



# 数字图像处理 Digital Image Processing

## 彩色图像分割 Color Image Segmentation





# 彩色图像分割及处理

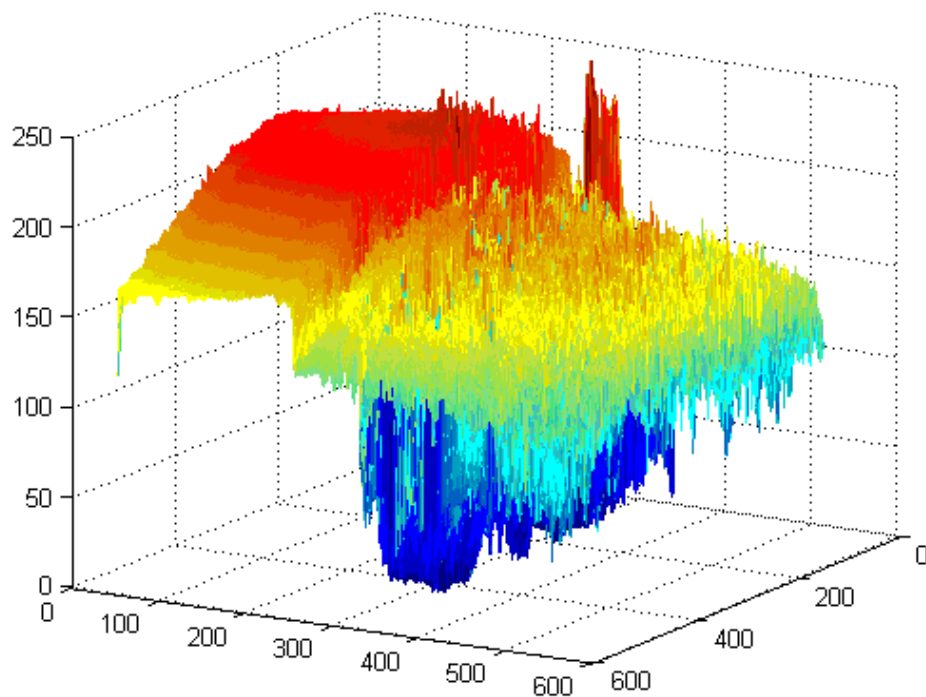
1. 分水岭算法
2. Mean shift分割
3. Normalized cuts(Ncuts)分割
4. Ncuts分割改进算法
5. Graph cuts(GC) 优化
6. GC与交互式分割
7. Graph cut与变分模型





# 分水岭算法

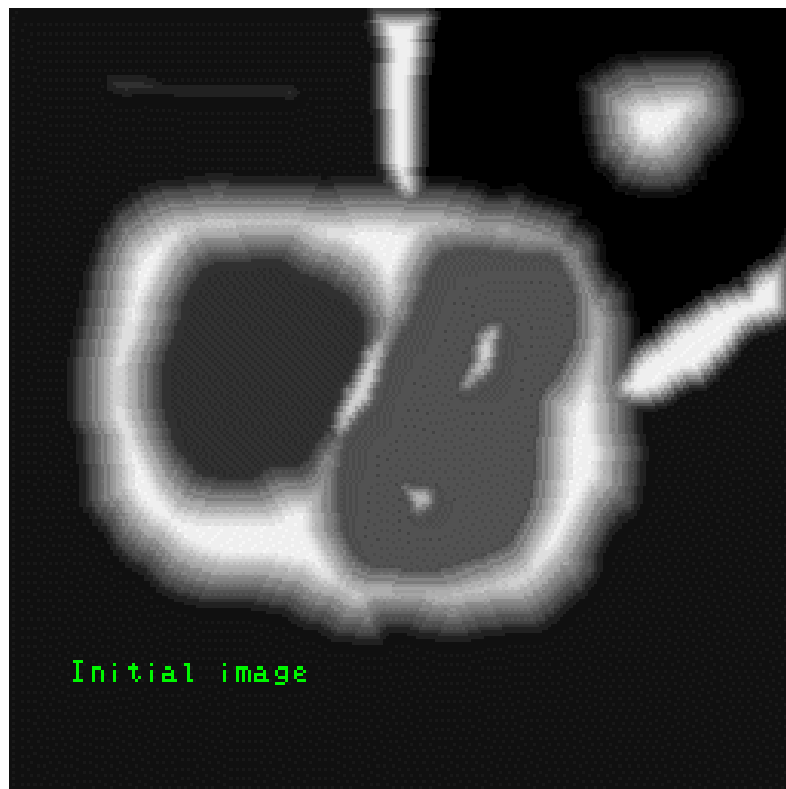
## Watershed algorithm



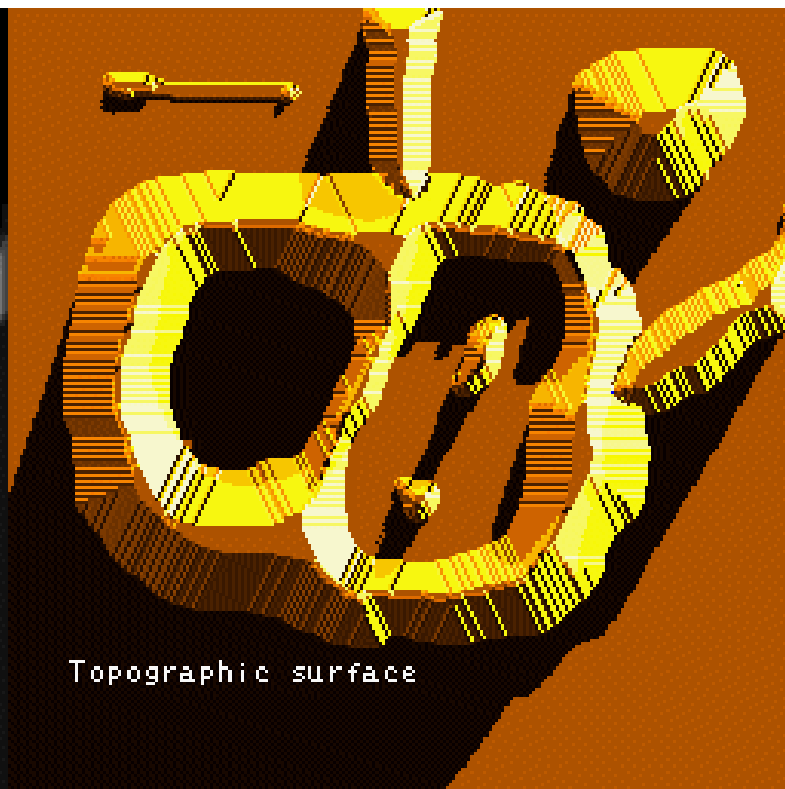


# 分割原理

(1) 任何的灰度级图像都可以被看做是一个地形图



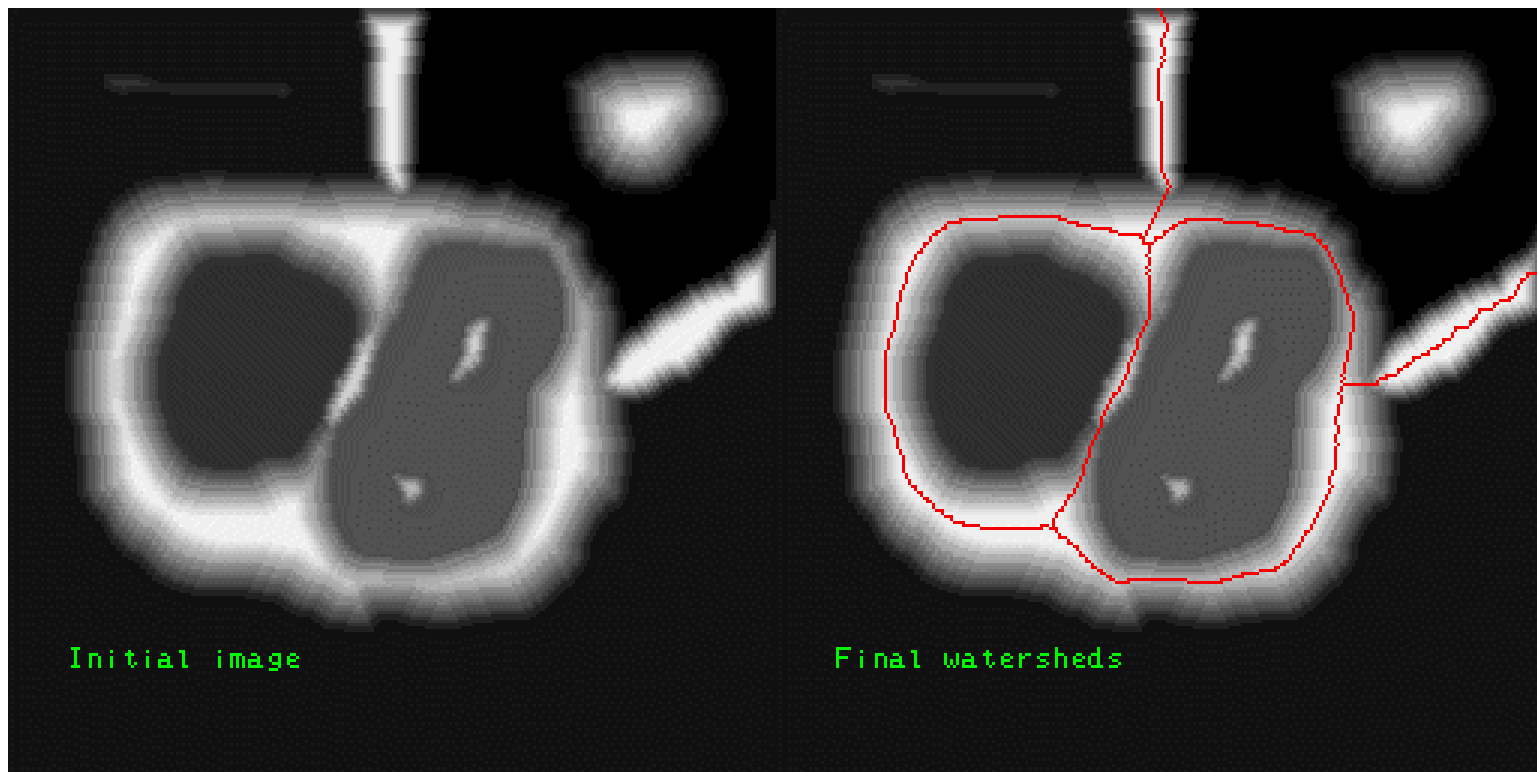
Initial image



Topographic surface

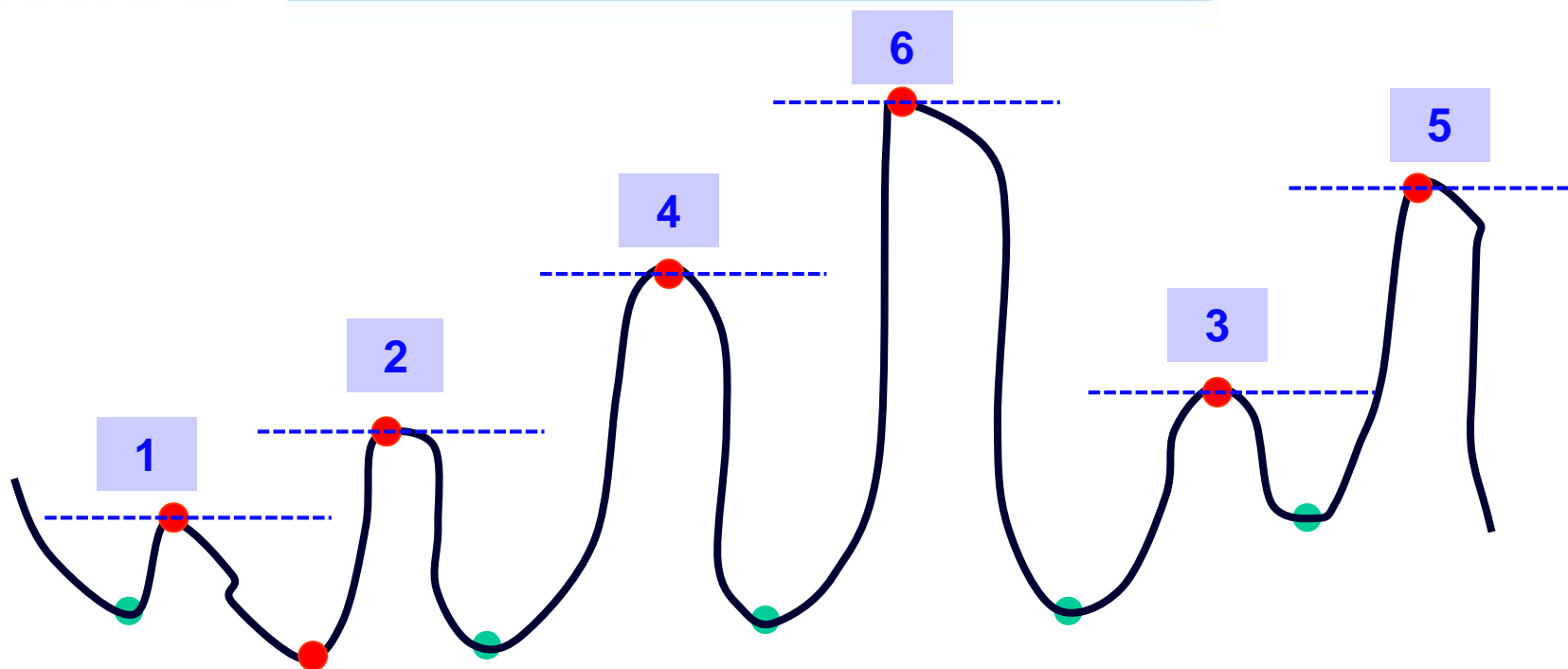


(2) 假设我们在每个区域最小值位置地方打个洞，让水以均匀的速度上升，从低到高淹没整个地形。当处在不同的汇聚盆地中的水将要聚合在一起时，修建大坝将阻止聚合，最后得到的水坝边界就是分水岭的分割线。



Initial image

Final watersheds



- 获取**局部极小**
- 统计**连通分量**
- 以最小的局部极小为基准**提升高度**
- 如果出现某两个连通分量合并，则**记录分界线**
- 逐渐提升高度，直至整个图像**合并一个连通分量**
- 记录的所有的分界线即为**分割边界**

**递归过程**



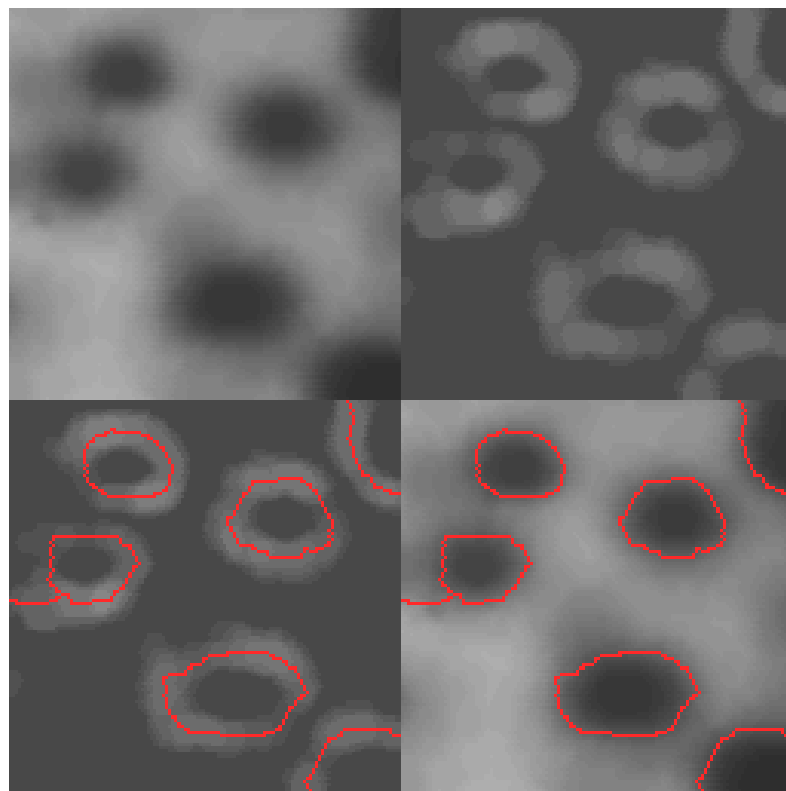


## ◆ 分水岭算法用于梯度图像

分水岭方法应用在**梯度图像**，那么**集水处**对应**灰度变化最小的区域**，而**分水岭**对应**灰度变化相对最大的区域**。

从上到下，从右到左

- 原始图
- 梯度图
- 梯度图的分水岭
- 最终轮廓

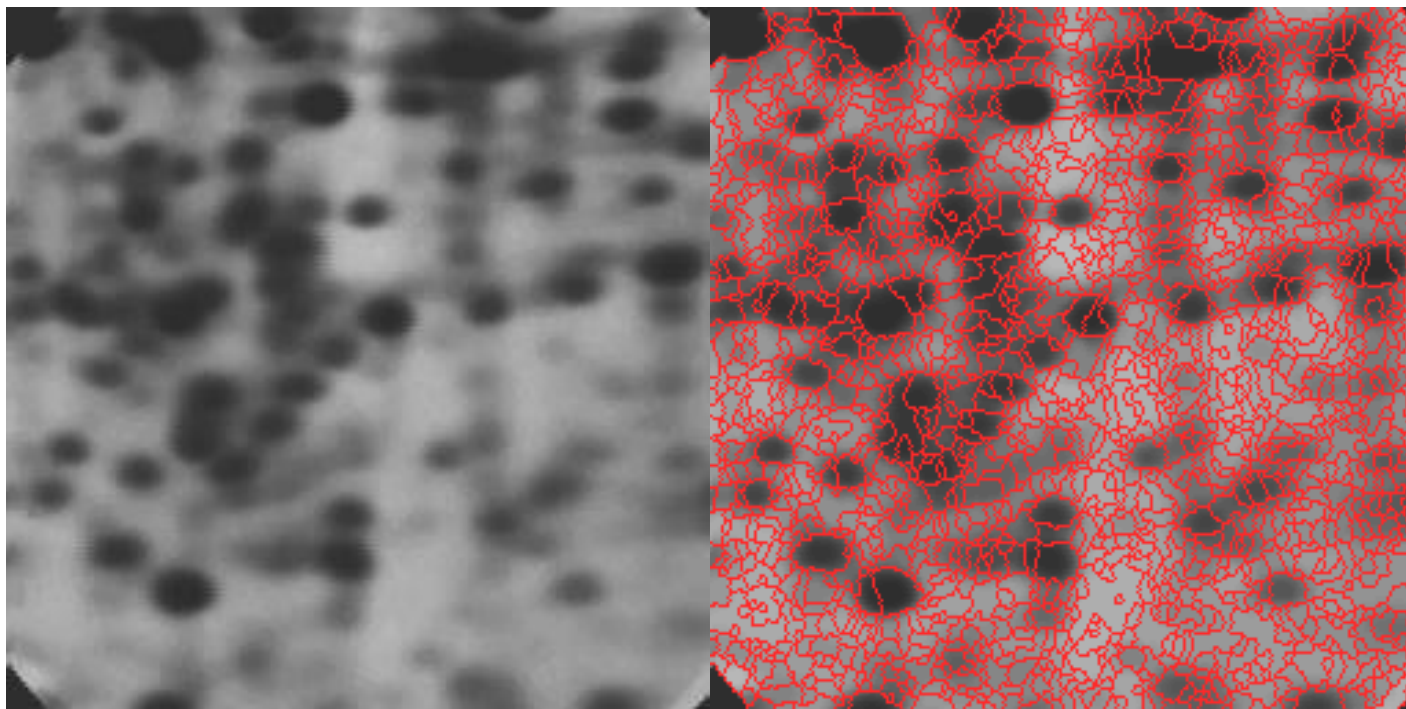






## ➤ 分水岭由于噪声或者局部不规则而引起“过度分割”

电泳凝胶图像与经过分水岭转变的分割图



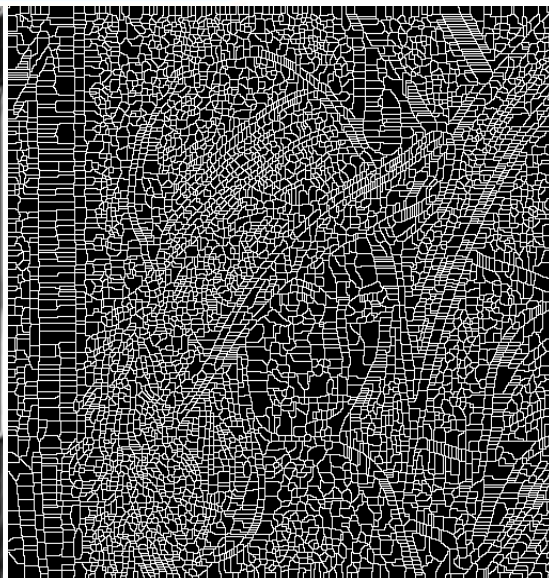




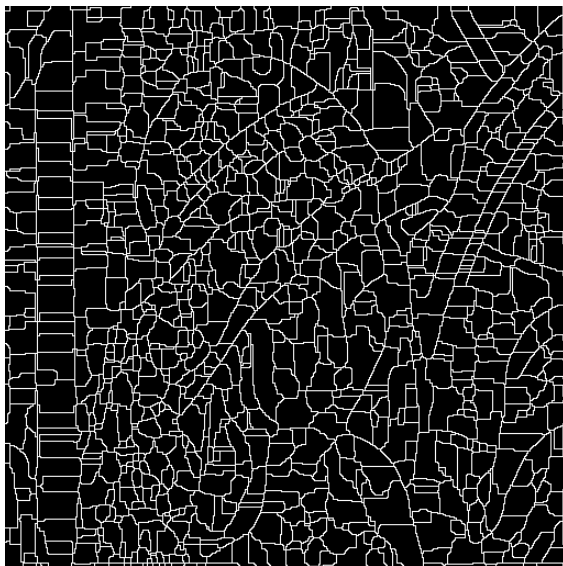
Image



Gradient



Old Watershed



New Watershed

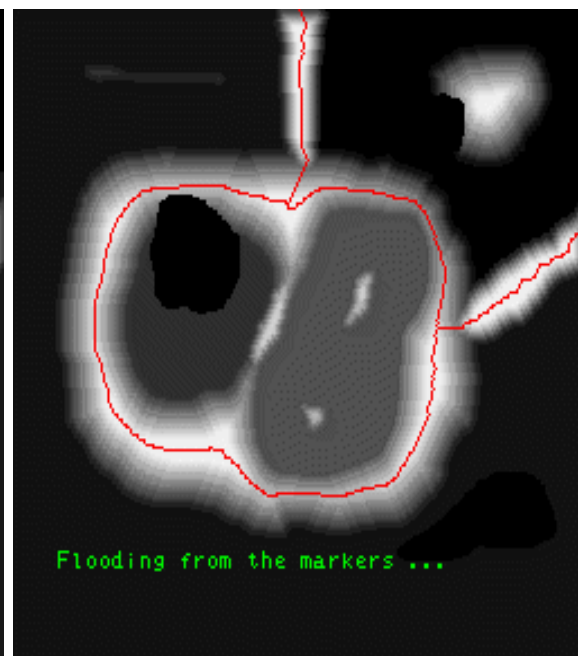
## ◆ 分水岭之前平滑去噪

采用高斯滤波或者中值滤波先处理  
去掉一些局部极小，减少区域个数



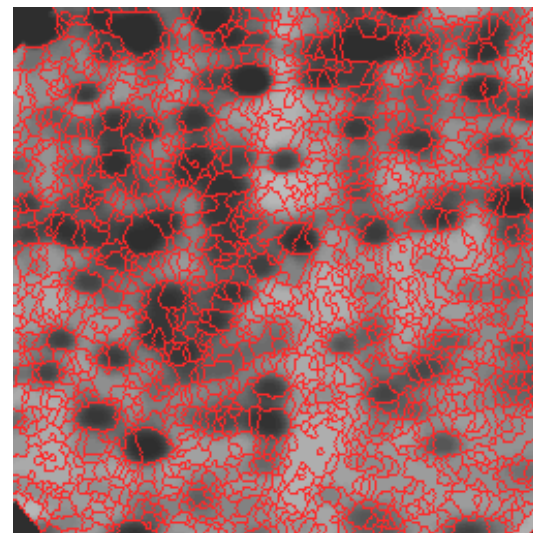
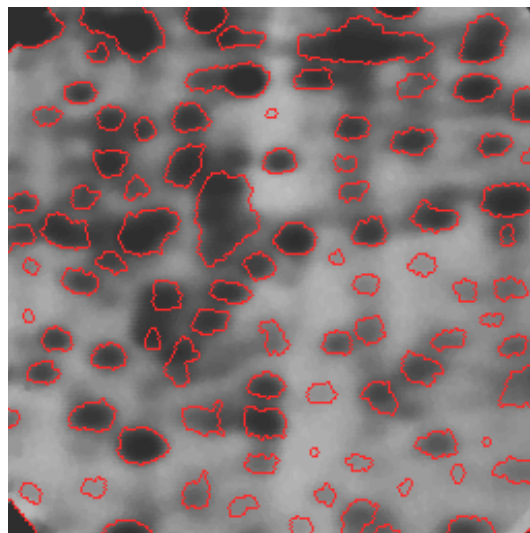
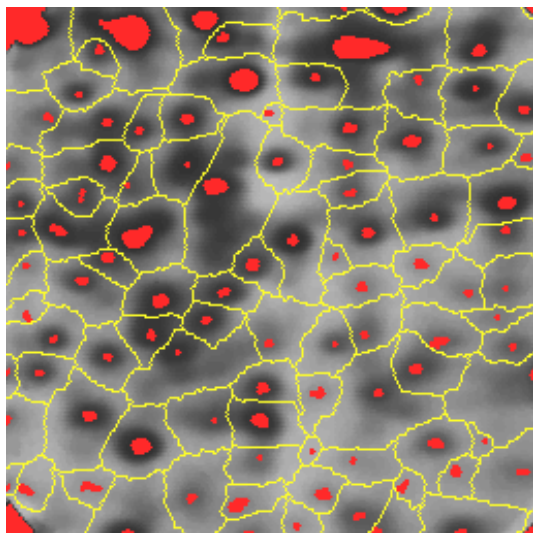
## ◆ 标记约束分水岭算法

从先前已经定好的区域开始浸水，防止过度分割

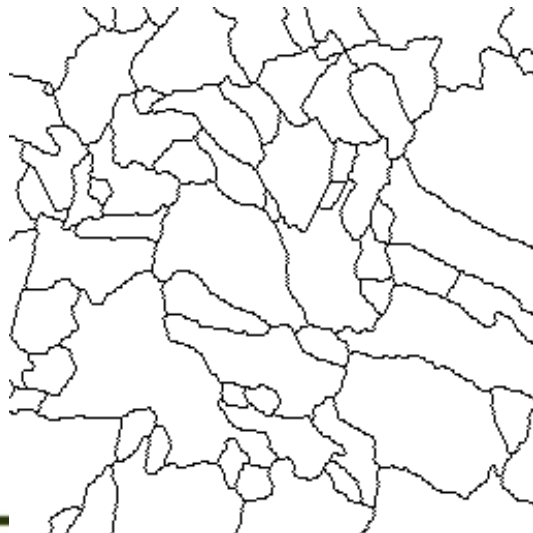
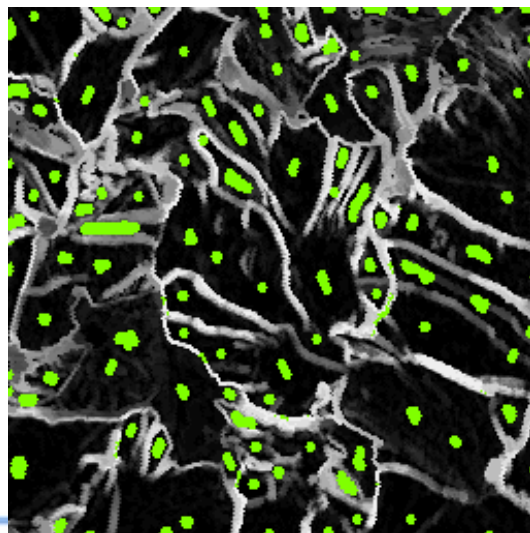
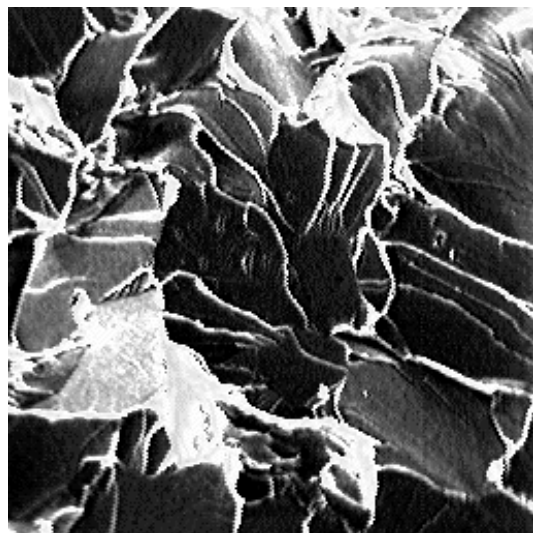




## 电泳凝胶图像与经过标记约束分水岭转变的分割图



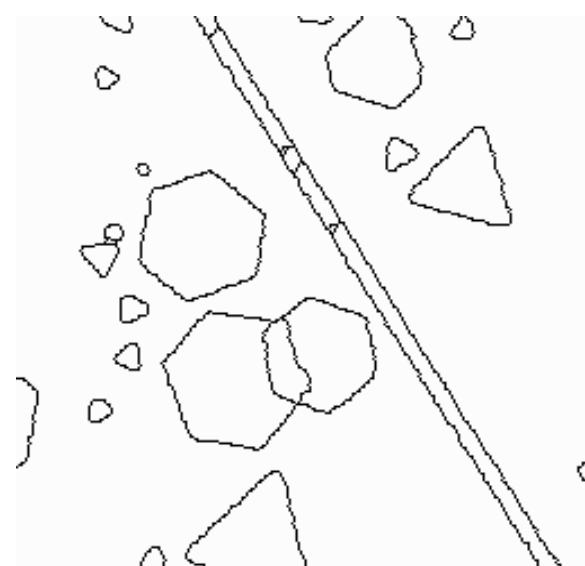
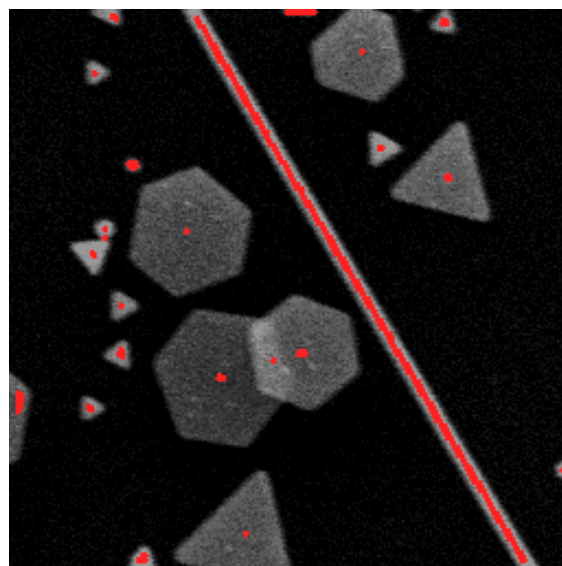
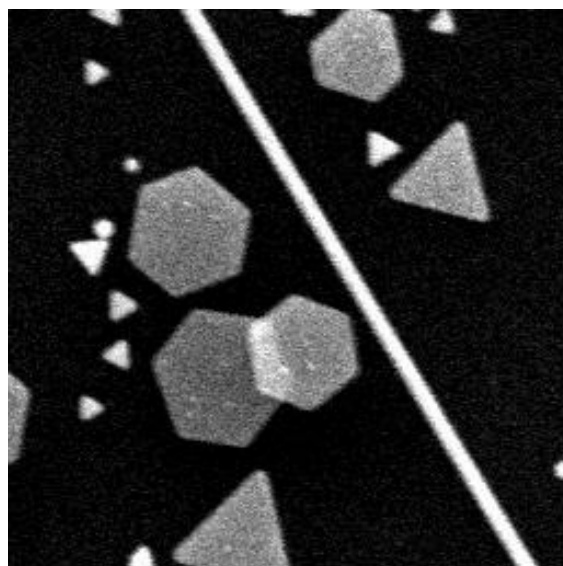
## 钢的断裂面的提取







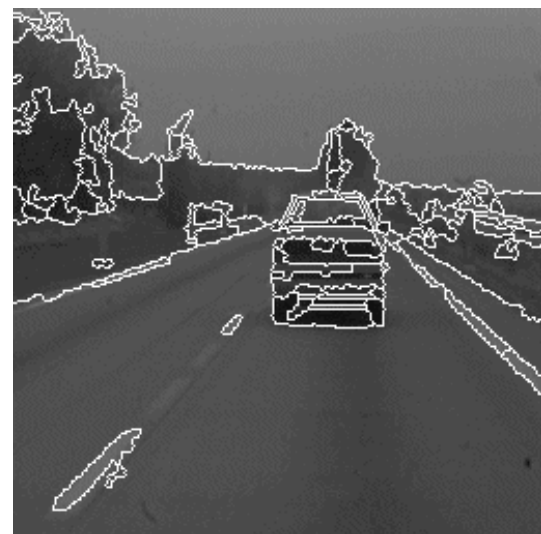
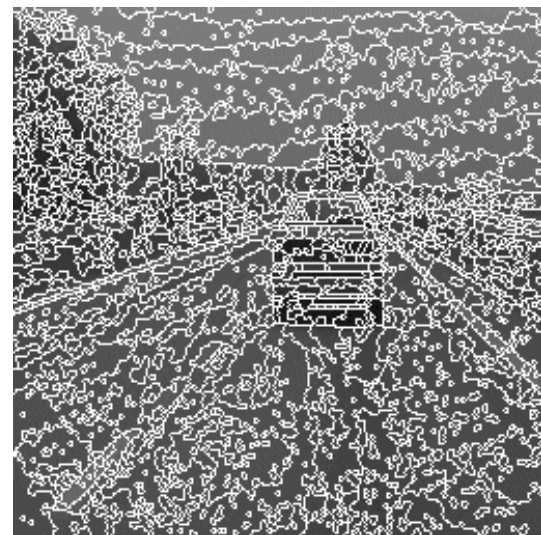
## 银色木纹的提取





## ◆ 分级分割

- 通过分水岭算法，得到一张初始的分割图片
- 以这些相对高度为基础，再次用分水岭算法，可达下一级的分割图如下





# Meyer(1991)分水岭分割算法

1. 选择所有局部最小作为区域种子
2. 将其所有邻域点加到优先级队列，按照值的大小排序
3. 从队列中取出优先级最高的像素点：
  1. 如果该像素所有的被标记的邻域点都有相同的标记，那么就把这个标记给该像素
  2. 将它的所有未被标记的点加入优先级队列
4. 重复步骤3直到结束

**Matlab: `seg = watershed(bnd_im)`**







# 分水岭分割算法的优点和缺点

- 优点
  - Fast ( $< 1$  sec for  $512 \times 512$  image)
  - Among best methods for hierarchical segmentation
- 缺点
  - 容易“过分割”
  - No top-down information
- Usage
  - Preferred algorithm for hierarchical segmentation





# 彩色图像分割及处理

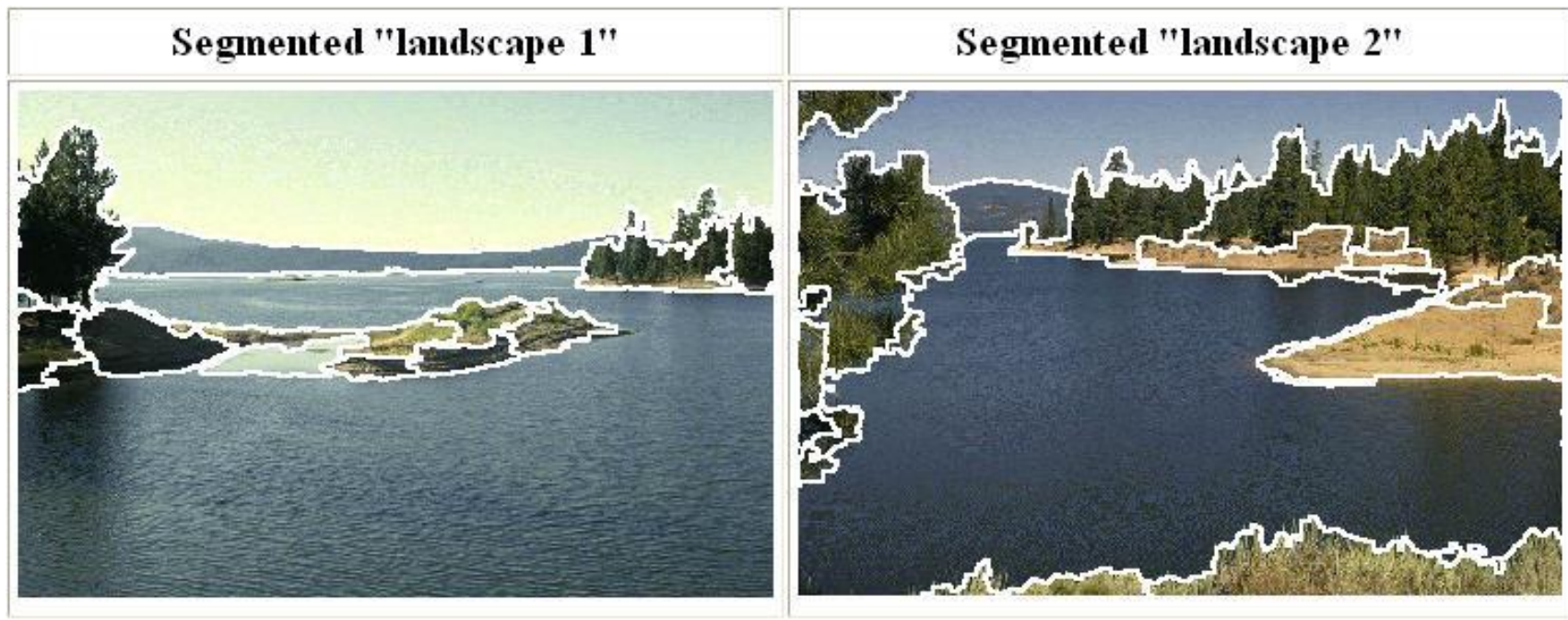
1. 分水岭算法
2. Mean shift分割
3. Normalized cuts(Ncuts)分割
4. Ncuts分割改进算法
5. Graph cuts(GC) 优化
6. GC与交互式分割
7. Graph cut与变分模型





# Mean shift segmentation

- Versatile technique for clustering-based segmentation



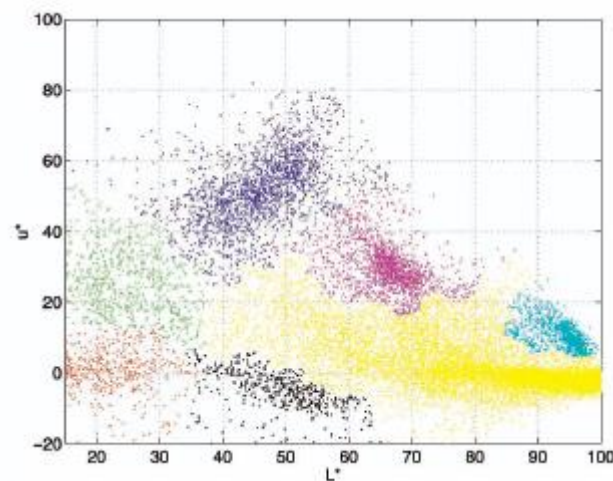
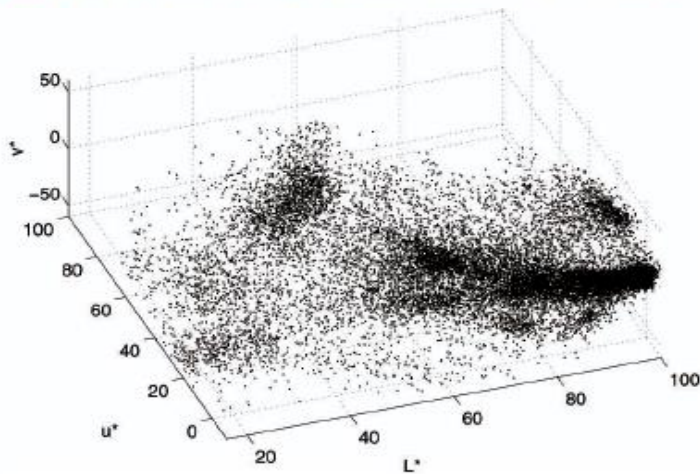
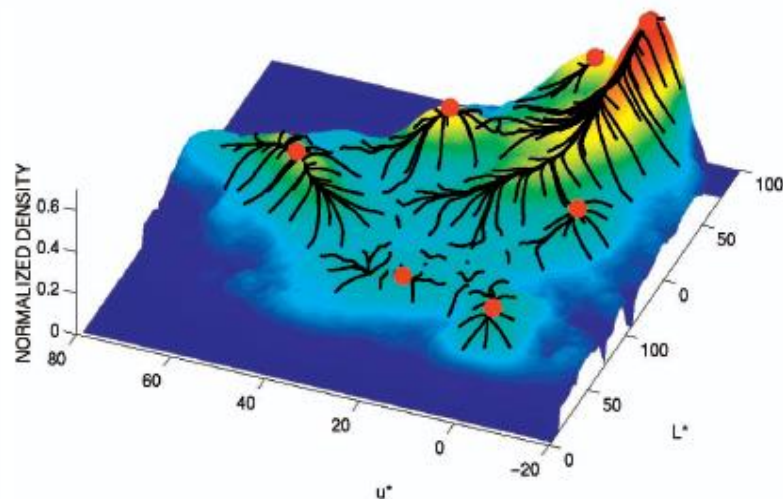
- ◆ D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.





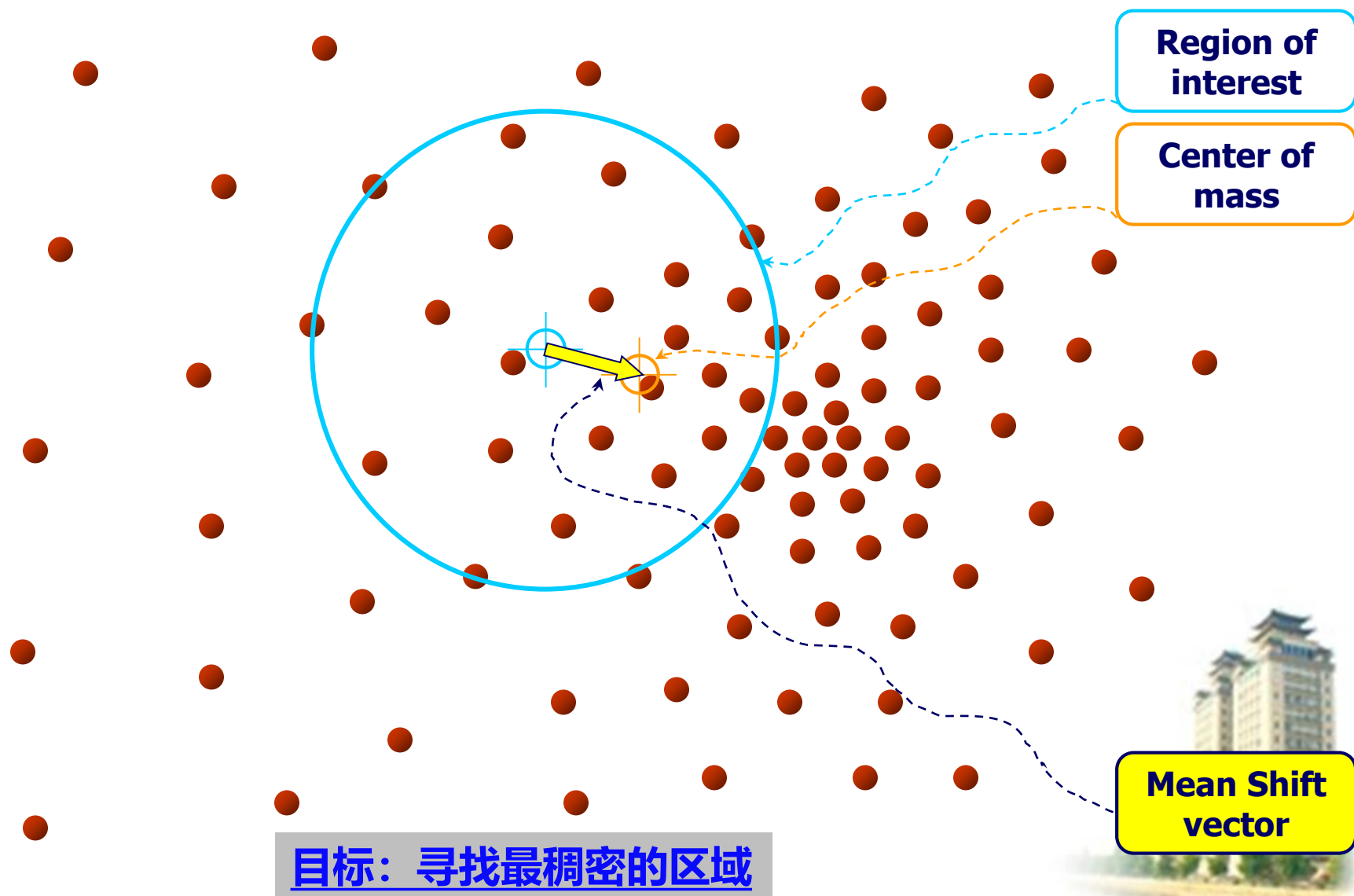
# Mean shift algorithm

- Try to find *modes* of this non-parametric density





# Intuitive Description

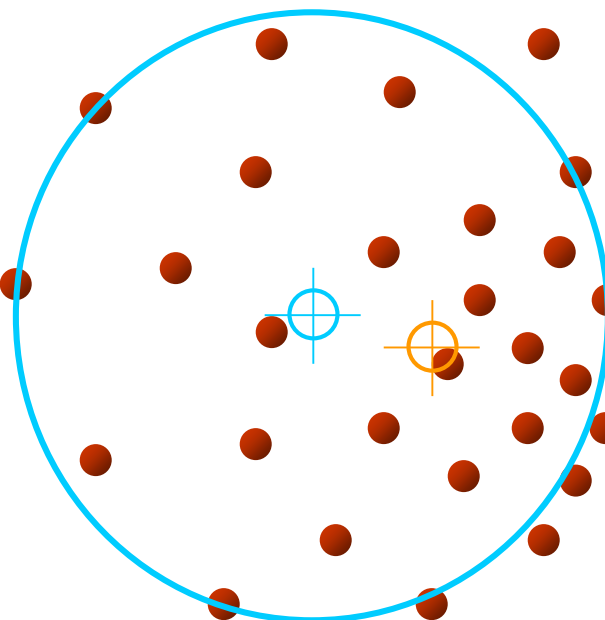




# Intuitive Description

Region of  
interest

Center of  
mass



Mean Shift  
vector

目标：寻找最稠密的区域

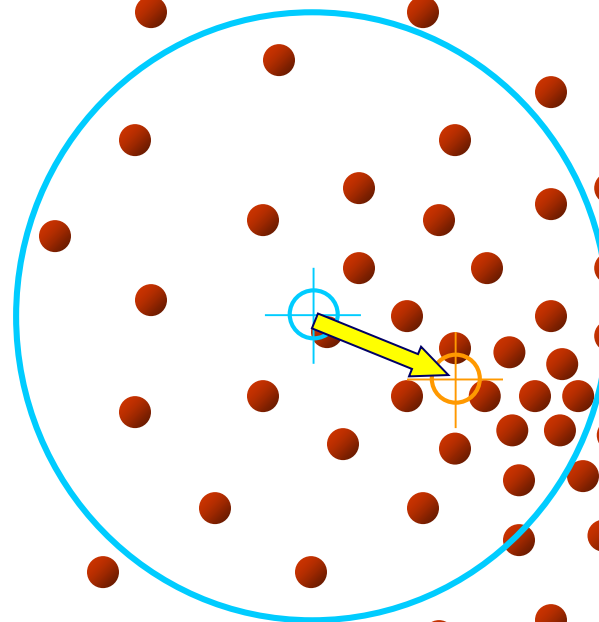




# Intuitive Description

Region of  
interest

Center of  
mass



Mean Shift  
vector

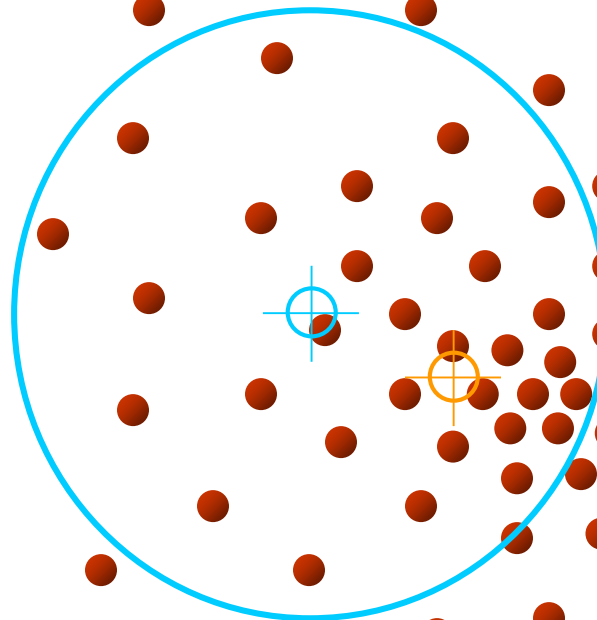
目标：寻找最稠密的区域



# Intuitive Description

Region of  
interest

Center of  
mass



Mean Shift  
vector

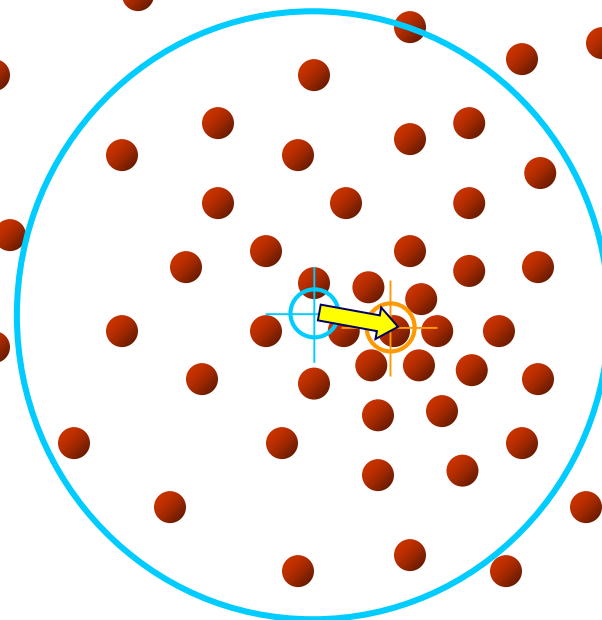
目标：寻找最稠密的区域



# Intuitive Description

Region of  
interest

Center of  
mass



Mean Shift  
vector

目标：寻找最稠密的区域



# Intuitive Description

Region of  
interest

Center of  
mass

Mean Shift  
vector

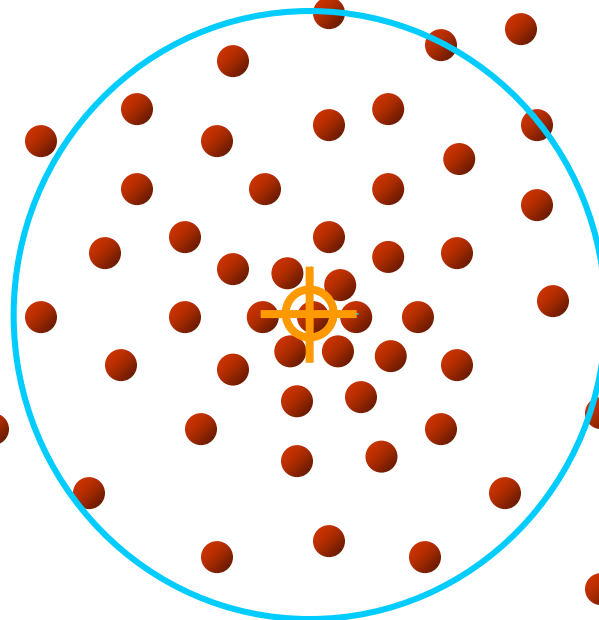
目标：寻找最稠密的区域



# Intuitive Description

Region of  
interest

Center of  
mass



目标：寻找最稠密的区域





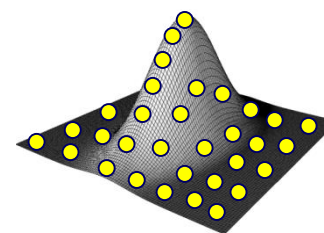
# What is Mean Shift ?

## A tool for:

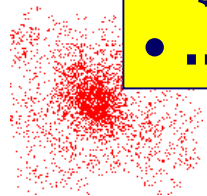
在样本集合中寻找模型，确定 $N$ 维空间 $R^N$ 里面一个潜在的概率密度函数 (PDF- probability density function)

特征空间的概率密度函数

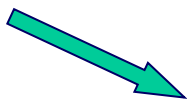
- 颜色空间 (**color space**)
- 尺度空间 (**Scale space**)
- 事实上我们可以设想的任意特征空间
- ...



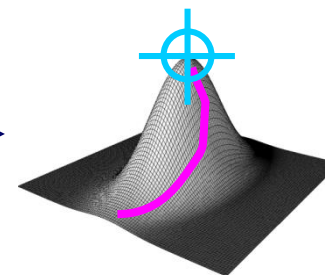
Discrete PDF Representation



Data



非参数密度**梯度**估计  
Non-parametric  
Density **GRADIENT** Estimation  
(Mean Shift)



PDF Analysis

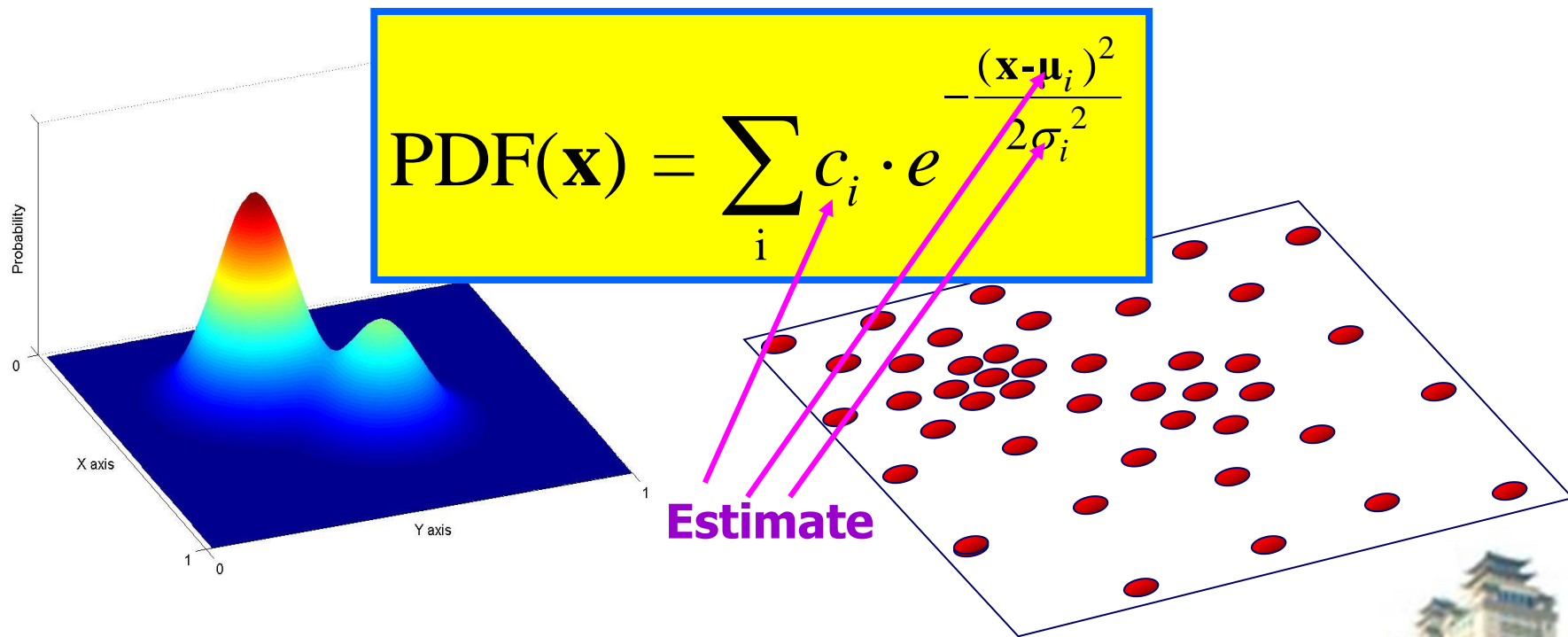






# *Parametric* Density Estimation

Assumption : The data points are sampled from an underlying PDF



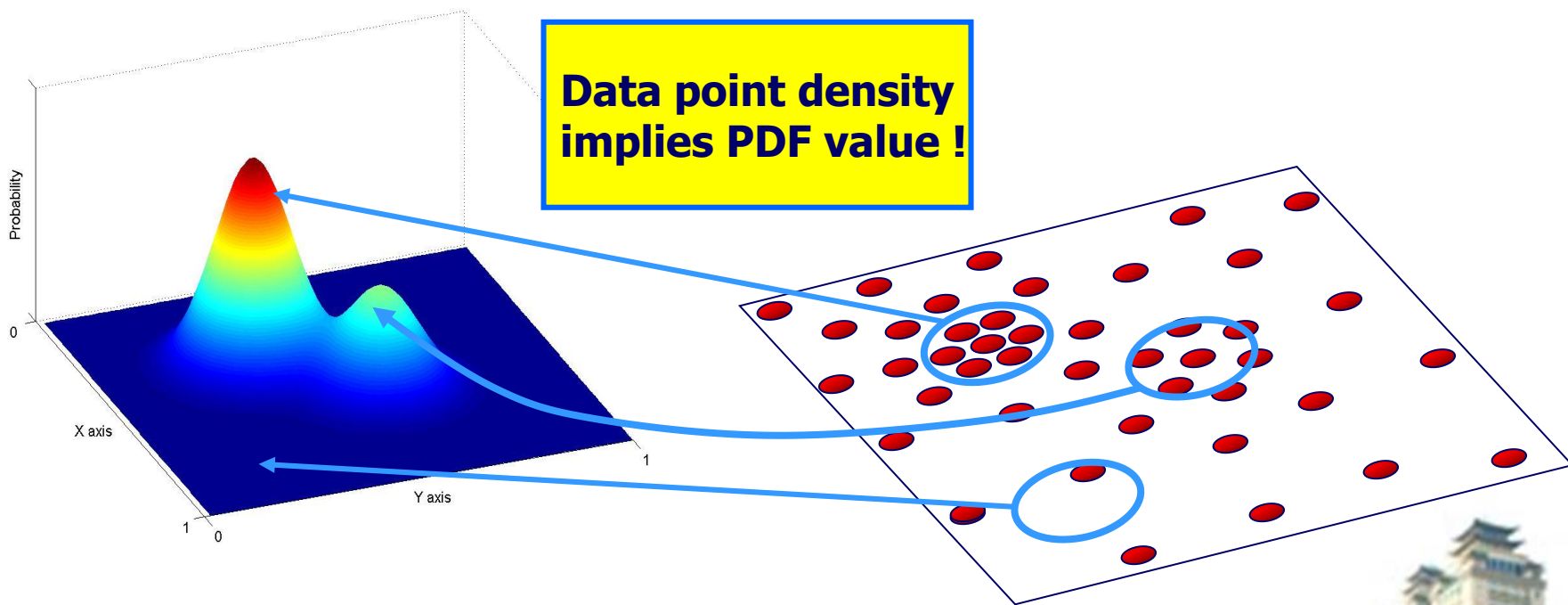
Assumed Underlying PDF

Real Data Samples



# Non-Parametric Density Estimation

假设：数据点是从一个隐含的概率密度函数PDF进行采样

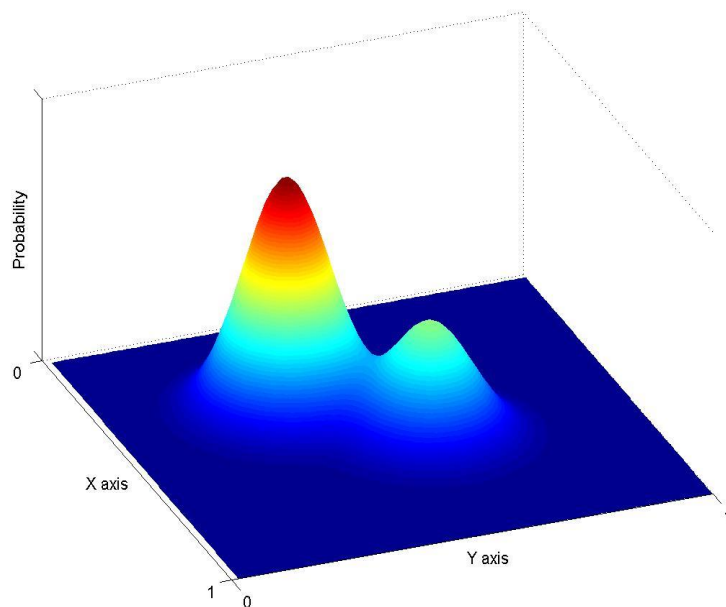


Assumed Underlying PDF

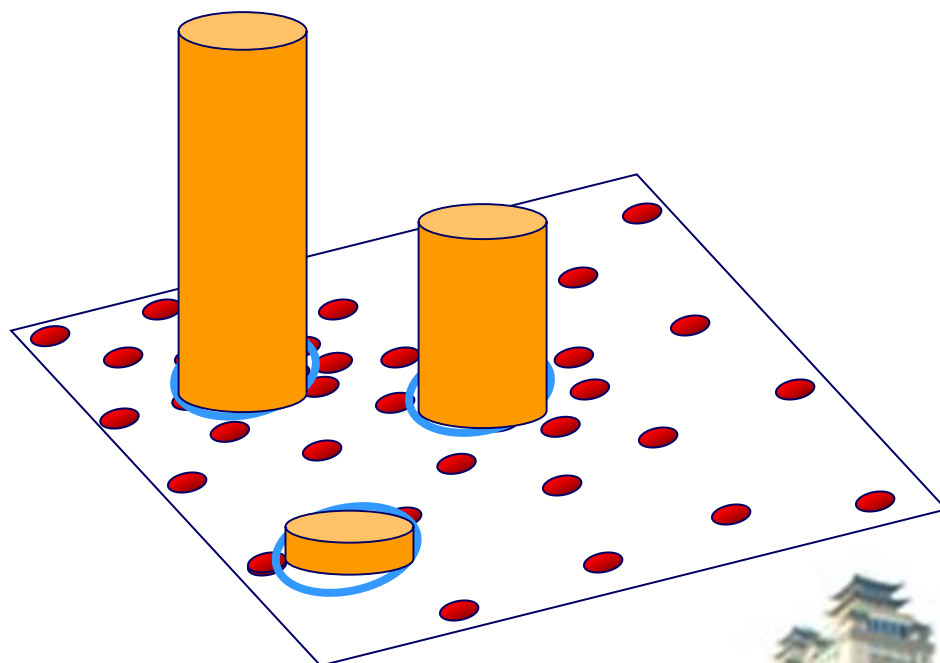
Real Data Samples



# Non-Parametric Density Estimation



Assumed Underlying PDF

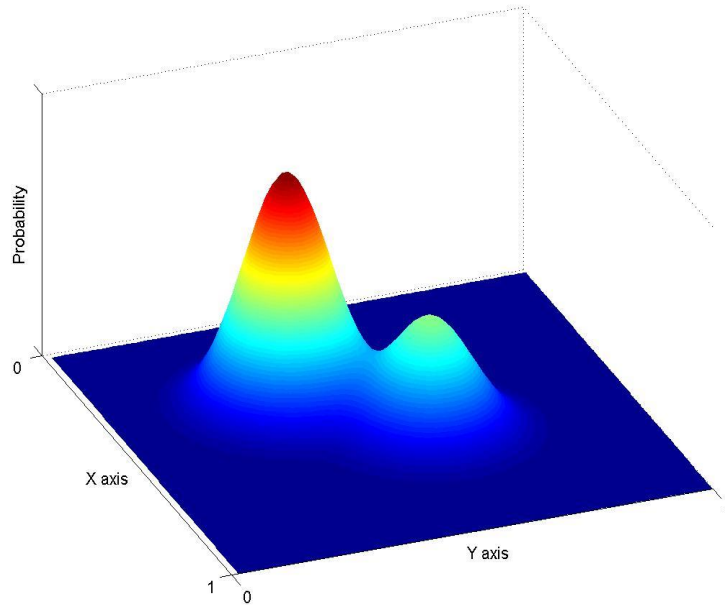


Real Data Samples

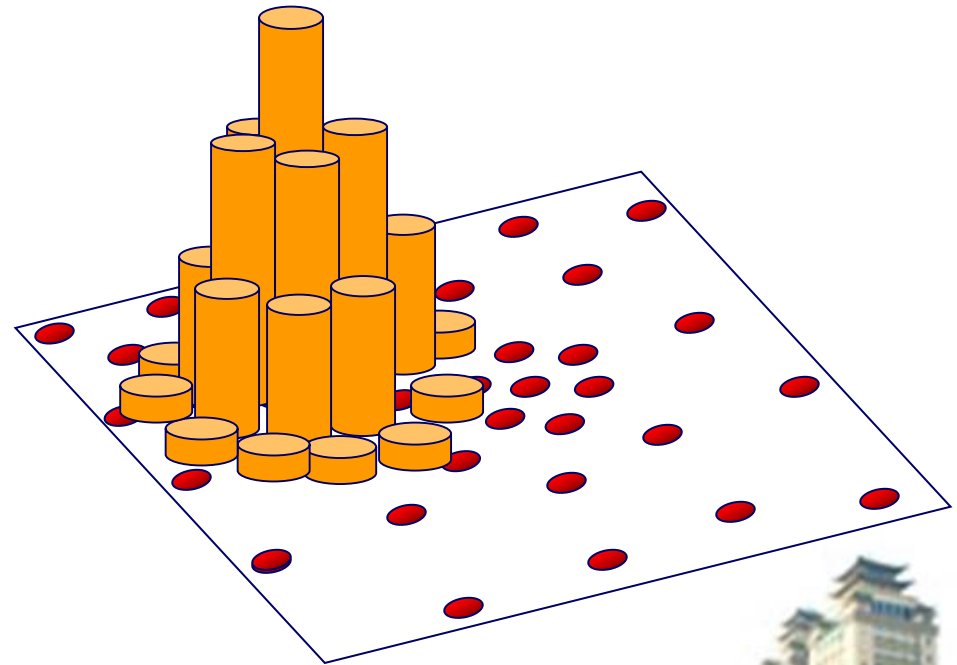




# Non-Parametric Density Estimation



Assumed Underlying PDF



Real Data Samples

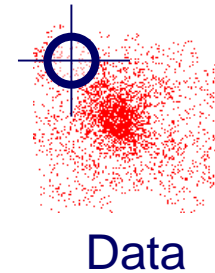


# Kernel Density Estimation

## 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  
 $x_1 \dots x_n$



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



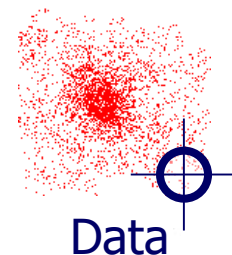


# Kernel Density Estimation

## Various Kernels

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

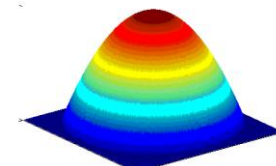
A function of some finite number of data points  
 $\mathbf{x}_1 \dots \mathbf{x}_n$



### Examples:

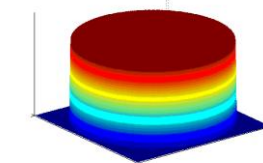
- Epanechnikov Kernel

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



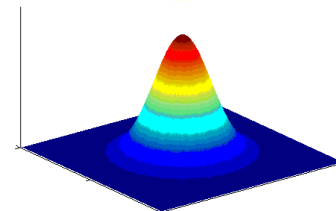
- Uniform Kernel

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



- 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

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

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



# Computational Estimation *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

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

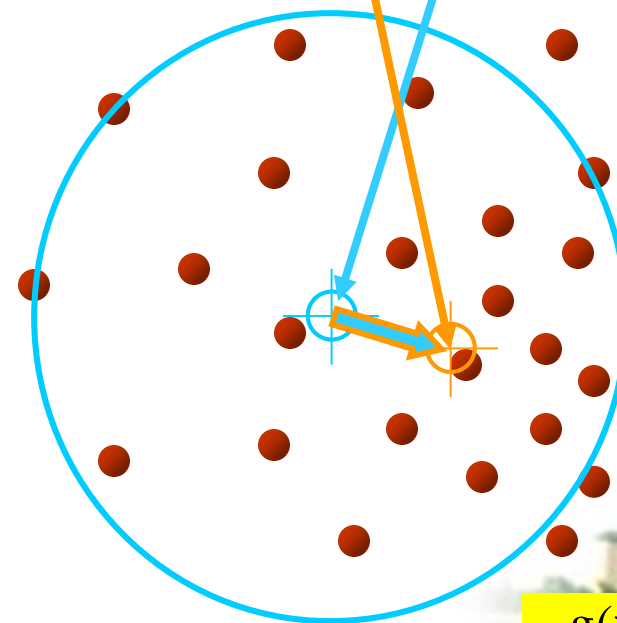
Yet another Kernel  
density estimation !

Simple Mean Shift procedure:

- Compute mean shift vector

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

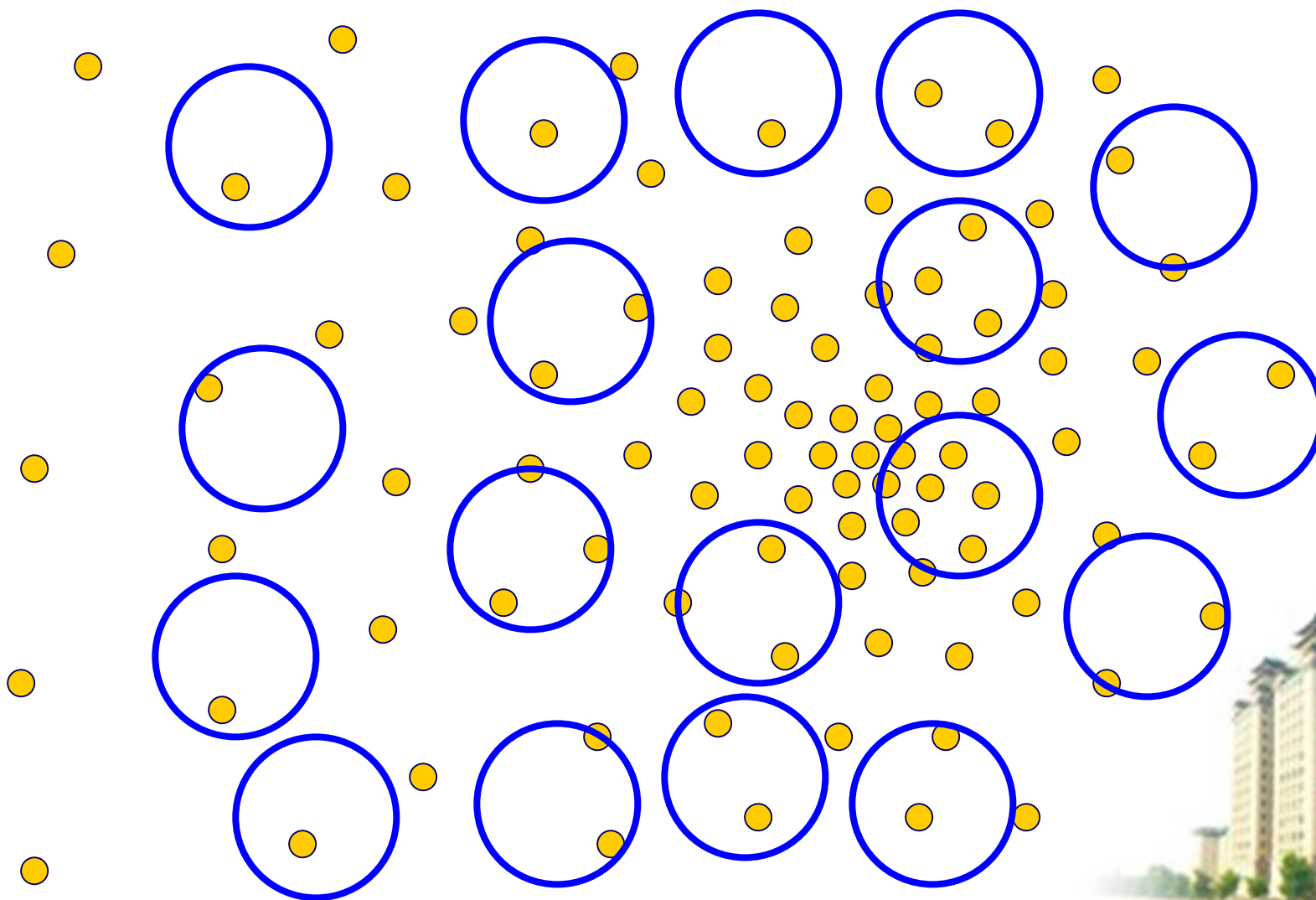
- Translate the Kernel window by  $\mathbf{m}(\mathbf{x})$



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



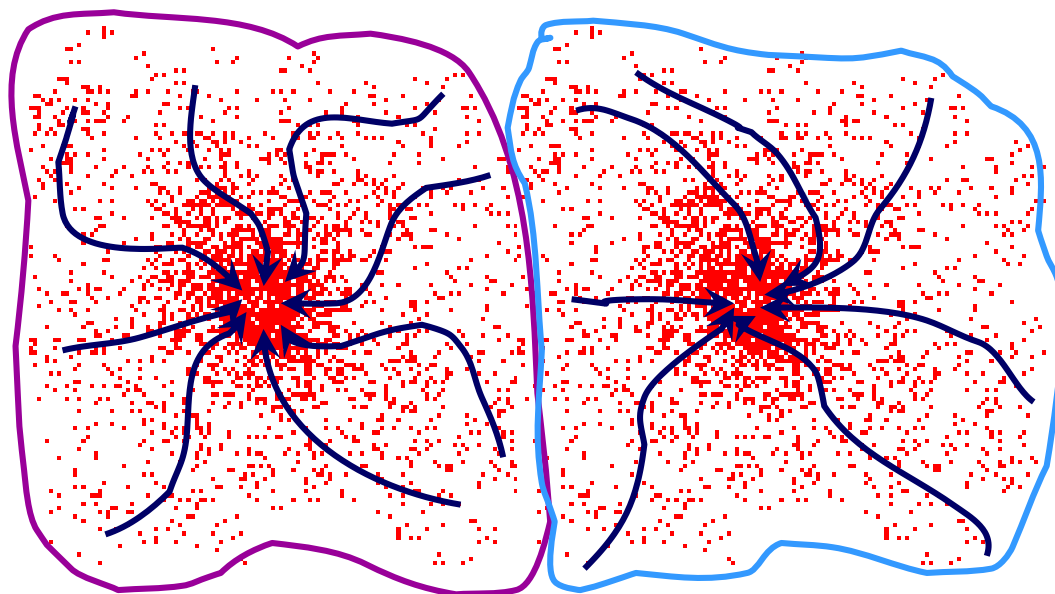
# Real Modality Analysis





# Attraction basin

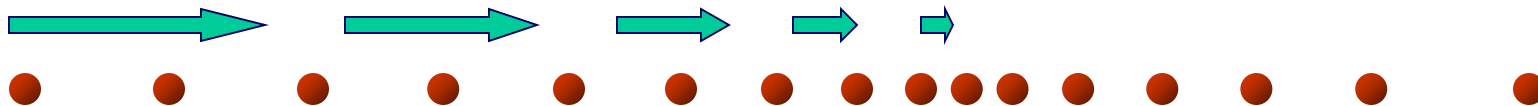
- **Attraction basin:** the region for which all trajectories lead to the same mode
- **Cluster:** all data points in the attraction basin of a mode







# Mean Shift Properties



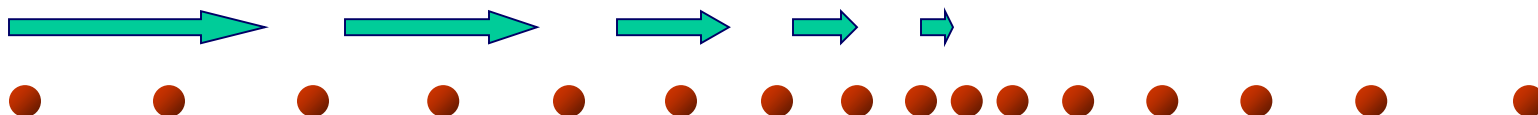
- Automatic convergence speed – the mean shift vector size depends on the gradient itself.
- Near maxima, the steps are small and refined
- Convergence is guaranteed for infinitesimal steps only  $\rightarrow$  infinitely convergent, (therefore set a lower bound)
- For Uniform Kernel (🍷), convergence is achieved in a finite number of steps
- Normal Kernel (📐) exhibits a smooth trajectory, but is slower than Uniform Kernel (🍷).

**Adaptive  
Gradient  
Ascent**





# Mean Shift Strengths & Weaknesses



## Strengths :

- Application independent tool
- Suitable for real data analysis
- Does not assume any prior shape (e.g. elliptical) on data clusters
- Can handle arbitrary feature spaces
- Only ONE parameter to choose
- $h$  (window size) has a physical meaning, unlike K-Means

## Weaknesses :

- The window size (bandwidth selection) is not trivial
- Inappropriate window size can cause modes to be merged, or generate additional “shallow” modes → Use adaptive window size





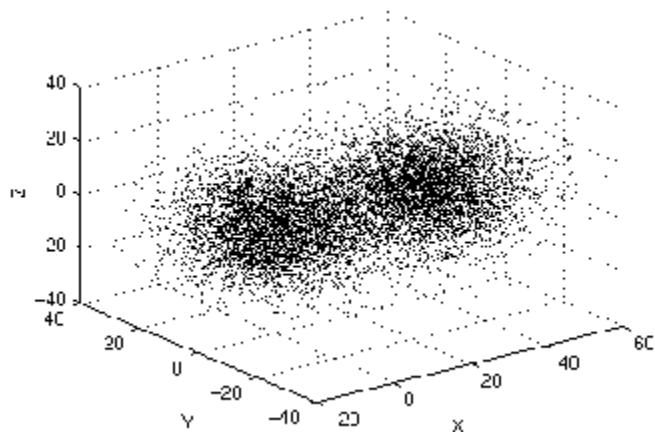
# Mean Shift 的收敛性？





# Clustering

## Synthetic Examples



Simple Modal Structures

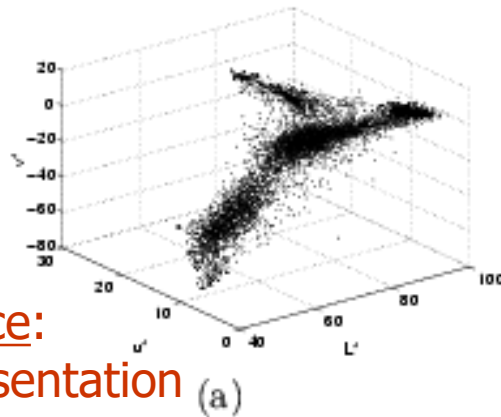
Complex Modal Structures





# Clustering

## Real Example



Initial window  
centers

Modes found

Modes after  
pruning

Final clusters

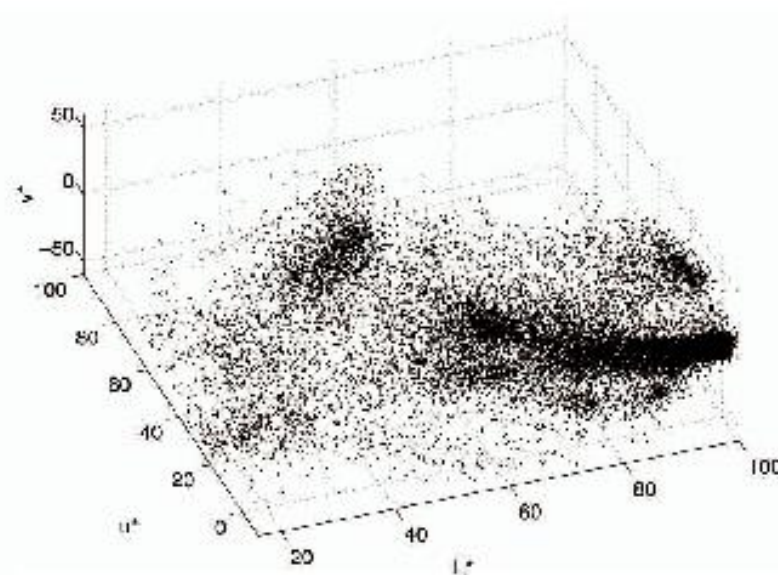






# Clustering

## Real Example



$L^*u^*v$  space representation

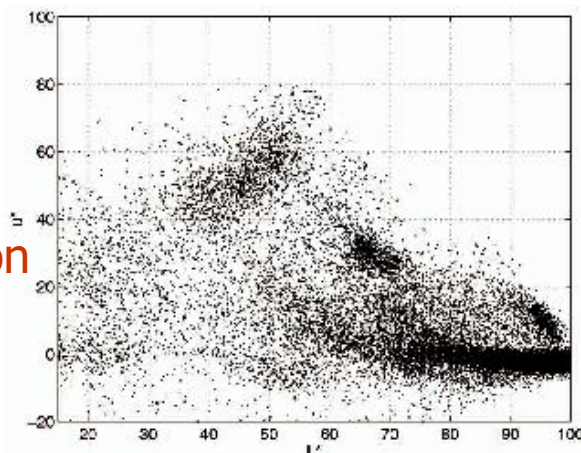




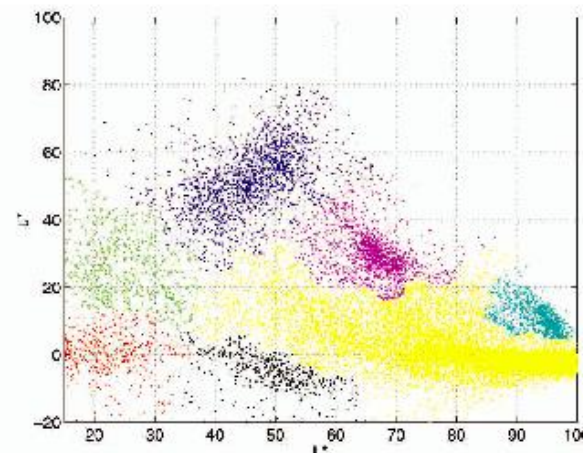
# Clustering

## Real Example

2D ( $L^*u$ )  
space  
representation



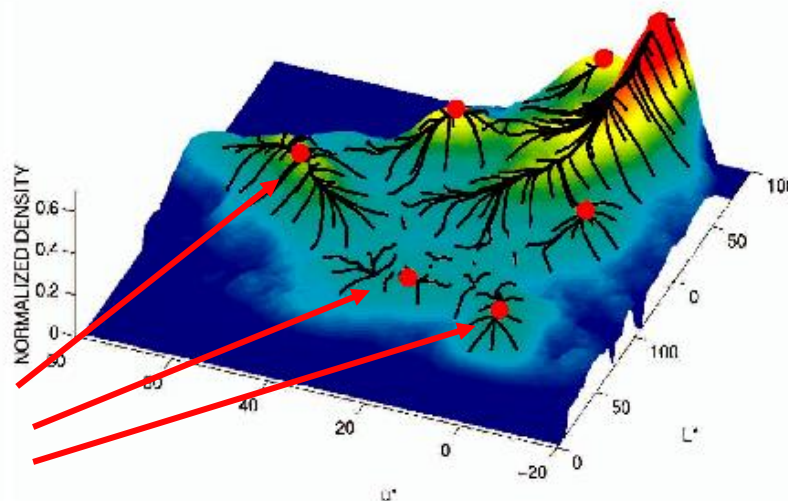
(a)



(b)

Final clusters

Not all trajectories  
in the attraction basin  
reach the same mode



(c)





# Discontinuity Preserving Smoothing

Feature space : Joint domain = spatial coordinates + color space

$$K(\mathbf{x}) = C \cdot k_s \left( \left\| \frac{\mathbf{x}^s}{h_s} \right\| \right) \cdot k_r \left( \left\| \frac{\mathbf{x}^r}{h_r} \right\| \right)$$

Meaning : treat the image as data points in the spatial and gray level domain





# Discontinuity Preserving Smoothing

## Example





# Segmentation

Segment = Cluster, or Cluster of Clusters

## Algorithm:

- Run Filtering (*discontinuity preserving smoothing*)
- Cluster the clusters which are closer than window size





# Things to remember

- Uses of segmentation
  - Efficiency
  - Better features
  - Want the segmented object
- Mean-shift segmentation
  - Good general-purpose segmentation method
  - Generally useful clustering, tracking technique
- Watershed segmentation
  - Good for hierarchical segmentation
  - Use in combination with boundary prediction

