

计算机视觉—图像分割

申抒含

中国科学院自动化研究所
模式识别国家重点实验室

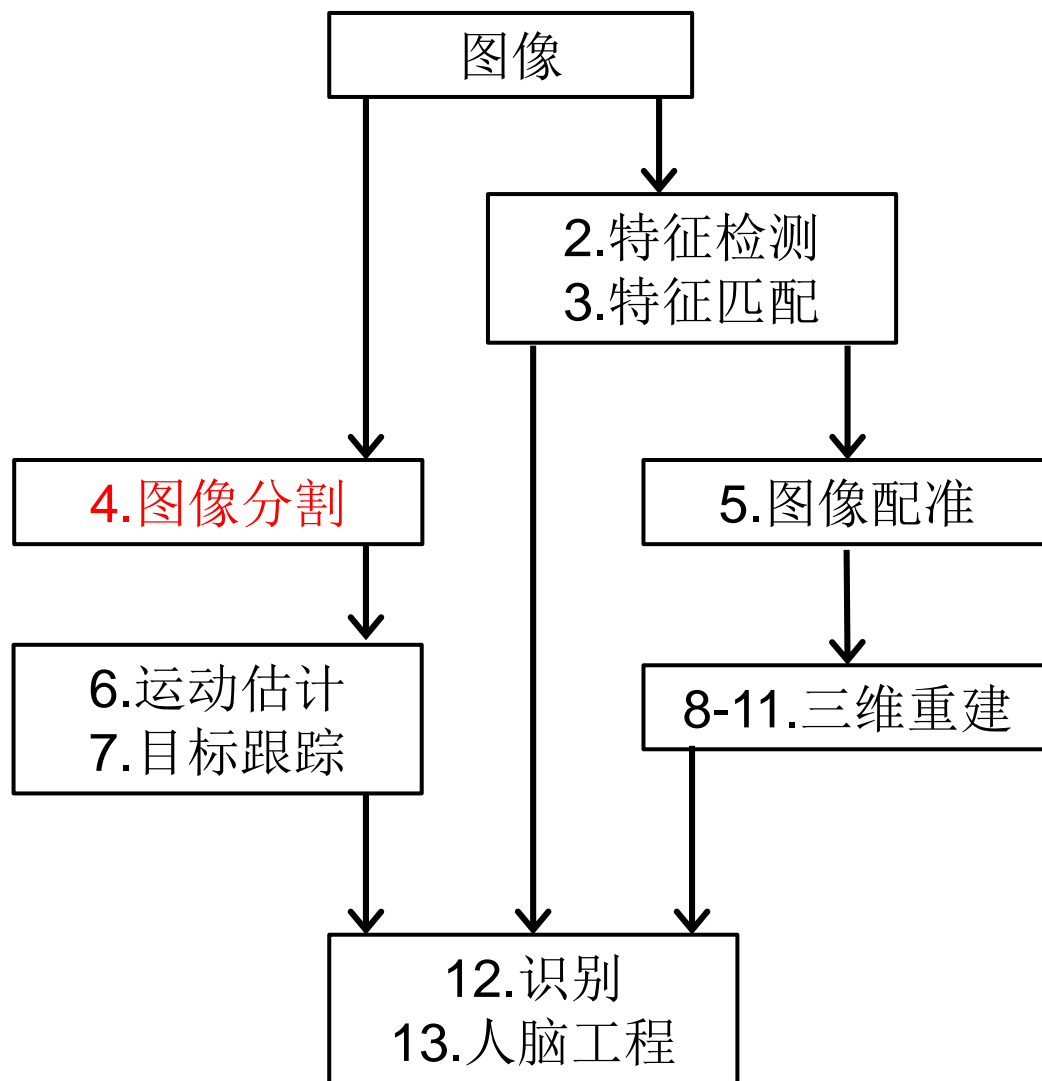


Robot Vision Group

National Laboratory of Pattern Recognition

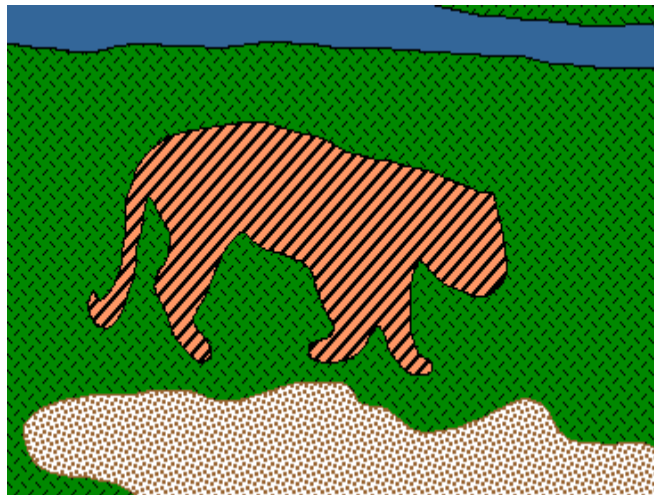
Institute of Automation, Chinese Academy of Sciences

计算机视觉课程结构图



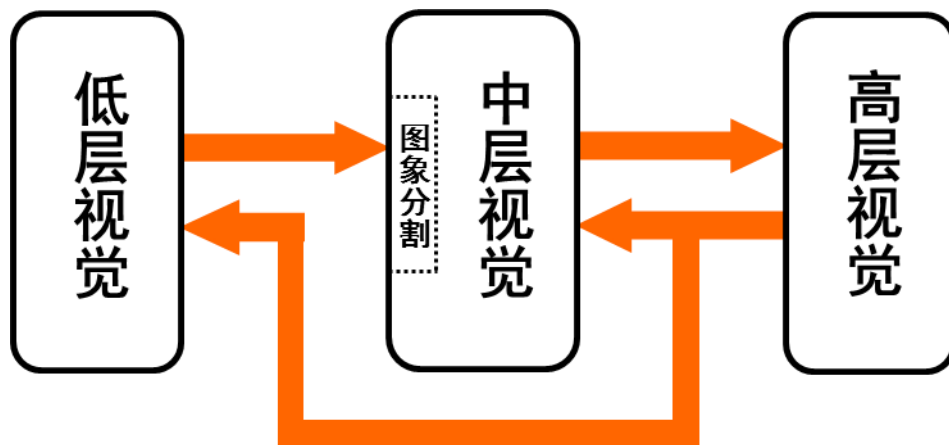
什么是图像分割

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

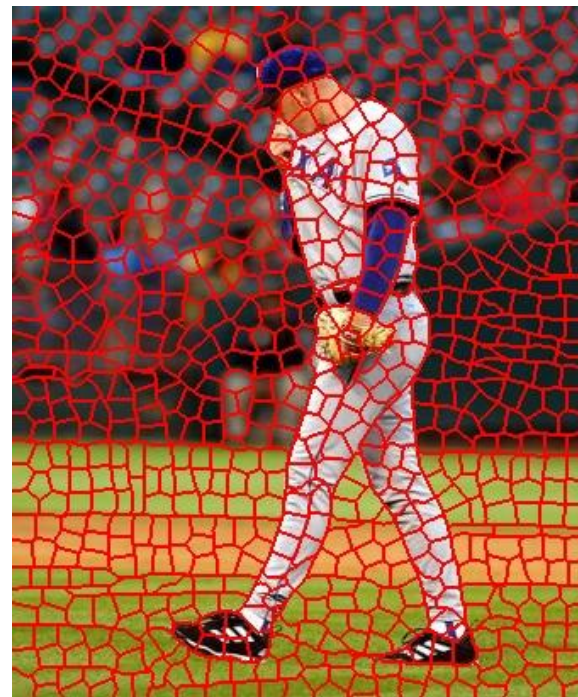
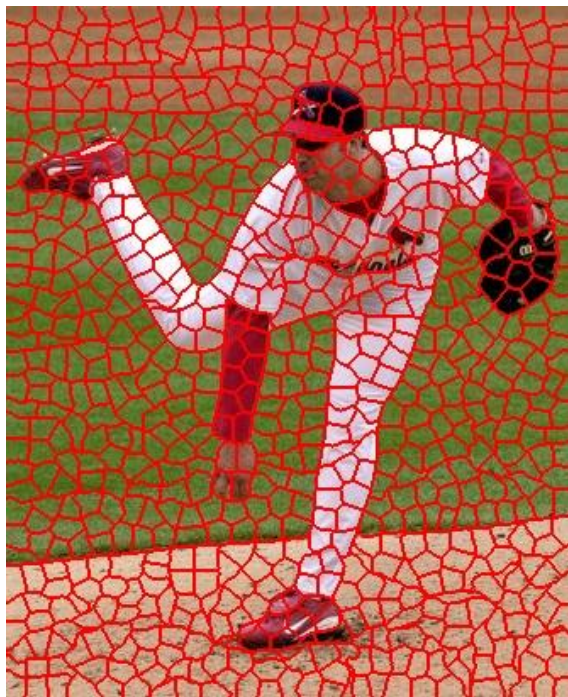
图像分割是由图像处理进到图像分析的关键步骤。它是目标表达的基础，使得更高层的图像分析和理解成为可能。



图像分割的难点

图像分割是将图像中相似的像素组合在一起的过程，这是一个自底向上的过程，依赖于图像本身。

“superpixels”



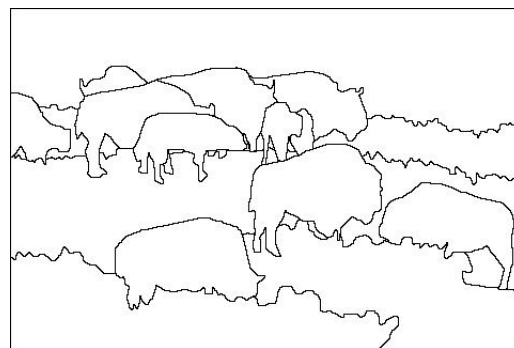
Ren and Malik, 2003

图像分割的难点

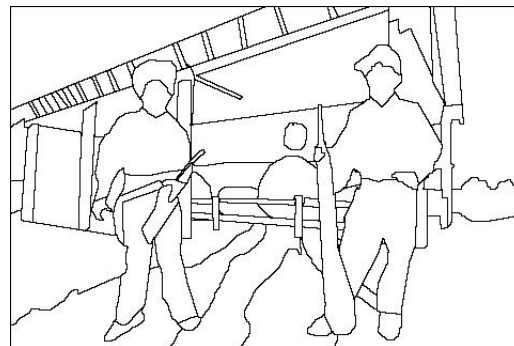
图像分割是将图像划分为不同语义区域的过程，这是一个自顶向下的过程，依赖于高层知识。



image



human segmentation



Berkeley segmentation database

图像分割的难点

格式塔学派（Gestalt）：人脑的运作原理是整体的，整体不同于其部件的总和。



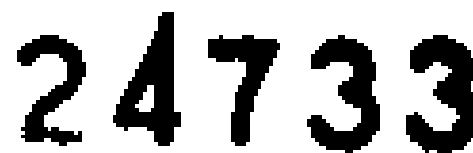
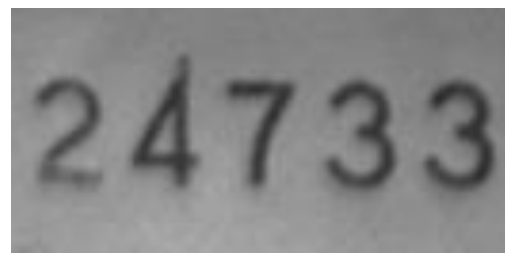
图像分割的难点

- 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像特征提取问题，它还涉及到各种图像特征的知觉组织。

阈值法

阈值法基本原理：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。

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



- 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。

阈值法— p 分位数法

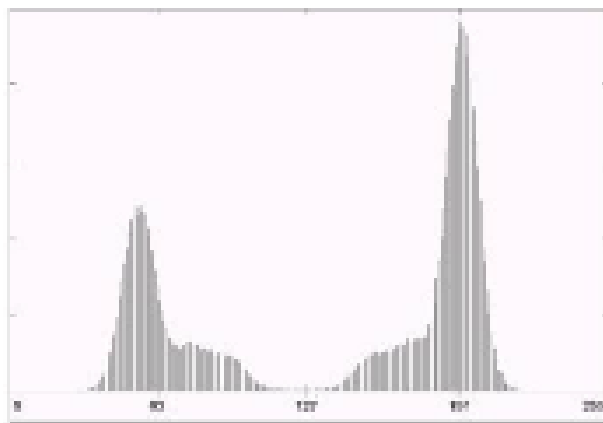
1962年Doyle提出 p -分位数法：使目标与整个图像的像素比例等于其先验概率 $p\%$ 来设定阈值。



但对于据大多数图像，我们无法预先估计图像中目标所占的像素比例。

阈值法—双峰法

1966年Prewitt等提出双峰法：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。



对于直方图中波峰不明显或者波谷平坦宽阔的图像无法使用。

阈值法—迭代法

基本流程：

1. $K=0$ ，初始阈值选取为图像的平均灰度 T_K ；
2. 用 T_K 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_K 的部分为 T_A ，大于 T_K 的部分为 T_B ；
3. 计算 $T_{K+1}=(T_A + T_B) / 2$ ，将 T_{K+1} 作为新的全局阈值代替 T_K ；
4. $K=K+1$ ；
5. 重复过程2-5，如此迭代，直至 T_K 收敛，即 $T_{K+1} = T_K$ 。

阈值法—Otsu法

最大类间方差法是由日本学者大津于1979年提出的，是一种自适应的阈值确定的方法，又叫大津法，简称Otsu法。

Otsu法按图像的灰度特性将图像分成背景和目标两部分，背景和目标之间的类间方差越大，说明构成图像的两部分的差别越大。当部分目标错分为背景或部分背景错分为目标都会导致两部分差别变小。

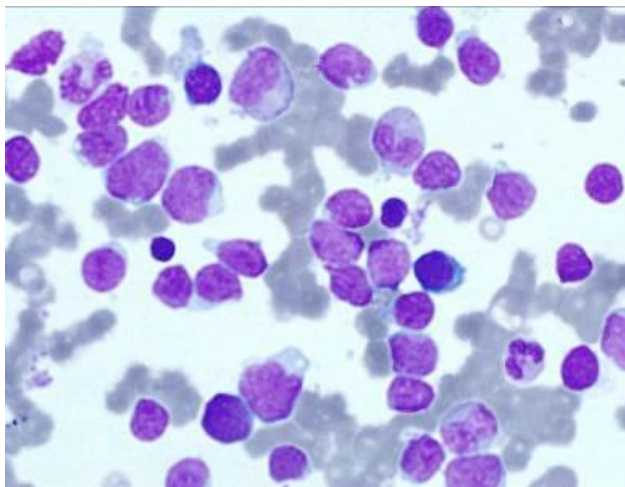
因此，使类间方差最大的分割意味着错分概率最小。

阈值法—Otsu法

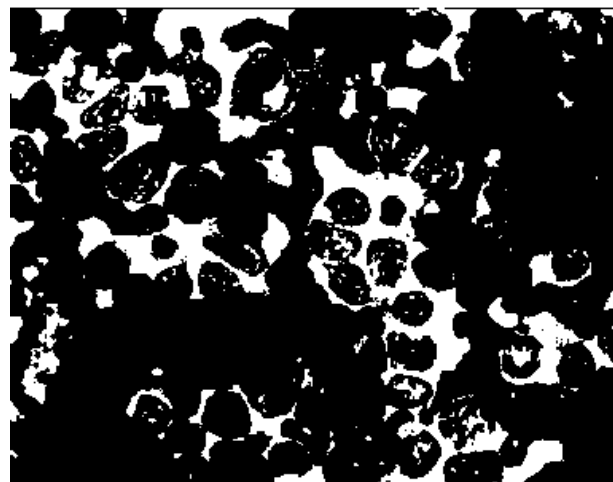
Otsu法基本流程:

- 记 t 为前景与背景的分割阈值，前景点数占图像比例为 w_0 ，平均灰度为 u_0 ；背景点数占图像比例为 w_1 ，平均灰度为 u_1 。则图像的总平均灰度为： $u=w_0 \times u_0 + w_1 \times u_1$ 。
- 从最小灰度值到最大灰度值遍历 t ，当 t 使得值 $g=w_0 \times (u_0 - u)^2 + w_1 \times (u_1 - u)^2$ 最大时 t 即为分割的最佳阈值。
- 直接应用Otsu法计算量较大，因此实现时采用了等价的公式 $g=w_0 \times w_1 \times (u_0 - u_1)^2$ 。

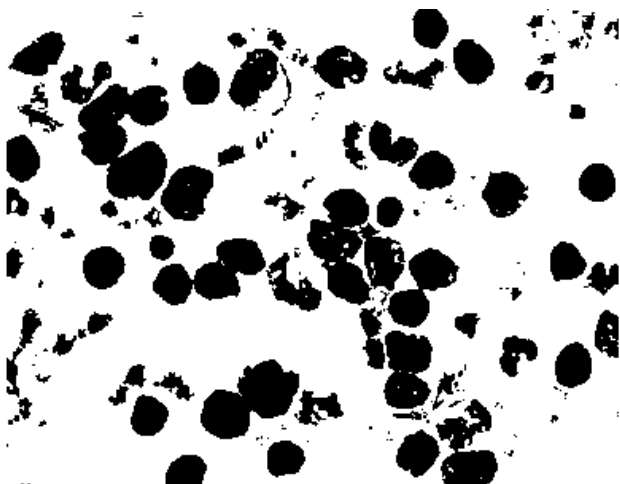
阈值法—比较



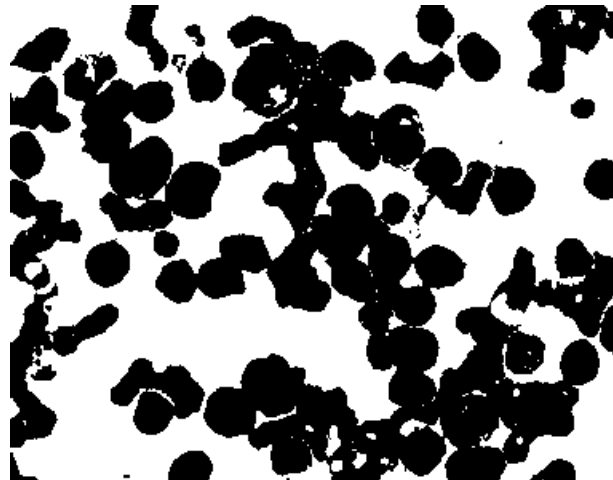
图像



阈值偏低



迭代法



Otsu法

区域分割—Watershed

阈值法没有考虑图像中的区域信息，分割中还有一类直接针对图像区域进行分割的方法，如分水岭算法（Watershed）。

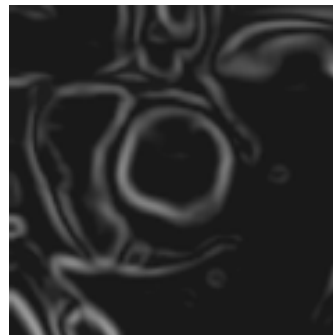
分水岭算法的基本思想为：

1. 把梯度图像看作是测地学上的拓扑地貌；
2. 在每一个地形极小点处打一个孔；
3. 把整个模型慢慢浸入水中，随着浸入的加深，每一个局部极小值的影响域慢慢向外扩展；
4. 在两个集水盆汇合处构筑大坝，即形成分水岭。

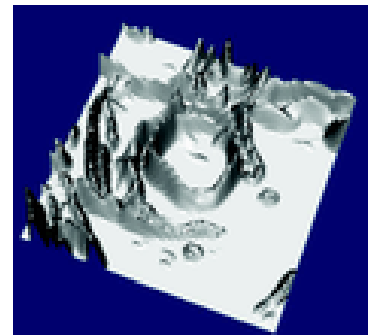
区域分割— Watershed

分水岭算法流程：

1. 选择梯度图中的一些局部极小点，每个赋予不同的标签；
2. 将每一个局部极小点的 h 阶高度影响域内的像素点根据灰度级次序放入一个先进先出(FIFO)队列；
3. 从队列中取出一个像素，如果它的邻域已标记像素均有同样标签，则将这个像素点也标记为这一标签。邻域内所有未标记且不在队列的像素点放入FIFO队列；
4. 重复步骤3直到队列为空；
5. 所有剩余未标记像素构成分水岭。



梯度图



梯度图(3D)



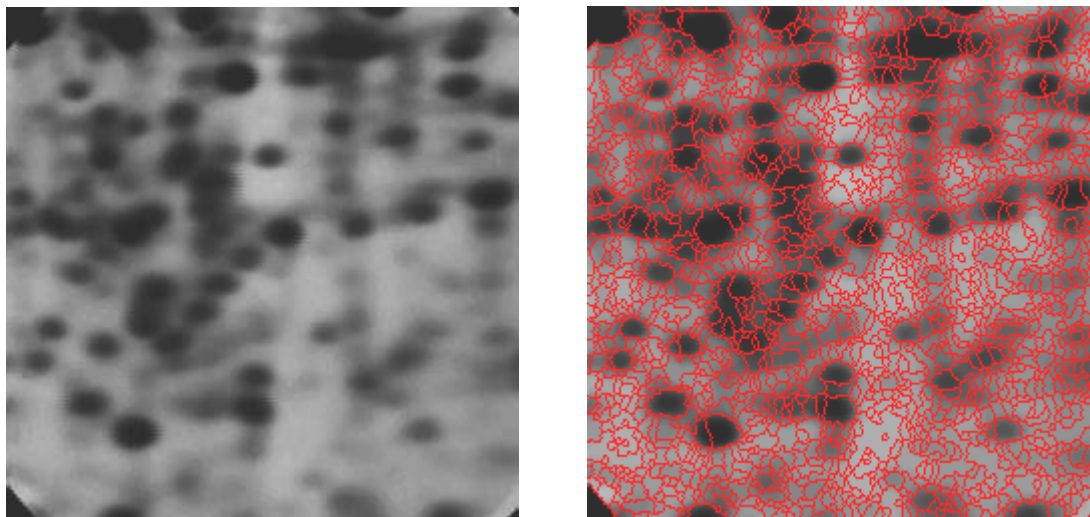
分水岭



分水岭(3D)

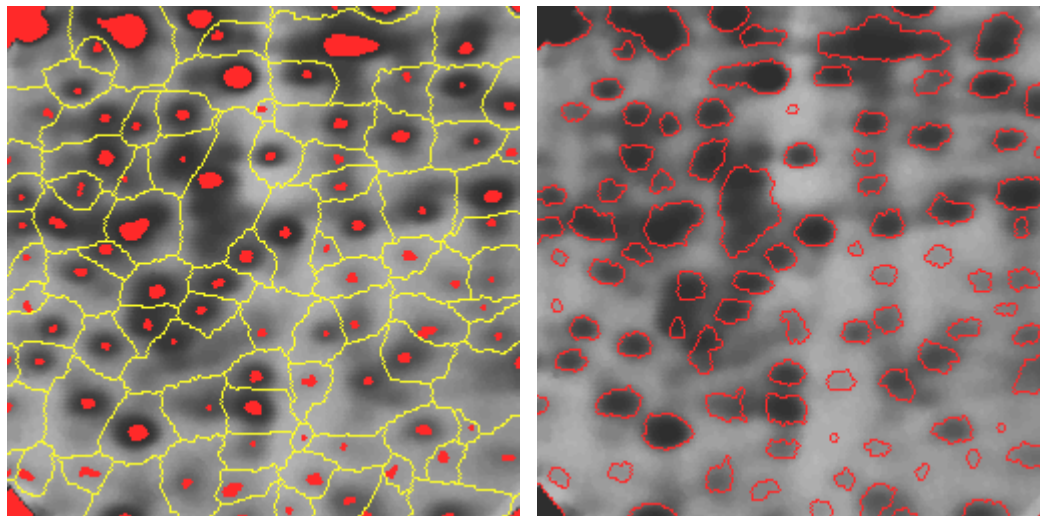
区域分割— Watershed

分水岭算法将每一个局部极值点当作一个独立的区域，因此容易产生过分割现象。



因此分水岭算法通常是作为一种交互式算法使用，由用户自行选择种子点位置。

区域分割— Watershed

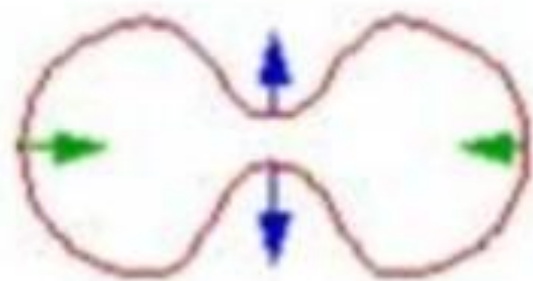
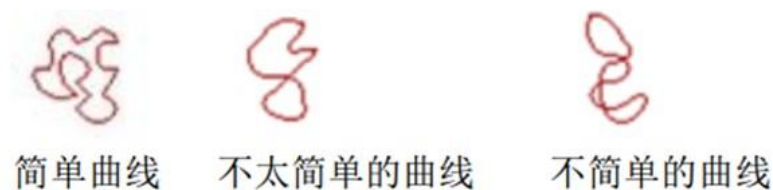


- 1) 滤波;
- 2) 选择平缓的“盆地”底部作为标记 (红斑);
- 3) 调用分水岭算法确定分水线 (黄线);
- 4) 在每个分水线分出的小区域内完成分割。

活动轮廓

活动轮廓模型（Active Contour Model）的基本思想是使用连续曲线来表达目标边缘，并定义一个能量泛函使得其自变量包括边缘曲线，因此分割过程就转变为求解能量泛函的最小值的过程，一般可通过求解函数对应的欧拉方程来实现，能量达到最小时的曲线位置就是目标的轮廓位置。

简单曲线在曲率力（也就是曲线的二次导数）的驱动下演化所具有的性质：一切简单曲线，无论被扭曲得多么严重，只要还是一种简单曲线，那么在曲率力的推动下最终将退化成一个小圆，然后消逝。



曲率力

活动轮廓

活动轮廓模型（Active Contour Model）的基本思想是使用连续曲线来表达目标边缘，并定义一个能量泛函使得其自变量包括边缘曲线，因此分割过程就转变为求解能量泛函的最小值的过程，一般可通过求解函数对应的欧拉方程来实现，能量达到最小时的曲线位置就是目标的轮廓位置。

简单曲线在曲率力（也就是曲线的二次导数）的驱动下演化所具有的性质：一切简单曲线，无论被扭曲得多么严重，只要还是一种简单曲线，那么在曲率力的推动下最终将退化成一个小圆，然后消逝。



曲率力

活动轮廓—Snakes

Snakes是Active Contour Model的一种，它以构成一定形状的一些控制点为模板（轮廓线），通过模板自身的弹性形变，与图像局部特征相匹配达到调和，即某种能量函数极小化，完成对图像的分割。

$$f(s) = (x(s), y(s))$$

$$E_{total} = \int (\alpha \|f'(s)\|^2 + \beta \|f''(s)\|^2 + P(f(s))) ds$$



弹性能量

弯曲能量

外部能量

活动轮廓—Snakes

$$f(s) = (x(s), y(s))$$

$$E_{total} = \int (\alpha \|f'(s)\|^2 + \beta \|f''(s)\|^2 + P(f(s))) ds$$

最小化能量函数 E_{total} 是一个典型的变分问题（自变量是函数 $f(i)$ ，因变量是函数的函数，即数学上所谓的泛函。对泛函求极值的问题，数学上称之为变分法）。这一问题可以通过欧拉方程迭代求解（局部最小值满足下面的欧拉方程）：

$$\underbrace{-(\alpha f')' + (\beta f'')''}_{\text{曲线内力}} = \underbrace{-\nabla P}_{\text{图像外力}}$$

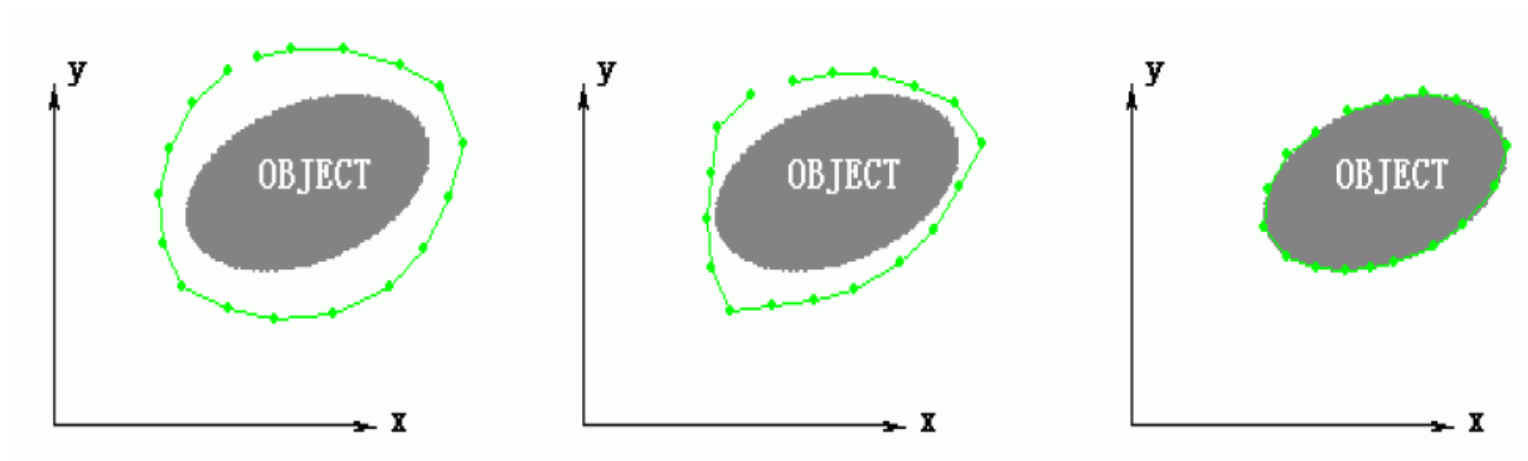
曲线**内力**
使曲线连续平滑

图像**外力**
使曲线趋近边缘

$$P = -\|\nabla I(f(s))\|^2$$

活动轮廓—Snakes

实际应用中，一般先在物体周围手动点出控制点，作为Snakes模型的起始位置，然后对能量函数迭代求解。



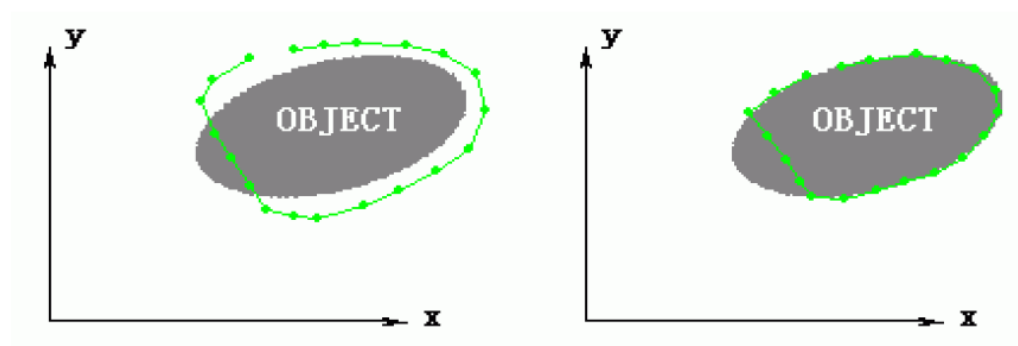
初始位置

第8次迭代

第32次迭代

活动轮廓—Snakes改进

在没有图像力平衡的条件下，内部力将把所有控制点收缩为一点或一条直线。也就是说，被分割物体必须完全包含在 Snakes 的初始位置之内，否则陷在内部的控制点将无法回到物体的边界。



解决方案：添加向外膨胀的力（ballooning force）

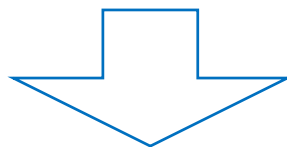
活动轮廓—Snakes改进

$$\underbrace{-(\alpha f')' + (\beta f'')''}_{\text{曲线内力}} = \underbrace{-\nabla P}_{\text{图像外力}}$$

曲线内力
使曲线连续平滑

图像外力
使曲线趋近边缘

$$P = -\|\nabla I(f(s))\|^2$$



$$\underbrace{-(\alpha f')' + (\beta f'')''}_{\text{曲线内力}} = \underbrace{k_1 \vec{n}(s) - k \frac{\nabla P}{\|\nabla P\|}}_{\text{图像外力+气球外力}}$$

曲线内力
使曲线连续平滑

图像外力+气球外力
使曲线趋近边缘，且向外膨胀

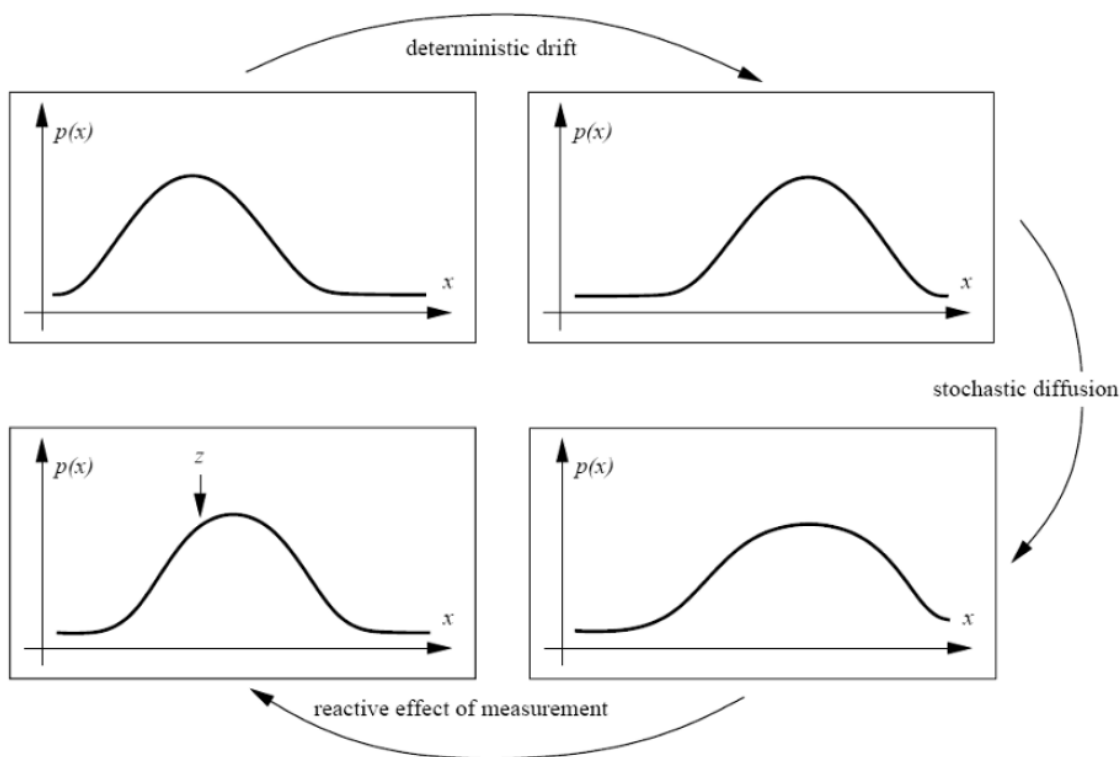
$\vec{n}(s)$ 是控制点 $f(s)$ 处单位法向量

活动轮廓—Condensation

在许多应用中，物体在图像序列中形态变化，此时可以用上一帧Snake分割结果来预测下一帧中物体轮廓的位置。

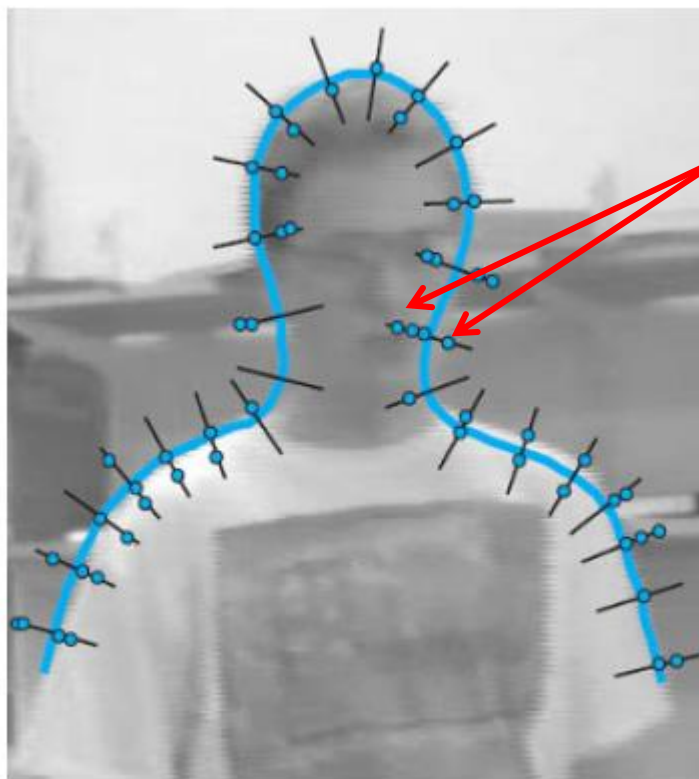
一种解决方法是使用卡尔曼滤波，称为Kalman Sankes。

$$x_t = Ax_{t-1} + \omega_t$$



活动轮廓—Condensation

在大多数情况下，由于背景干扰等因素的影响，物体轮廓并不满足高斯分布。



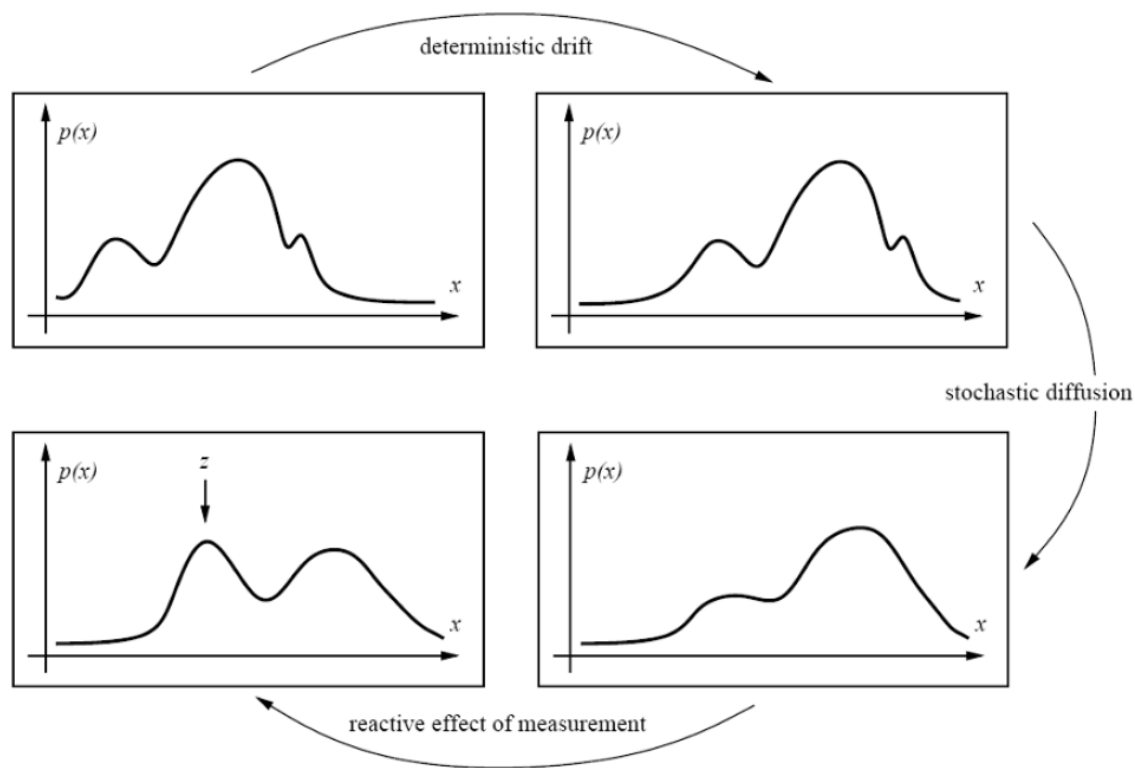
多峰分布

活动轮廓—Condensation

在大多数情况下，由于背景干扰等因素的影响，物体轮廓并不满足高斯分布。

这时的解决方法是使用粒子滤波，Particle Filtering。

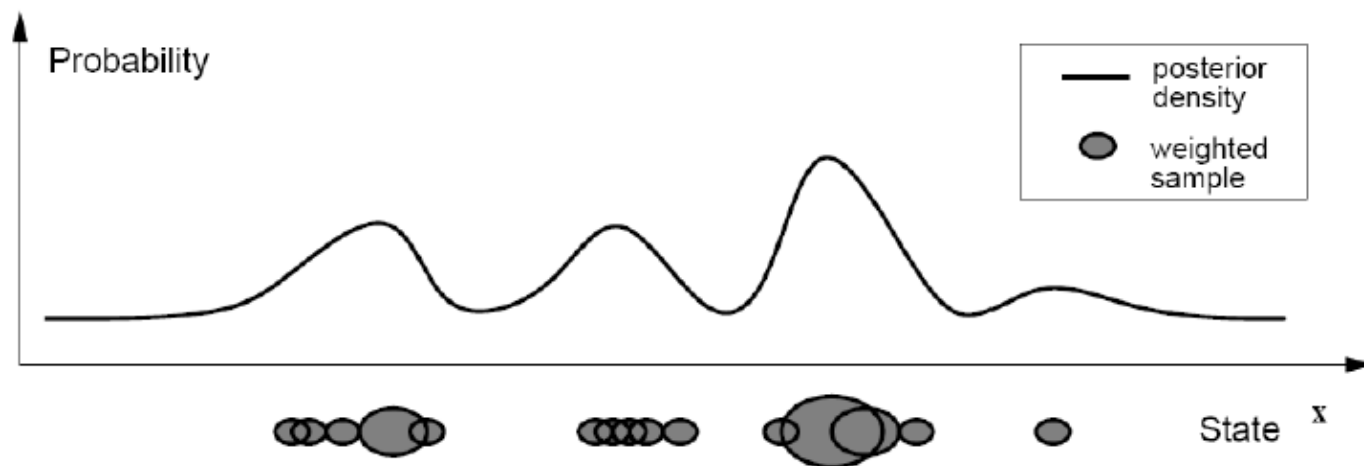
粒子滤波是一种对非高斯非线性状态进行跟踪的有效方法，粒子滤波的思想基于蒙特卡洛方法，它是利用粒子集来表示概率，可以用在任何形式的状态空间模型上。



活动轮廓—Condensation

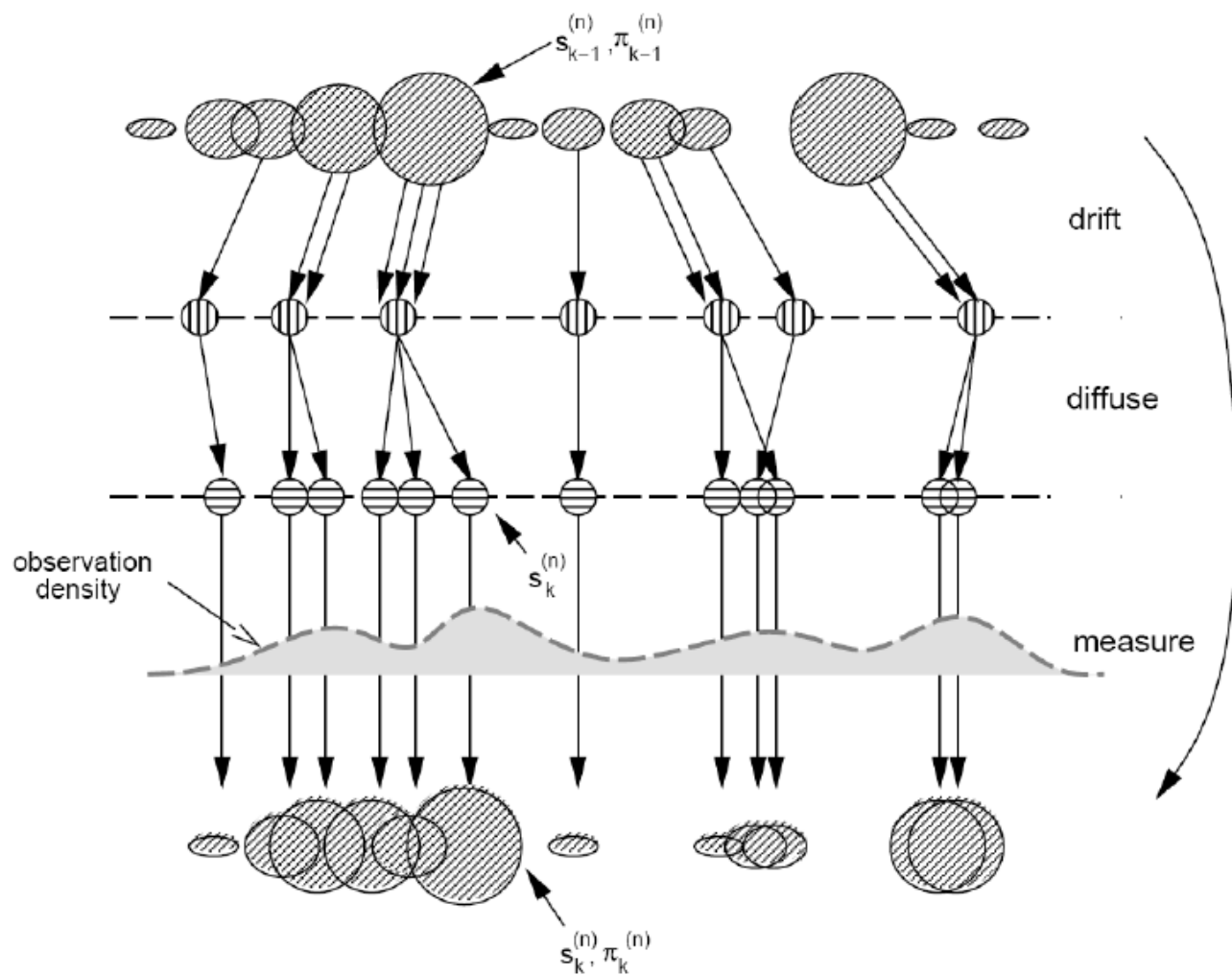
粒子滤波方法最早由Isard和Blake于1998引入计算机视觉，并命名为CONDENSATION (CONditional DENSity propagATION)

粒子滤波的基本思想：使用一组带有权重的采样点近似状态的后验概率。



Isard and Blake 1998

活动轮廓—Condensation



采样



转移



扰动

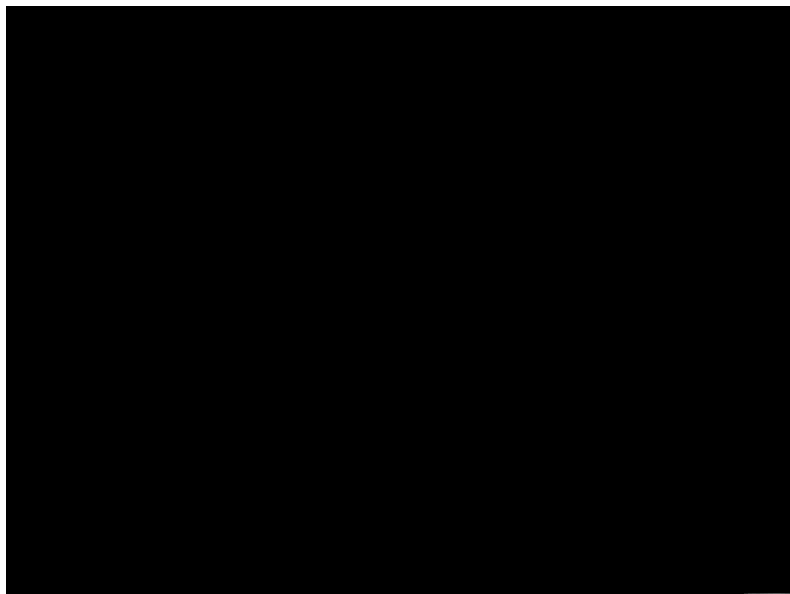


观测



重采样

活动轮廓—Condensation

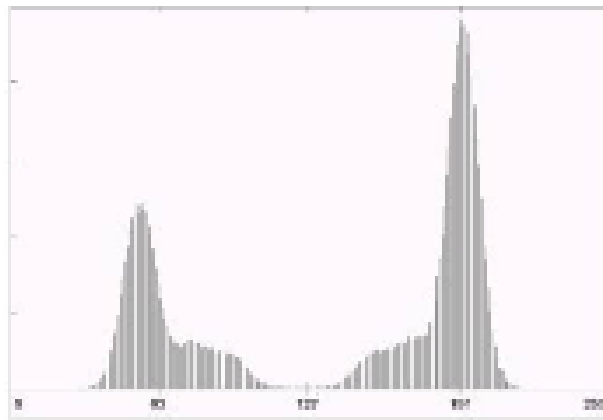


聚类分割

最直接的分割方法是直接根据像素特征进行聚类（Clustering），将相似的像素合为一类。

两种典型的聚类方法：

- K均值（**K-means**）聚类：像素特征符合混合高斯分布，通过估计各个高斯分布的均值和方差进行聚类；
- 均值漂移（**Mean Shift**）聚类：直接对像素特征分布进行建模，并寻找分布模型中的局部极值点进行聚类；



聚类分割

最直接的分割方法是直接根据像素特征进行聚类（Clustering），将相似的像素合为一类。

两种典型的聚类方法：

- K均值（**K-means**）聚类：像素特征符合混合高斯分布，通过估计各个高斯分布的均值和方差进行聚类；
- 均值漂移（**Mean Shift**）聚类：直接对像素特征分布进行建模，并寻找分布模型中的局部极值点进行聚类；
- K均值（**K-means**）聚类使用混合高斯模型，并计算每一个高斯分布的参数，是一种参数化（**parametric**）方法；
- 均值漂移（**Mean Shift**）聚类直接对特征分布进行建模，不考虑分布的参数，是一种非参数化（**non-parametric**）方法；

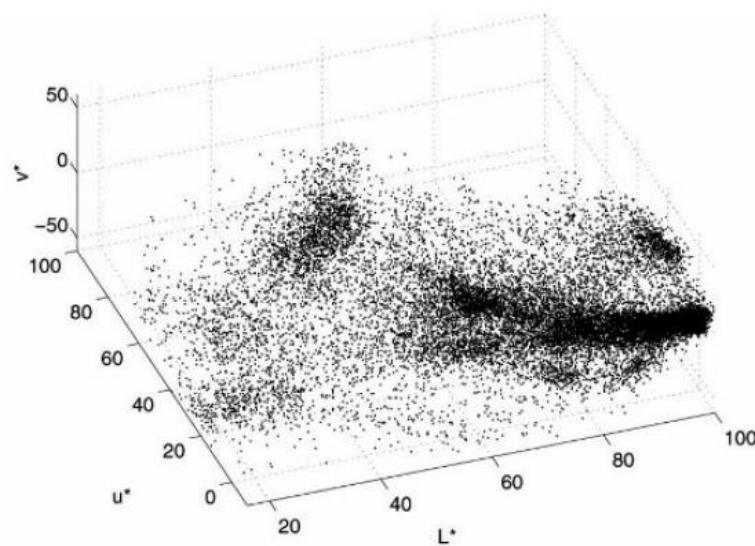
聚类分割—K-means

可以使用的像素特征：灰度、(R,G,B)、(R,G,B,X,Y)、(L^*u^*v)



(a)

图像



(b)

L^*u^*v 颜色空间分布

聚类分割—K-means

K-means基本流程：

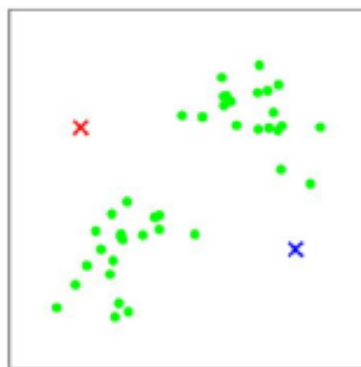
1. 给定聚类中心个数 k ;
2. 随机选取 k 个聚类中心点 $\mu_1, \mu_2, \dots, \mu_k$;
3. 对于每一个特征点 x_i ，根据到聚类中心的距离计算其所属类别 c_i ，其中 $c_i = \operatorname{argmin}_j \|x_i - \mu_j\|^2$;
4. 对每一类 j ，根据其包含的所有特征点均值更新 μ_j ;
5. 重复步骤3-5，直到 $\mu_1, \mu_2, \dots, \mu_k$ 不再改变。

聚类分割—K-means

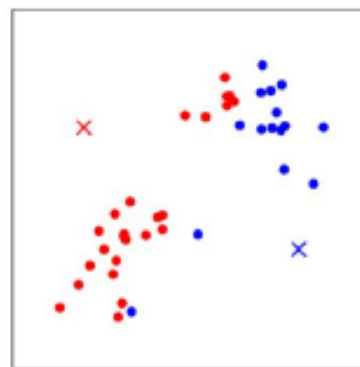
K-means聚类举例 ($k=2$) :



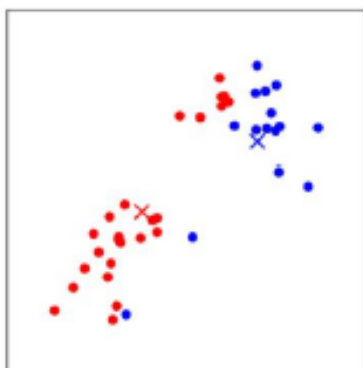
(a)



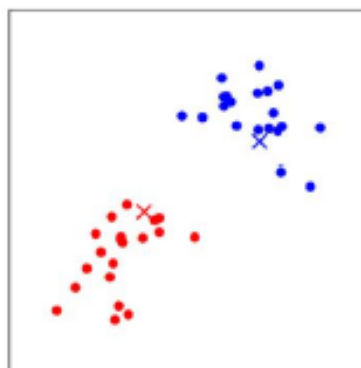
(b)



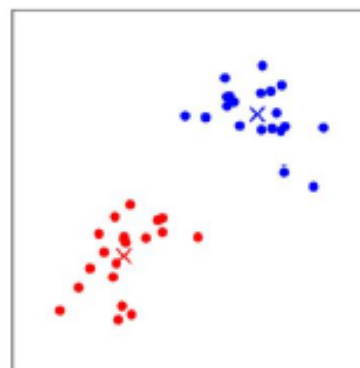
(c)



(d)



(e)



(f)

聚类分割—K-means

期望最大化（Expectation Maximization, EM）算法：

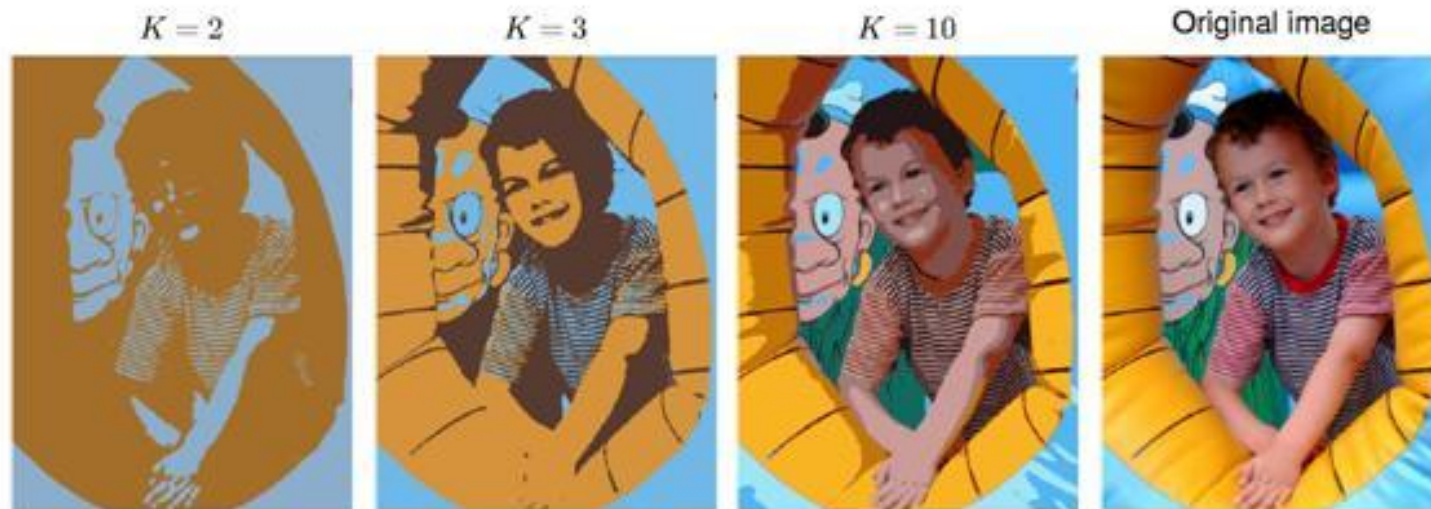
- E-step: 计算似然值 $z_{ik} = \frac{1}{Z_i} \pi_k \mathcal{N}(x_i | \mu_k, \Sigma_k)$, $\sum_k z_{ik} = 1$ 。 z_{ik} 表明特征 x_i 属于第 k 个高斯模型的可能性；
- M-step: 更新高斯分布参数, $N_k = \sum_i z_{ik}$ 表示每个类别的样本数；

$$\mu_k = \frac{1}{N_k} \sum_i z_{ik} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_i z_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

聚类分割—K-means



K-means图像分割举例

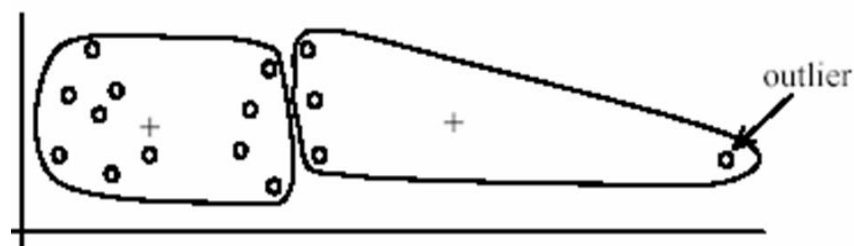
聚类分割—K-means

K-means的优势:

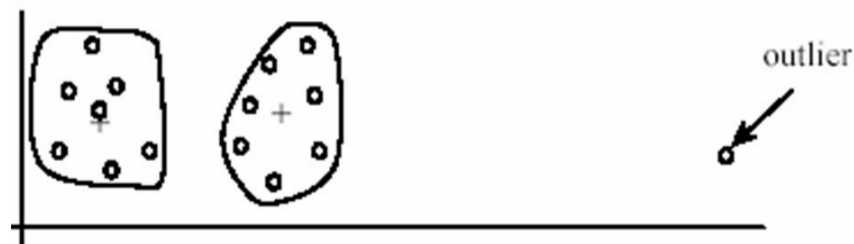
- 1) 算法简单;
- 2) 能收敛到局部极值。

K-means的不足:

- 1) Memory-intensive
- 2) 需要指定聚类中心数 k ;
- 3) 对初始值敏感;
- 4) 对外点敏感;
- 5) 只能进行“球形”聚类。



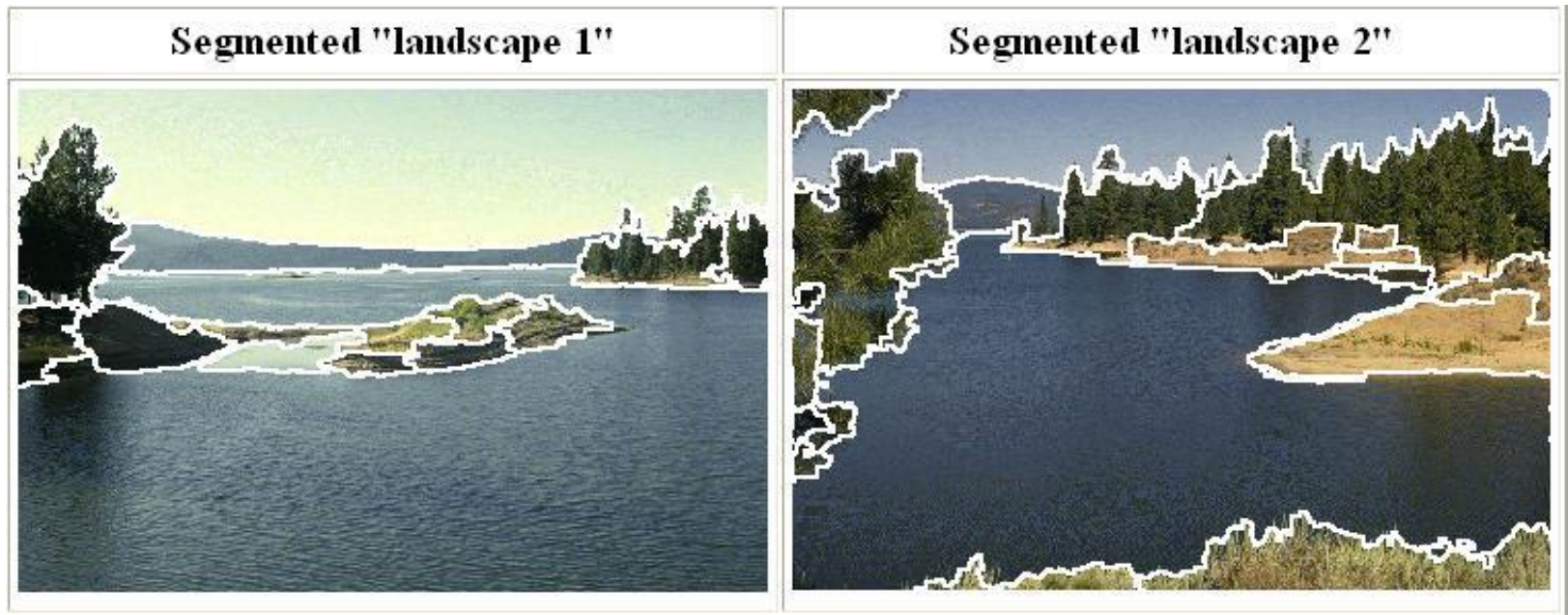
(A): Undesirable clusters



(B): Ideal clusters

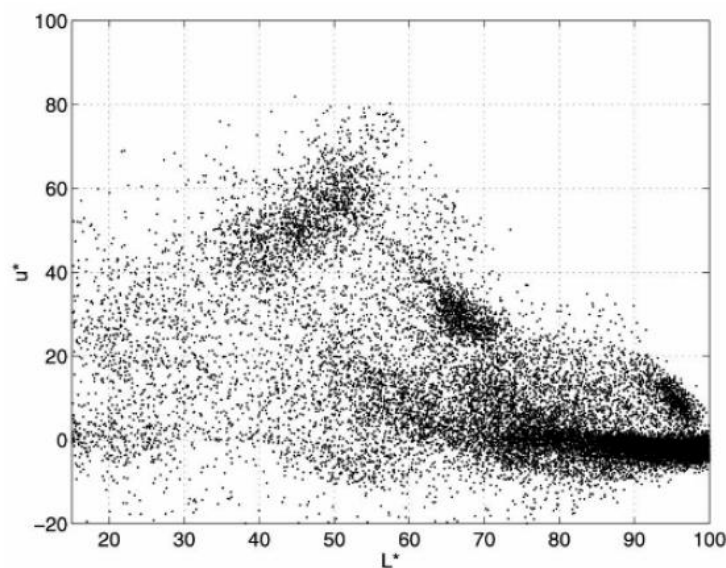
聚类分割—Mean Shift

Mean Shift是一种比K-means更加有效、适用范围更广的聚类分割算法。

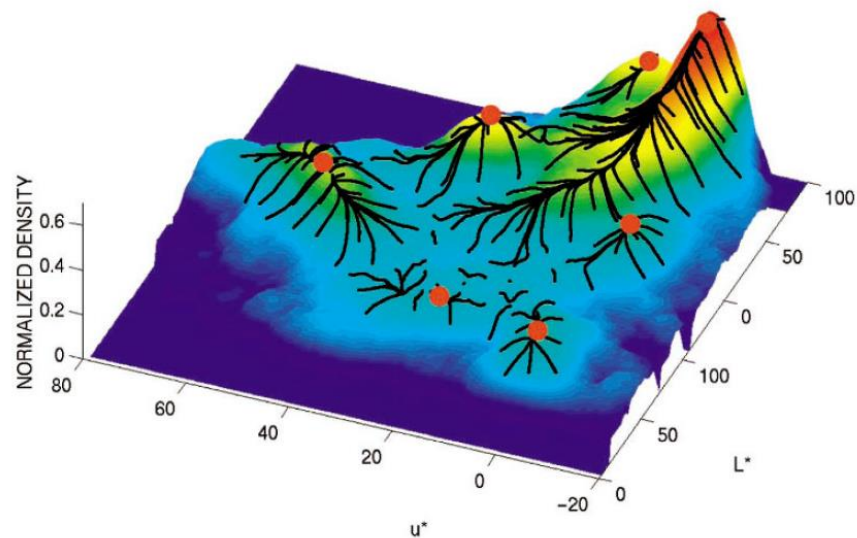


聚类分割—Mean Shift

Mean Shift的基本思想是：不考虑特征分布函数的参数，不对特征分布函数显式计算，而是直接寻找特征密度分布函数上的局部极值点。



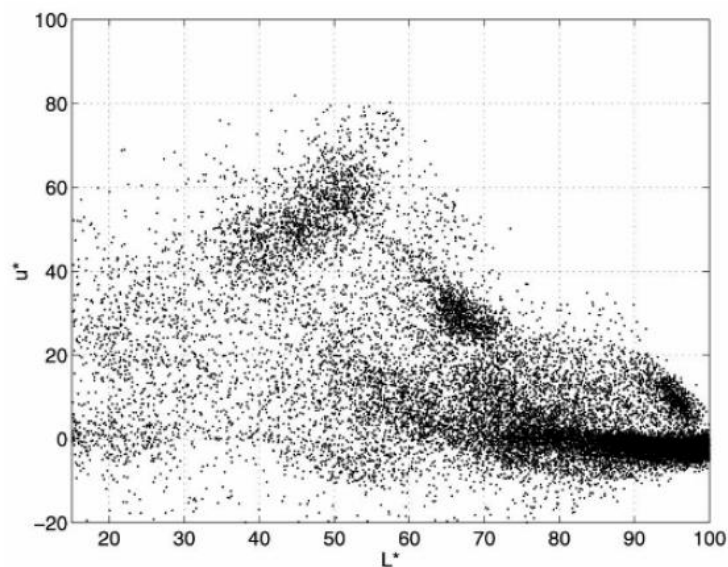
像素 L^*u^* 颜色空间分布



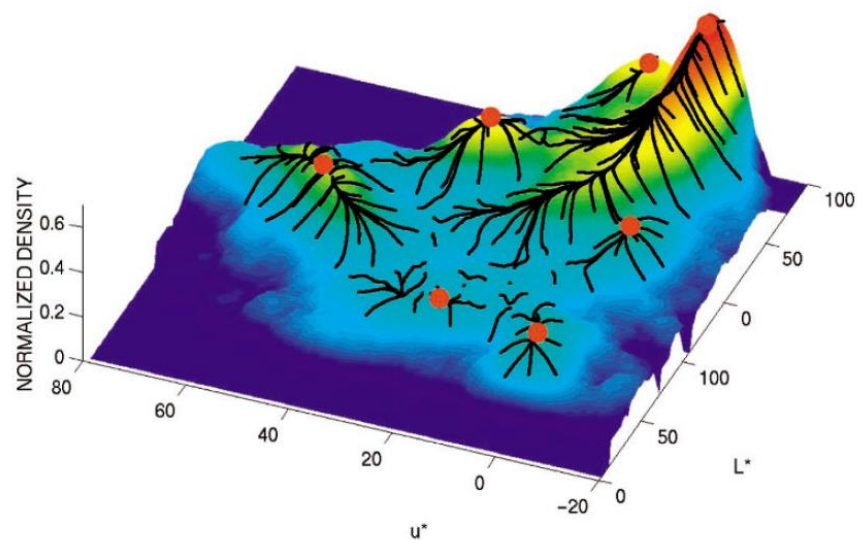
L^*u^* 颜色空间密度分布

聚类分割—Mean Shift

我们现在只有稀疏的特征点，如何获得特征空间上的密度分布函数？



像素 L^*u^* 颜色空间分布



L^*u^* 颜色空间密度分布

聚类分割—Mean Shift

我们现在只有稀疏的特征点，如何获得特征空间上的密度分布函数？

最简单的方法：根据已有的稀疏特征点数据平滑插值

$$f(x) = \sum_i K(x - x_i) = \sum_i k\left(\frac{\|x - x_i\|^2}{h^2}\right)$$

x_i 是特征点， $k()$ 是核函数（也称Parzen window）

$$K_E(r) = \max(0, 1 - r) \quad \text{Epanechnikov kernel}$$

$$K_N(r) = \exp\left(-\frac{1}{2}r\right) \quad \text{Gaussian kernel}$$

聚类分割—Mean Shift

直接在 $f(x)$ 上寻找局部极值通常是比较困难的，尤其当特征维数较高时。因此Mean shift随机采样和局部移动的方式寻找极值点。

$$f(x) = \sum_i k\left(\frac{\|x - x_i\|^2}{h^2}\right)$$

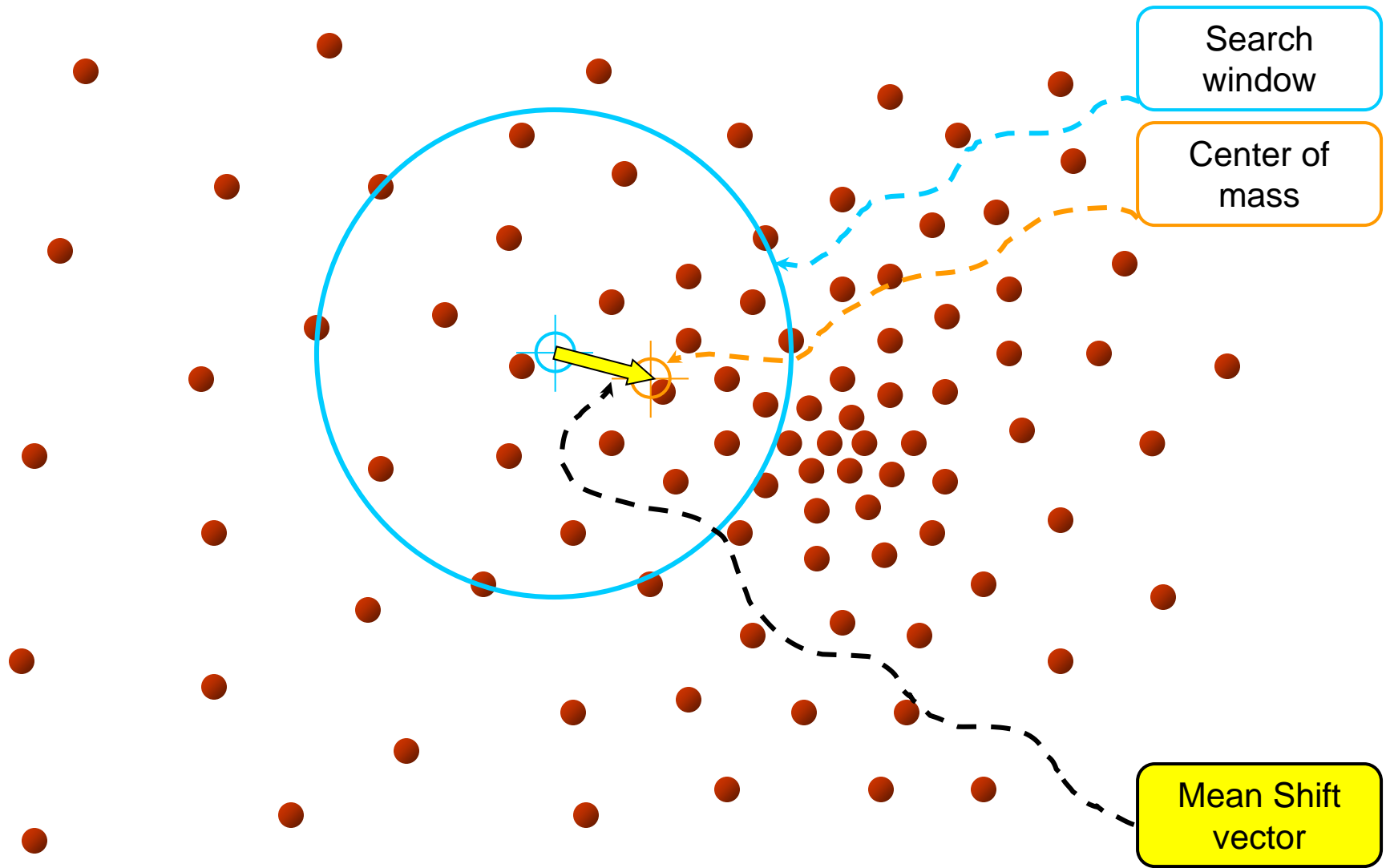
$$\nabla f(x) = \sum_i (x_i - x)G(x - x_i) = \sum_i (x_i - x)g\left(\frac{\|x - x_i\|^2}{h^2}\right), \quad g(r) = -k'(r)$$

$$= \left(\sum_i G(x - x_i) \right) m(x)$$

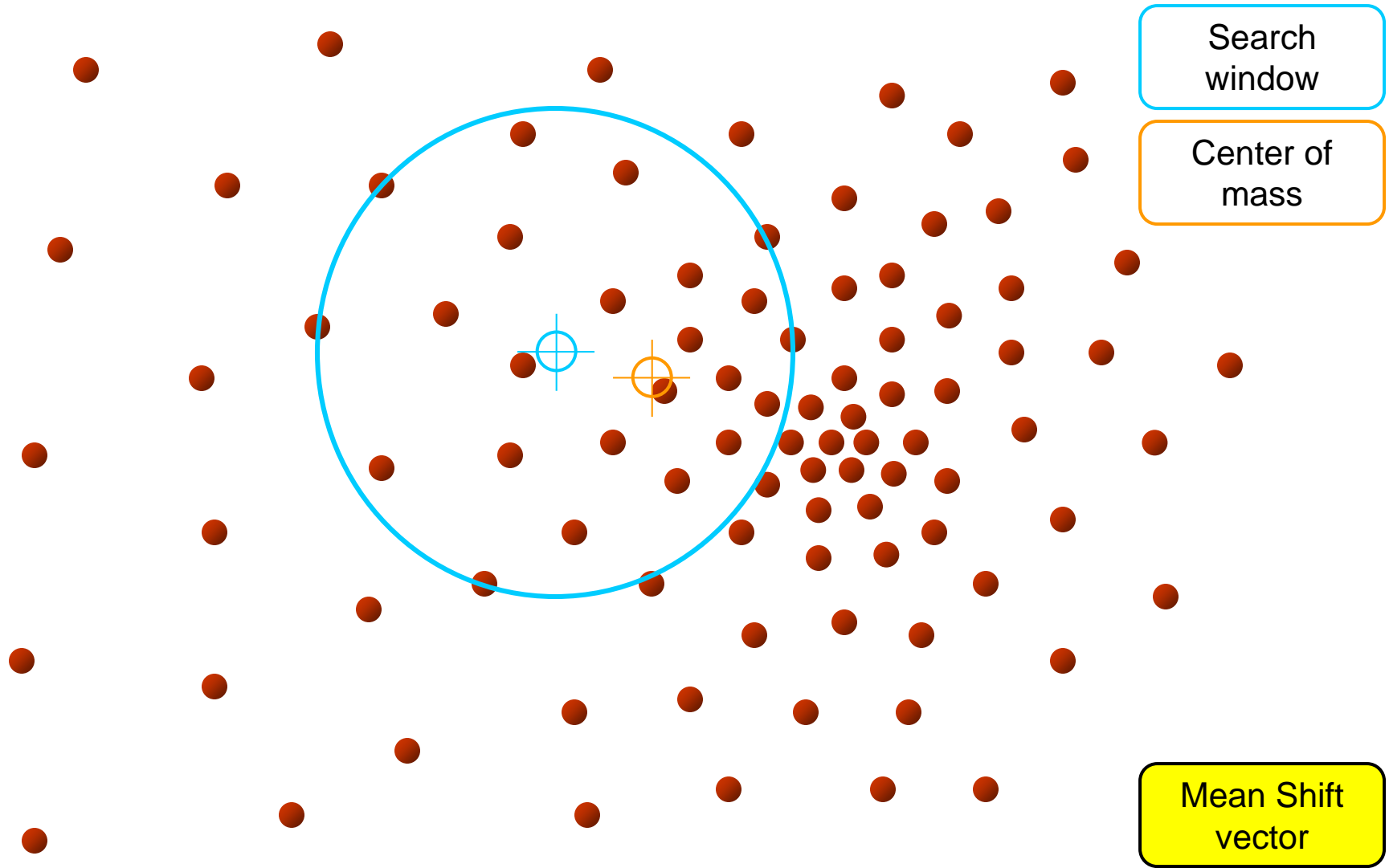
$$m(x) = \frac{\sum_i x_i G(x - x_i)}{\sum_i G(x - x_i)} - x$$

mean shift

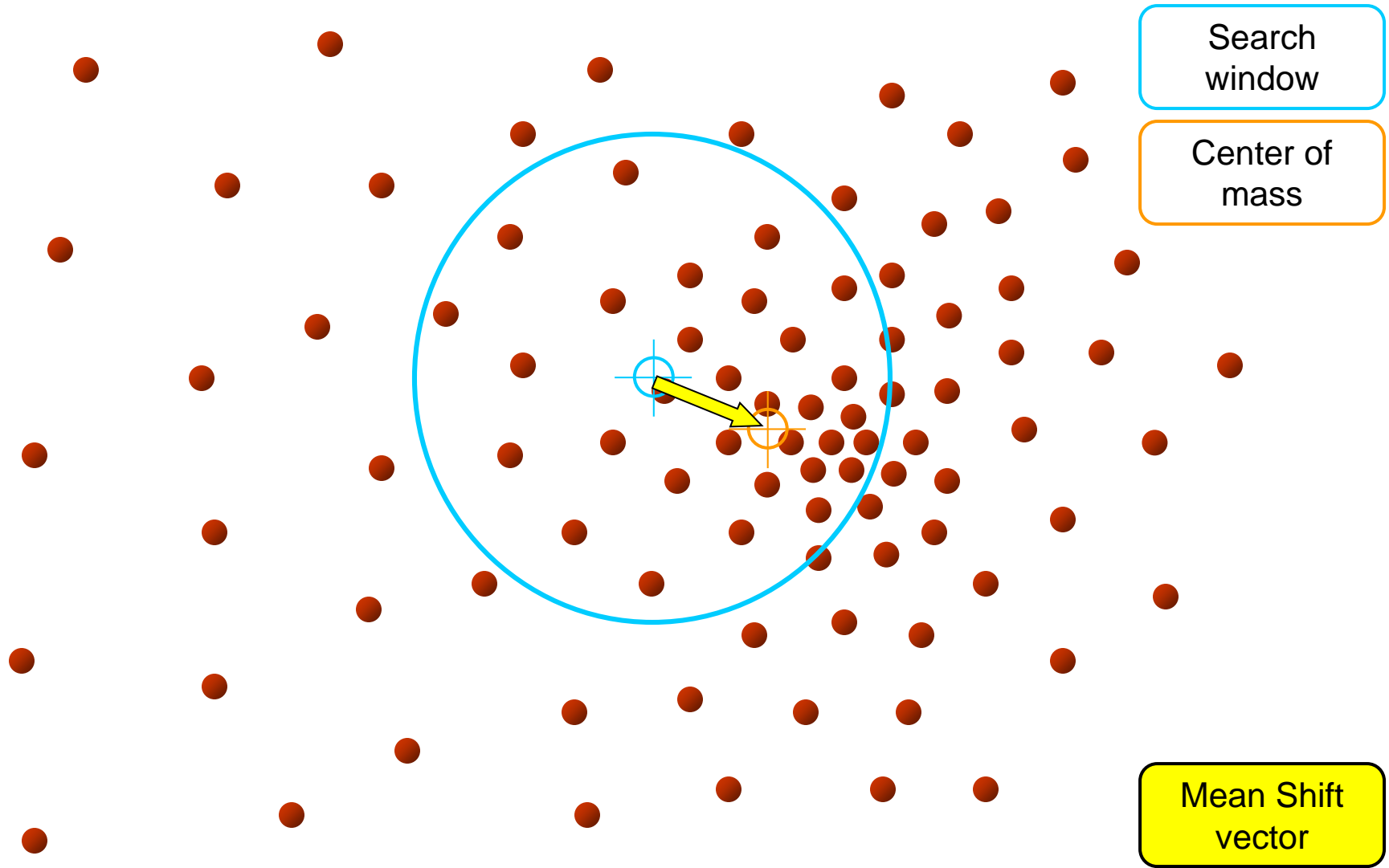
聚类分割—Mean Shift



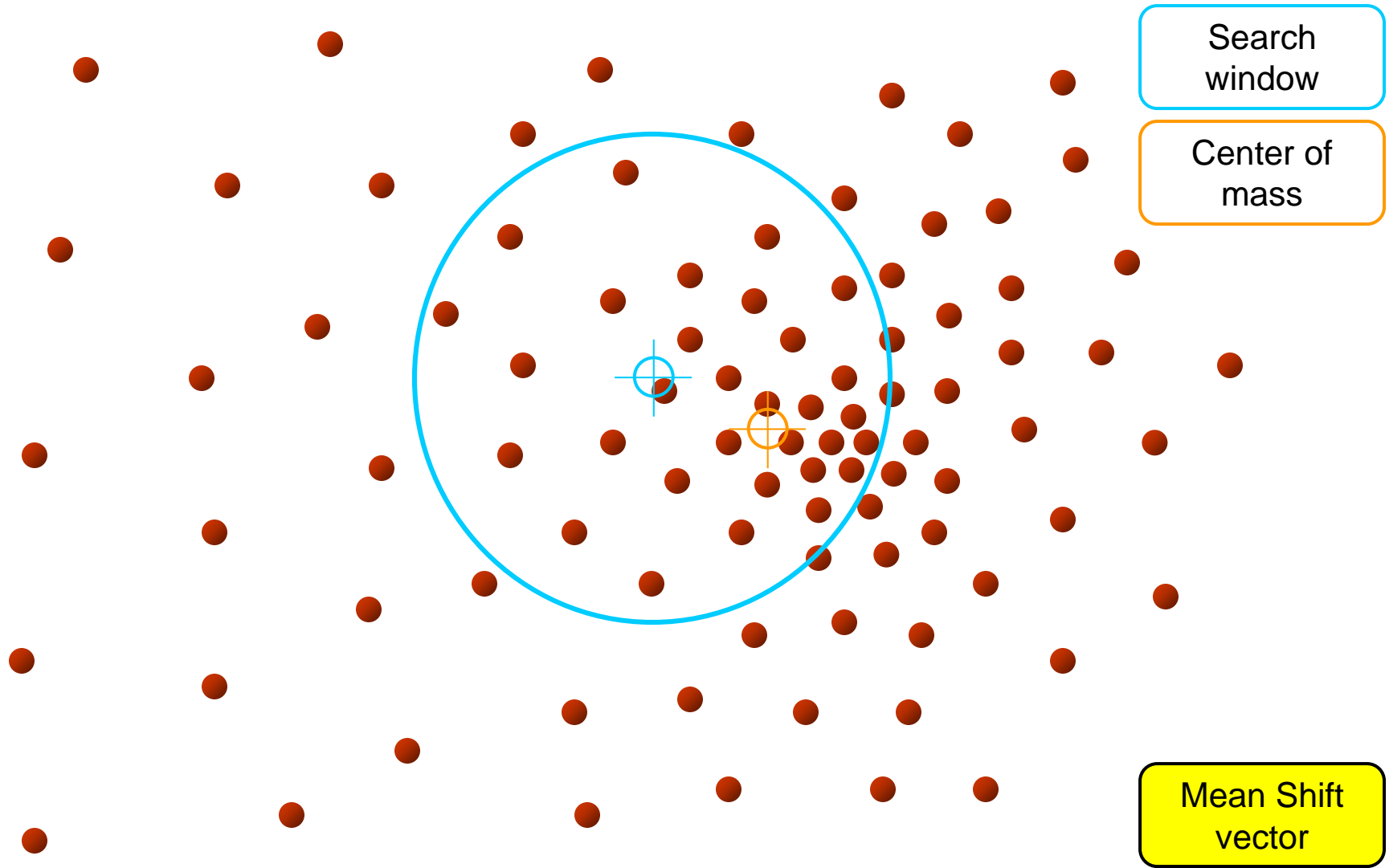
聚类分割—Mean Shift



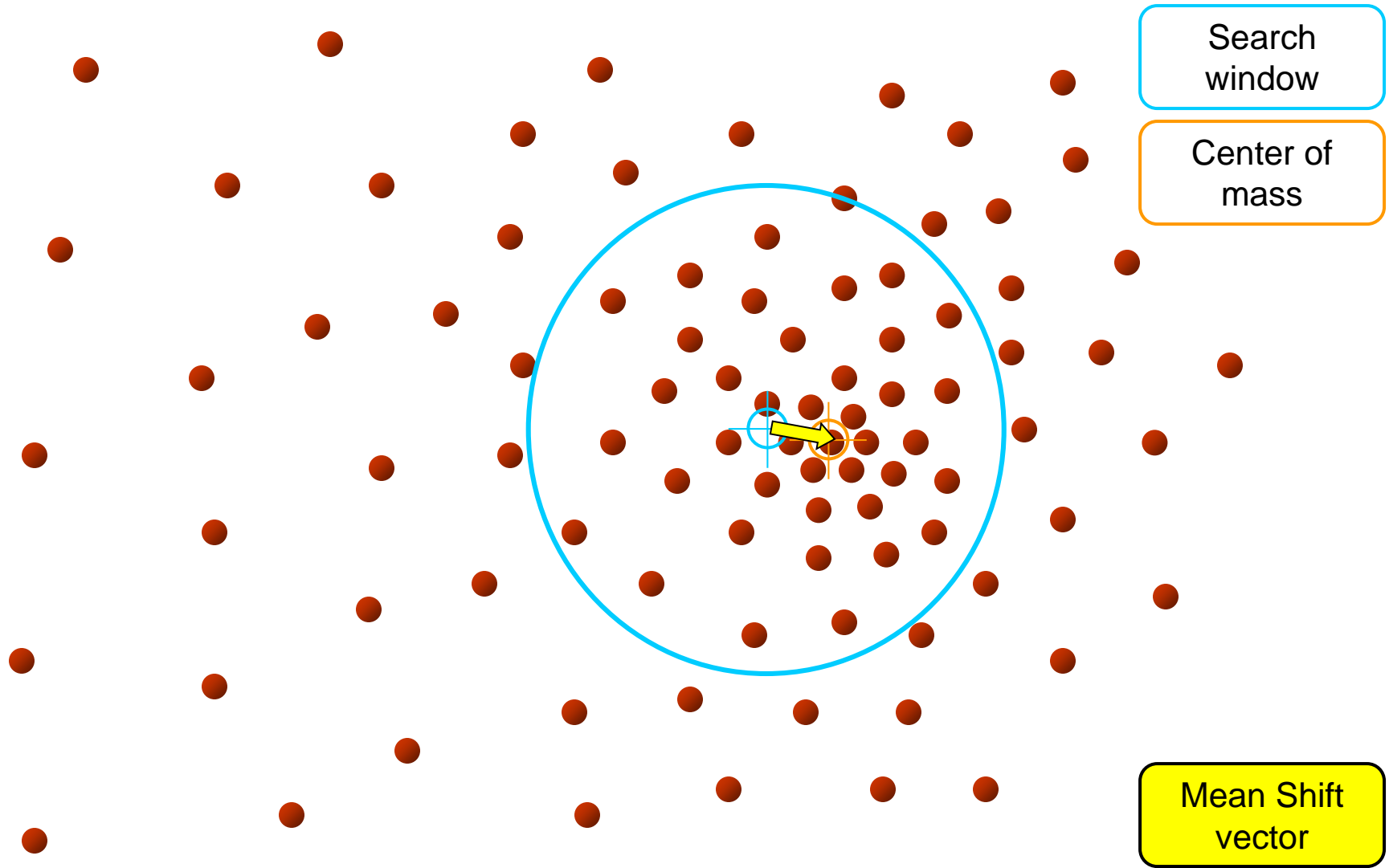
聚类分割—Mean Shift



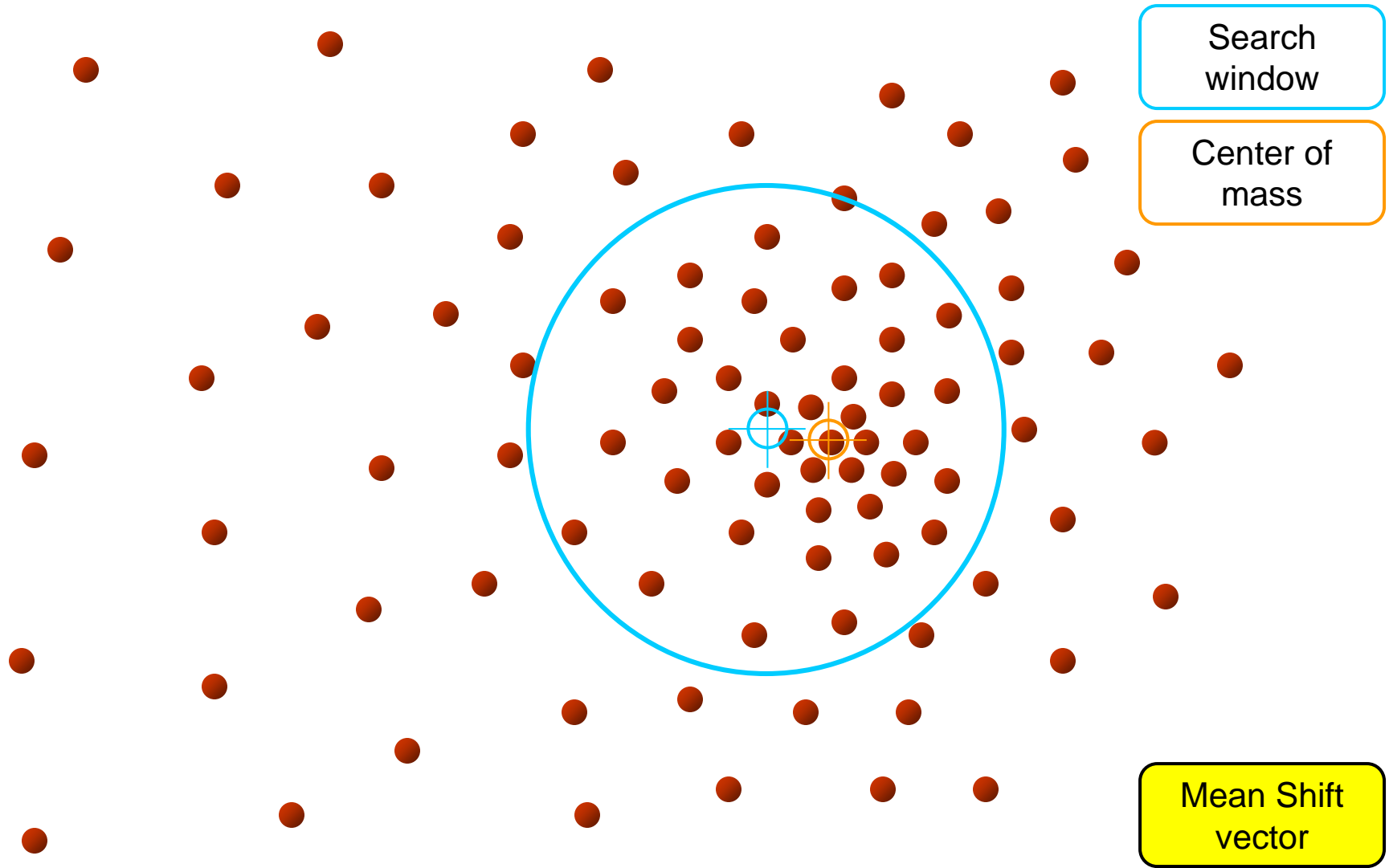
聚类分割—Mean Shift



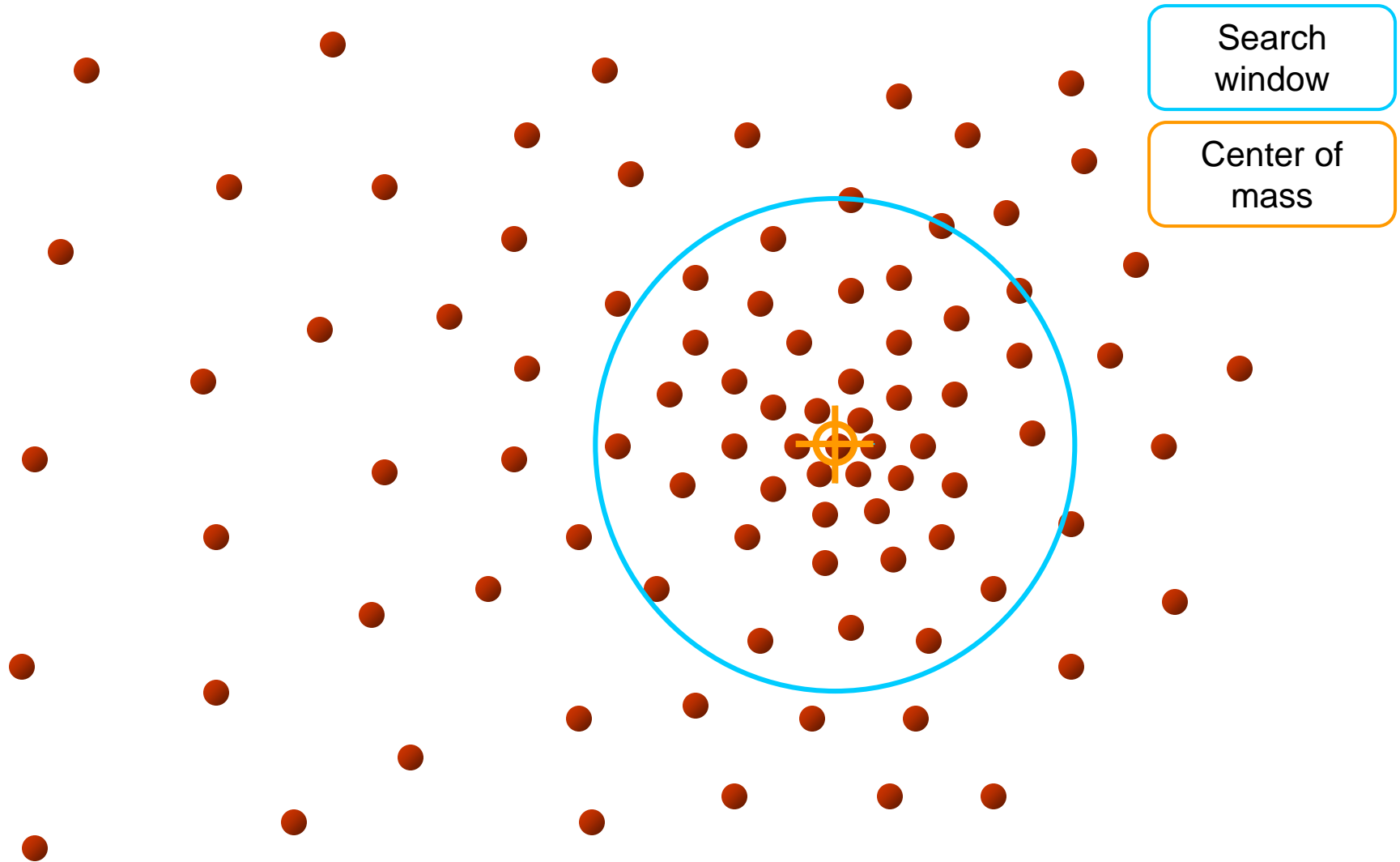
聚类分割—Mean Shift



聚类分割—Mean Shift



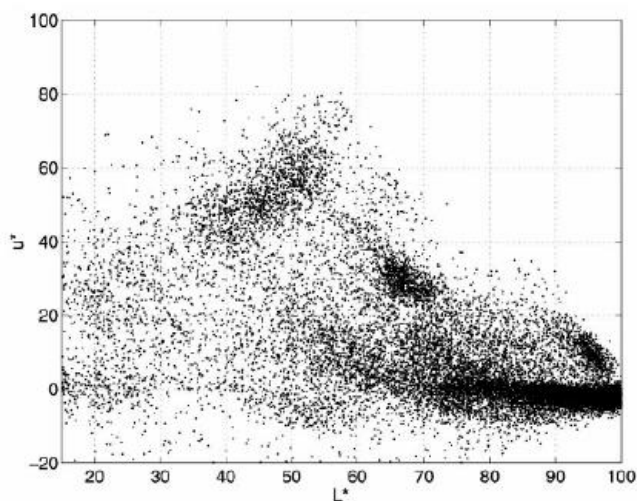
聚类分割—Mean Shift



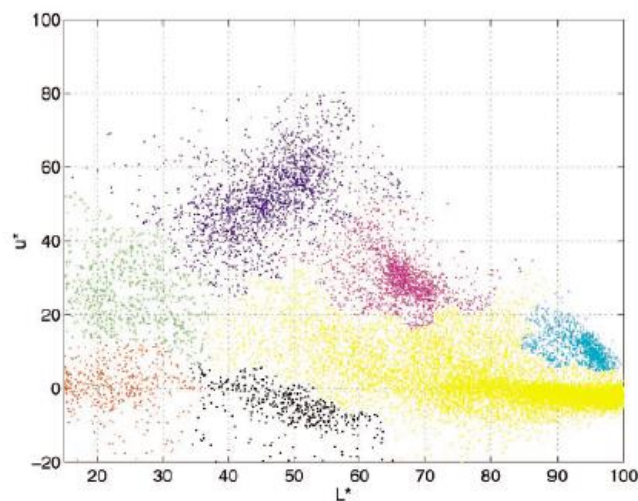
聚类分割—Mean Shift

Mean shift图像分割流程：

1. 选取像素特征（颜色，梯度，位置等）；
2. 在每一个像素特征处计算概率密度；
3. 迭代计算Mean shift并不断移动初始点位置直到收敛；
4. 收敛时位置相近的聚类中心合并。



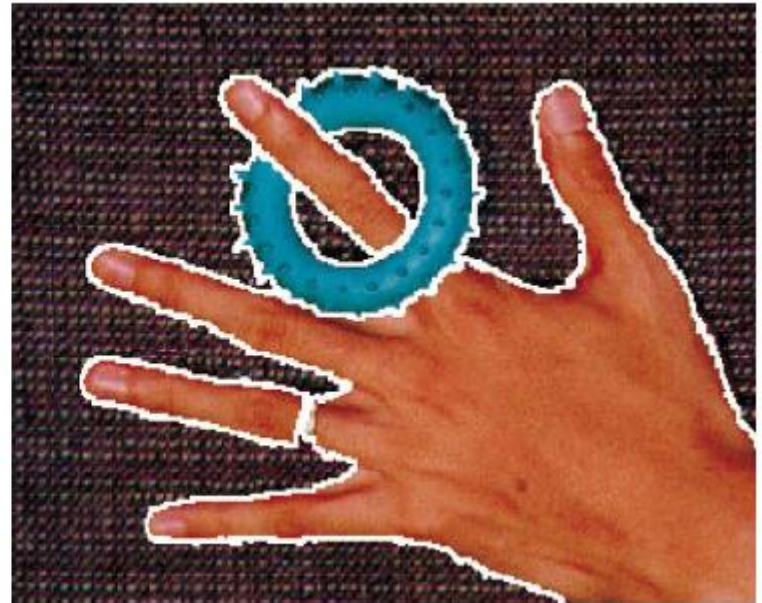
像素 L^*u 颜色空间分布



聚类结果（159次mean shift）

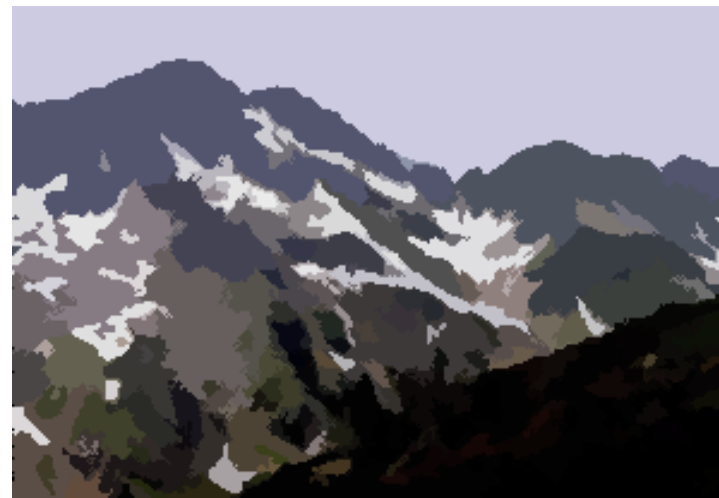
聚类分割—Mean Shift

同时使用使用颜色和位置作为像素特征：



聚类分割—Mean Shift

更多Mean shift分割结果举例：



聚类分割—Mean Shift

更多Mean shift分割结果举例：



聚类分割—Mean Shift

Mean shift优势：

- 1) 聚类结果可以任意形态（不需要球形聚类区域假设）；
- 2) 只有一个参数（窗口大小）；
- 3) 聚类数量可变；
- 4) 对外点鲁棒。

Mean shift不足：

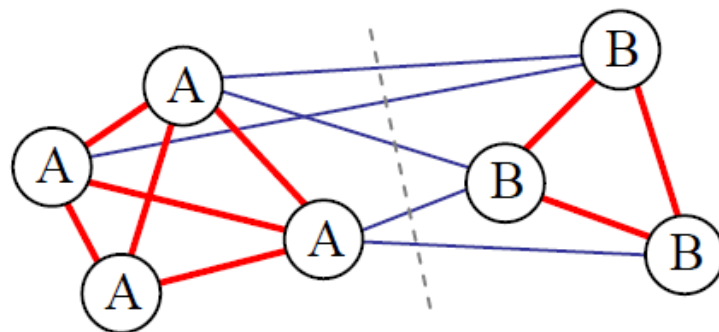
- 1) 输出结果随窗口大小改变而改变；
- 2) 计算复杂度高。

图割集—Normalized cuts

如果将图像看作是顶点和边构成的图，每一个像素是图一个顶点，像素之间的关系是图中的边，那么分割问题可以表达为求图像对应图的割集。

如右图中A和B为顶点，顶点间连线为边，线的粗细表示权重，则虚线对应一个cut，cut值定义为被分割边的权重和：

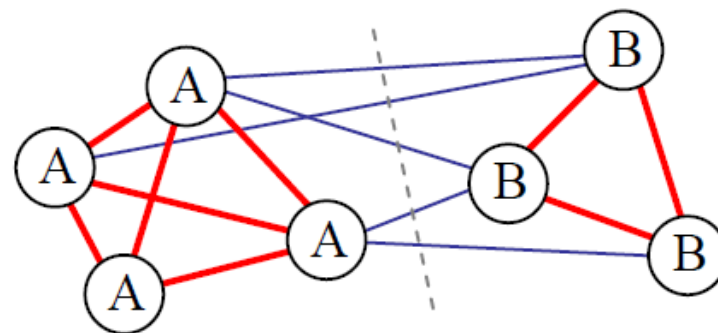
$$cut(A, B) = \sum_{i \in A, j \in B} \omega_{ij}$$



图像分割问题转换为图的minimum cut问题。

图割集— Normalized cuts

$$cut(A, B) = \sum_{i \in A, j \in B} \omega_{ij}$$



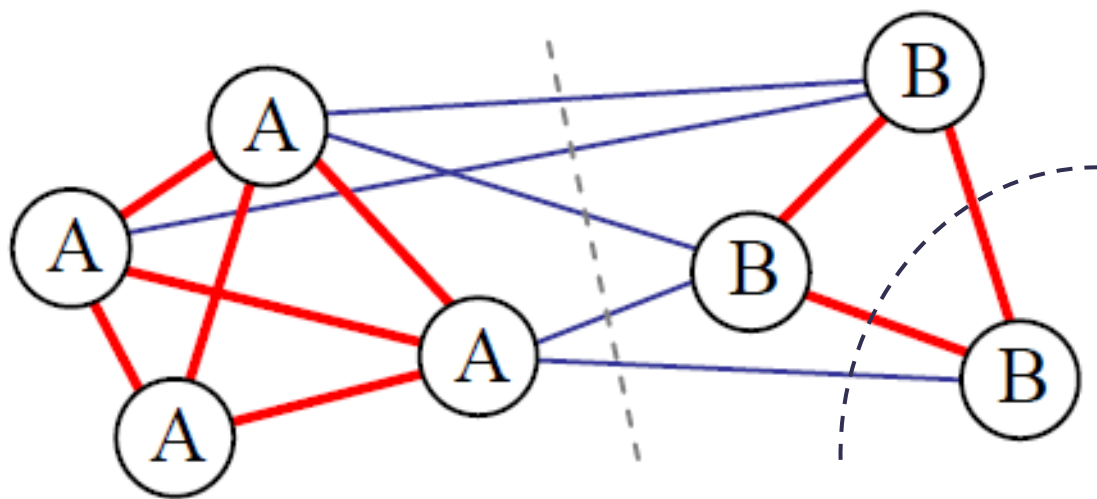
权重的定义: $w_{ij} = \exp\left(-\frac{\|F_i - F_j\|^2}{\sigma_F^2} - \frac{\|x_i - x_j\|^2}{\sigma_s^2}\right)$

像素特征

像素位置

图割集— Normalized cuts

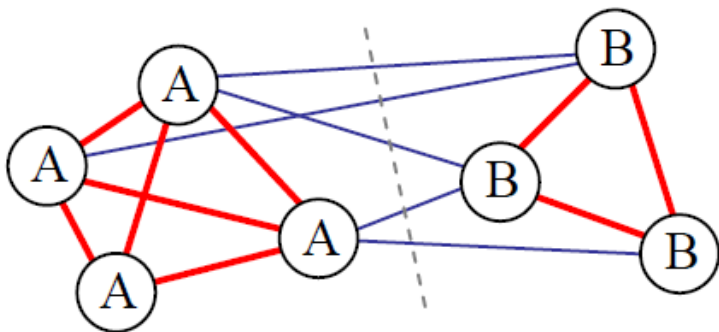
图的minimum cut容易导致分割出单像素点，因此一个更好的cut方式是使用Normalized cut。



图割集— Normalized cuts

Normalized cut定义为:

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



	<i>A</i>	<i>B</i>	sum
<i>A</i>	$assoc(A, A)$	$cut(A, B)$	$assoc(A, V)$
<i>B</i>	$cut(B, A)$	$assoc(B, B)$	$assoc(B, V)$
sum	$assoc(A, V)$	$assoc(B, v)$	

$$assoc(A, V) = \sum_{i \in A, j \in A} \omega_{ij}$$

最小化 $Ncut(A, B)$ 的本质是通过划分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。

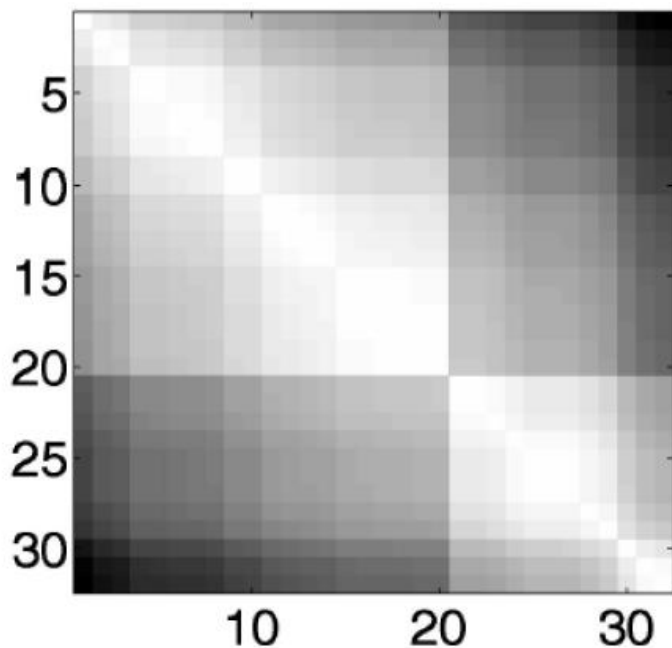
图割集— Normalized cuts

Normalized cut通过最小化 $Ncut(A,B)$ 可以使同时最小化类间相似性和最大化总类内相似性，因此其对应一个最优的cut。

Normalized cut是一个NP-hard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。

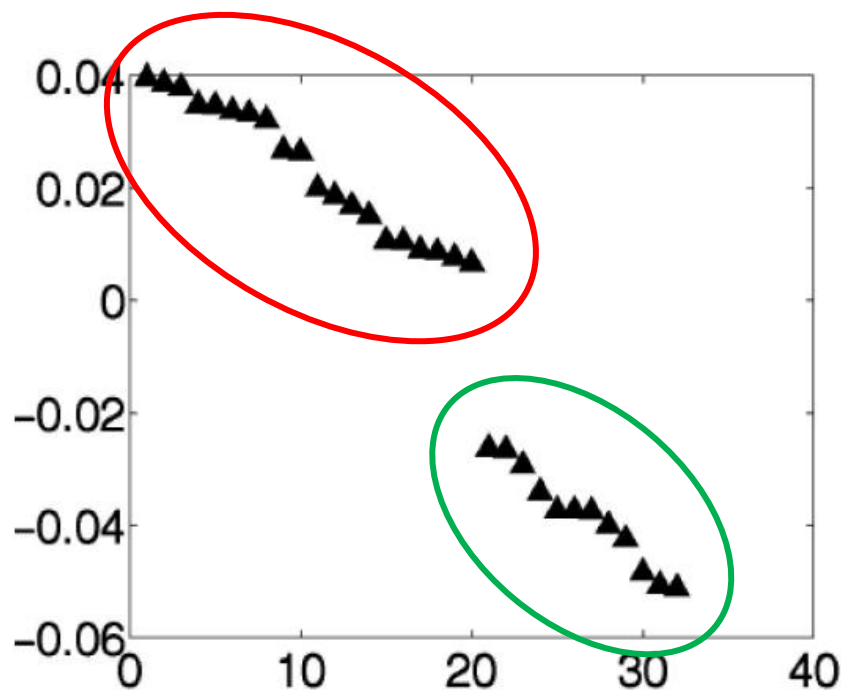
定义 $W=[\omega_{ij}]$ 为权重矩阵， $D=\text{diag}(d(1),\dots,d(N))$ ， $d(i)$ 表示 W 的第 i 行求和，则广义特征值问题： $(D-W)y=\lambda Dy$ 的次小特征值是最小 $Ncut$ 对应的实数解。该特征值所对应的特征向量对应于最优划分。

图割集— Normalized cuts



(a)

一个 32×32 的权重矩阵



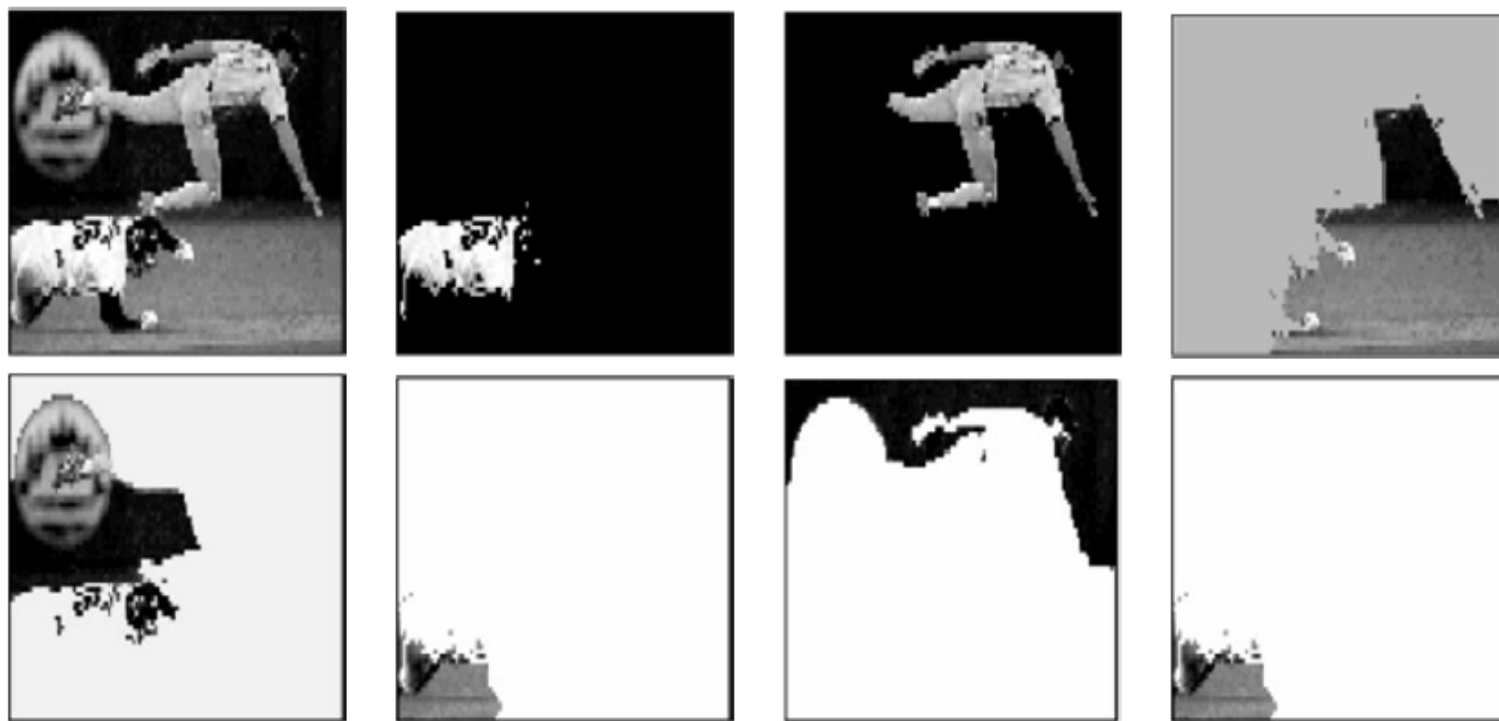
(b)

次小特征值对应的特征向量

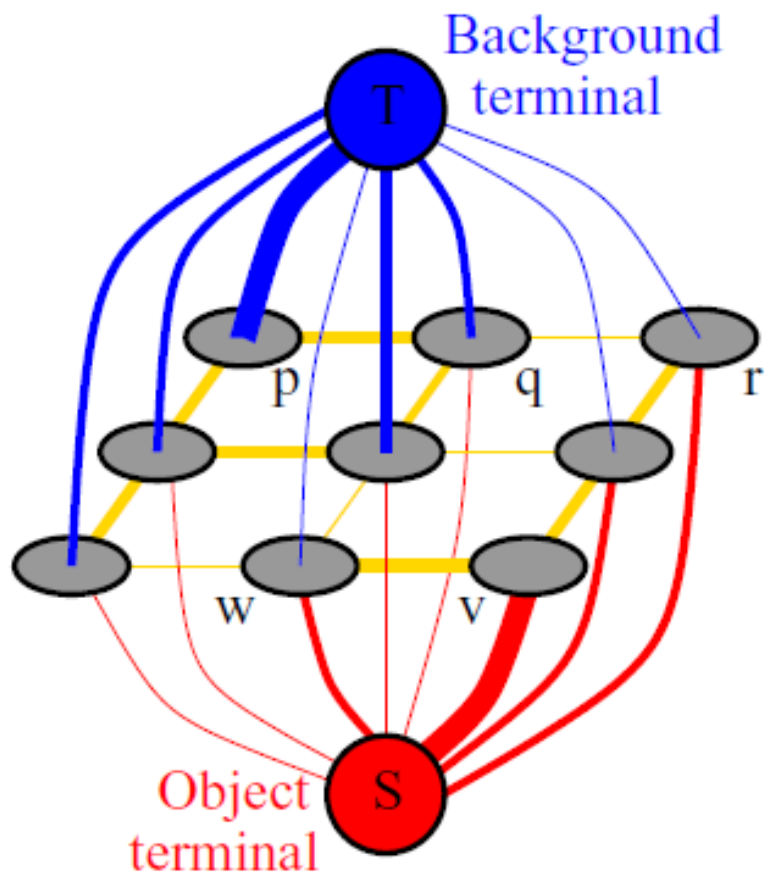
特征向量中每个值对应图中一个节点，可以根据对应值的正负性将节点分为两类。

图割集— Normalized cuts

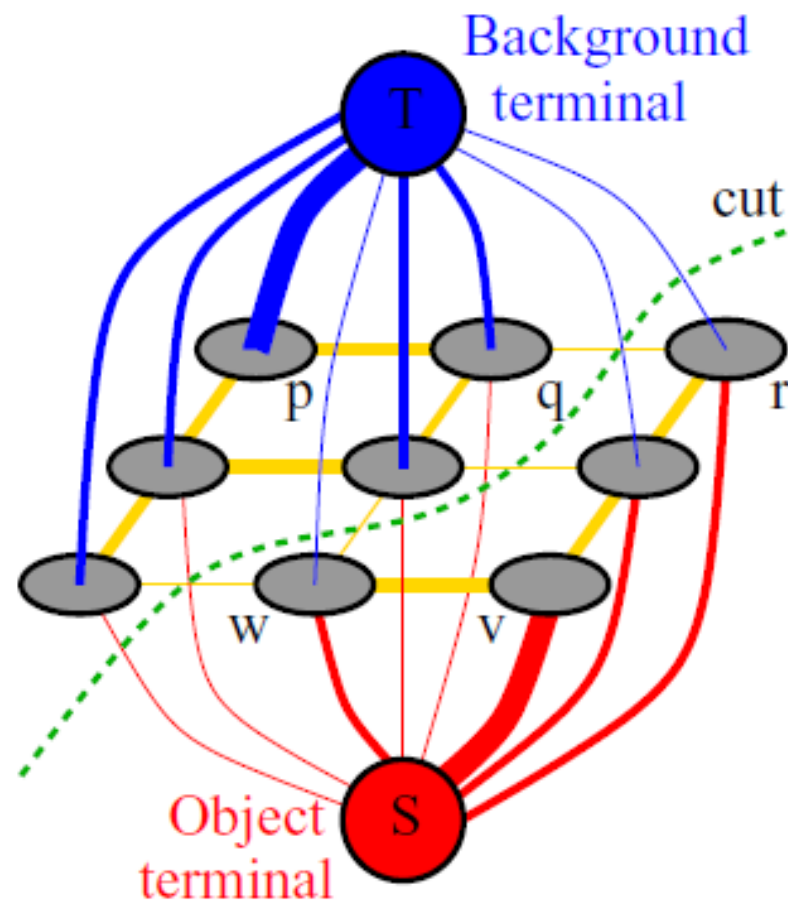
如果需要在二分类基础上进一步分割，则构造子区域的权重矩阵 W ，对角阵 D ，并继续求解其对应的广义特征值问题。



图割集— Graph cuts



Maximum flow



Minimum cut

图割集— Graph cuts

设定S和T节点的方法:




图割集— Graph cuts


定义一个指标集 L ， L 包含图像中每一个像素的0-1标签（0为背景，1为前景）。

那么需要找一个最优指标集 L ，能够最小化下面的能量函数：

$$E(L) = E_d(L) + \lambda E_s(L)$$



数据项，衡量每一个像素与前景/背景的差异性




平滑项，相邻像素之间标签的不一致性

图割集— Graph cuts



$\tilde{L}(x, y)$

$$E(L) = E_d(L) + \lambda E_s(L)$$

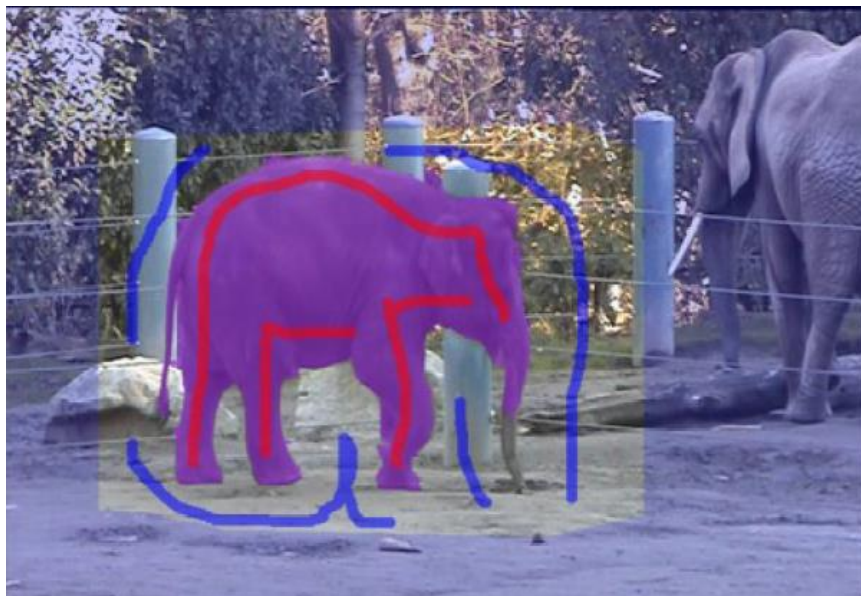

$$E_d(L) = \sum_{(x,y)} C(x, y, L(x, y))$$

$$C(x, y, L(x, y)) = \begin{cases} \infty & \text{if } L(x, y) \neq \tilde{L}(x, y) \\ C'(x, y, L(x, y)) & \text{otherwise} \end{cases}$$

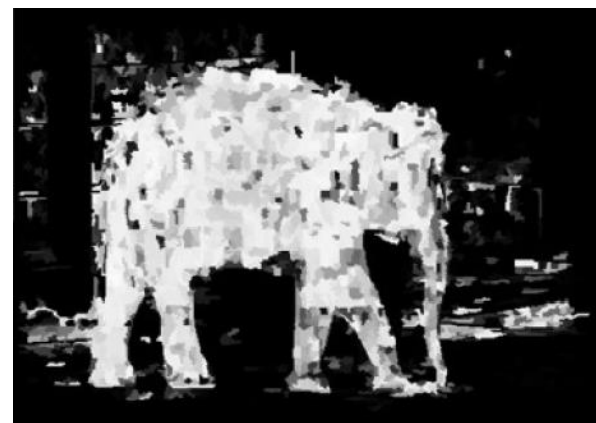
$C'(x, y, 0)$: 像素(x,y)与背景点的相似度

$C'(x, y, 1)$: 像素(x,y)与前景点的相似度

图割集— Graph cuts



$C'(x, y, 0):$



$C'(x, y, 1):$

$C'(x, y, 0):$ 像素(x,y)与背景点的相似度

$C'(x, y, 1):$ 像素(x,y)与前景点的相似度

图割集— Graph cuts

相邻像素点标签应当尽量一致，除非灰度值（或色彩）差异很大。

$$E(L) = E_d(L) + \lambda E_s(L)$$



$$E_s(L) = \sum_{\text{neighbors}(p,q)} \omega_{pq} |L(p) - L(q)|$$

ω_{pq} 表示像素点p和q的图像相似性

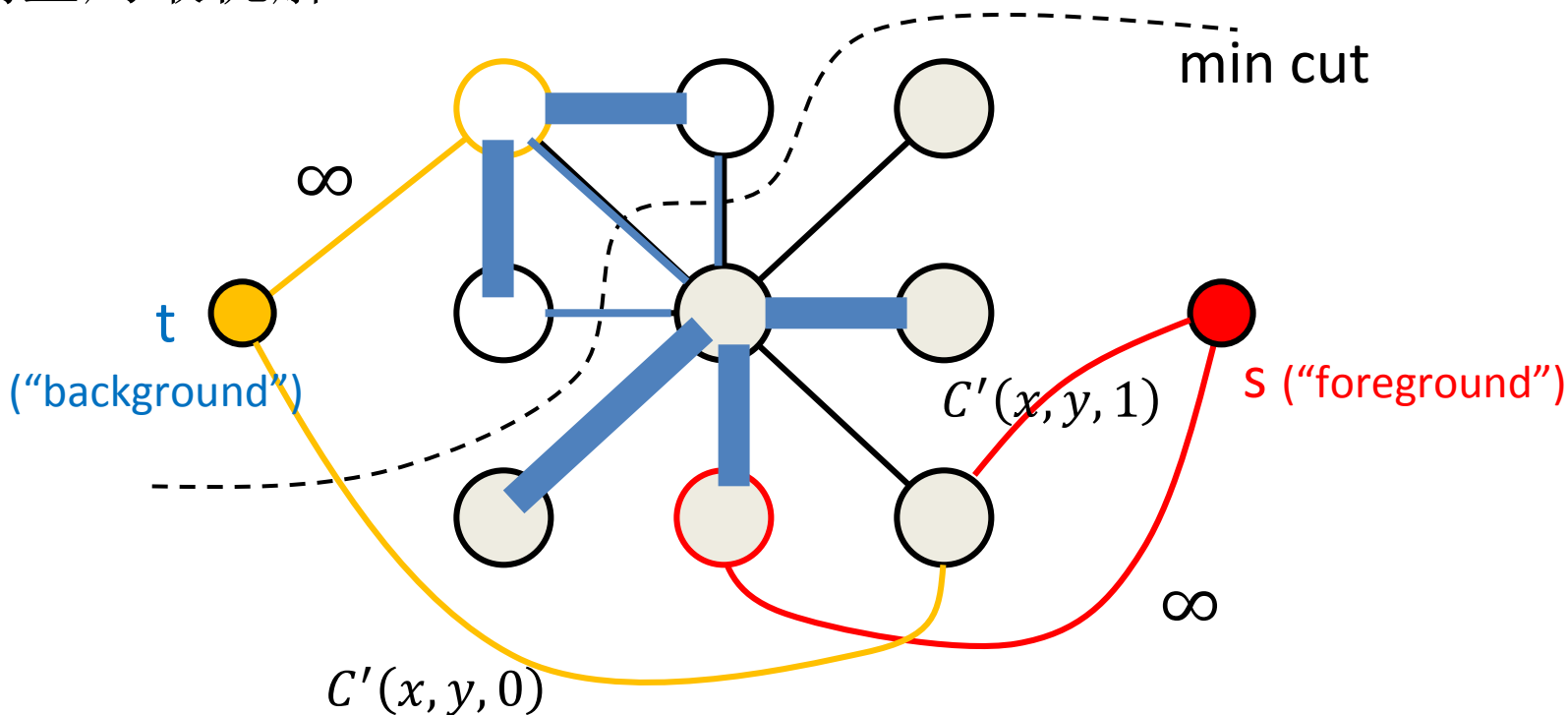


$$\omega_{pq} = 0.1$$

$$\omega_{pq} = 10.0$$

图割集— Graph cuts

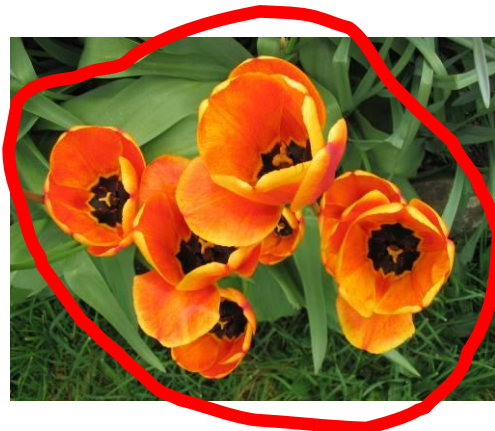
最小化 $E(L) = E_d(L) + \lambda E_s(L)$ 的问题可以通过max flow/min cut找到全局最优解。



min cut给出了s-t的最优分割结果，这一结果对应了前景和背景的最优划分。

图割集— Graph cuts

Graph cuts分割结果



总结—本节课内容回顾

阈值分割— p 分位数法

阈值分割—双峰法

阈值分割—迭代法

阈值分割—Otsu法

区域分割—Watershed

★ 活动轮廓—Snake

★ 活动轮廓—Condensation

★ 聚类分割—K-means

★ 聚类分割—Mean shift

图割集—Normalized cuts

★ 图割集—Graph cuts

总结—参考文献

- Kass, M., Witkin, A., and Terzopoulos, D. [Snakes: Active contour models](#). IJCV, 1(4):321–331, 1988.
- Isard, M. and Blake, A. [CONDENSATION—conditional density propagation for visual tracking](#). IJCV, 29(1):5–28, 1998.
- Comaniciu, D. and Meer, P. [Mean shift: A robust approach toward feature space analysis](#). IEEE T-PAMI, 24(5):603–619, 2002.
- Shi, J. and Malik, J. [Normalized cuts and image segmentation](#). IEEE T-PAMI, 8(22):888–905, 2000.
- Boykov, Y. and Kolmogorov, V. [An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision](#). IEEE T-PAMI, 26(9):1124–1137, 2004.