = -k'(\mathbf{x})$$

1. 计算 $m(x)$;
2. 如果 $\|m(x) - x\|$ 小于一个给定的阈值, 结束循环; 否则, 将 $m_y(x)$ 赋给 x , 继续执行(1)。



- Find features (color, gradients, texture, etc)
- Set kernel size for features K_f and position K_s
- Initialize windows at individual pixel locations
- Perform mean shift for each window until convergence
- Merge windows that are within width of K_f and K_s





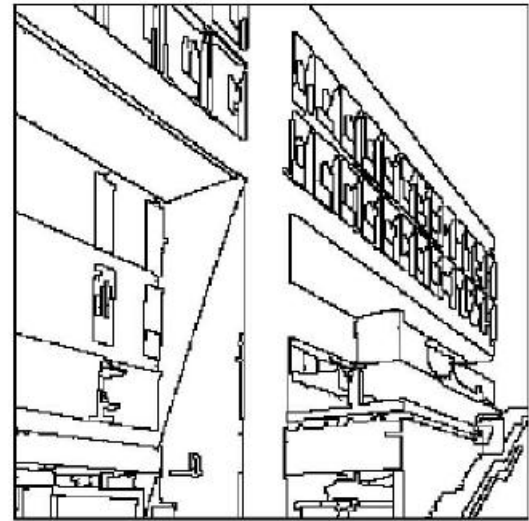
Results



(a)



(b)



(c)

Results

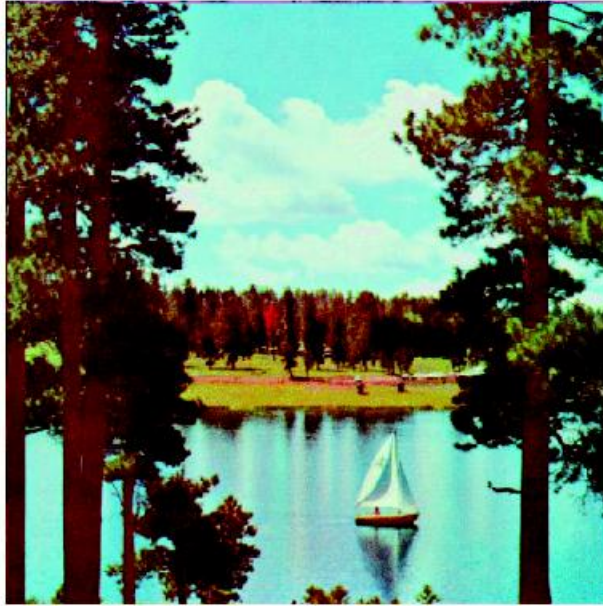


(a)



(b)

Results



(a)



(b)



Graph Cut (图割)

- 基本思想：
 1. 将图像用图的方式表示，顶点表示像素，边表示像素之间的关系。图像分割对应图的割集。
 2. 确定图中边的权值，使图像分割目标（能量最小化）对应图的最小割。
 3. 用最大流算法求解最小割问题。

图论的相关知识

• 图 (Graph) :

由点集和边集构成的集合 $G = (V, E)$ 。

V -----点集

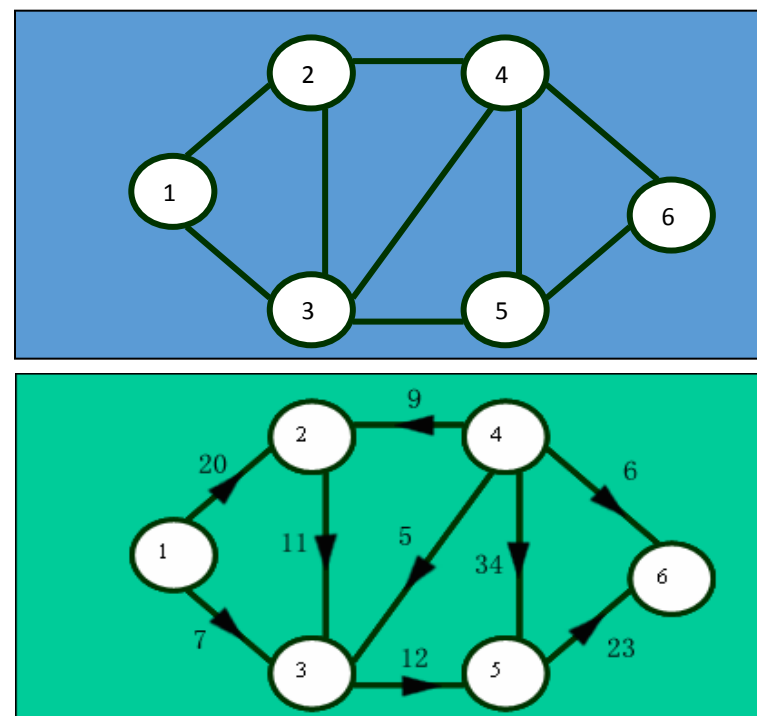
E -----边集

赋权图：每条边赋有一个权值 $W(p, q)$

有向图：每条边有方向

有向赋权图： 每条边既有方向又有一个权值

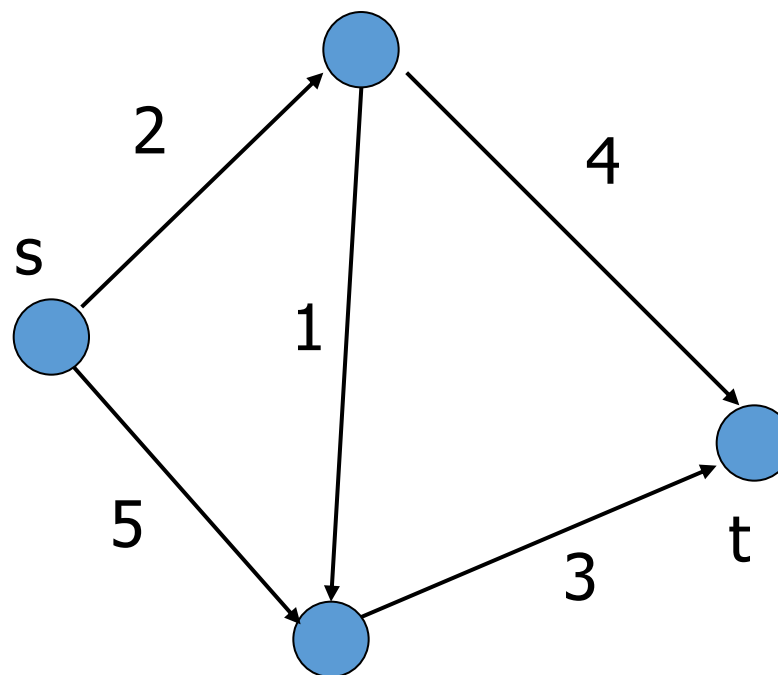
无向赋权图.....



S-T 图

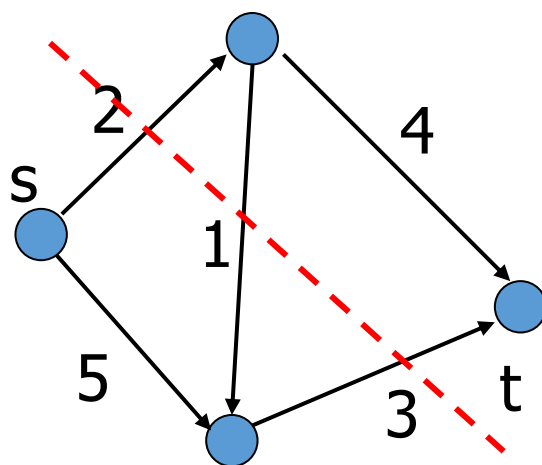
- 有源节点 (s) 和终节点 (t)
- 每条边有一个非负的容量 $\text{Cap}(i,j)$
- 对于不存在的边, 其容量为0

$$G = (V, E)$$

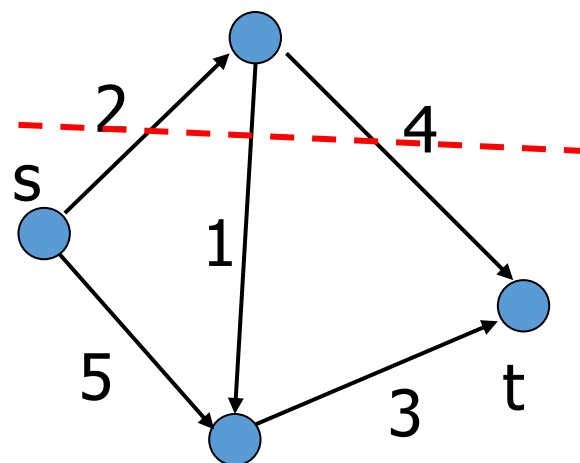


S-T割

- 割：将s-t图分成两个子集 S 和 T
- s-t割：当且仅当 s 属于 S ， t 属于 T



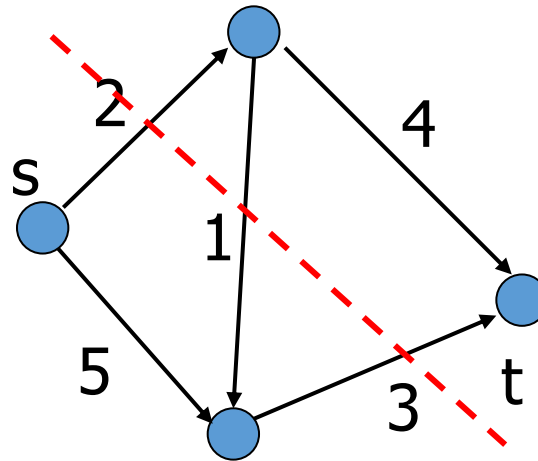
s-t割



非s-t割

最小割 (min cut)

- $Cap([S, T]) = \sum_{\{i,j\} \in E, i \in S, j \in T} Cap(i, j)$



s-t图中容量最小的s-t割

- 两种基于Graph cut的算法在计算视觉中的应用
 - 1 Normalized cut及其在图像分割中的应用
 - 2 用graph cut算法求解计算机视觉中的能量极小化问题

- 1 Normalized cut及其在图像分割中的应用
 - (1) 图论在分类问题中的应用
 - (2) Normalized cut
 - (3) 求解最小Normalized cut的近似算法
 - (4) Normalized cut 在图像分割中应用

(1) 图论在分类问题中的应用

- 一般的分类问题

给定一个点集 V ，按照一定的相似度量（距离）寻求一种划分，将点集 V 划分成不相交的若干子集合 $V_1, V_2 \dots V_m$ 。使得每一子集内部的相似度尽量高，而子集之间的相似度尽量低。

两个问题：

- 什么是最优划分准则？
- 有没有有效算法？

- 用图论的方法来解决分类问题

在给定点集 V 中的每一点对 i, j 之间，建立一条边 (i, j) ，给这条边赋权 $w(i, j)$ 相似度量。这样就建立了一个赋权图（无向）。

对于这样的图我们可以用邻接矩阵来表示 ($w(i, j) = w_{ij}$)

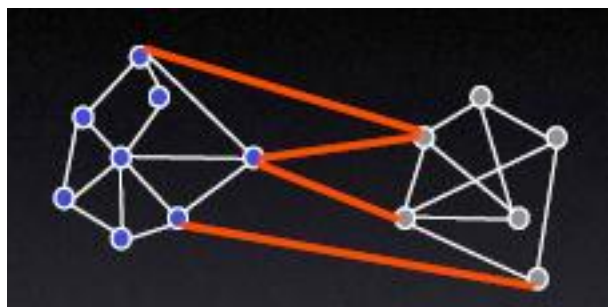
$$\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix}$$

考虑二分类问题

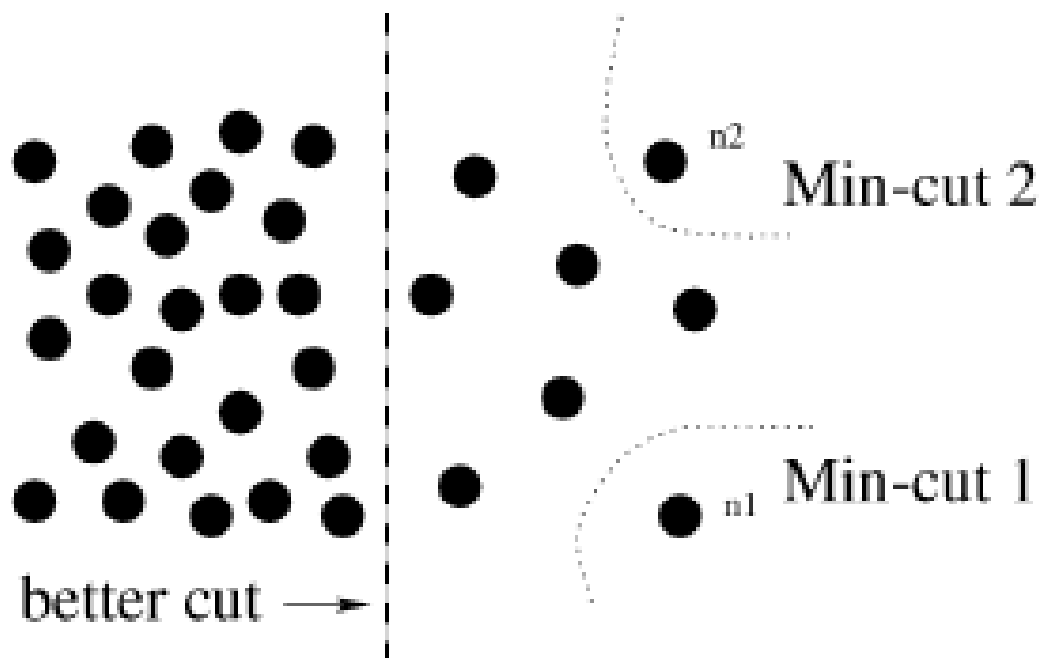
将点集 V 分成不相交的两部分 A 、 B 。则两类别之间的相似性我们可以用图割来度量。

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

显然最优划分应使 $cut(A, B)$ 取最小值



- 但是仅考虑用割集的权值之和来度量两个集合之间的相关性，会容易出现孤立分割的问题。如下图所示：



(2) Normalized cut

一个解决上述问题的办法是通过定义新的类间相似性度量。

Normalized cut ($Ncut$):

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

这里 $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$

这样包含孤立点的 $Ncut$ 值不会小。

再定义总的类内相似性度量

$$Nassoc(A, B) = \frac{assoc(A, A)}{assoc(A, V)} + \frac{assoc(B, B)}{assoc(B, V)}$$

- 通过简单的推导可以证明

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} = 2 - Nassoc(A, B)$$

可见最小化类间相似性和最大化总类内相似性是等价的。这样就解决了划分准则的问题，即

最优划分对应于最小 $Ncut$

通过求最小 $Ncut$ ，就可以得到最优划分。

下一个问题是有没有求最小 $Ncut$ 的有效算法？

(3) 求解最小 $Ncut$ 的近似算法

不幸的是求一个图的 $Ncut$ 是一个NP问题。

但是通过精巧的构造，可以通过求解如下广义特征值问题来得到最小 Normalized cut 近似解²。

令 $W = [w_{ij}]$

$$D = \text{diag}(d(1), d(2), \dots, d(N)) , \quad d(i) = \sum_j w_{ij}$$

则广义特征值问题

$$(D - W)x = \lambda Dx$$

的次小特征值是最小 $Ncut$ 对应的实数解。该特征值所对应的特征向量对应于最优划分

- 特征向量离散化

由于我们需要特征向量是仅含有不同符号的两个值（见推导过程^[2]），故还需要对所得特征向量做离散化处理。即需要选择一个分界点。有两种方法：

（1）取中点

（2）取0

- 多分类问题

首先用次小特征值所对应特征向量进行二分类。然后用再次小特征值所对应的特征向量对已分好的两类再次细分。或者在每个分好的类别中，重复用上述算法进行分类

具体算法：

- 给定一个点集，构建图 $G(V, E)$ ，边的权为对应两端点的相似度
- 求解 $(D - W)x = \lambda Dx$ 的特征值及其所对应的特征向量
- 用次小特征值所对应的特征向量进行二分类
- 若需再分，则在每个分好的类别中重复上述过程。否则终止。

(4) Normalized cut 在图像分割中应用

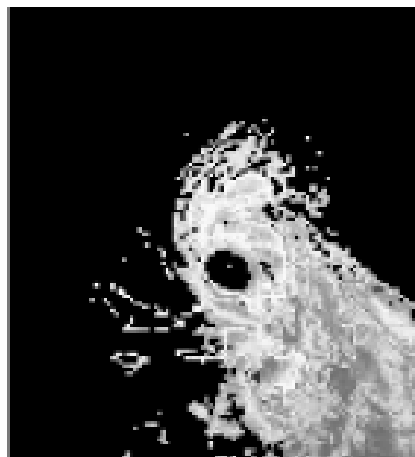
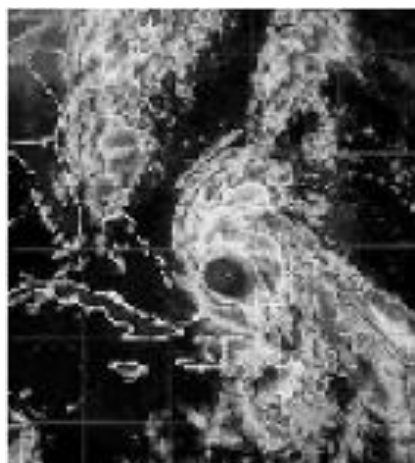
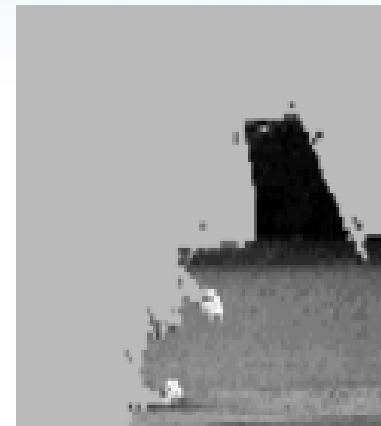
将一幅图像上所有像素点看作点集 V ，每两个点之间都建立一条边，得到边集 E 。为每条边按下面方法赋权

$$w_{ij} = e^{\frac{-\|F_i - F_j\|^2}{\sigma_I}} * \begin{cases} e^{\frac{-\|X_i - X_j\|^2}{\sigma_X}}, & \text{if } \|X_i - X_j\|_2 < r \\ 0, & \text{otherwise} \end{cases}$$

这样就建立一个赋权无向图 $G = (V, E)$

按照前述算法，我们就可以完成对该幅图像分割操作。

实验结果[2]

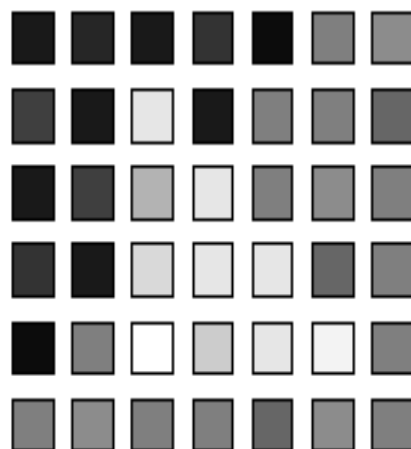


- 2 用graph cut算法求解计算机视觉中的能量极小化问题
 - (1) 计算机视觉中的多标记问题
 - (2) 多标记问题的能量极小化模型
 - (3) 运用Graph cut算法求解这类问题

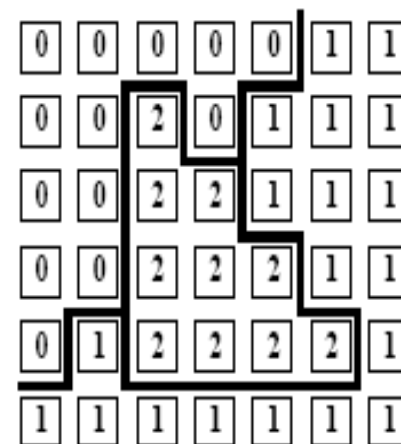
(1) 计算机视觉中的多标记问题

计算机视觉的很多问题可以看作一个最优标记 (labeling) 问题。如：

- 图像分割
- 图像恢复
- 立体视觉
- 三维重建



(a) An image



(b) A labeling

同样面临的两个问题：

- 什么是最优标记准则？
- 有没有有效求解算法

(2) 多标记问题的能量极小化模型

我们可以通过构造一个如下的能量函数来得到最优标记准则。

$$E(f) = E_{data}(f) + E_{smooth}(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in P} V_{p,q}(f_p, f_q)$$

$f: P \rightarrow L$ 的映射。 P 是像素点集， L 是标记集。

Data项表示给每个像素点赋予标记（label）的代价

Smooth项表示每两个相邻的像素分别赋予标记 f_p 和 f_q 的代价

我们希望最优标记应使得能量函数取最小值，即可建立下面最优化模型

$$\begin{aligned} \min E(f) \\ s. t. \quad f_p \in L \end{aligned}$$

下面具体建立各问题的能量极小化模型。

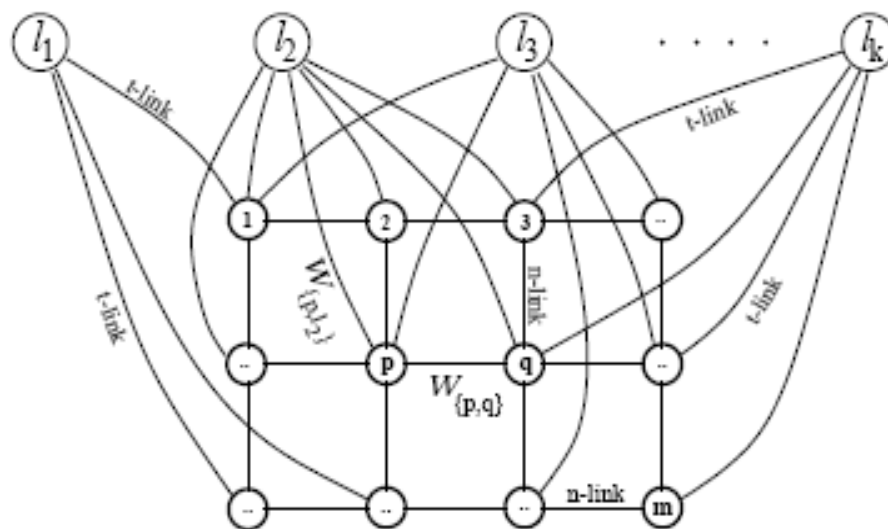
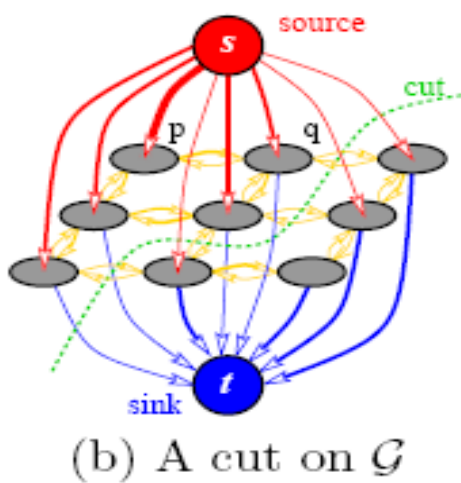
(3) 运用Graph cut 算法求解这类问题

- graph cut与能量函数的对应关系
- 运用graph cut算法求解能量极小化问题
 - 两标记问题
 - 多标记问题

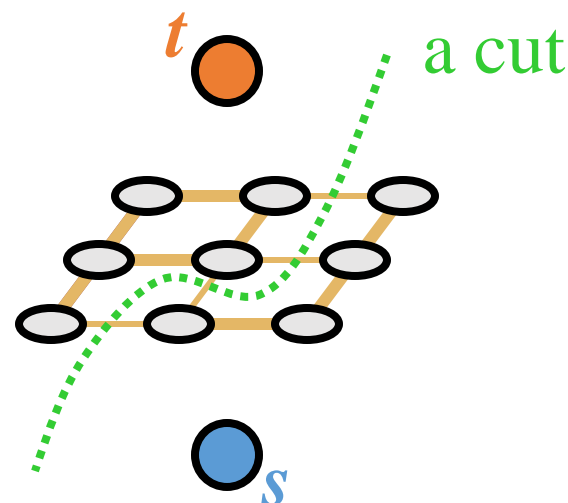
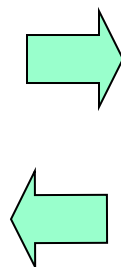
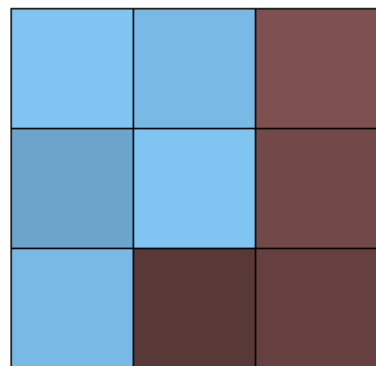
图割与能量函数的对应关系

• 能量函数

$$E(f) = E_{data}(f) + E_{smooth}(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} V_{p,q}(f_p, f_q)$$



图割与能量函数的对应关系

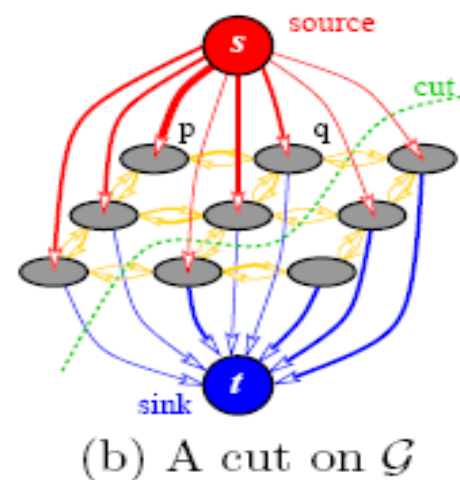


$$E(f) = E_{data}(f) + E_{smooth}(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} V_{p,q}(f_p, f_q)$$

- 运用graph cut算法求解能量极小化问题

- 两标记问题

对于两标记问题，最小能量对应于图的最小割。图论中已有经典的算法，可以求得一个图的最小割，从而得到极小能量。具体算法见参考文献[1]、[3]。



- 多标记问题

当标记数量大于2时，已经证明该问题是NP-hard问题。故很难求得该问题的全局极小值。

Boykov等^[4]构造了两个运用最小割求解该类能量函数的近似极小值的算法：

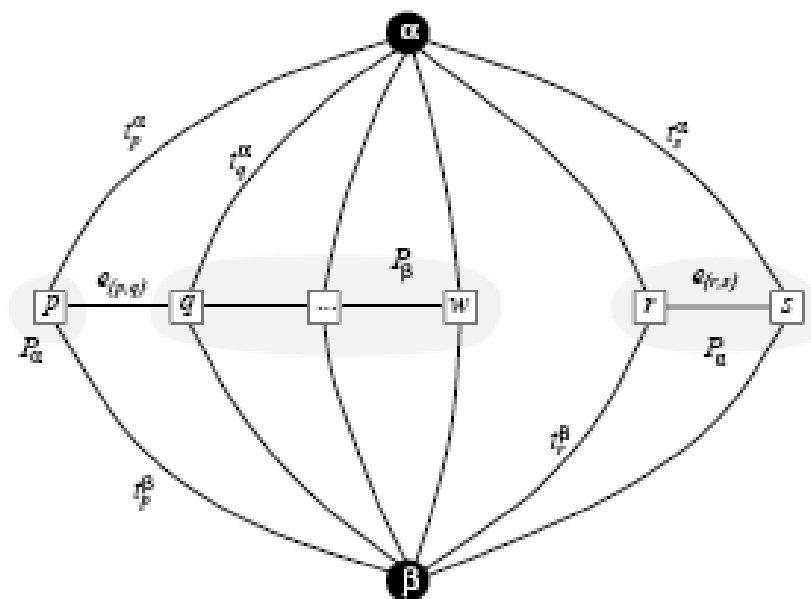
$\alpha - \beta$ swap

α expansion

这两个算法运算速度快，且能得到比较好的结果，从而得到了广泛应用，并使得用能量极小化模型和图割来处理计算机视觉中的一些问题成为目前的一个研究热点。

α - β swap

1. Start with an arbitrary labeling f
2. Set $\text{success} := 0$
3. For each pair of labels $\{\alpha, \beta\} \subset \mathcal{L}$
 - 3.1. Find $\hat{f} = \arg \min E(f')$ among f' within one α - β swap of f (Section 3)
 - 3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and $\text{success} := 1$
4. If $\text{success} = 1$ goto 2
5. Return f



edge	weight	for
t_p^α	$D_p(\alpha) + \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_{\alpha\beta}}} V_{\{p,q\}}(\alpha, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
t_p^β	$D_p(\beta) + \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_{\alpha\beta}}} V_{\{p,q\}}(\beta, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
$e_{\{p,q\}}$	$V_{\{p,q\}}(\alpha, \beta)$	$\{p,q\} \in \mathcal{N}$ $p, q \in \mathcal{P}_{\alpha\beta}$

$$f_p^{\mathcal{C}} = \begin{cases} \alpha & \text{if } t_p^\alpha \in \mathcal{C} \text{ for } p \in \mathcal{P}_{\alpha\beta} \\ \beta & \text{if } t_p^\beta \in \mathcal{C} \text{ for } p \in \mathcal{P}_{\alpha\beta} \\ f_p & \text{for } p \in \mathcal{P}, p \notin \mathcal{P}_{\alpha\beta} . \end{cases}$$

α expansion

1. Start with an arbitrary labeling f
2. Set $\text{success} := 0$
3. For each label $\alpha \in \mathcal{L}$
 - 3.1. Find $\hat{f} = \arg \min E(f')$ among f' within one α -expansion of f (Section 4)
 - 3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and $\text{success} := 1$
4. If $\text{success} = 1$ goto 2
5. Return f

α expansion

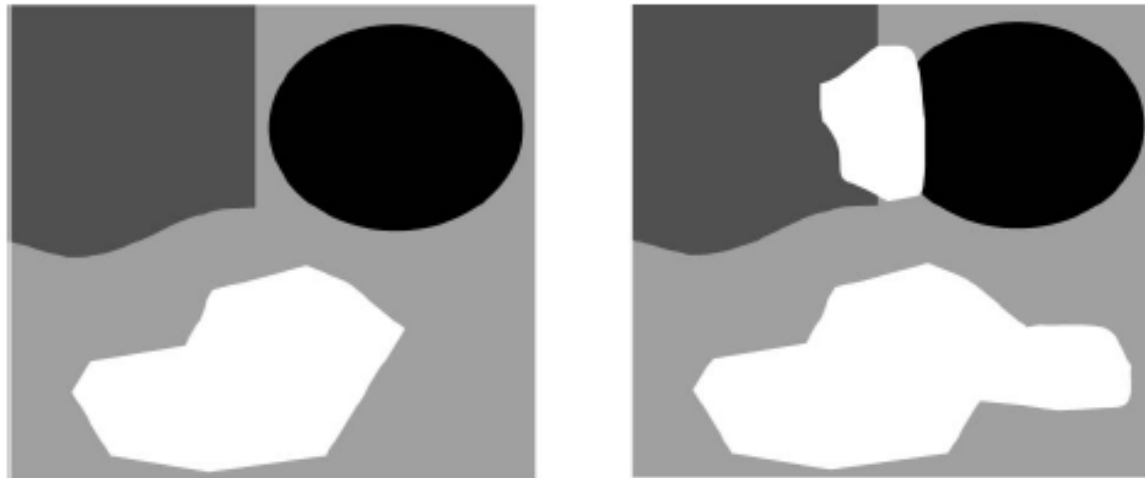
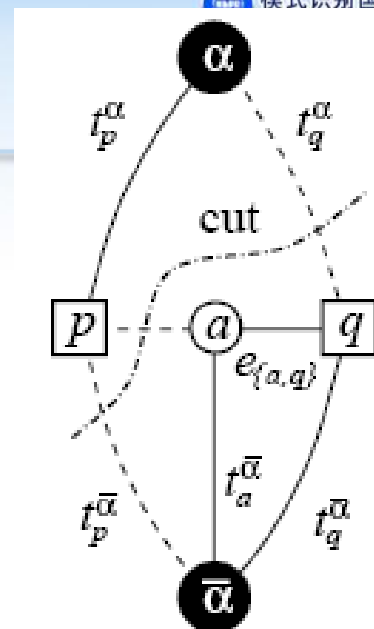
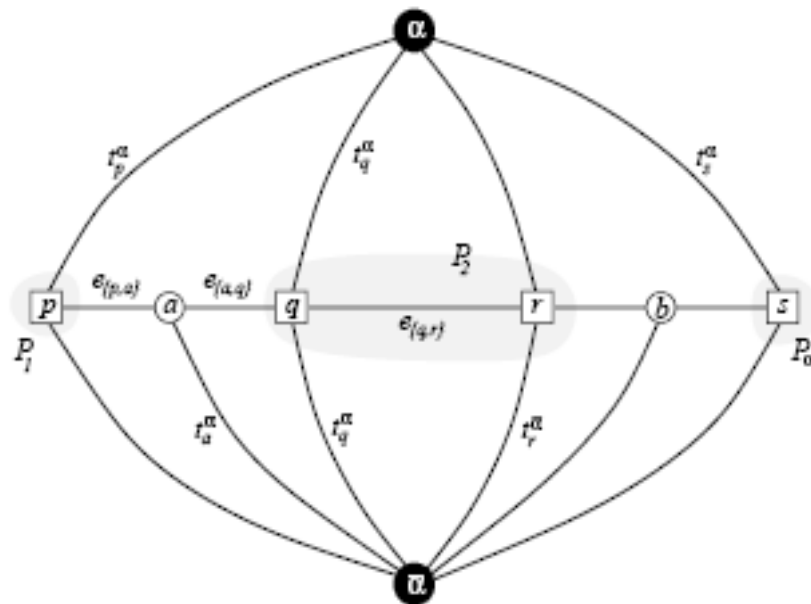


Fig. 1. An example of an expansion move. The labeling on the right is a white-expansion move from the labeling on the left.



edge	weight	for
$t_p^{\bar{\alpha}}$	∞	$p \in \mathcal{P}_\alpha$
$t_p^{\bar{\alpha}}$	$D_p(f_p)$	$p \notin \mathcal{P}_\alpha$
t_p^α	$D_p(\alpha)$	$p \in \mathcal{P}$
$e_{\{p,a\}}$	$V_{\{p,q\}}(f_p, \alpha)$	$\{p, q\} \in \mathcal{N}, f_p \neq f_q$
$e_{\{a,q\}}$	$V_{\{p,q\}}(\alpha, f_q)$	
$t_a^{\bar{\alpha}}$	$V_{\{p,q\}}(f_p, f_q)$	
$e_{\{p,q\}}$	$V_{\{p,q\}}(f_p, \alpha)$	

$$f_p^{\mathcal{C}} = \begin{cases} \alpha & \text{if } t_p^\alpha \in \mathcal{C} \\ f_p & \text{if } t_p^{\bar{\alpha}} \in \mathcal{C} \end{cases} \quad \forall p \in \mathcal{P}.$$

Graph Cut 参考文献

- [1] 楼世博等, 图论及其应用. 人民邮电出版社, 1982.
- [2] Jianbo Shi, Jitendra Malik, Normalized cuts and image segmentation[J]. IEEE Trans on PAMI, 2000; 22(8): 888-905.
- [3] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), number 2134 in LNCS, pages 359–374, Sophia Antipolis, France, September 2001. SpringerVerlag.
- [4] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11):1222–1239, November 2001.

小节

- 在图象分割领域还没有出现对任意图象都可以分割的算法，需要根据问题的不同设计和采用不同的算法，还可能要考虑时空复杂度的可接受性。

谢谢！