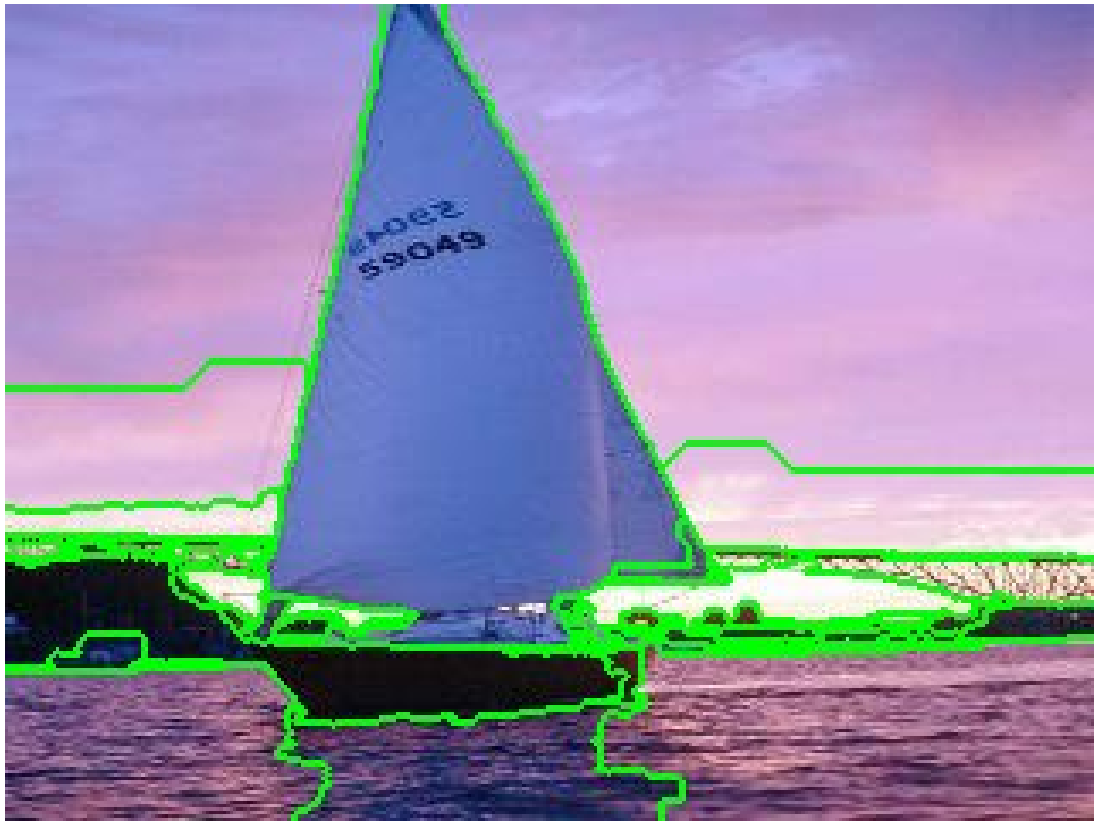
第五讲 目标分割

周文晖

计算机学院

课本：目标分割的概念和目的

•概念：将图像划分成若干具有特征一致性且互不重叠的图像区域的过程。



Aim: to partition an image into a collection of set of pixels

- Meaningful regions (coherent objects)
- Linear structures (line, curve, ...)
- Shapes (circles, ellipses, ...)

Other variants： What is segmentation?

- *Segmentation* = ***partitioning***
  - ✓ Carve dense data set into (disjoint) regions
  - ✓ Divide image based on pixel similarity
  - ✓ Divide spatiotemporal volume based on image similarity (shot detection)
  - ✓ Figure / ground separation (background subtraction)
  - ✓ Regions can be overlapping (layers)

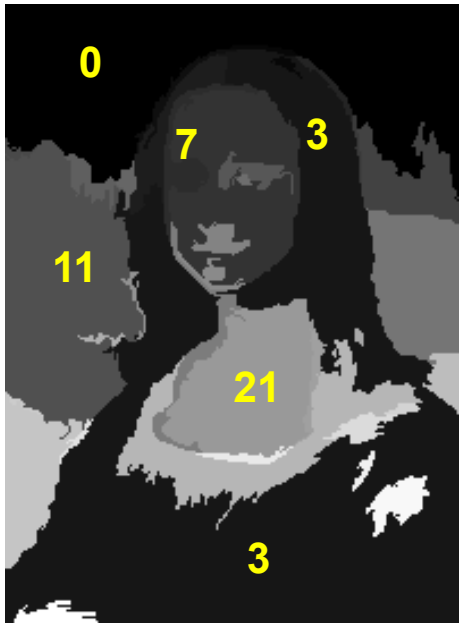
- *Grouping* = ***clustering***
  - ✓ Gather sets of items according to some model
  - ✓ If items are dense, then essentially the same problem as left. (e.g., clustering pixels)
  - ✓ If items are sparse, then problem has a slightly different flavor:
    - Collect tokens that lie on a line (robust line fitting)
    - Collect pixels that share the same fundamental matrix (independent 3D rigid motion)
    - Group 3D surface elements that belong to the same surface

S. Birchfield, Clemson Univ., ECE 847, http://www.ces.clemson.edu/~stb/ece847

Other variants: What is segmentation?

- Segmentation divides an image into groups of pixels
- Pixels are grouped because they share some local property (gray level, color, texture, motion, etc.)

boundaries



labels

pseudocolors

mean colors

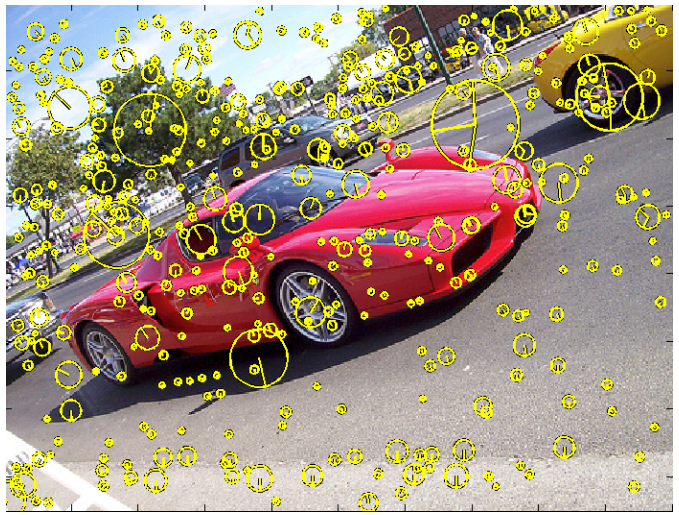
(different ways of displaying the output)

algorithm used: Pedro F. Felzenszwalb and Daniel P. Huttenlocher, Efficient Graph-Based Image Segmentation, IJCV, 59(2), 2004

图像分割的意义

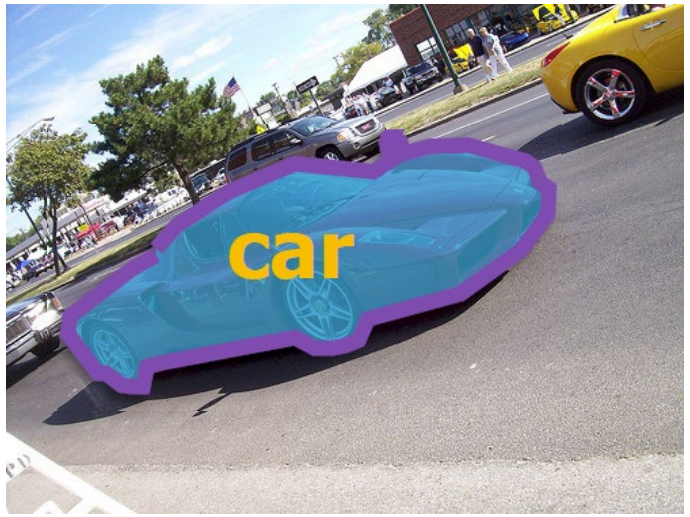
•区域对于图像理解和识别非常重要，往往表征场景中的目标，或部分目标。

•一幅图像可以包含多个目标，每个目标包含多个区域，每个区域对应目标的不同部分。



Low level features

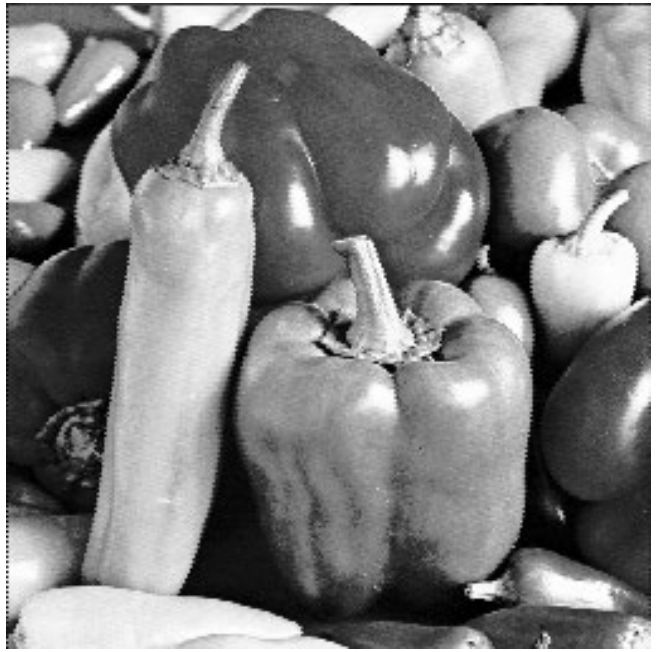
inference

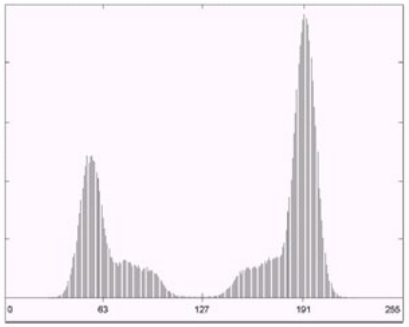


Higher level inference

图像分割举例

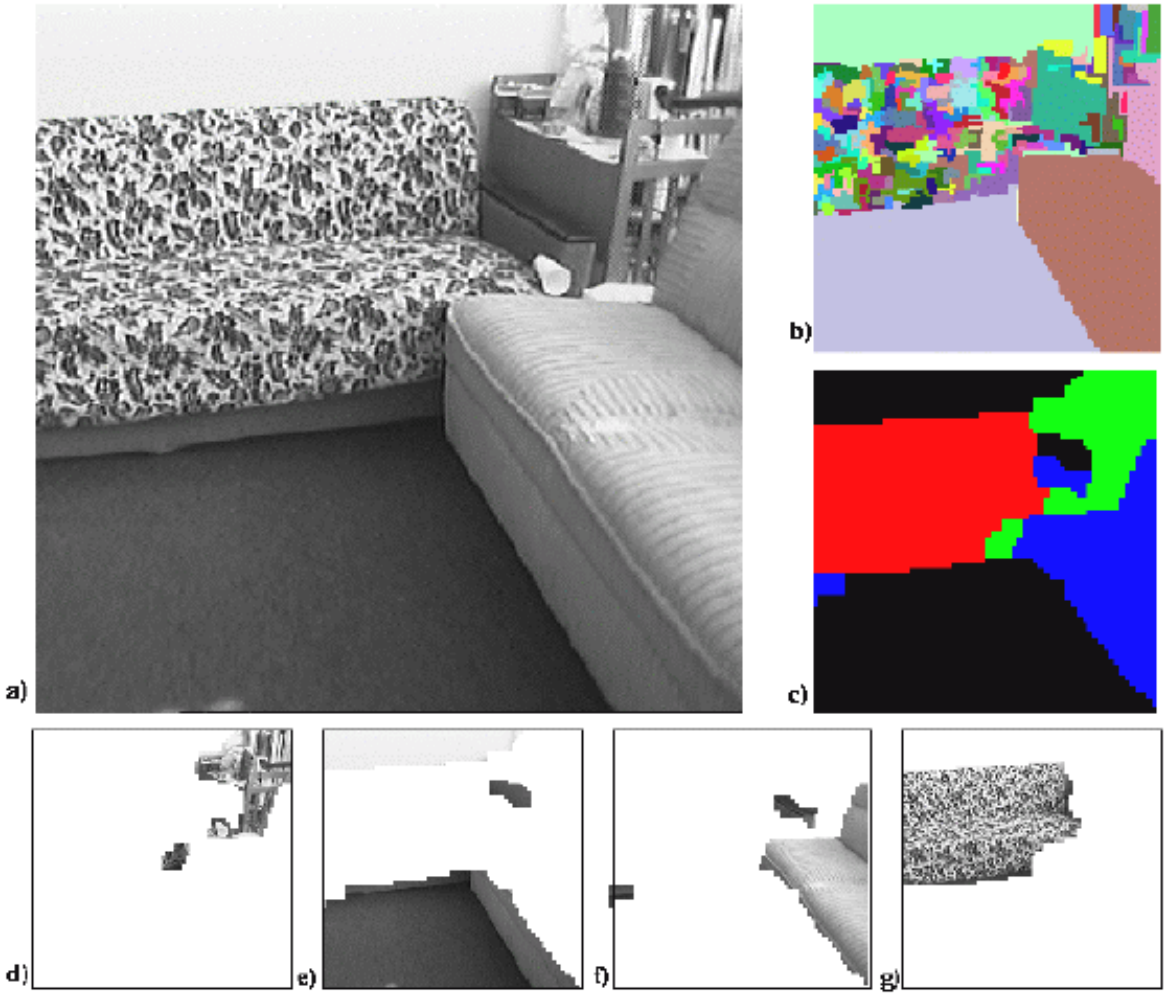
- 基于图像亮度





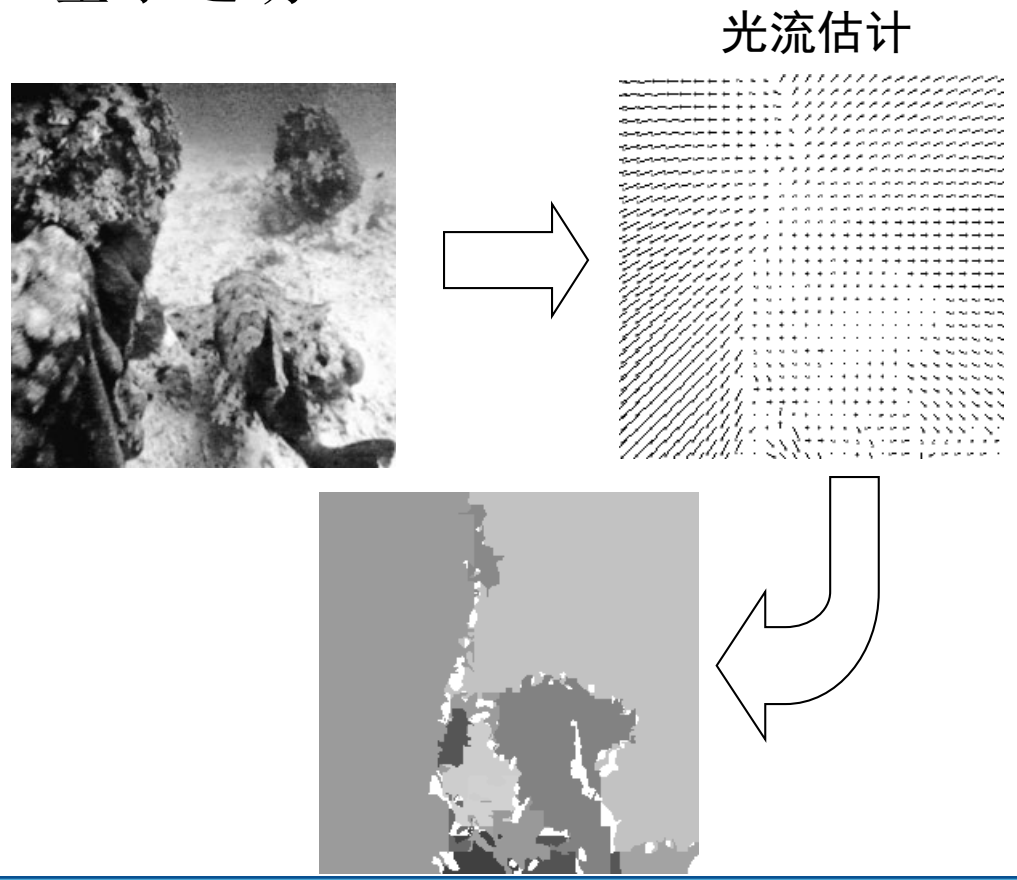
图像分割举例

- 基于纹理



图像分割举例

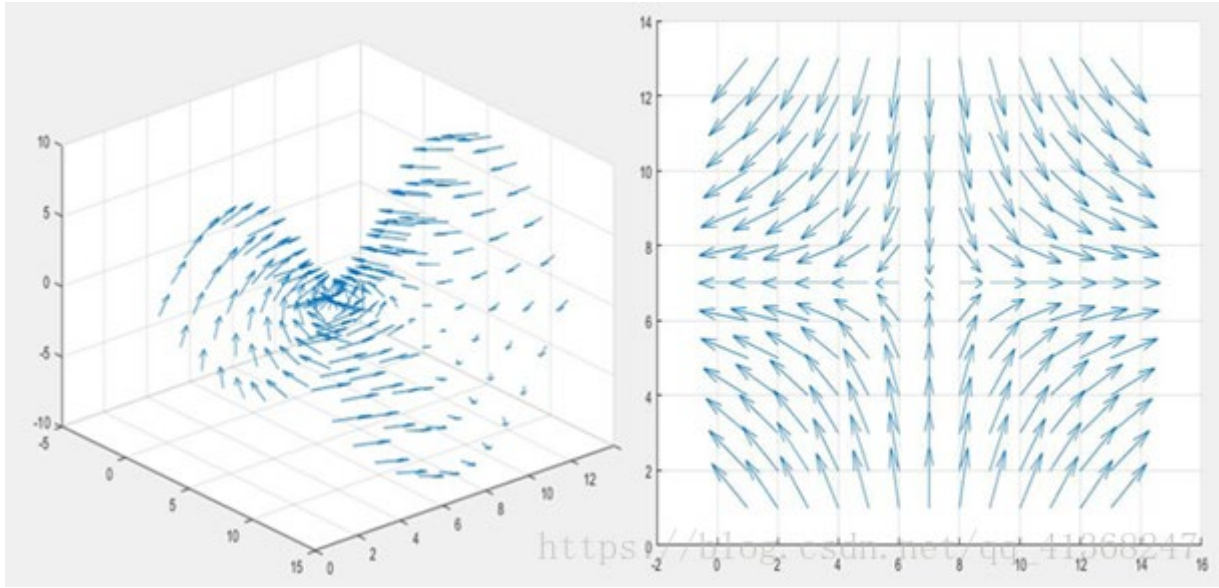
•基于运动



光流的概念是Gibson在1950年首先提出来的；

是空间运动物体在图像平面上的像素运动的瞬时速度；

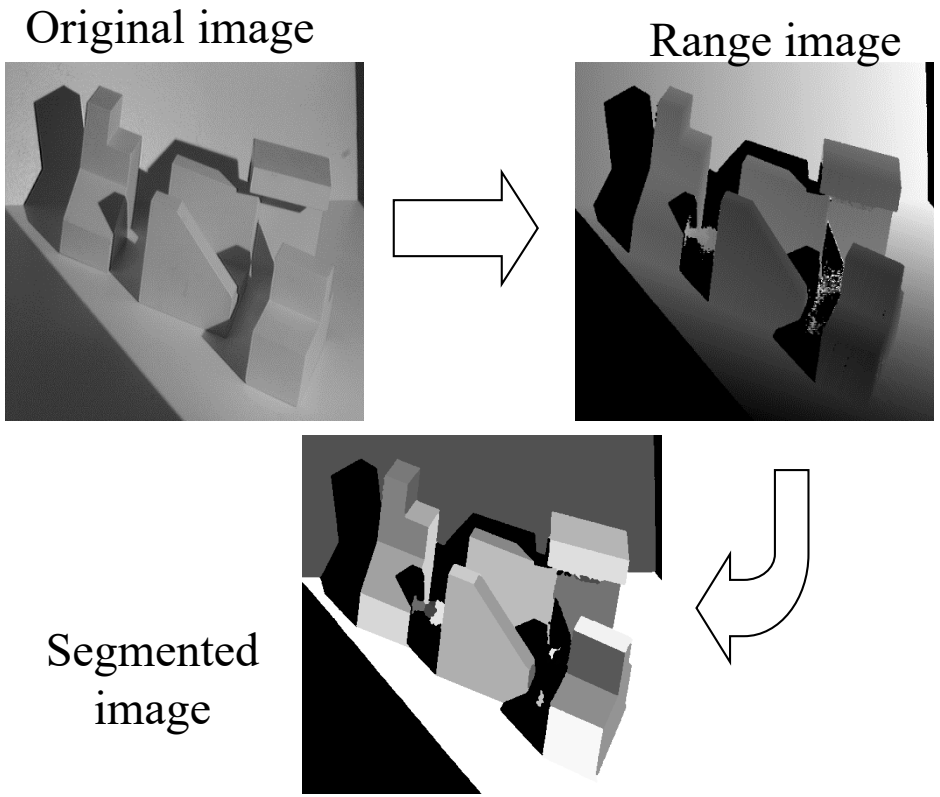
图像序列中像素在时间域上的变化以及相邻帧之间的相关性来找到上一帧跟当前帧之间存在的对应关系，从而计算出相邻帧之间物体的运动信息的一种方法。



三维空间的矢量场及其在二维平面内的投影

图像分割举例

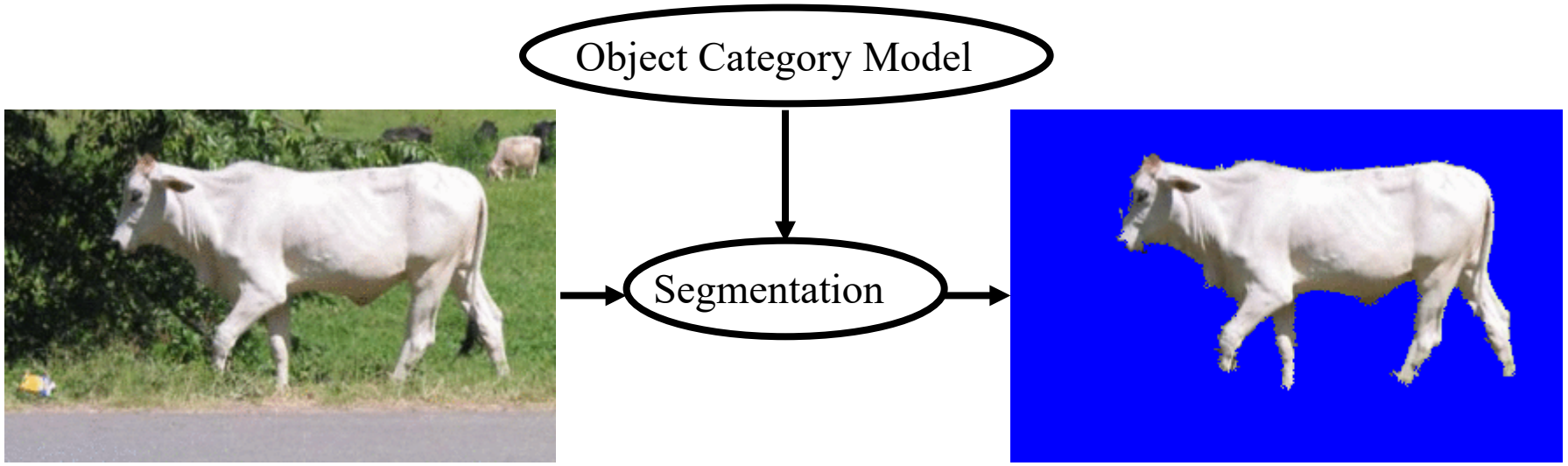
- 基于深度



(a) Original Image (b) Ground Truth (c) Ours

Figure 8. Failure cases on NYUD2 test set

图像分割应用



•理想的图像分割：

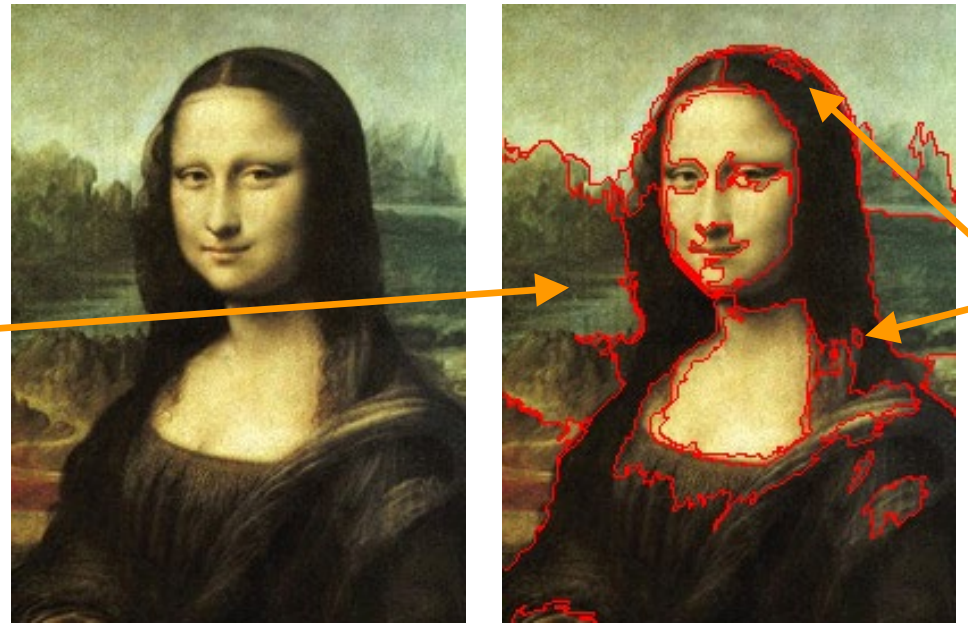
•能够无指导地、自动分割出完整目标。

对传统方法而言，图像分割至今仍是难题

- 图像分割是计算机视觉技术中首要的、重要的关键步骤，但又面临着诸多十分困难。
- 受各类因素（噪声、光照、深度信息丢失等）的影响，准确分割是一个十分困难的问题。
- 至今仍未能提出一种通用（off-the-shelf ）的图像分割模型。

Two errors：

undersegmentation (water should be separated from trees)



oversegmentation (hair should be one group)

图像分割至今仍是难题

- 图像分割的目标定义也往往是模糊的

"What, for example, is an object, and what makes it so special that it should be recoverable as a region in an image? Is a nose an object? Is a head one? ..."

-- D. Marr, 1982

"... [Automatic] *object segmentation for broad domains of general images is not likely to succeed, with a possible exception for sophisticated techniques in very narrow domains.*"

--Smeulders et al, 2000

"*automatic strong segmentation is hard to achieve, ..., the brittleness of strong segmentation is a mostly unsurpassable obstacle when describing the content of images by describing the content of its object.*"

-- Jain et al, 2000

图像分割的方法

- 图像分割通常基于亮度、颜色、纹理、深度或运动。
- 基于边缘的分割方法：
  - 先提取区域边界，再确定边界限定的区域；
- 基于区域的分割方法：
  - 确定每个像素的归属区域，从而完成分割

图像分割的方法

- 图像分割方法分类：

  a）自动分割算法

  –聚类方法 ：**目标应具有相似的亮度、颜色、纹理等**

  –基于边缘的方法 ：**目标轮廓等**

  –区域融合和区域增长：**目标区域应具有连续性和相邻性等**

  –混合优化方法

  b）交互式图像分割算法

  – “Snake” 或 “主动轮廓法”：**指导轮廓收缩的方向**

  – “魔棒” 或 “魔笔”：**提供背景或前景目标的样本信息**

基于聚类的图像分割方法

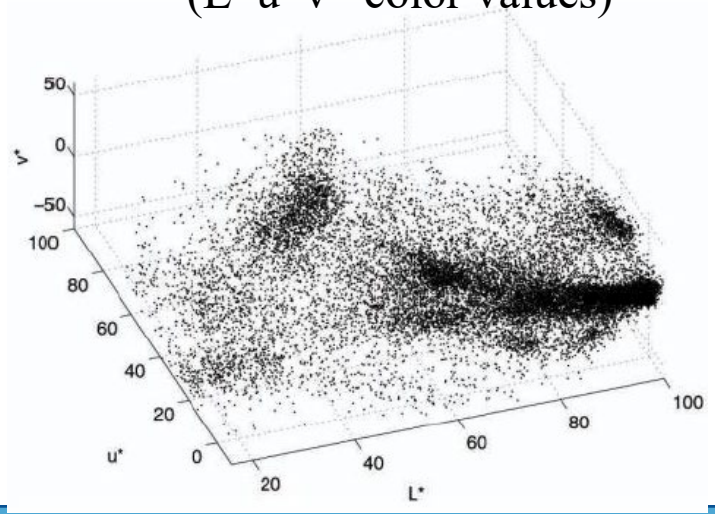
D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis. IEEE Transactions On Pattern Analysis And Machine Intelligence, 2002. 24: p. 603-619.

- 目标应具有相似的亮度、颜色、纹理等

The mean shift algorithm seeks modes or local maxima of density in the feature space

Image space

Feature space
(L*u*v* color values)



基于聚类的图像分割方法

- 目标应具有相似的亮度、颜色、纹理等

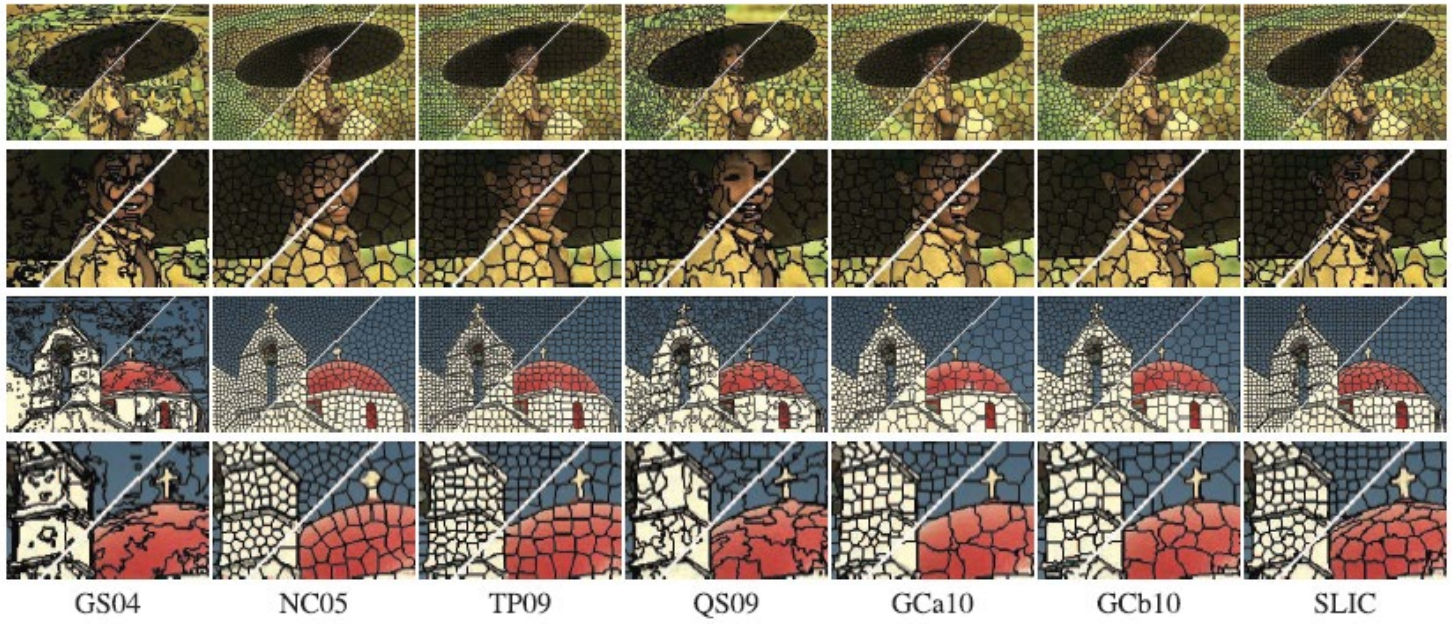
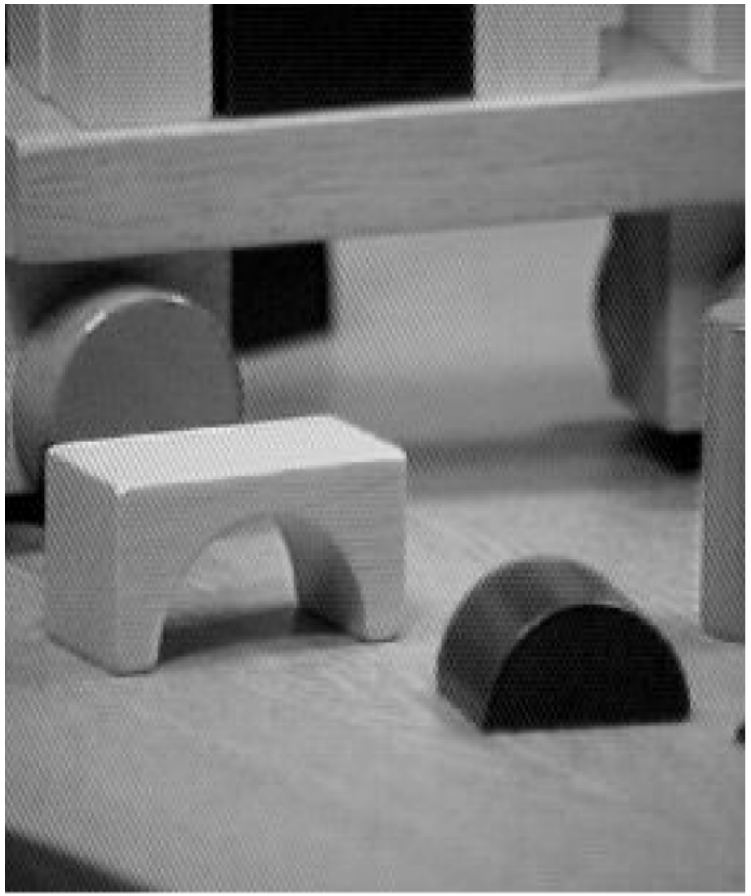


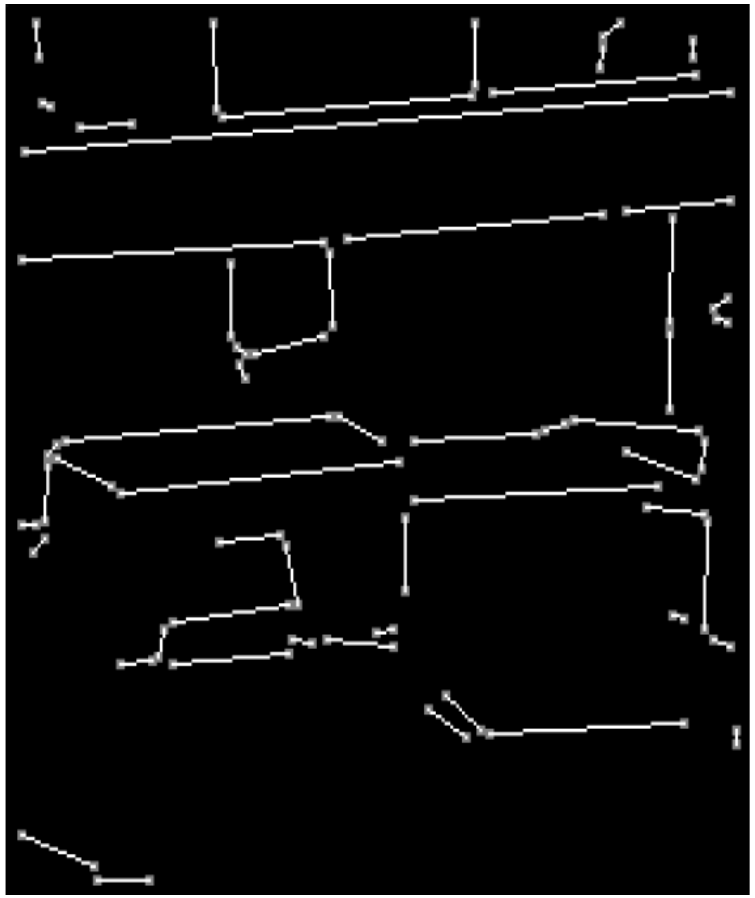
Fig. 7. Visual comparison of superpixels produced by various methods. The average superpixel size in the upper left of each image is 100 pixels and is 300 in the lower right. Alternating rows show each segmented image followed by a detail of the center of each image.

Fig. 1. Images segmented using SLIC into superpixels of size 64, 256, and 1,024 pixels (approximately).

R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, SLIC Superpixels Compared to State-of-the-art Superpixel Methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012. 34(11): p. 2274-2282.

基于轮廓和边界的图像分割方法





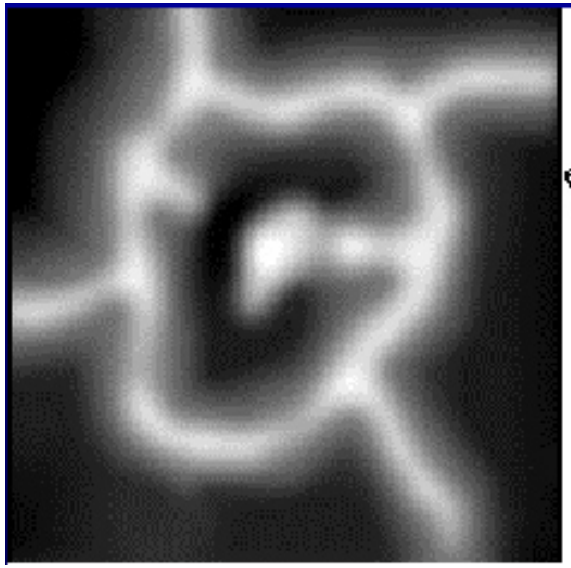
CSE 803 Fall 2008 Stockman

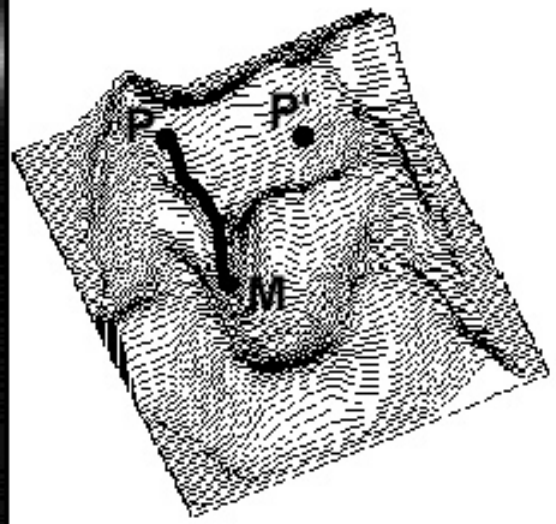
区域融合和区域增长方法

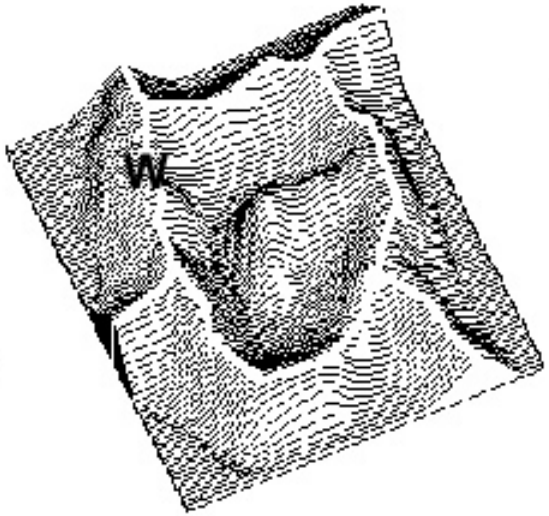
- 目标区域应具有连续性和相邻性等

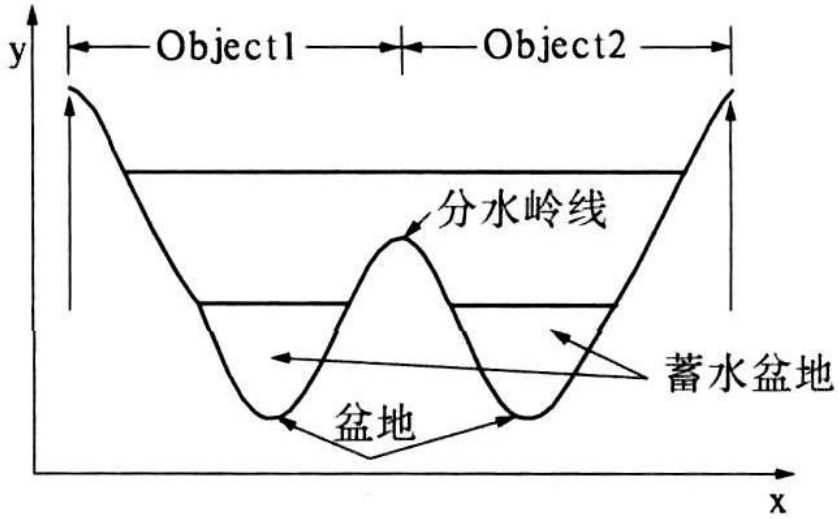
把图象看成3-D地形的表示，即2-D的地基（对应图像空间）加上第3维的高度（对应图像灰度）

图像的梯度图也可以视为3D地形，图像梯度图3D地形中的峰岭对应目标的边界



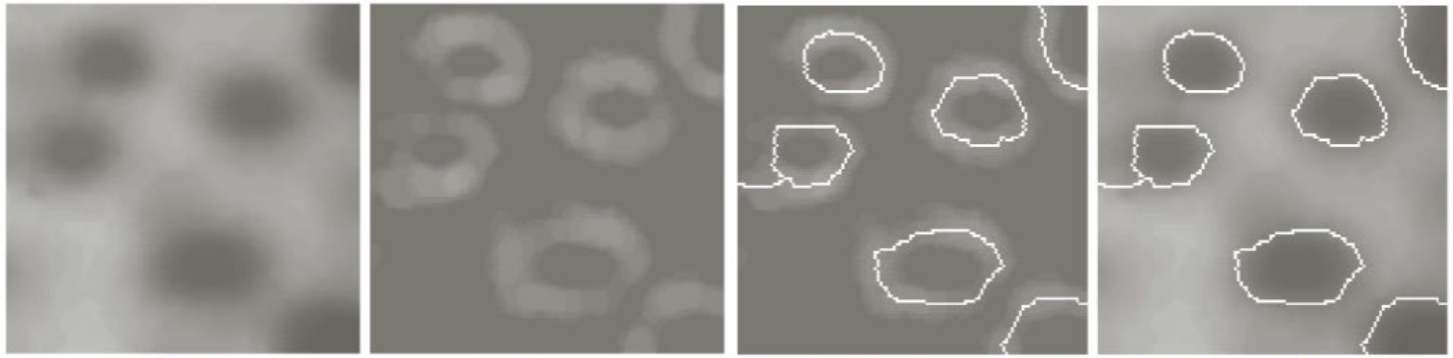




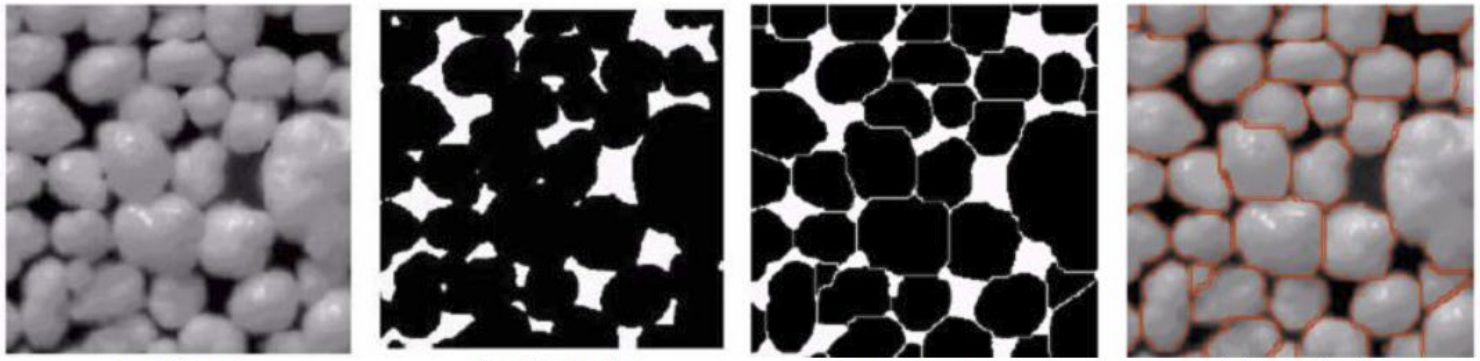


http://en.wikipedia.org/wiki/Watershed_(image_processing)

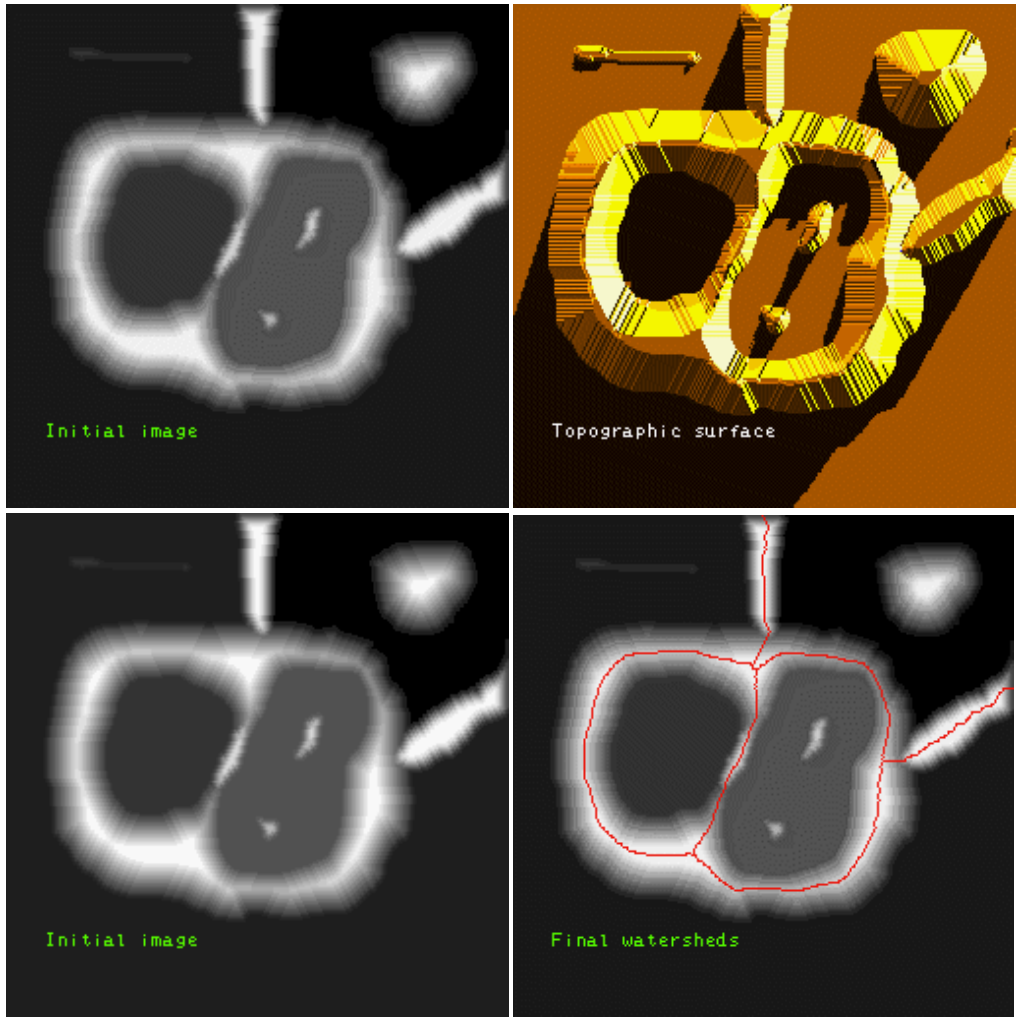
分水岭分割实例



原始图　梯度图　梯度图上的分水岭　原图上的分水岭



原始图　阈值分割　分水岭　叠加轮廓



主动轮廓方法

Given: initial contour (model) near desirable object

Goal: evolve the contour to fit exact object boundary

参数化主动轮廓： Snakes

Snakes: Active contour models

M Kass, A Witkin, D Terzopoulos - International journal of computer vision, 1988 - Springer

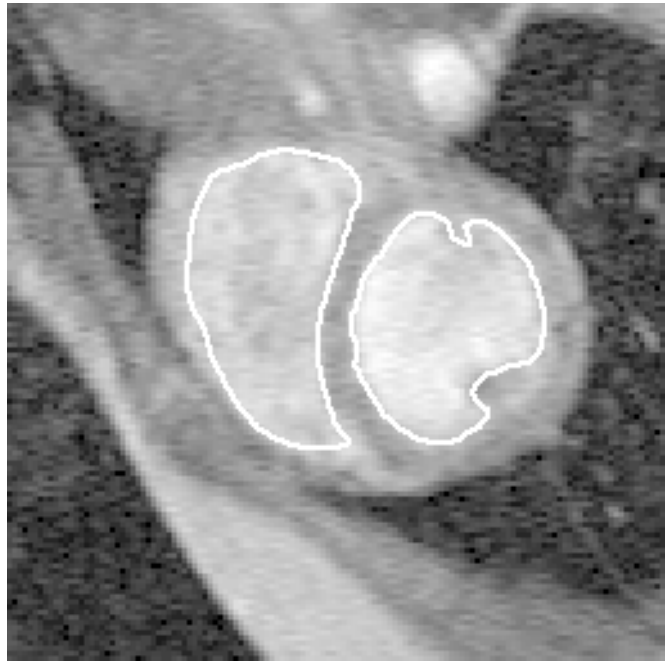
A snake is an energy-minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. Snakes are active contour models: they lock onto nearby edges, localizing them accurately. Scale-space …

☆ 99 被引用次数：21696 相关文章 所有 47 个版本 »

几何主动轮廓： 水平集方法 (level set)

- From Snake to Level Set
- Classical Level Set Model

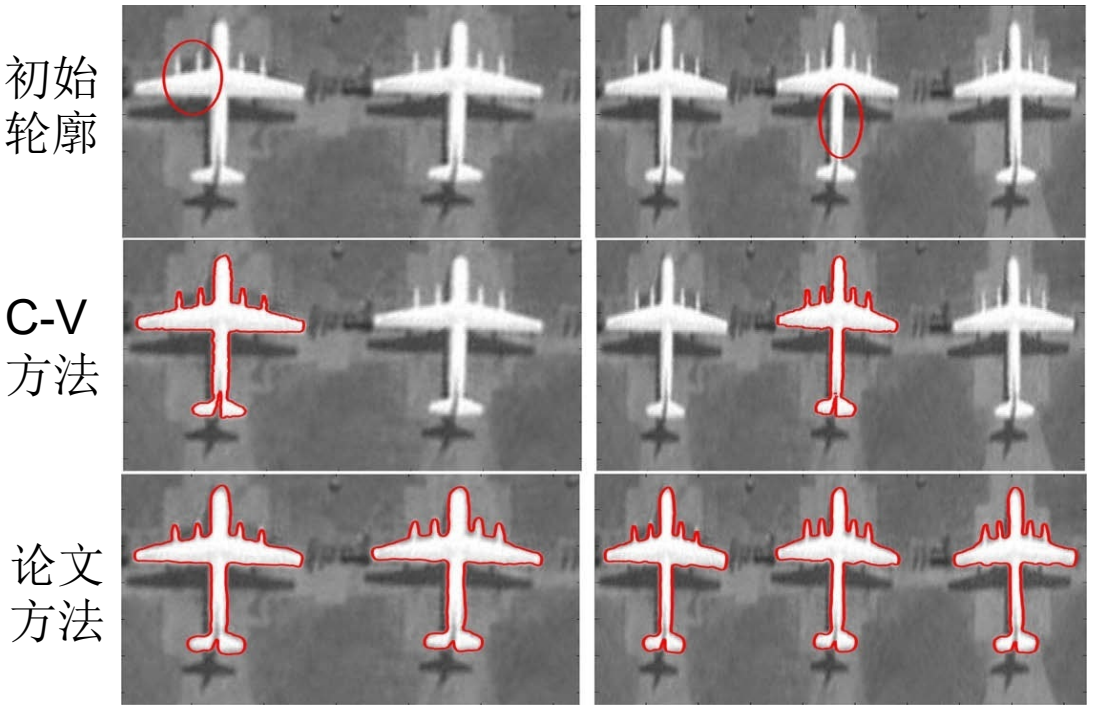
[Kass, Witkin, Terzopoulos 1988]

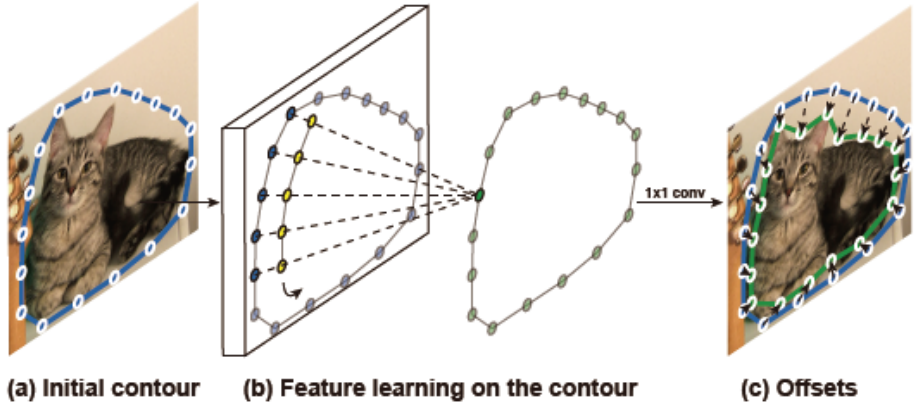


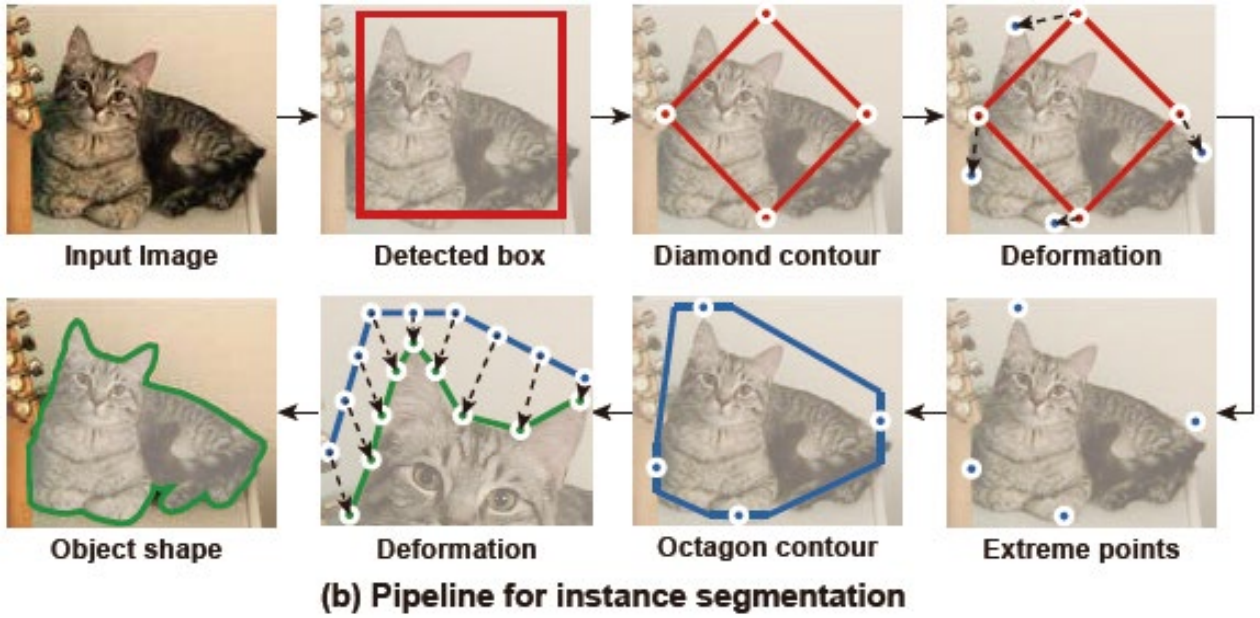
Tracking Heart Ventricles

主动轮廓方法

提供初始轮廓





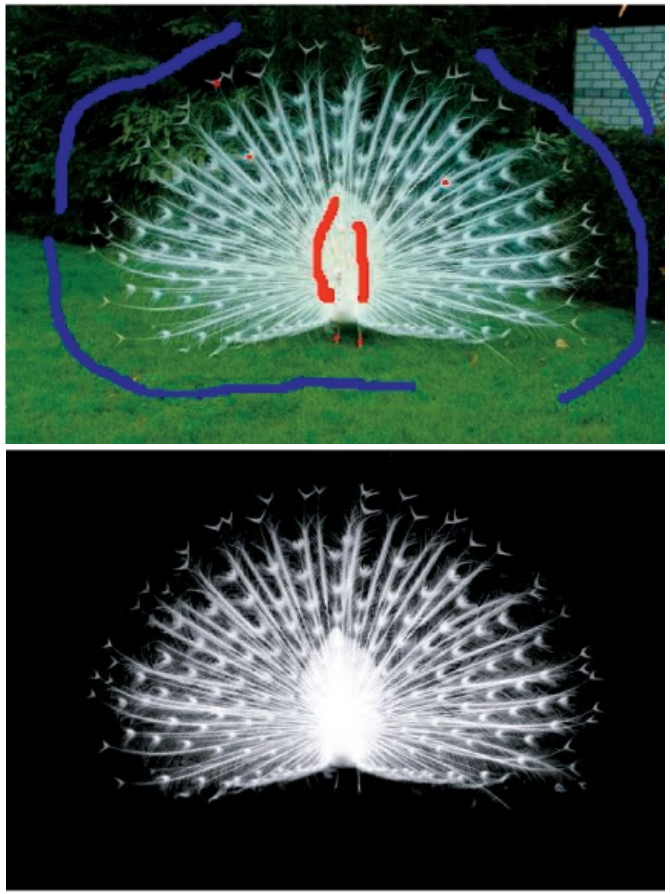


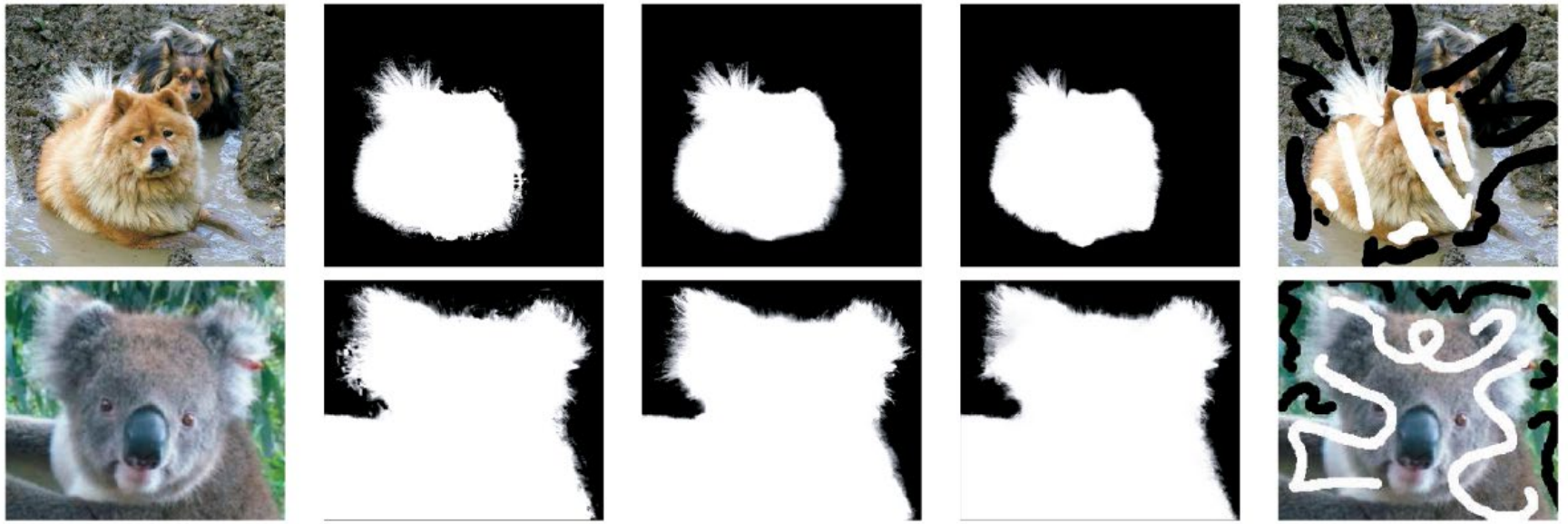
- K. Zhang, L. Zhang and H. Song, et al., Active Contours with Selective Local or Global Segmentation: A New Formulation and Level Set Method. Image And Vision Computing, 2010. 28: p. 668-676.

Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, Xiaowei Zhou. Deep Snake for Real-Time Instance Segmentation. CVPR2020

“魔棒”或“魔笔”

提供背景和前景的部分信息





(a) (b) (c) (d) (e)

Fig. 13. Result on Poisson matting examples. (a) Input image. (b) Bayesian matting (obtained from the Poisson matting paper). (c) Poisson matting (obtained from the Poisson matting paper). (d) Our result. (e) Scribbles.

A. Levin, D. Lischinski and Y. Weiss, A Closed Form Solution to Natural Image Matting. IEEE Transactions On Pattern Analysis And Machine Intelligence, 2008. 30: p. 228-242.

基于边缘的分割方法

轮廓搜索

基于轮廓的图像分割

- 图像中的目标轮廓往往是图像边缘所在。
- 对应于目标轮廓的边缘意味着一个区域的终结和另一个区域的开始，表现出图像局部特征不连续的特性。
- 目标轮廓信息在图像分析和人的视觉中都是十分重要的，是图像识别中提取图像特征的一个重要属性。

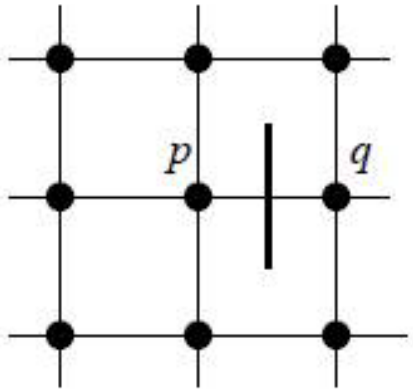
然而，轮廓 ≠ 边缘

轮廓搜索

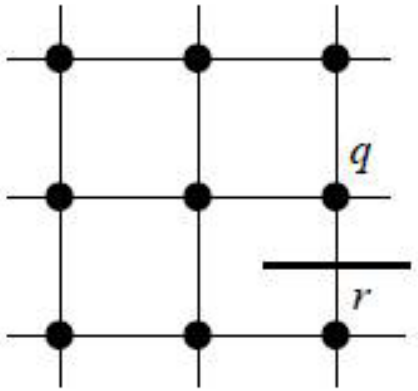
- 基于边界跟踪的目标分割方法：
  - 先提取局部边缘点；
  - 利用上一讲提出的Hough变换，以及后续的链码跟踪；
  - 先检测局部边缘点，再连接边缘点构成目标边界。
- 基于轮廓搜索的目标分割方法
  - 不同于边界跟踪，轮廓搜索是将局部边缘点检测和边界连接同时完成。
  - 是全局搜索策略。

轮廓搜索的基本思想

- 轮廓搜索的基本思想：沿大梯度方向搜索和延伸。
- 最佳轮廓应是该轮廓所经过的所有像素梯度和最大。目标边界或轮廓是由一系列边缘元素构成。
- 将轮廓搜索转为状态图中搜索代价最小的路径，这是一种全局最优求解，而非局部最优。

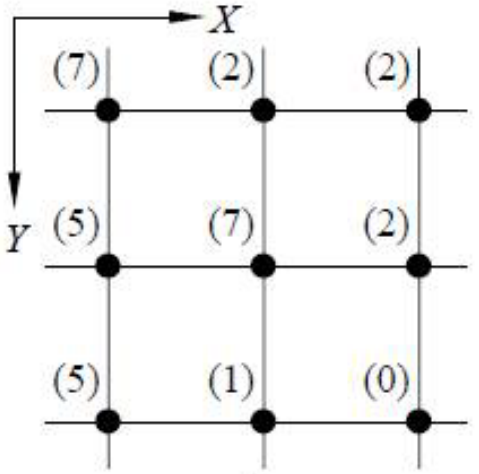


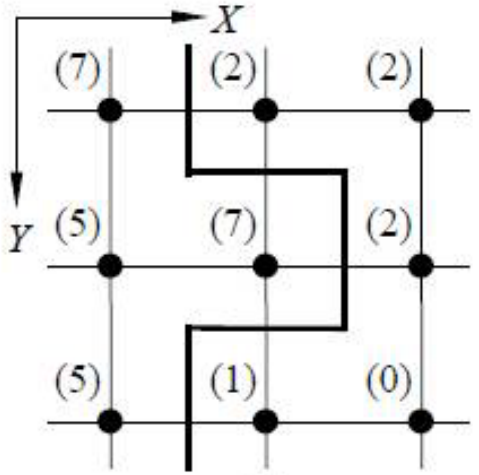
(a)



(b)

边缘元素





轮廓搜索示例

基于图的轮廓搜索思想

- 一个图可表示为： $\mathcal{G}=\langle \mathcal{V},\mathcal{E}\rangle$
- 其中节点集$\mathcal{V}$为像素构成
- 边集合$\mathcal{E}$代表像素间邻域连接
- 边的代价为相邻像素梯度的函数

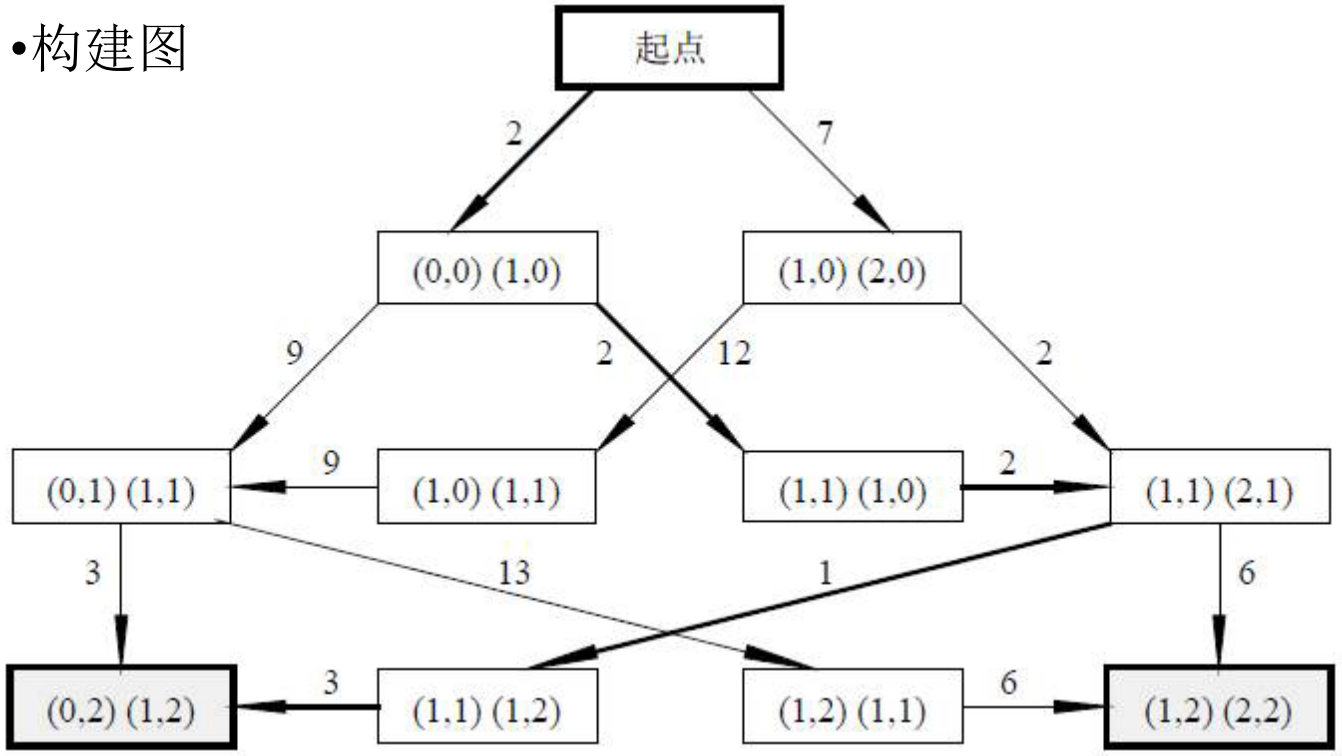
$$c(p,q)=H-[f(p)-f(q)]$$

Why?

- 一次图割的代价为：

$$C=\sum_{i=2}^{K}c(n_{i-1},n_i)$$

- 构建图



- 轮廓搜索转换为最大流最小割问题。

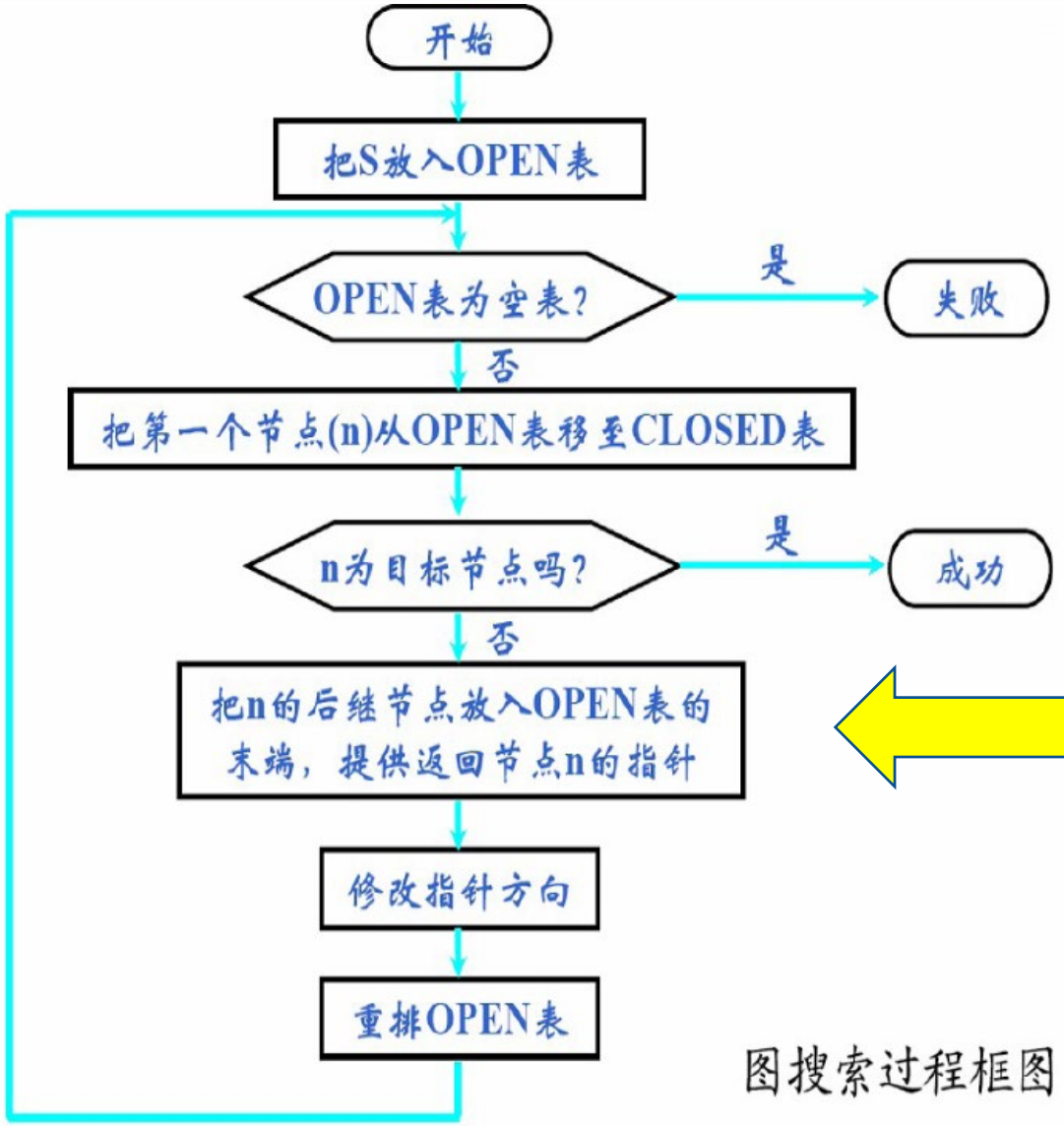
图搜索策略

- **图搜索方法：**
  - 广度优先搜索　**特点？**
  - 深度优先搜索　**特点？**
  - 启发式：启发函数，代价函数　**特点？与广度和深度优先搜索的联系？**
  - A 算法和A*算法　**特点？**

图搜索的实现

- 图搜索的实现采用两张存放节点的表组成：
  - OPEN表：用于存放已经生成,且已被估计或评价过但尚未产生后继节点的那些节点,也称未考察节点。
  - CLOSED表:用于存放已经生成且已被考察过节点。
- 扩展一个节点生成出该节点的所有后继节点，并给出它们之间的耗散值。该过程称为“扩展一个节点”。

图搜索的实现



图搜索过程框图

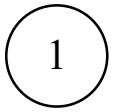
Open表中新增节点的排序方式决定不同的搜索策略。

图搜索中不同策略的实现

- Open表中新增节点的排序方式对应不同搜索策略：
  - 广度优先搜索 — **新增节点加入Open表末端**
  - 深度优先搜索 — **新增节点加入Open表前端**
  - 启发式：启发函数，代价函数 — **新增节点排序后加入Open表前端或末端**

    新增节点加入后，Open表中所有节点重新排序

广度优先搜索策略



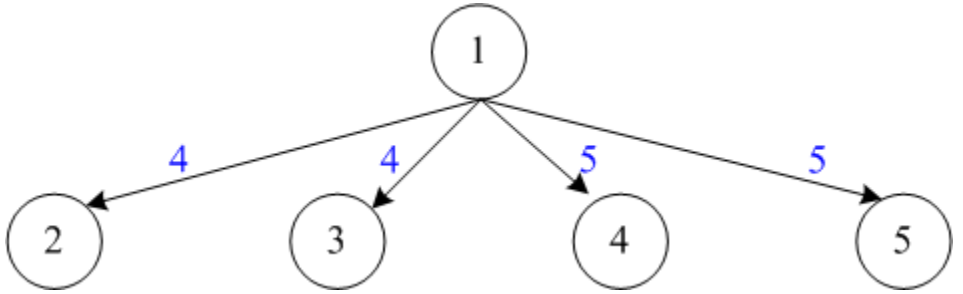
第一步：Open: 1

Closed: 空

第二步：

Open: **2, 3, 4, 5**

Closed: **1**

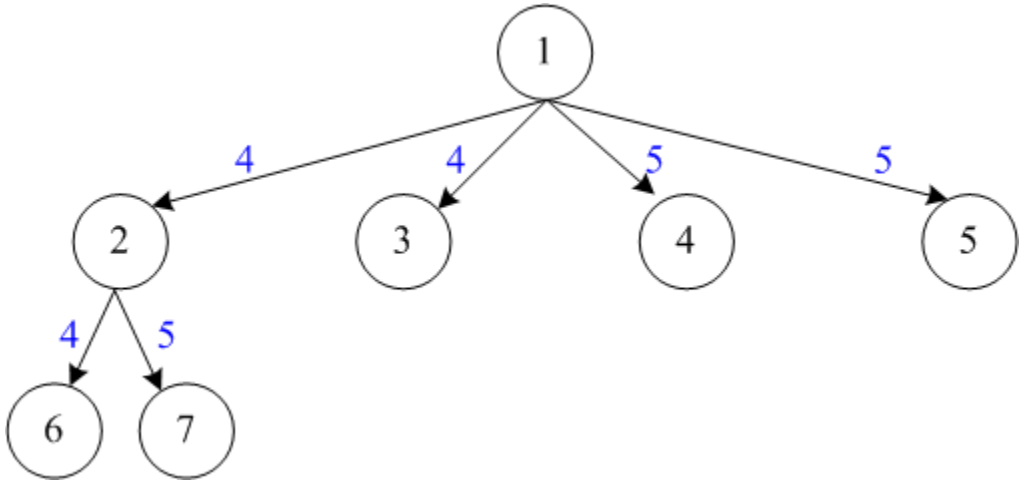


第三步：

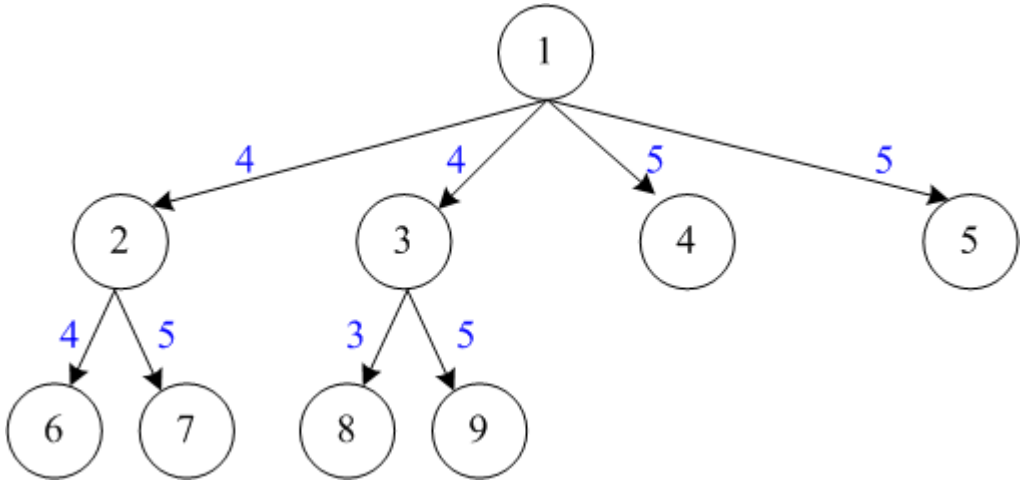
Open：3, 4, 5, **6, 7**

Closed: 1, **2**

排列到Open表最后



广度优先搜索策略



第四步：

Open： 4, 5, 6, 7, **8, 9**

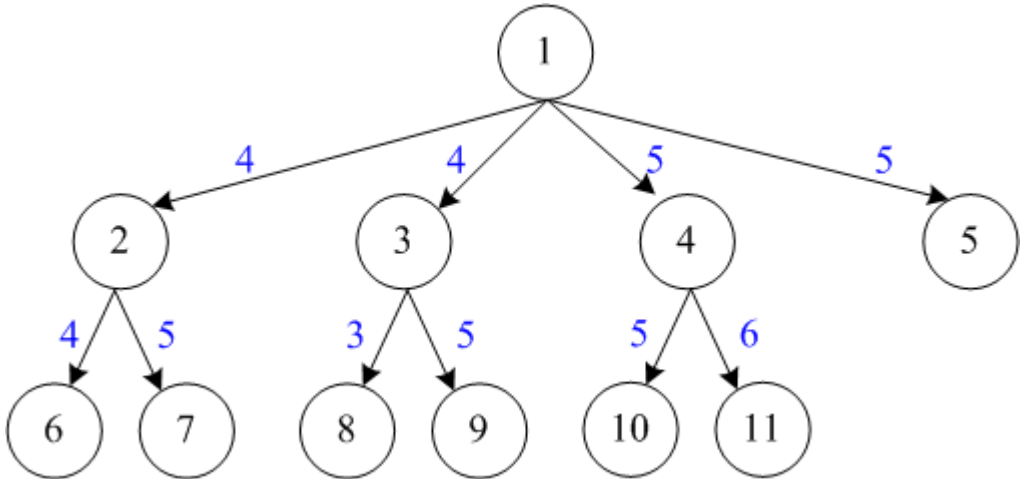
Closed: 1, 2, **3**

排列到Open表最后

第五步：

Open: 5,6,7,8,9,**10,11**

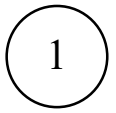
Closed: 1, 2, 3, **4**



深度优先搜索策略

第一步：Open: 1

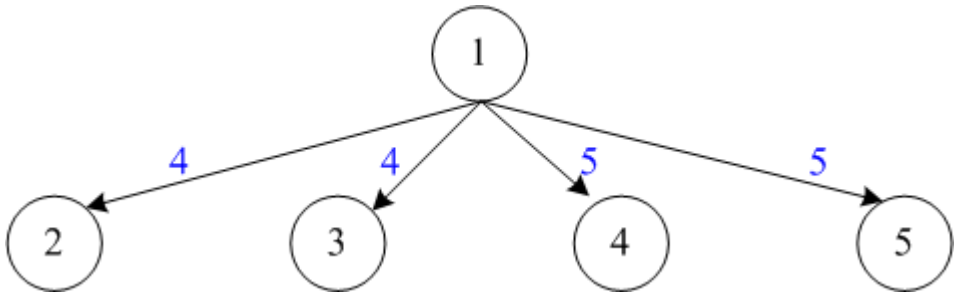
Closed: 空



第二步：

Open: **2, 3, 4, 5**

Closed: **1**

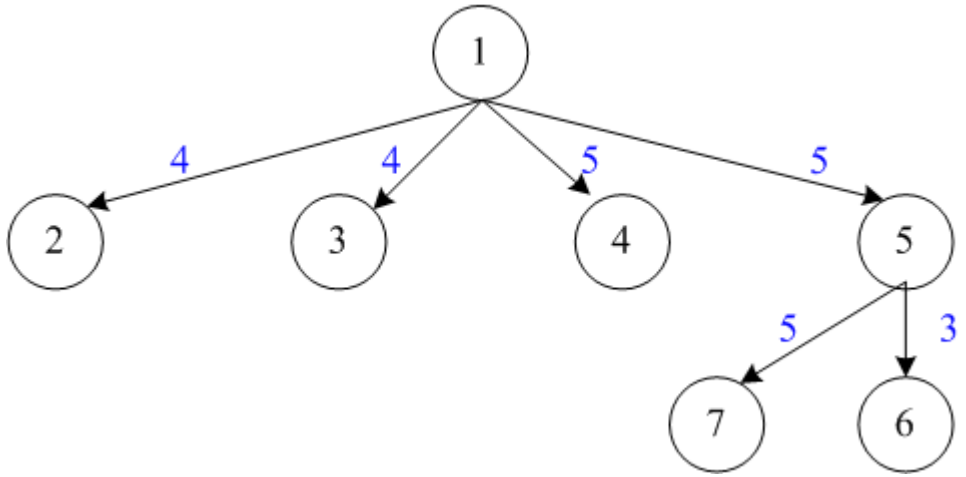


排列到Open表之前

第三步：

Open： **6, 7**, 3, 4, 5,

Closed: 1, **2**



深度优先搜索策略

第四步：

Open: **8**,7,3,4, 5

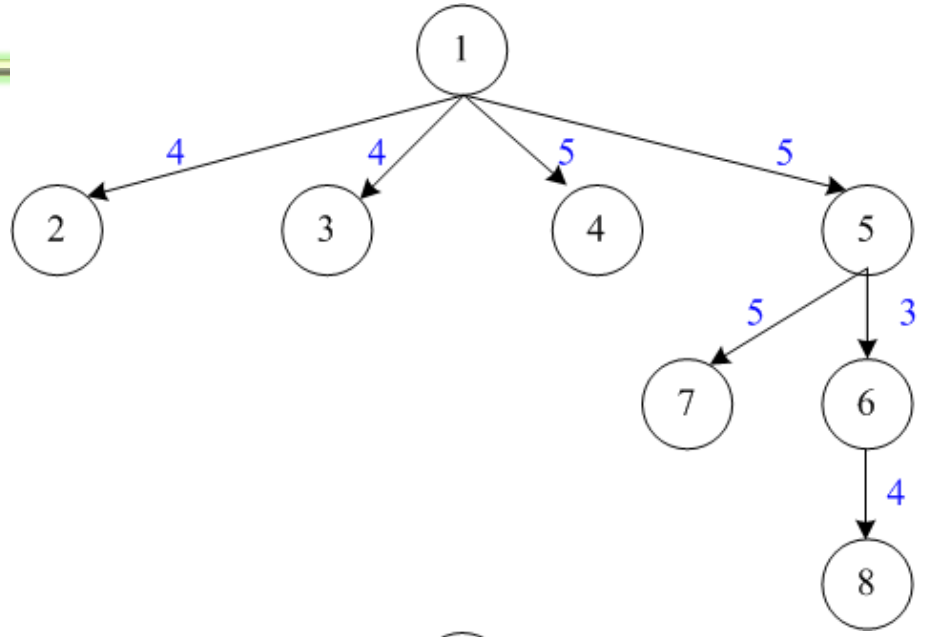
Closed: 1, 2, **6**

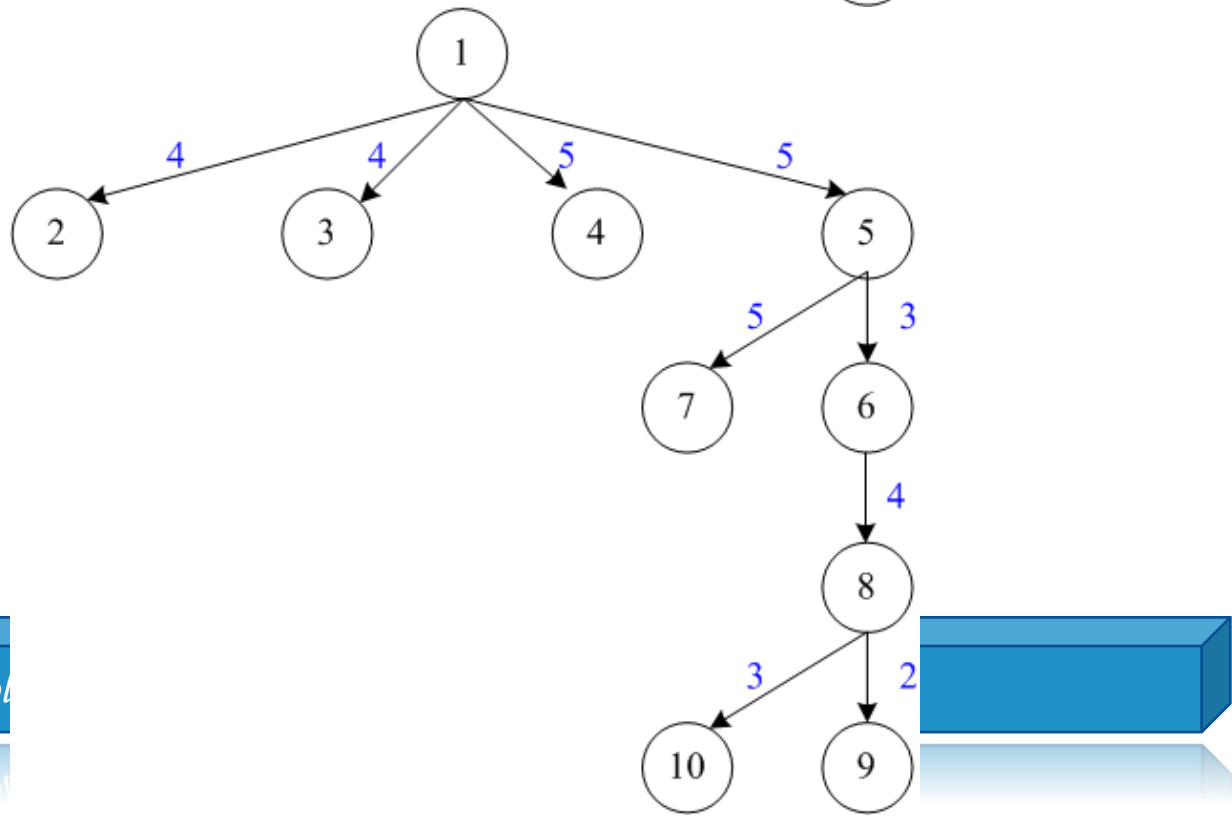
排列到Open表之前

第五步：

Open: **9,10,**7,3,4,5

Closed: 1, 2, 6, **8**

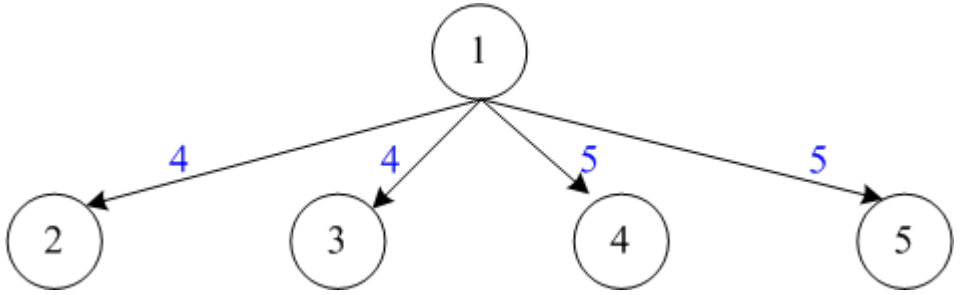




全局择优搜索策略

第二步：

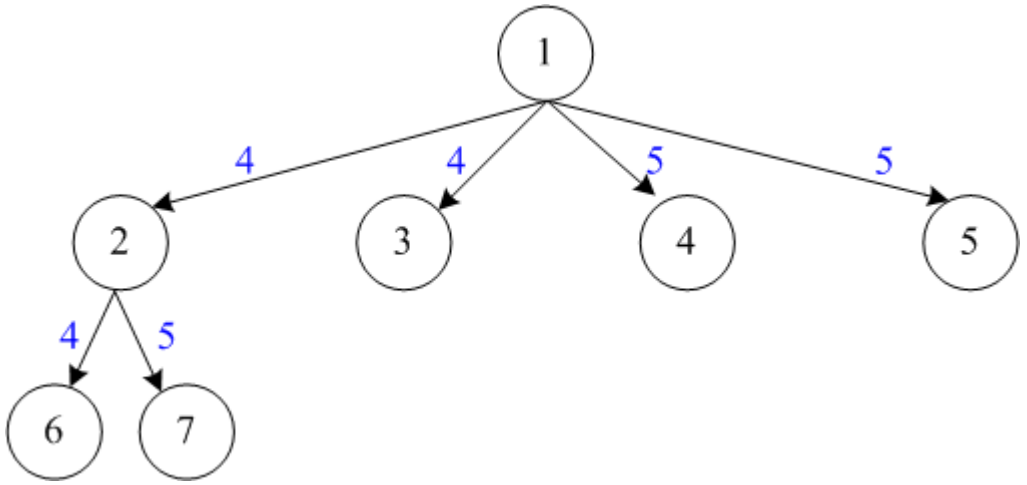
Open: 2,3,4,5



第三步：

Open: **6,3,7,4,5**

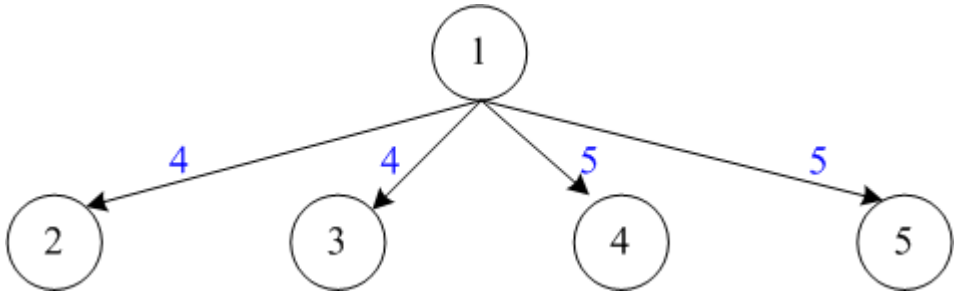
Open表中所有节点排序



局部择优搜索策略

第二步：

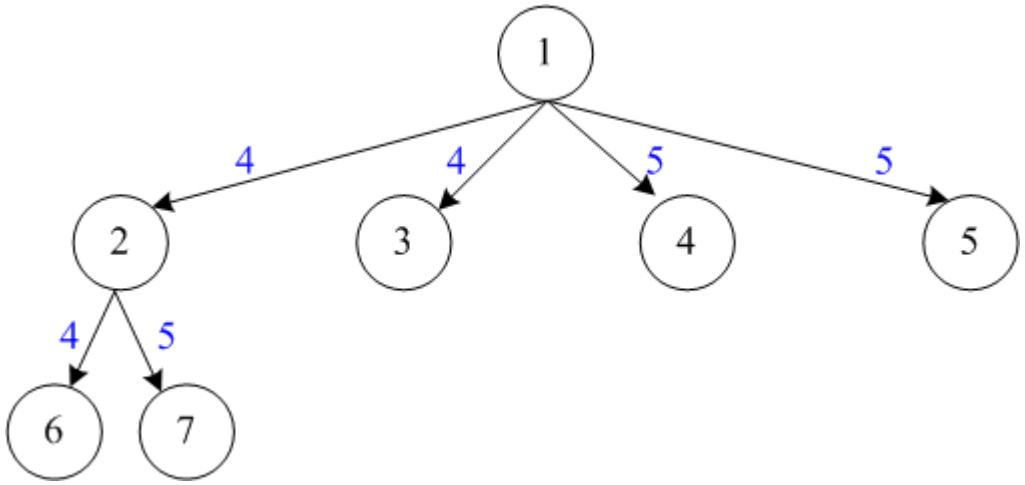
Open: 2,3,4,5



第三步：

Open: **6,7,**3,4,5

Open表中新增节点排序，并放在前面



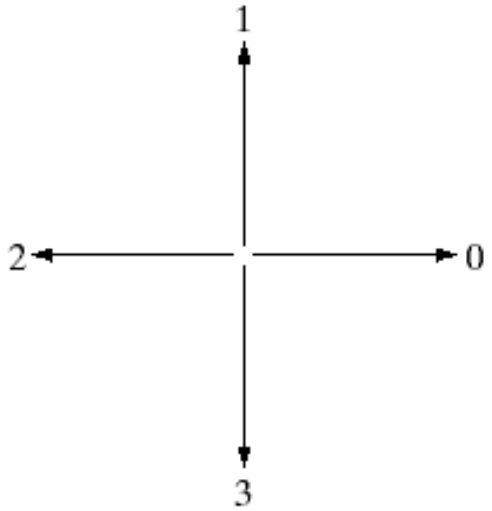
基于边界跟踪的分割方法

边界表示和链码跟踪

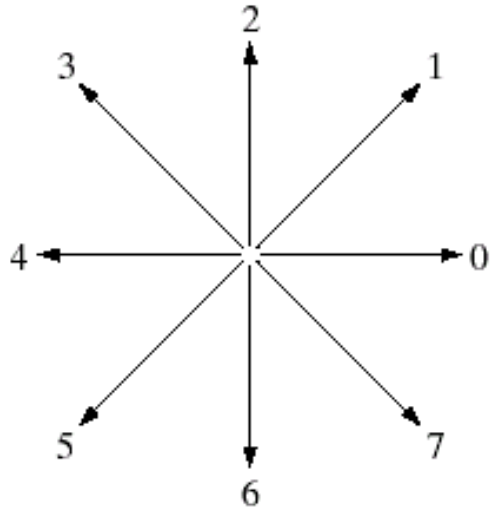
Why we focus on a boundary?

- The boundary is a good representation of an object shape and also requires a few memory.
- **Chain codes:** represent an object boundary by a connected sequence of straight line segments of specified length and direction.

4-方向和8-方向链码的共同特点是直线段的长度固定，方向数有限。



4-directional chain code



8-directional chain code

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Chain Codes

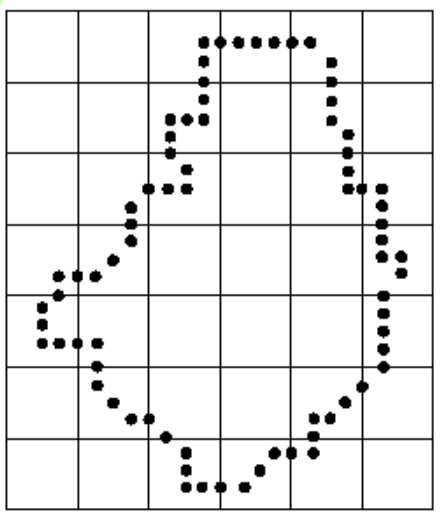
•在**链码表达**中，只有边界的起点需用（绝对）坐标表示，其余点都只用接续方向来代表偏移量。

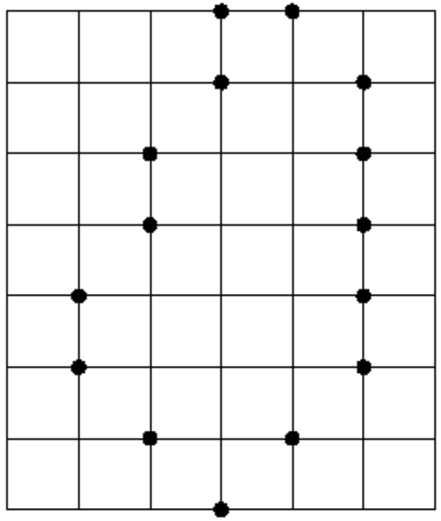
•Problem of a chain code:

a chain code sequence depends on a starting point.

解决方法：链码归一化

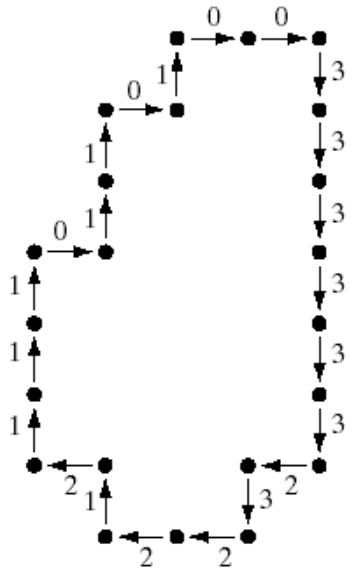
Object boundary (resampling)

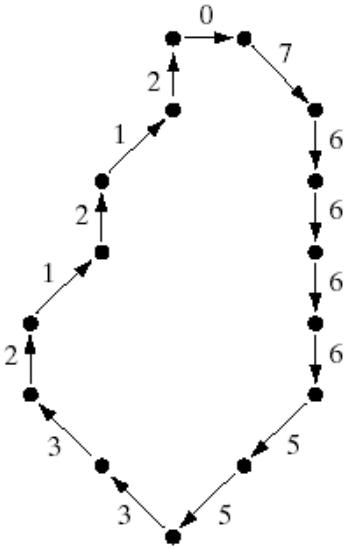




Boundary vertices

4-directional chain code

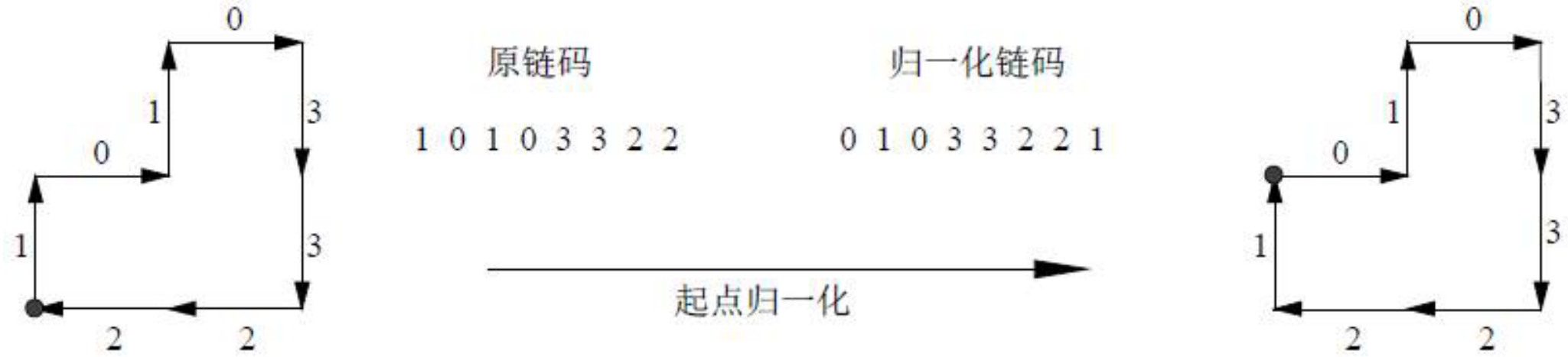




8-directional chain code

链码归一化——起点归一化

- 链码起点归一化
  - 给定一个从任意点开始而产生的链码，把它看作一个由各个方向数构成的**自然数**。
  - 将这些方向数依一个方向循环以使它们所构成的**自然数**的值最小；
  - 将这样转换后链码起点作为归一化链码的起点

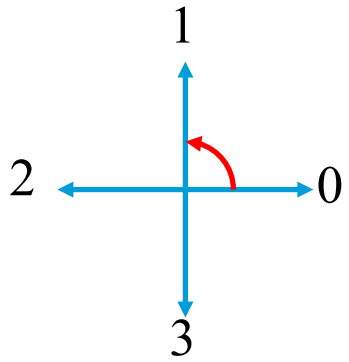


链码归一化——旋转归一化

- 链码旋转归一化
  - 利用链码的**一阶差分**来重新构造一个序列：一个表示原链码各段之间方向变化的新序列

The first difference of a chain code:

counting the number of direction change (in counterclockwise) between 2 adjacent elements of the code.



Chain code :	The first difference
0 → 1	1
0 → 2	2
0 → 3	3
2 → 3	1
2 → 0	2
2 → 1	3
…	

Example: - a chain code: 10103322

- The first difference = 3133030

- Treating a chain code as a circular sequence,

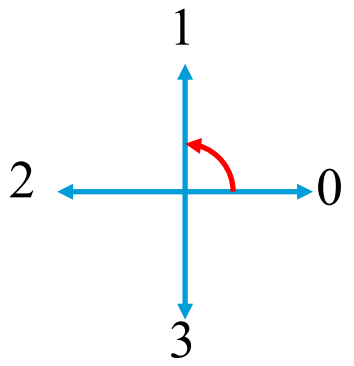
(2)→1→0→1→0→3→3→2→2

the first difference: 3 3 1 3 3 0 3 0

用一阶差分作为旋转归一化形式，是因为它具有旋转不变性

链码归一化——旋转归一化

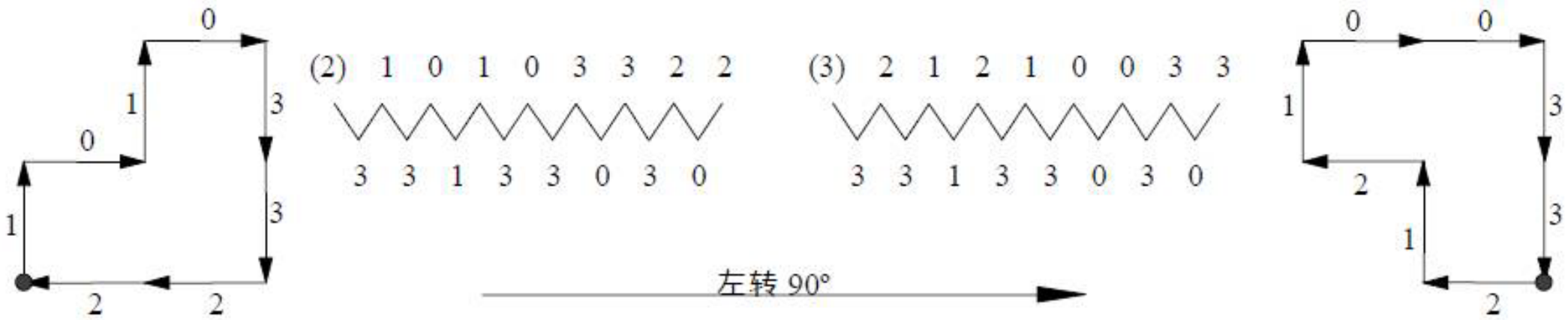
- **The first difference is rotational invariant.**



Chain code : The first difference

0 → 1	1
0 → 2	2
0 → 3	3
2 → 3	1
2 → 0	2
2 → 1	3
…	

原链码发生旋转，但差分码没有变化



链码旋转归一化

基于链码的边界形状描述符

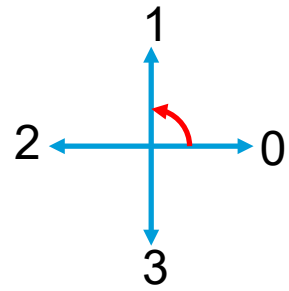
- 链码起点归一化：具有平移不变性和唯一性；
- 链码旋转归一化（差分码）：具有旋转不变性；
- 基于链码的边界形状描述符：对差分码进行（起点）归一化，可得到归一化（唯一）的差分码；
- 归一化差分码具有平移和旋转不变性，也具有唯一性，可用来表示边界，称为“**形状数**”；
- 形状数序列的长度(位数)称为**形状数的阶**，可作为闭合边界的周长。

Shape number of the boundary definition: the first difference of smallest magnitude

The order *n* of the shape number: the number of digits in the sequence

基于链码的边界形状描述符

•例： Shape numbers of order 4, 6 and 8



	Order 4	Order 6
Chain code:	0 3 2 1	0 0 3 2 2 1
Difference:	3 3 3 3	3 | 0 3 3 0 3
Shape no.:	3 3 3 3	0 3 3 0 3 3

	Order 8		
Chain code:	0 0 3 3 2 2 1 1	0 3 0 3 2 2 1 1	0 0 0 3 2 2 2 1
Difference:	3 | 0 3 0 3 0 3 0	3 3 1 3 3 | 0 3 0	3 | 0 0 3 3 0 0 3
Shape no.:	0 3 0 3 0 3 0 3	0 3 0 3 3 1 3 3	0 0 3 3 0 0 3 3

基于链码的边界形状描述符

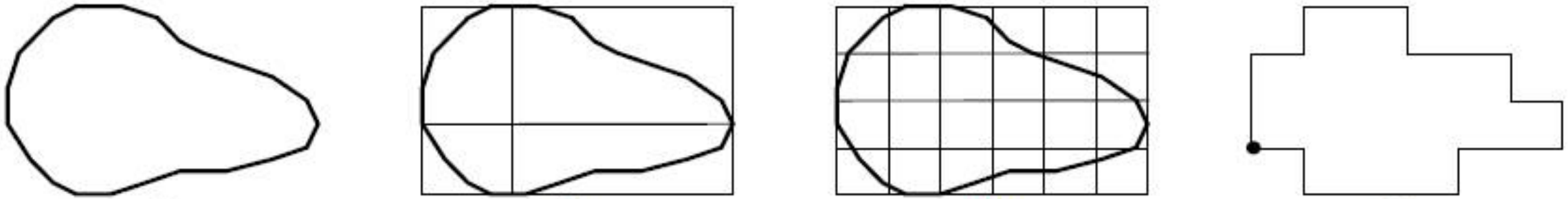
•例

Original boundary

Find the smallest rectangle that fits the shape

Create grid

Find the nearest Grid.



(a) (b) (c) (d)

(e) 链码： 1 1 0 1 0 0 3 0 0 3 0 3 2 2 3 2 2 2 1 2

(f) 微分码： 3 0 3 1 3 0 3 1 0 3 1 3 3 0 1 3 0 0 3 1

(g) 形状数： 0 0 3 1 3 0 3 1 3 0 3 1 0 3 1 3 3 0 1 3

基于区域的分割方法

全局阈值分割

基本思想

•基本思想：根据图像数据的特征将图像空间划分为互不重叠的区域，从而达到分割的目的。

•问题：选用哪种特征？

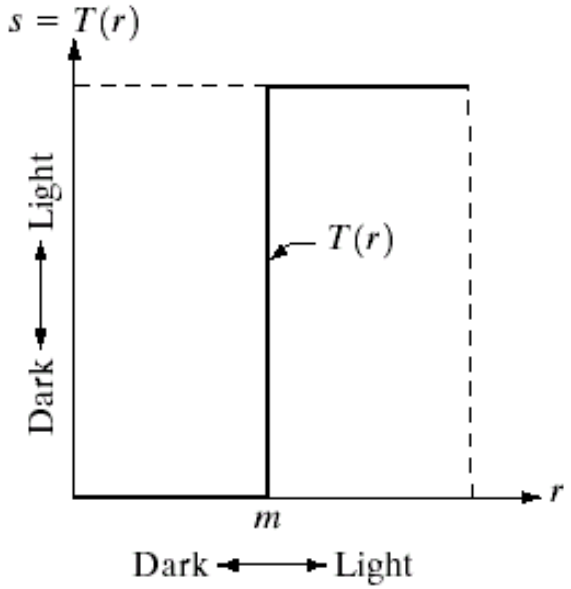
•灰度值、颜色值

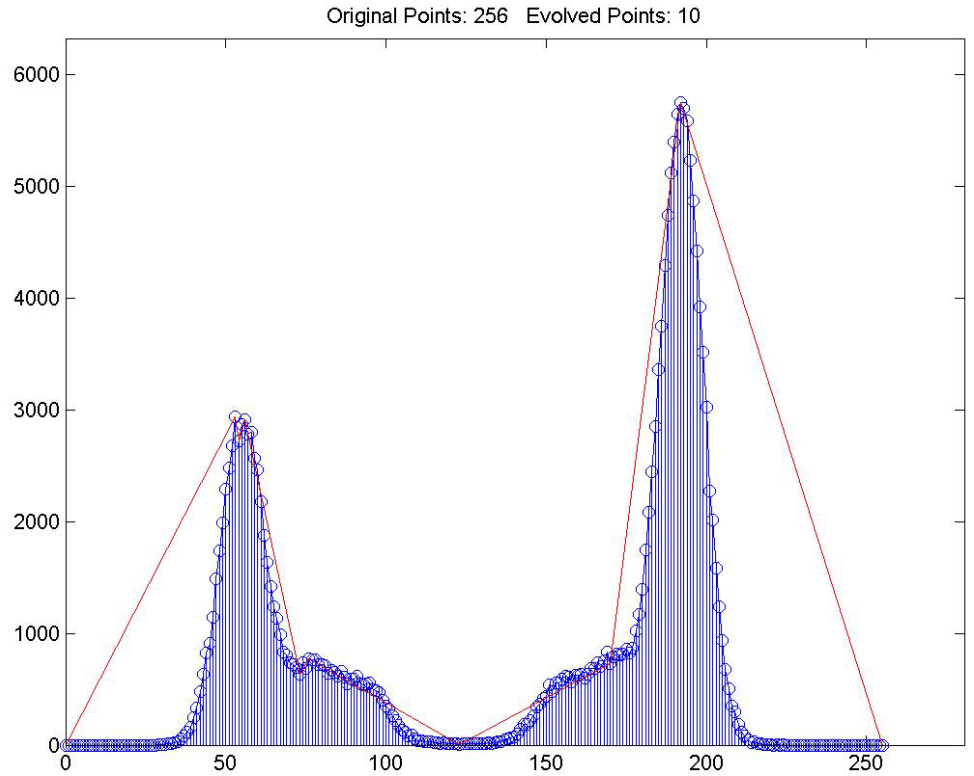
•隐含条件：同一区域的像素应**具有相同或相似的**特征，如灰度或颜色。

图像阈值分割

•阈值分割实质：灰度映射中的二值化方法，将图像像素分为两类：前景目标和背景

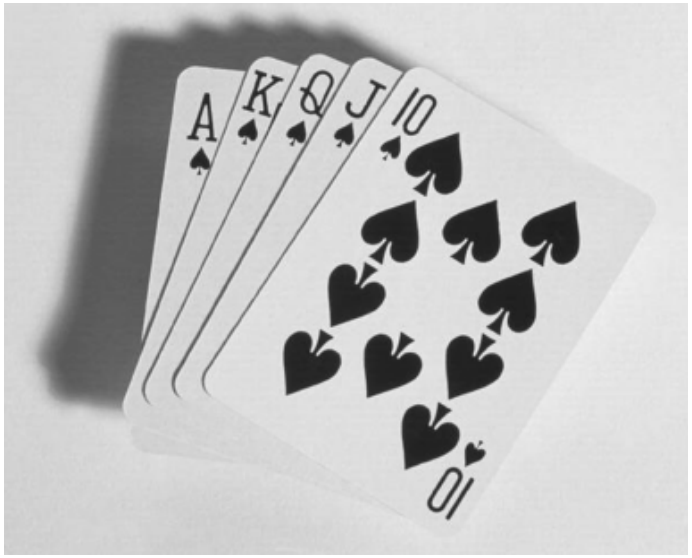
$$T_r = \begin{cases} 1 & r \geq m \\ 0 & r < m \end{cases}$$





图像阈值分割

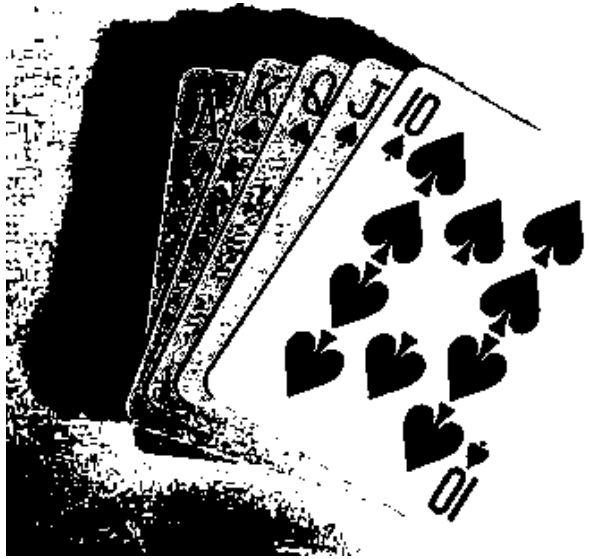
- 问题：如何确定合适的阈值？



Original Image

Thresholded Image

Threshold Too Low

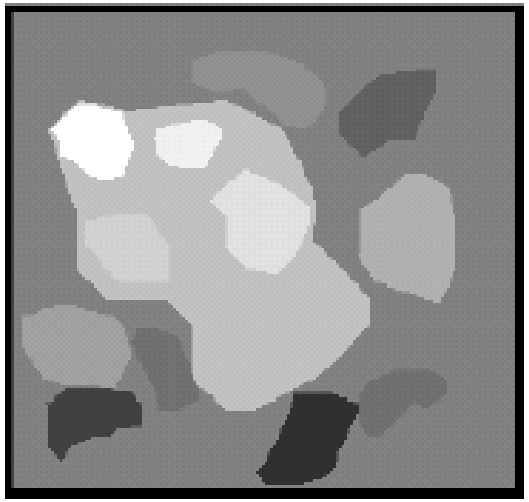


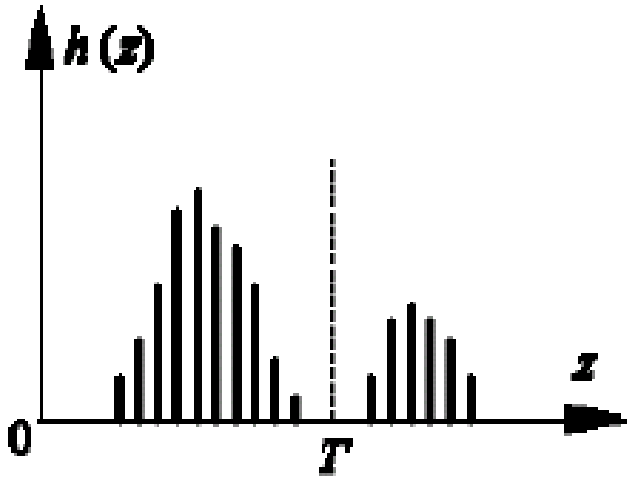
Threshold Too High

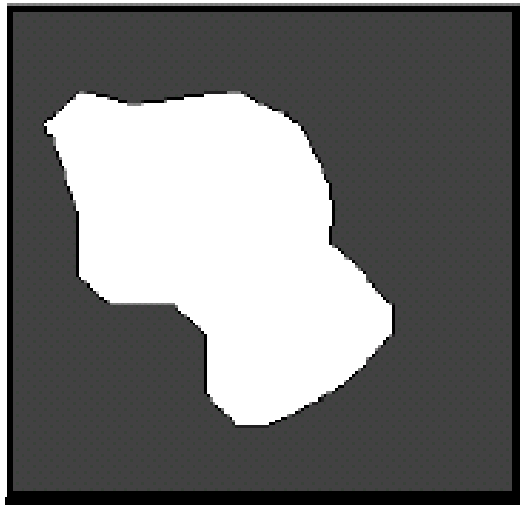
直方图阈值分割

•60年代中期，Prewitt提出了直方图双峰法

•如果灰度级直方图呈明显的双峰状，则选取两峰之间的谷底所对应的灰度级作为阈值。

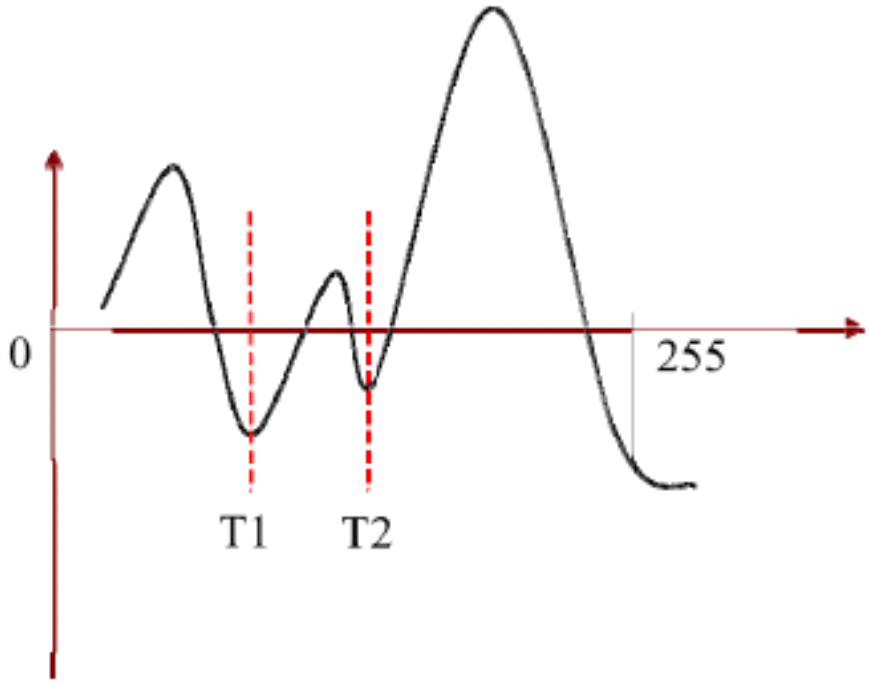






直方图阈值分割

•若灰度级直方图能呈现多个明显的峰值，如三个峰值，可取两个峰谷处的灰度值T1，T2作为阈值。同样，可以进行阈值化。



直方图阈值分割

•若在图像中存在背景和**n个有意义的目标S1，S2，…,Sn。背景的灰度最小，目标间的灰度**的差异较大，则可在两两之间差异度较大处设置门限： **T0，T1，…,Tn-1，则分割后的图像**为：

$$g(x,y)=\begin{cases} c_0 & f(x,y)\le T_0 \\ c_1 & T_0\le f(x,y)\le T_1 \\ \vdots & \vdots \\ c_n & T_{n-1}\le f(x,y)\le T_n \end{cases}$$

问题：如何自动获取最优阈值？

最优阈值

- 思路：使图像中目标物和背景分割错误最小的阈值。

- 设一幅图像只由目标和背景组成，已知像素灰度概率密度分别为 $p_{obj}(z)$ 和 $p_{bg}(z)$，目标像素占整幅图像像素比为 α，则图像总的灰度级概率密度函数为：

$$p(z)=\alpha p_{obj}(z)+(1-\alpha)p_{bg}(z)$$

- 对于某个阈值 z_k，有：

$$C_z=\begin{cases}\text{目标} & z\geq z_k\\ \text{背景} & z<z_k\end{cases}$$

最优阈值

- 公式 $p(z)=\alpha p_{obj}(z)+(1-\alpha)p_{bg}(z)$ 说明：

- 全概率公式： $B_1,B_2,\cdots,B_n$是S的一个划分，$P(B_i)>0$，$i=1,2,\cdots,n$,则

$$P(A)=P(A|B_1)P(B_1)+P(A|B_2)P(B_2)+\cdots+P(A|B_n)P(B_n)$$

$$=\sum_{i=1}^{n}P(A|B_i)P(B_i)$$

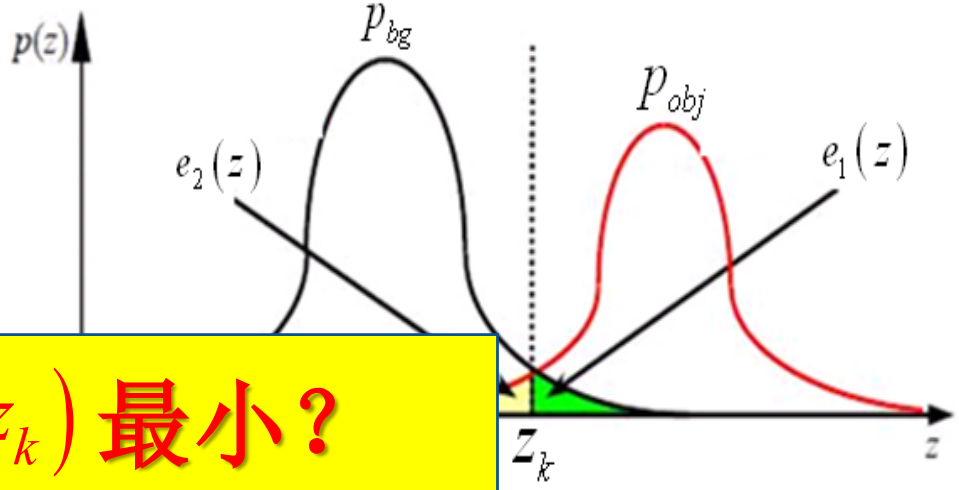
- 目标像素和背景像素是图像I的一个划分，有

$$p_{obj}(z)\rightarrow p(z|obj) \qquad p_{bg}(z)\rightarrow p(z|bg)$$

$$p(obj)=\alpha \qquad p(bg)=1-\alpha$$

最优阈值

- 背景错归为目标的概率：$e_1(z_k)=\int_{z_k}^{+\infty} p_{bg}(z)dz$
- 目标错归为背景的概率：$e_2(z_k)=\int_{-\infty}^{z_k} p_{obj}(z)dz$
- 总的错分概率 $e($



问题：如何找到 $e(z_k)$ 最小？

- 最佳阈值的目标：

  寻找阈值 z_k，使得 $e(z_k)$ 最小。

Why?

最优阈值

•最佳阈值的目标： $z_k = \arg\min e(z_k)$

•求导，并令 $\dfrac{\partial e(z_k)}{\partial z_k} = 0$

$$e_1(z_k) = \int_{z_k}^{+\infty} p_{bg}(z)dz \Rightarrow \frac{\partial e_1(z_k)}{\partial z_k} = -p_{bg}(z_k)$$

$$e_2(z_k) = \int_{-\infty}^{z_k} p_{obj}(z)dz \Rightarrow \frac{\partial e_2(z_k)}{\partial z_k} = p_{obj}(z_k)$$

$$\alpha \cdot p_{obj}(z_k) = (1-\alpha) \cdot p_{bg}(z_k)$$

•假设目标和背景的概率密度函数为高斯模型

$$p_{obj}(z) = \frac{1}{\sqrt{2\pi}\sigma_{obj}} e^{-\frac{(z-\mu_{obj})^2}{2\sigma_{obj}^2}} \qquad p_{bg}(z) = \frac{1}{\sqrt{2\pi}\sigma_{bg}} e^{-\frac{(z-\mu_{bg})^2}{2\sigma_{bg}^2}}$$

最优阈值

$$\alpha \cdot p_{obj}(z_k) = (1-\alpha) \cdot p_{bg}(z_k)$$

两边取对数，并化简

$$Az_k^2 + Bz_k + C = 0$$

其中

$$\begin{cases} A = \sigma_{obj}^2 - \sigma_{bg}^2 \\ B = 2\left(\mu_{obj}\sigma_{obj}^2 - \mu_{bg}\sigma_{bg}^2\right) \\ C = \mu_{bg}^2\sigma_{obj}^2 - \mu_{obj}^2\sigma_{bg}^2 + 2\sigma_{obj}^2\sigma_{bg}^2 \ln\left(\dfrac{\alpha\sigma_{bg}}{(1-\alpha)\sigma_{obj}}\right) \end{cases}$$

最优阈值

- 方程 $Az_k^2 + Bz_k + C = 0$ 在一般情况下存在两个解。
- 当且仅当 $\sigma_{obj}=\sigma_{bg} = \sigma$ 时，有唯一解：

$$z_k = \frac{\mu_{obj} + \mu_{bg}}{2} + \frac{\sigma^2}{\mu_{obj} - \mu_{bg}} \ln\left(\frac{\alpha}{1-\alpha}\right)$$

- 若 $\alpha = \frac{1}{2}$ ，则

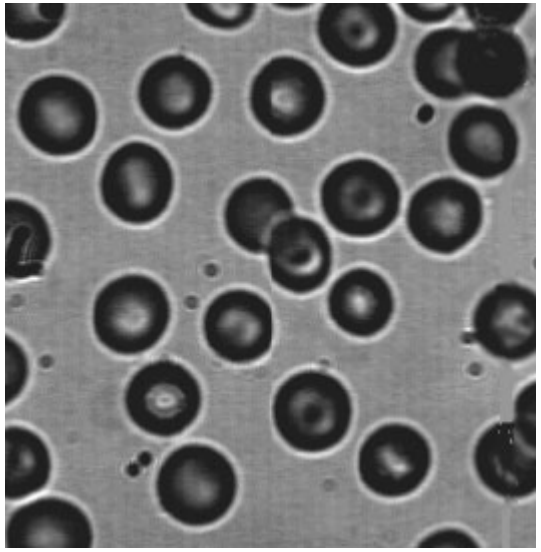
$$z_k = \frac{\mu_{obj} + \mu_{bg}}{2}$$

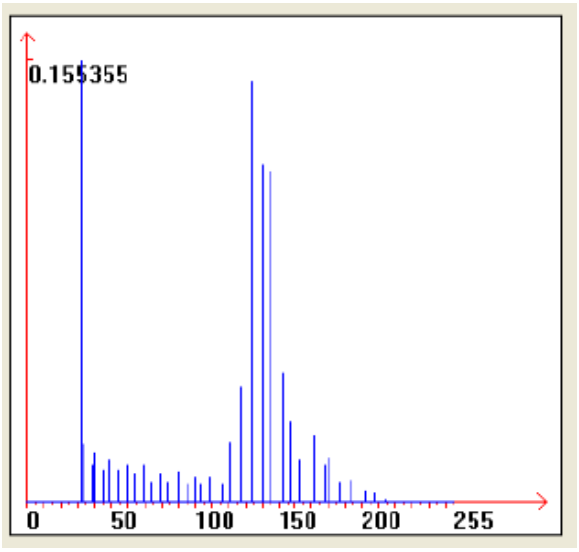
均值迭代阈值分割方法

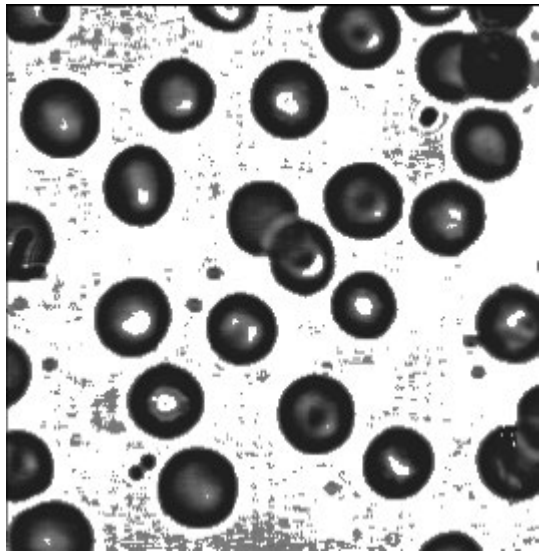
均值迭代阈值分割方法

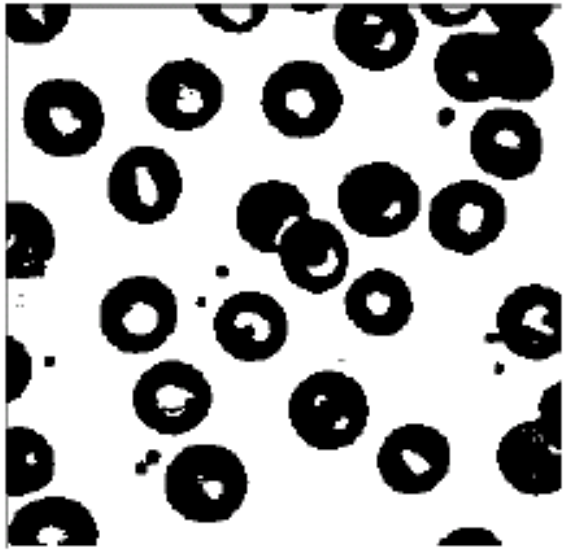
- Step1. 选择一个初始的估计阈值T（可以用图像的平均灰度值作为初始阈值）
- Step2. 用该阈值把图像分割成两个部分R1和R2
- Step3. 分别计算R1和R2的灰度均值μ1和μ2
- Step4. 选择一个新的阈值T＝(μ1＋μ2)/2
- Step5. 重复2-4直至后续迭代中平均灰度值μ1和μ2保持不变

均值迭代阈值分割结果









类间方差阈值分割

- 思路：
  - 基于二元统计分析理论，选取一个阈值k，构造两个统计量C_{bg}和C_{obj}，满足这两个统计量的类内方差最小，类间方差最大，则此时的 k 为最佳阈值。

最大类间方差法——OTSU方法

•给定一幅图像，其归一化直方图表示为：

$$p_i = n_i / N; \quad p_i \geq 0; \quad \sum_i p_i = 1$$

•设分割的阈值为k，则背景和目标类的概率为：

$$\omega_0 = \Pr\left(C_{bg}\right) = \sum_{i=0}^{k} p_i = \omega(k) \qquad \omega_1 = \Pr\left(C_{obj}\right) = \sum_{i=k+1}^{L-1} p_i = 1 - \omega(k)$$

•背景类的均值为：

$$\mu_0 = \sum_{i=0}^{k} i \Pr\left(i \mid C_{bg}\right) = \sum_{i=0}^{k} i \Pr\left(i, C_{bg}\right) / \Pr\left(C_{bg}\right) = \sum_{i=0}^{k} i\, p_i / \omega_0 = \mu(k) / \omega(k) \quad ; \quad \mu(k) = \sum_{i=0}^{k} i p_i$$

最大类间方差法——OTSU方法

- 目标类地均值为

$$\mu_1 = \sum_{i=k+1}^{L-1} i \Pr\left(i \mid C_{obj}\right) = \sum_{i=k+1}^{L-1} i\, p_i / \omega_1 = \frac{\mu_T - \mu(k)}{1-\omega(k)} \qquad \mu_T = \sum_{i=0}^{L-1} i p_i$$

- $\omega_0, \omega_1, \mu_0, \mu_1$ 满足

图像均值： $\omega_0\mu_0 + \omega_1\mu_1 = \mu_T$ $\omega_0 + \omega_1 = 1$

最大类间方差法——OTSU方法

- 背景类的方差

$$\sigma_0^2=\sum_{i=0}^{k}(i-\mu_0)^2\Pr(i\mid C_{bg})=\sum_{i=0}^{k}(i-\mu_0)^2\,p_i/\omega_0$$

- 目标类的方差

$$\sigma_1^2=\sum_{i=k+1}^{L-1}(i-\mu_1)^2\Pr(i\mid C_{obj})=\sum_{i=k+1}^{L-1}(i-\mu_1)^2\,p_i/\omega_1$$

最大类间方差法——OTSU方法

•类内方差为： $\sigma_w^2 = \omega_0\sigma_0^2 + \omega_1\sigma_1^2$

•类间方差为： $\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2$

$= \omega_0\omega_1(\mu_1 - \mu_0)^2$

寻找使得类间方差最大的阈值

参考文献：K, Fukunage, Introduction to Statisticul Pattern Recogniition. New York: Academic, 1972, pp. 260-267.

OTSU方法具体实现

- 步骤1：

```
BYTE* ptr = pimg;
for (i=0; i<imsize; i++)            // 统计直方图
        histogram[*ptr++]++;
```

- 步骤2：

```
float prob[256], miu[256], miuT = 0;
for (i=0; i<256; i++) {
        prob[i]= (float)histogram[i] / imsize;      // 各灰度级的概率
        miuT += miu[i] = i * prob[i];               // 各灰度级的质量矩，
}
```

$$\mu_T = \sum_{i=0}^{L-1} ip_i$$

OTSU方法具体实现

- 步骤3：寻找最大类间方差

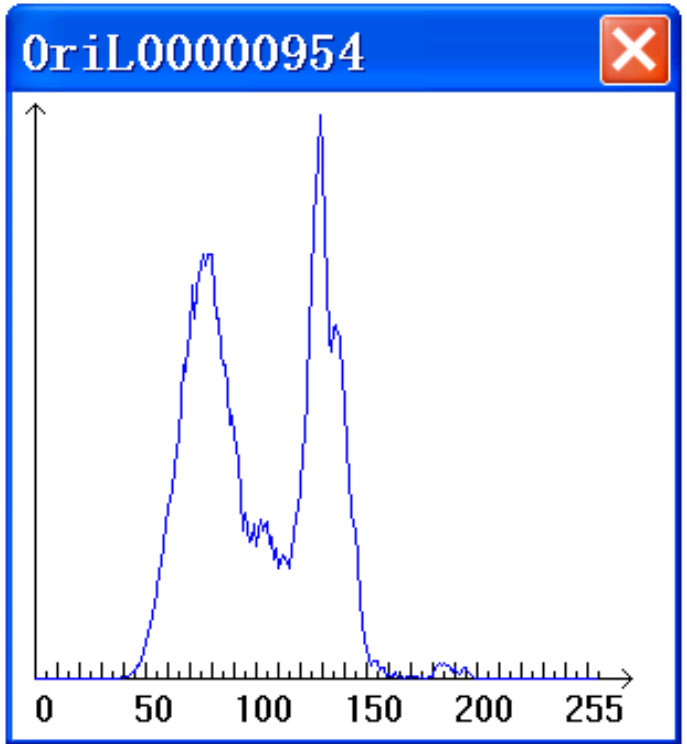
```
BYTE t = 0;          // 阈值t
float miu0 = 0, miu1, miuk = 0, wk = 0, w0, w1, sigma, sigma_max = -1;
for(i=0; i<256; i++) {
    wk += prob[i];       miuk += miu[i];
    w0 = wk;    w1 = 1 - wk;
    miu0 = miuk / w0;
    miu1 = (miuT - miuk) / w1;
    sigma = w0*w1*(miu1-miu0)*(miu1-miu0);
    // 寻找最大sigma值
    if ( sigma >= sigma_max ) {
        t = i;
        sigma_max = sigma;
    }
}
```

$$\mu_0 = \mu(k)/\omega(k)$$

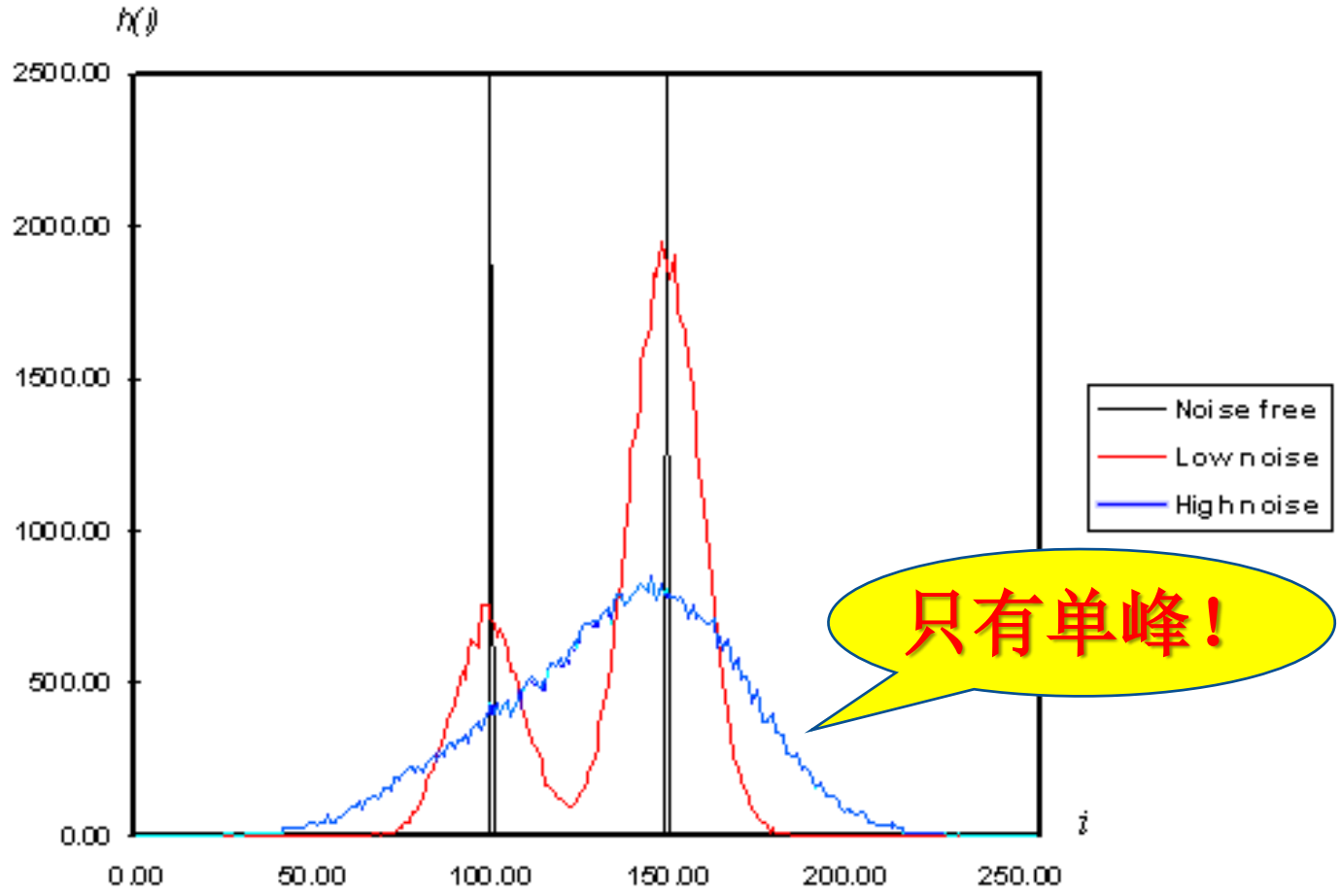
$$\mu_1 = \frac{\mu_T - \mu(k)}{1-\omega(k)}$$

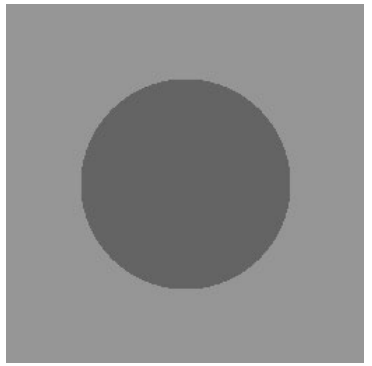
$$\sigma_B^2 = \omega_0\omega_1(\mu_1 - \mu_0)^2$$

OTSU分割结果

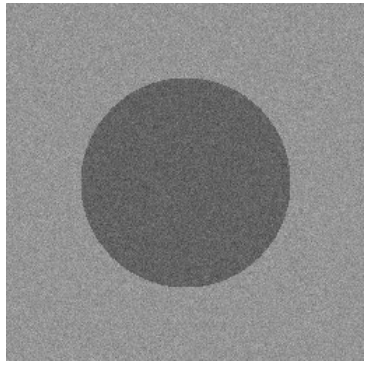


阈值分割的优缺点

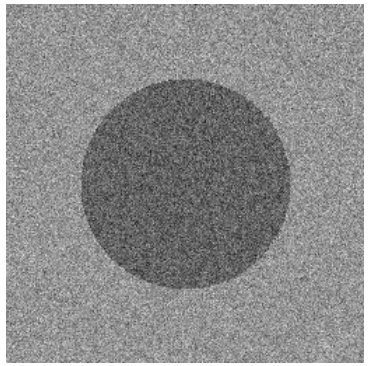




Noise free



Low noise



High noise

阈值分割的优缺点

- **阈值分割的优点：**
  - 实现简单，当不同类的物体灰度值或者其它特征值相差很大时，它能有效地对图像进行分割。
- **缺点：**
  - 对于图像中不存在明显灰度差异或灰度值范围有较大重叠的图像分割问题难以得到准确的结果。
  - 只是考虑了图像的灰度信息，而没有考虑图像的空间信息。

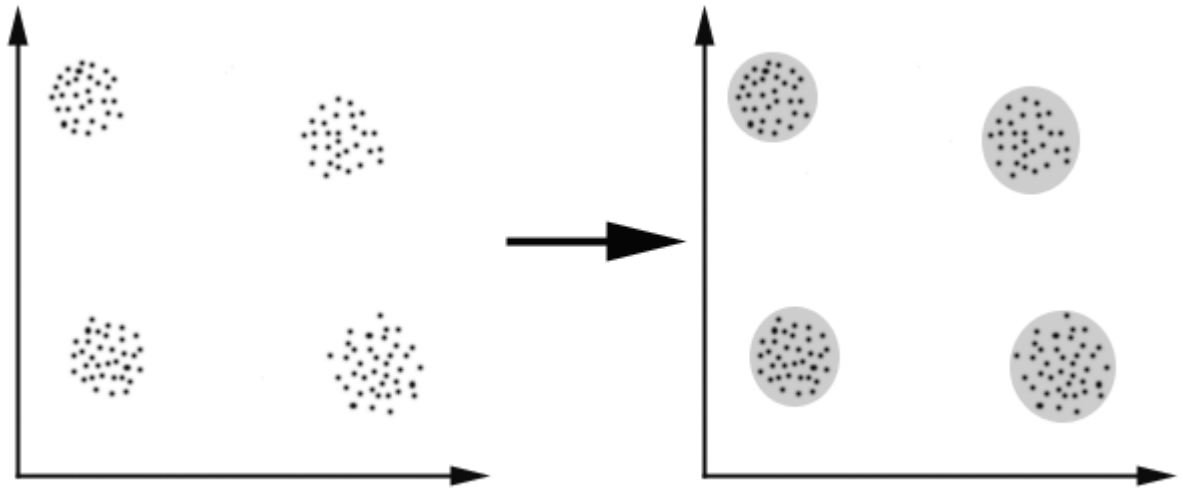
基于聚类的目标分割方法

K-Means 算法

K-Means算法

•在数据挖掘中，K-Means算法是一种 cluster analysis 的算法.

•其主要是来计算数据聚集的算法，主要通过不断地取离种子点最近均值的算法.

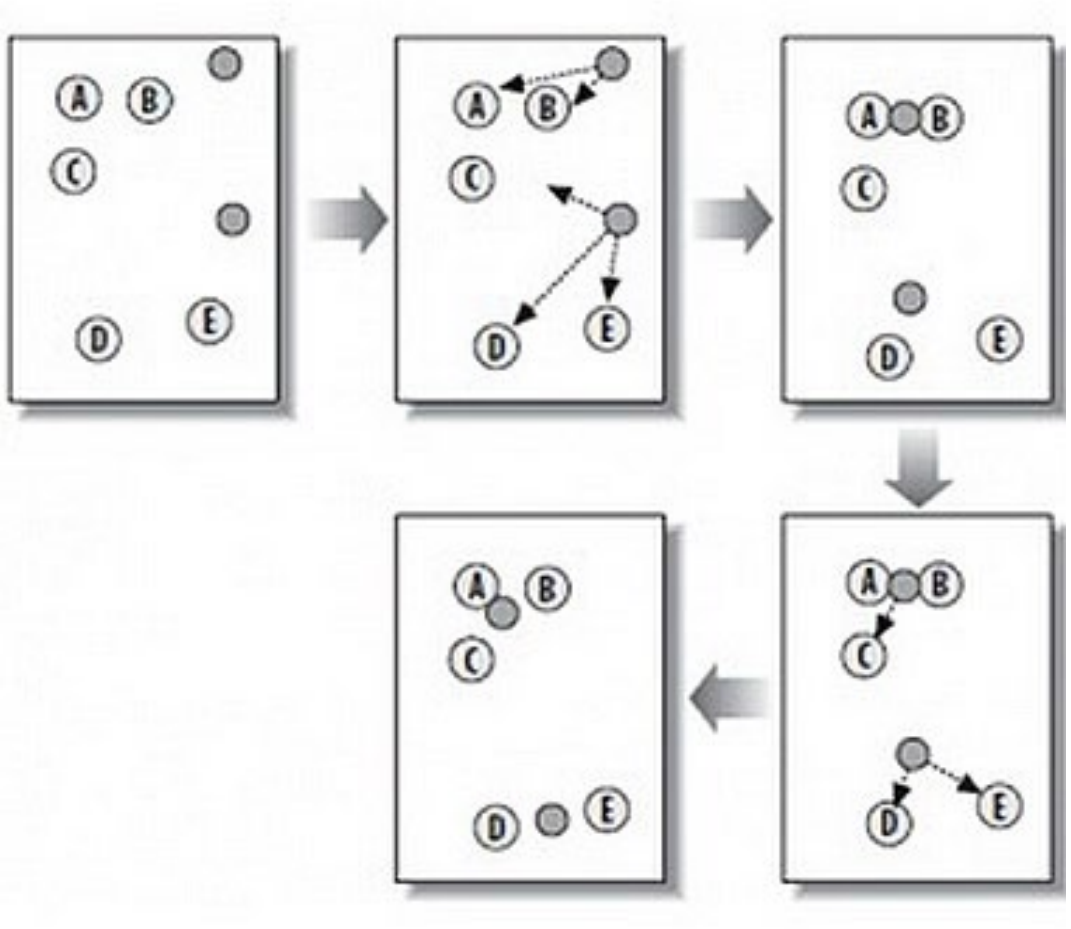


算法概要

1) 随机在图中取*K*个种子点。

2) 然后对图中的所有点求到这*K*个种子点的距离，假如点 P_i 离种子点S_i最近，那么 P_i 属于 S_i 点群。（上图中，我们可以看到A，B属于上面的种子点，C，D，E属于下面中部的种子点）

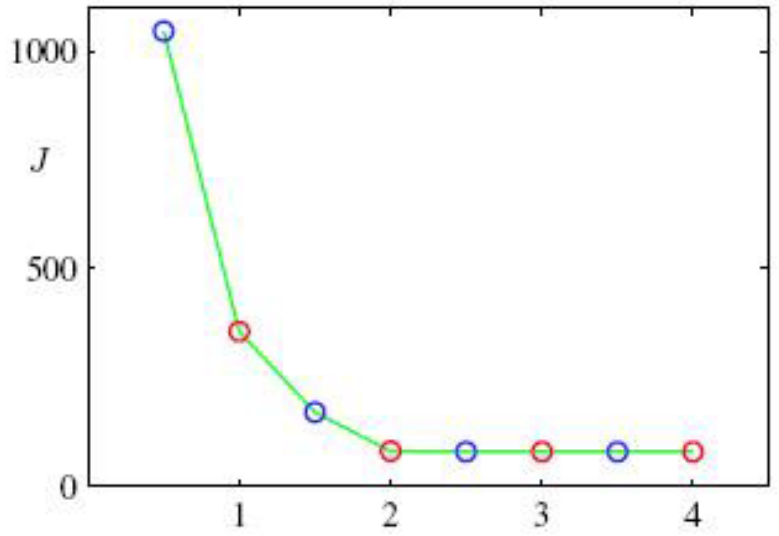
3) 接下来，我们要移动种子点到属于它的“点群”的中心。

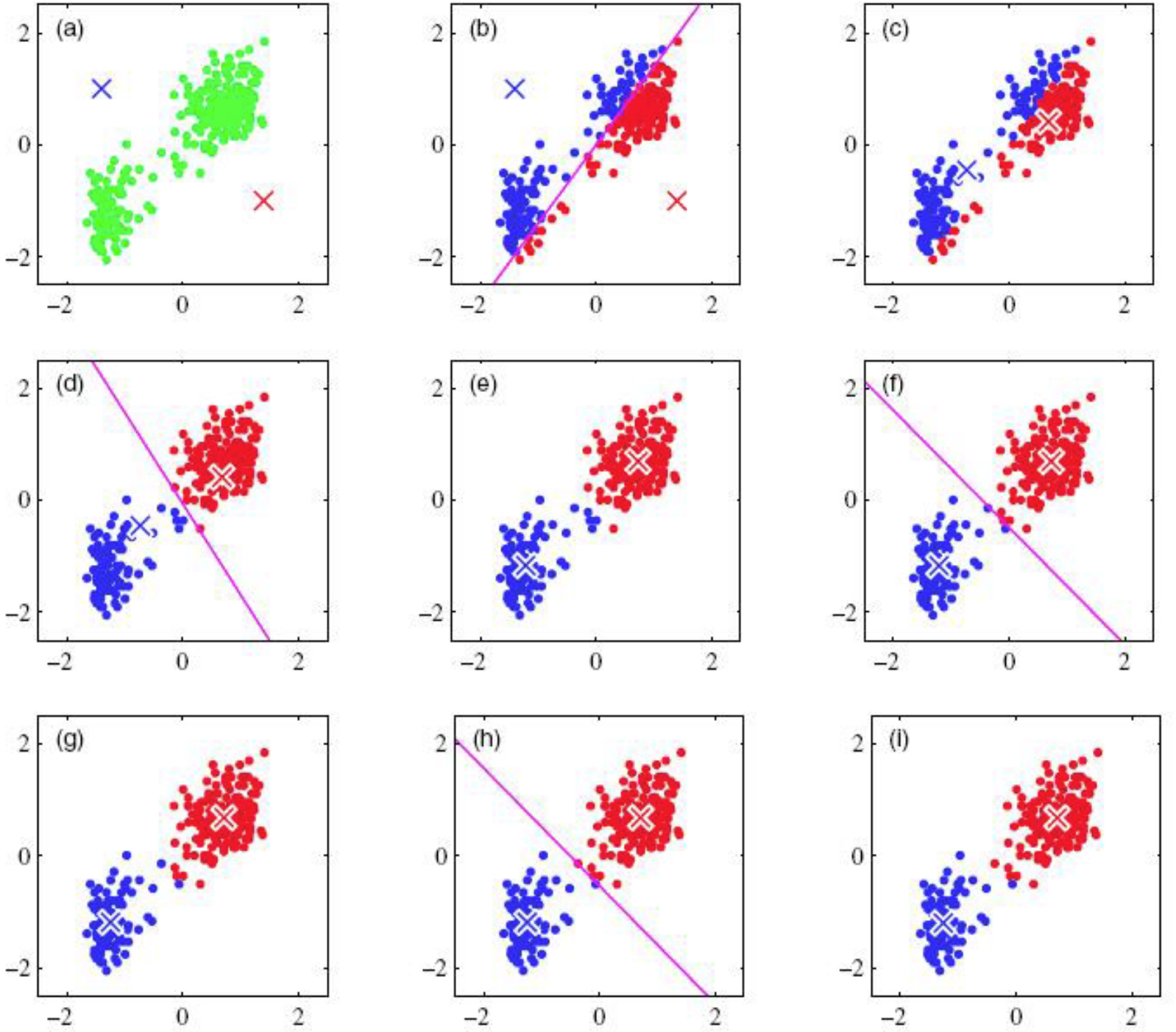


4) 然后重复第2）和第3）步，直到种子点没有移动（我们可以看到图中的第四步上面的种子点聚合了A，B，C，下面的种子点聚合了D，E）

K-Means聚类过程

•经过三次迭代，算法收敛



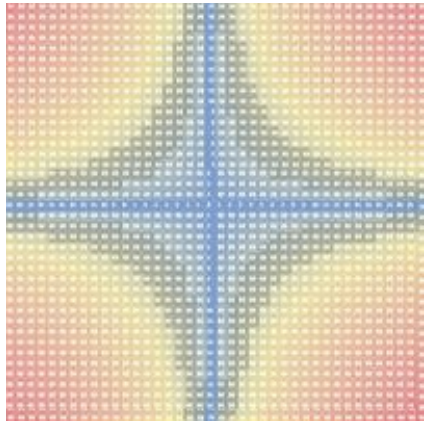


求点群中心的算法

•1）Minkowski Distance公式

λ可以随意取值，可以是负数，也可以是正数，或是无穷大

$$d_{ij} = \sqrt[\lambda]{\sum_{k=1}^{n} \left| x_{ik} - x_{jk} \right|^{\lambda}}$$

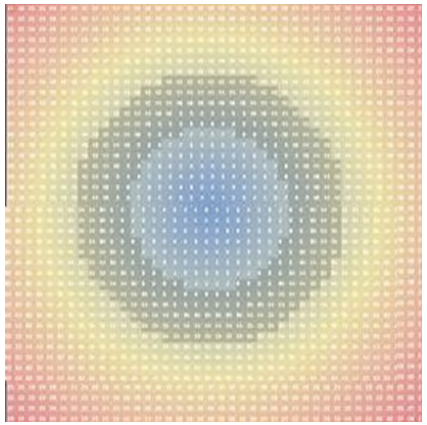


以星形的方式逼近中心

•2）Euclidean Distance公式

Minkowski Distance公式中λ=2的情况

$$d_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk} \right)^2}$$

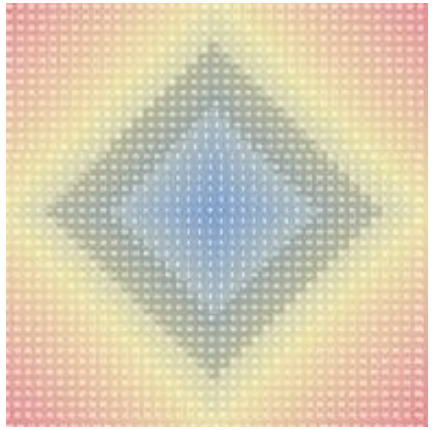


以同心圆的方式逼近中心

• 3）CityBlock Distance公式

Minkowski Distance公式中λ=1的情况

$$d_{ij} = \sum_{k=1}^{n} \left| x_{ik} - x_{jk} \right|$$



以菱形的方式逼近中心

K-Means算法：例

Original

K=5

K=11

基于全局能量函数的分割方法

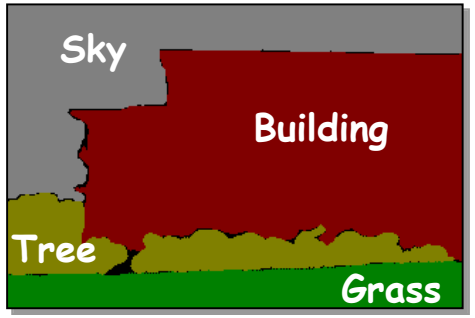
Graph Cuts分割

Graph Cuts 图割法

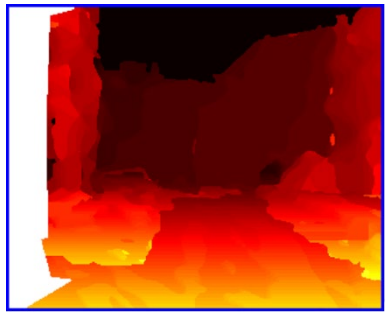
•Graph Cuts是一种流行的能量函数优化算法，普遍应用于分割（Image Segmentation）、立体匹配（stereo Matching）、抠图（Image Matting）等视觉任务中。

•**Image Labelling Problems：** **Assign a label to each image pixel**

Object Segmentation

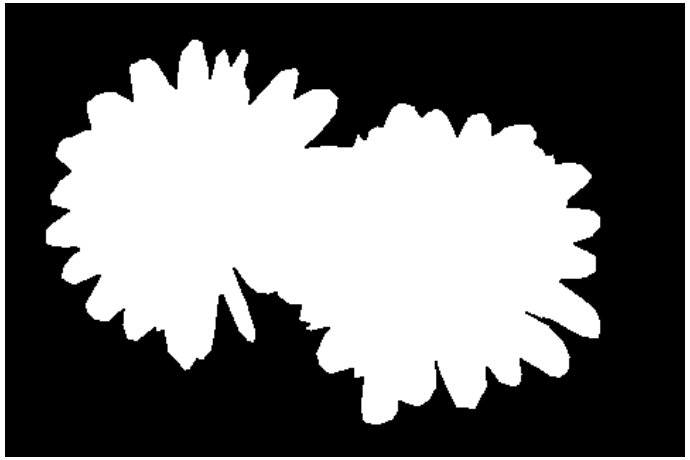


Depth Estimation



Graph Cuts 图割法

•例 Foreground / Background Estimation



Graph Cuts 图割法

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j)$$

Data term　　Smoothness term

- Foreground / Background Estimation

Data term

$$\psi_i(x_i = 0) = -\log(p(x_i \notin FG))$$
$$\psi_i(x_i = 1) = -\log(p(x_i \in FG))$$

Estimated using FG / BG colour models

Smoothness term

$$\psi_{ij}(x_i, x_j) = K_{ij}\delta(x_i \neq x_j)$$

where $K_{ij} = \lambda_1 + \lambda_2 \exp(-\beta(I_i - I_j)^2)$

Intensity dependent smoothness

Graph Cuts 图割法

•**Foreground / Background Estimation**

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j)$$

Data term **Smoothness term**

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbf{L}} E(\mathbf{x})$$

How to solve this optimisation problem?

- Transform into min-cut / max-flow problem
- Solve it using min-cut / max-flow algorithm

Graph Cuts 图割法

•根据给定的一幅图像，构建一张图**G=(V, E)**

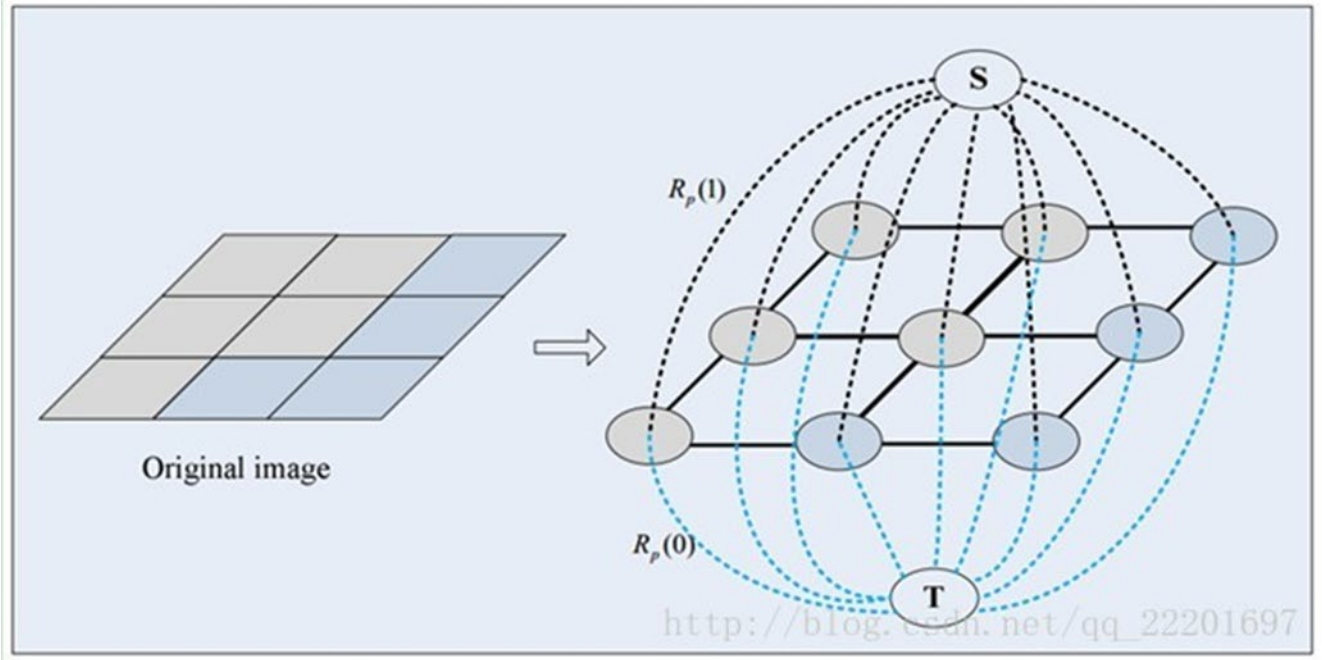
• **V**为图**G**的顶点，对应于图像中的各像素；

• **V**中还包含了特殊的两个终端顶点**S**和**T**；

• **S**称为源点（Source），**T**称为汇点（Sink）；

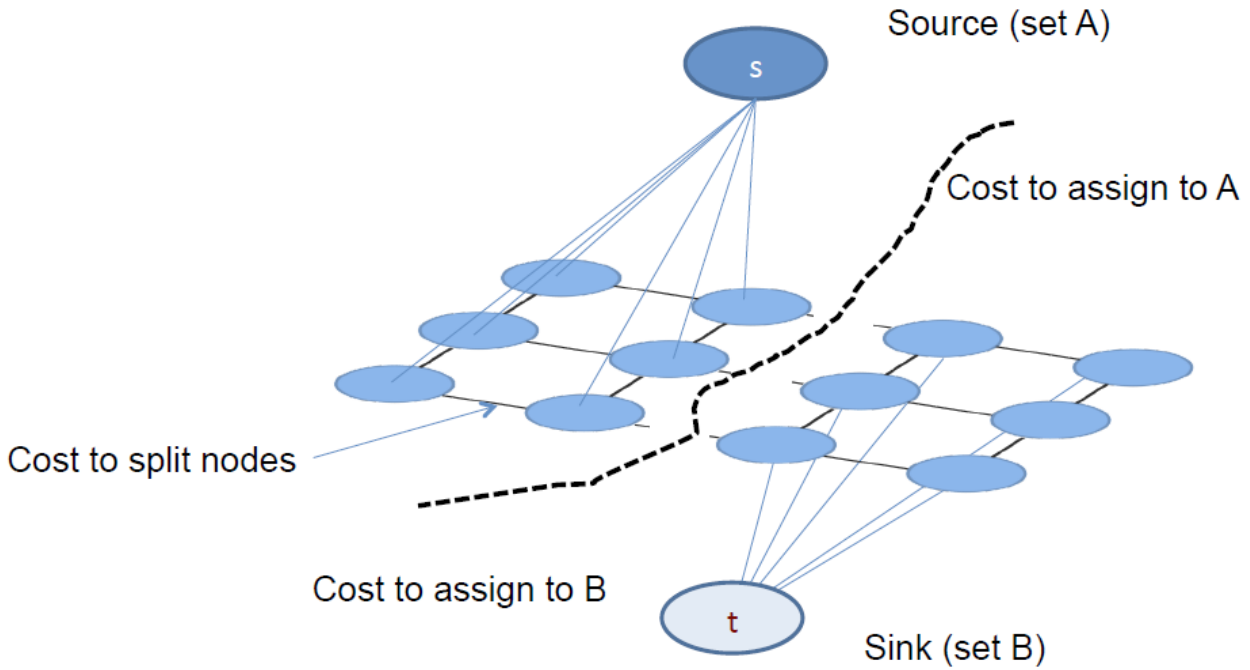
• 两邻域顶点（图像邻域像素）的连接边，称为n-links；

•普通顶点（图像像素）与终端顶点**S**和**T**的连接边，称为t-links；



Graph Cuts 图割法

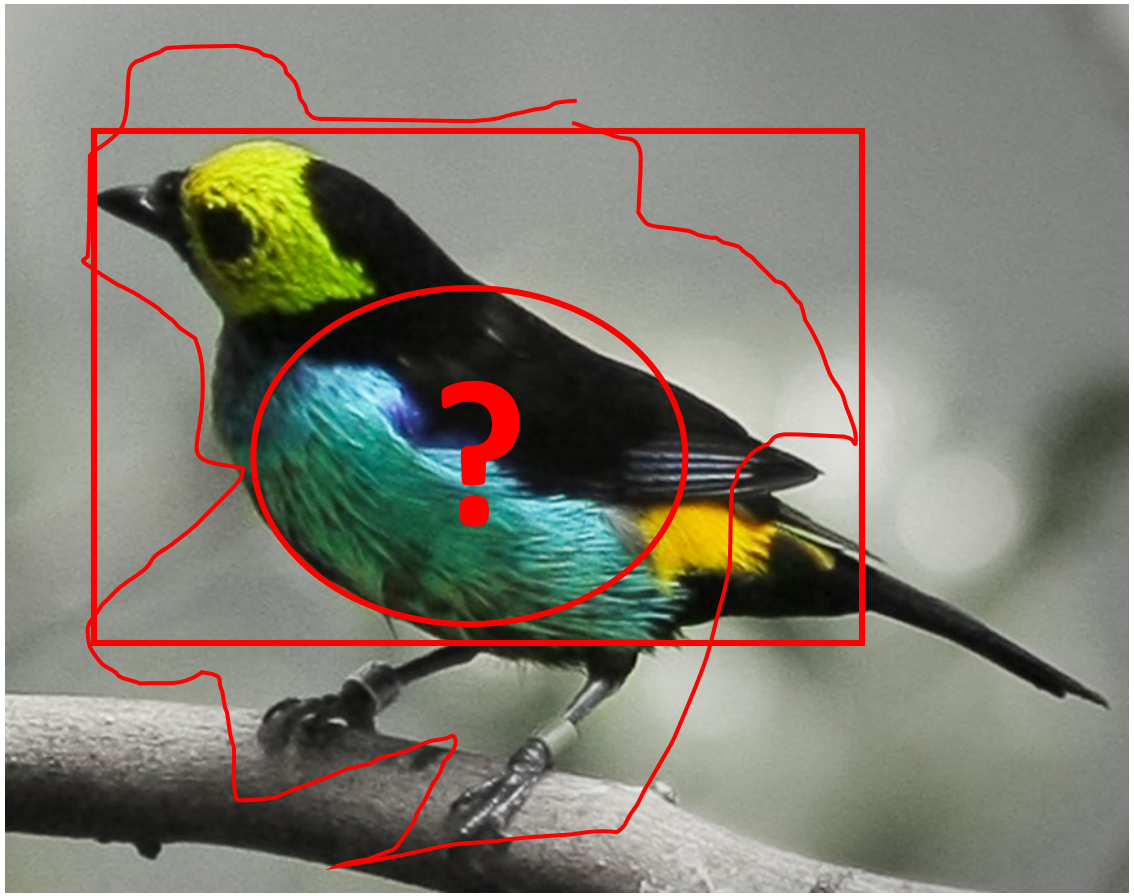
- **Cut（割）：** 图中n-links边集合**E**的一个子集**C**，Cut（割）的cost为边集**C**的所有边的权值总和。
- 如果一个割，其包含的所有边权值之和最小，则称为最小割，也就是图割的结果。
- 在图割中，最小割把图的顶点划分为两个不相交的子集**A**和**B**，其中**S**∈**A**，**T**∈**B**。
- 事实上，这两个子集**A**和**B**对应于图像的前景像素集和背景像素集，也就相当于完成了图像分割。



Graph Cuts 图割法

•例

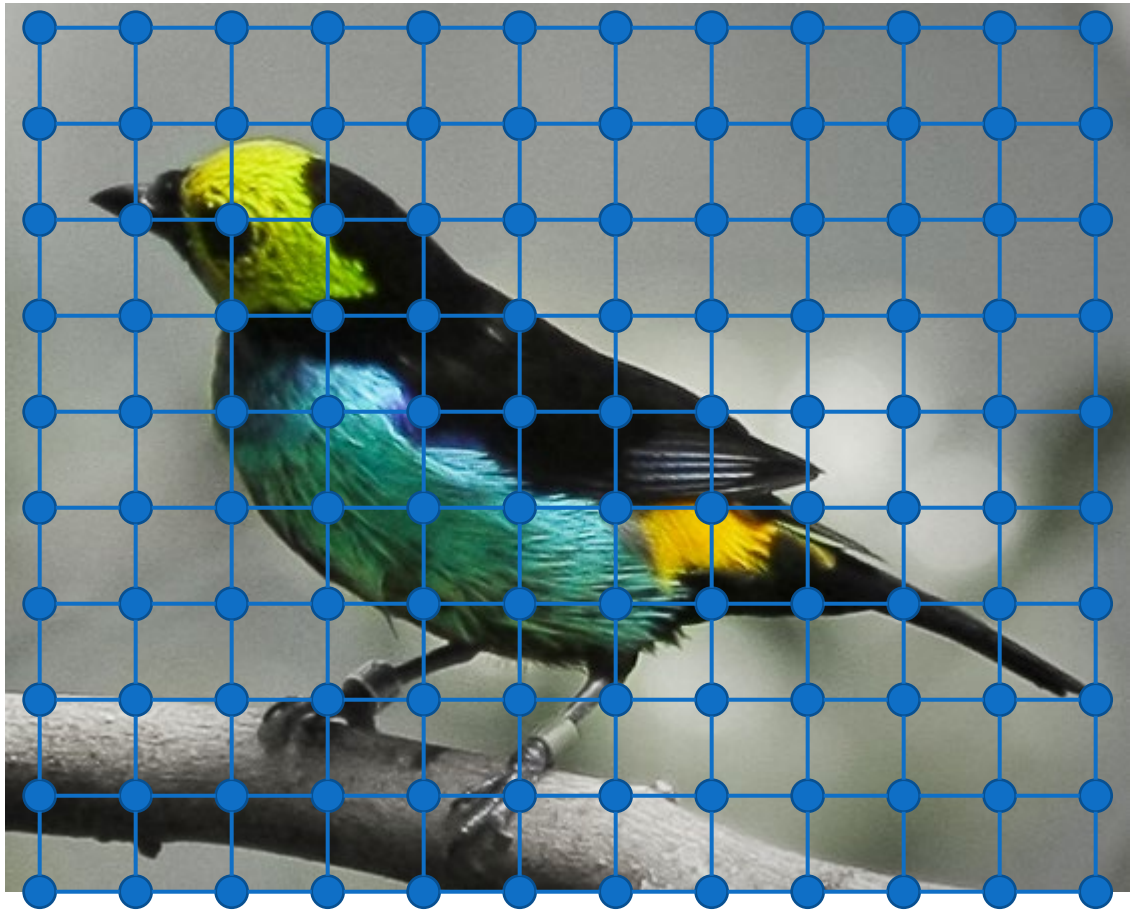
- First: select a region of interest
- How to select the object automatically?
- We care about two terms: graph and cuts



Graph Cuts 图割法

•What are graphs?

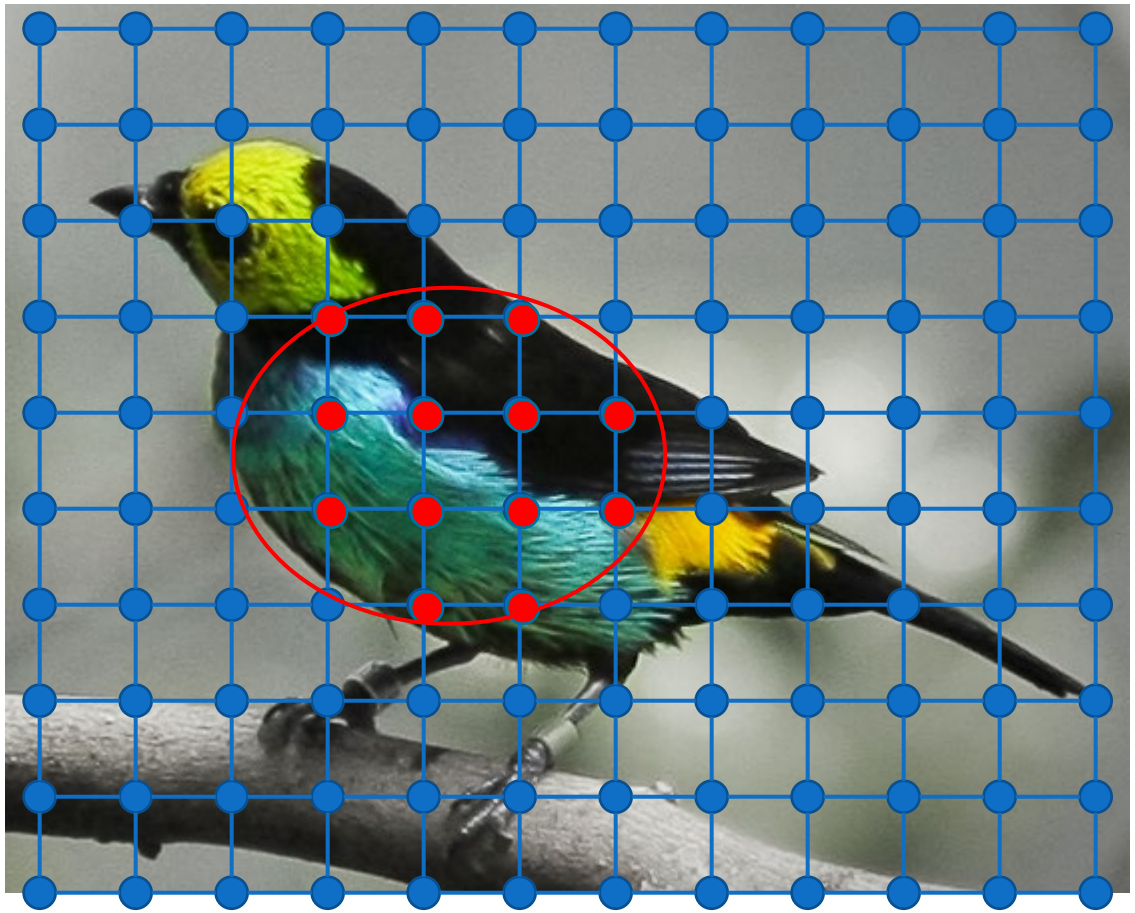
- **Nodes**
  - usually pixels
  - sometimes samples
- **Edges**
  - weights associated, W(i, j)
  - e.g. RGB value difference



Graph Cuts 图割法

- What are cuts?

- Each “cut” -> points, W(i, j)
- Optimization problem
- W(i, j) = 1/|RGB(i) – RGB(j)|

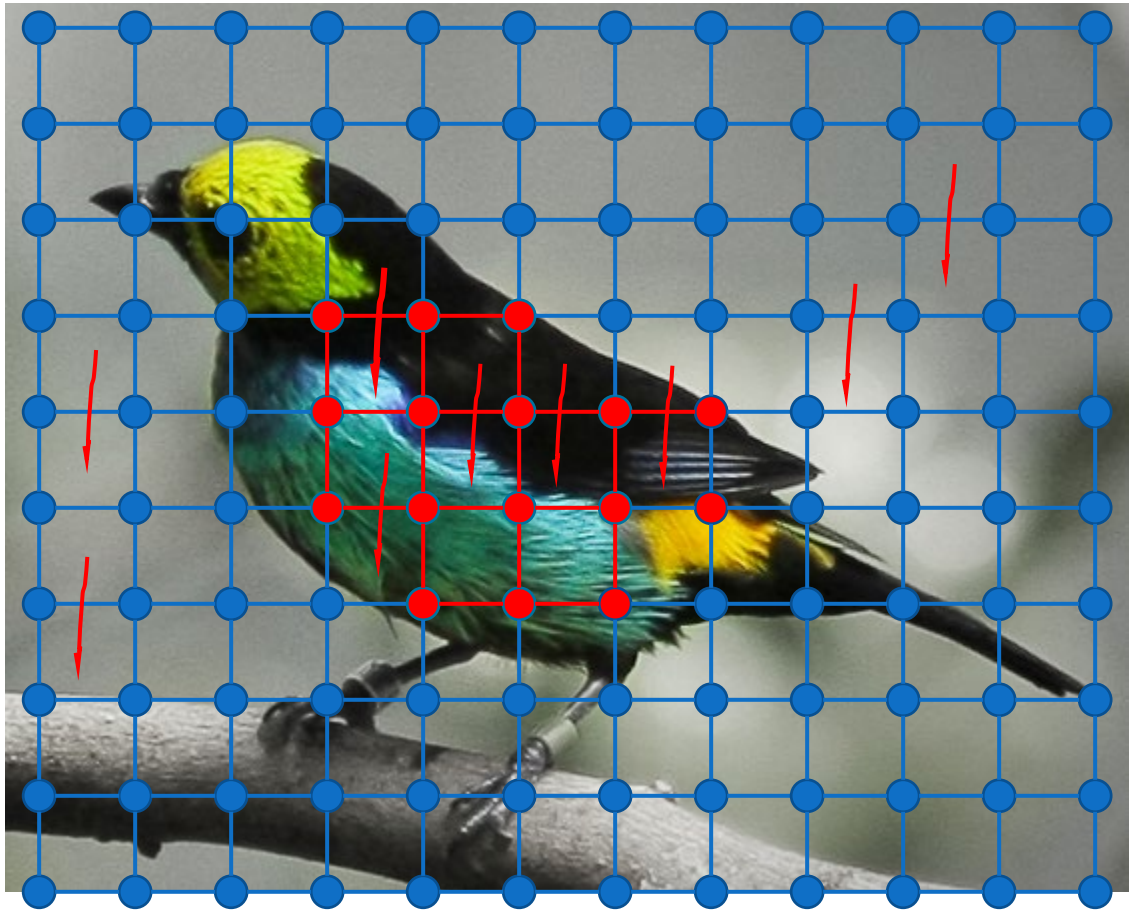


Graph Cuts 图割法

- We want highest sum of weights
- Each “cut” -> points, W(i, j)
- Optimization problem
- W(i, j) = 1/|RGB(i) – RGB(j)|

These cuts give high points

W(i, j) = 1/|RGB(i) – RGB(j)| is high

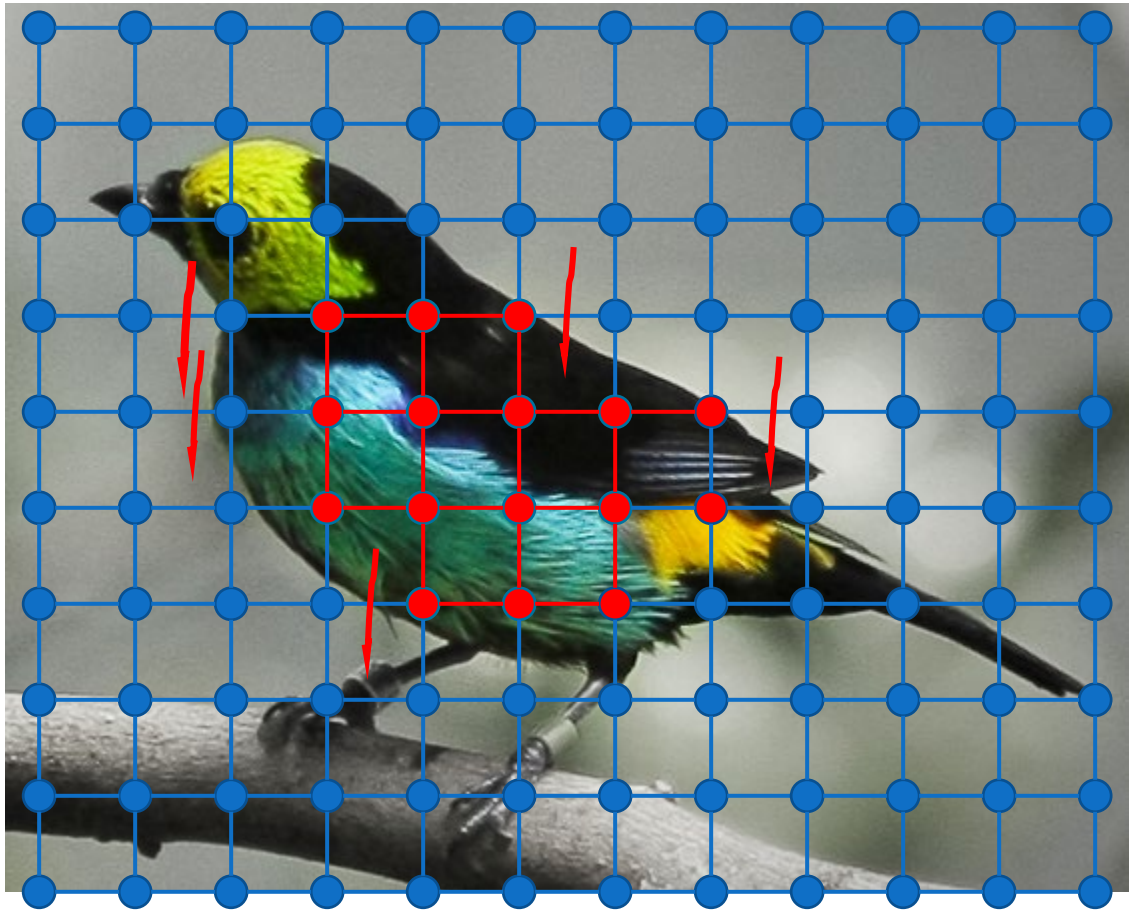


Graph Cuts 图割法

- We want highest sum of weights
- Each “cut” -> points, W(i, j)
- Optimization problem
- W(i, j) = 1/|RGB(i) – RGB(j)|

These cuts give **low** points

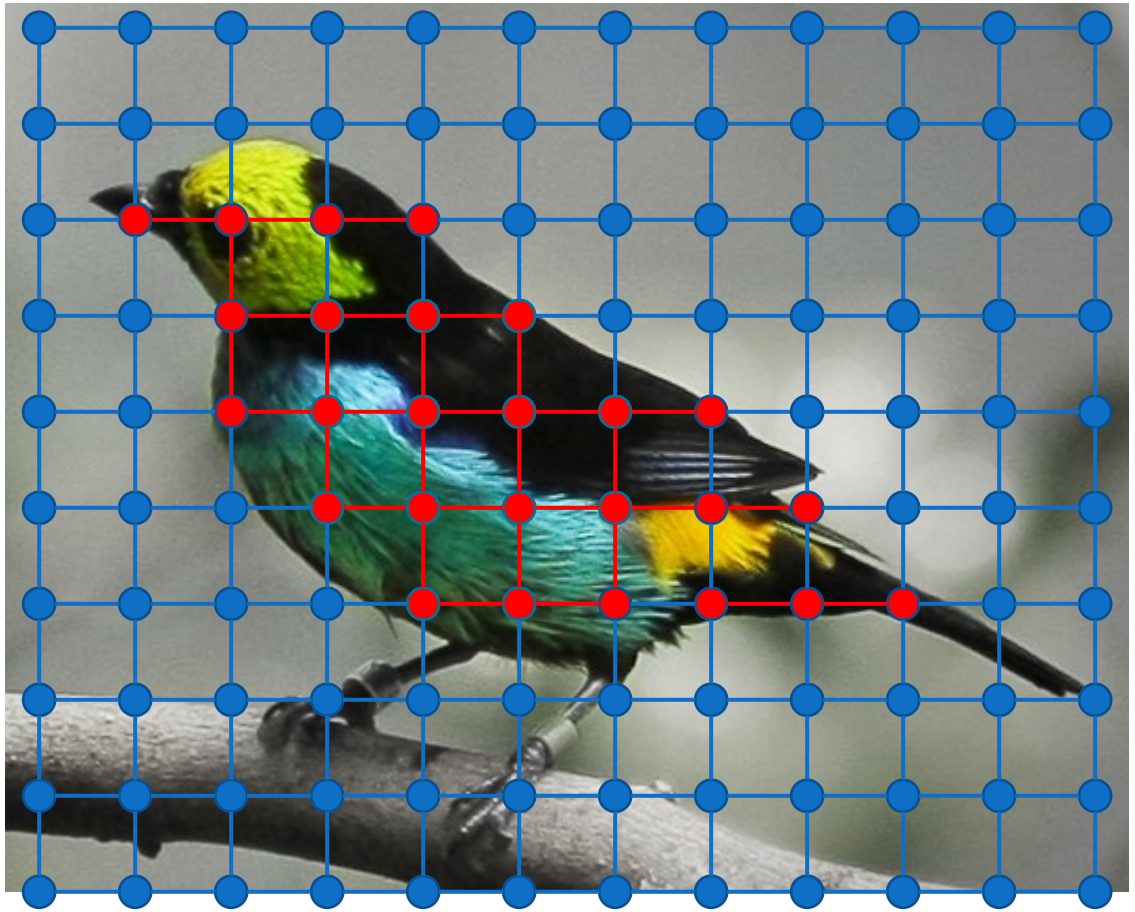
W(i, j) = 1/|RGB(i) – RGB(j)| is **low**



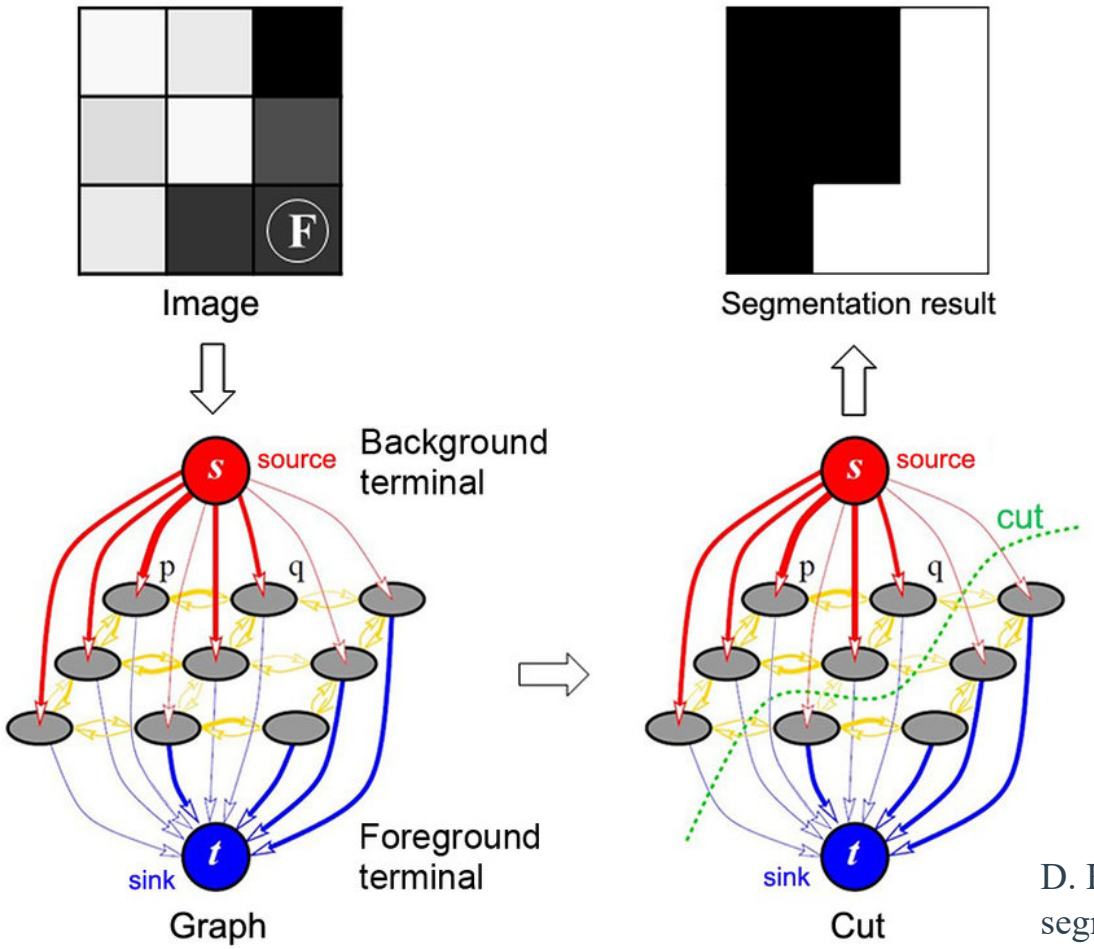
Graph Cuts 图割法

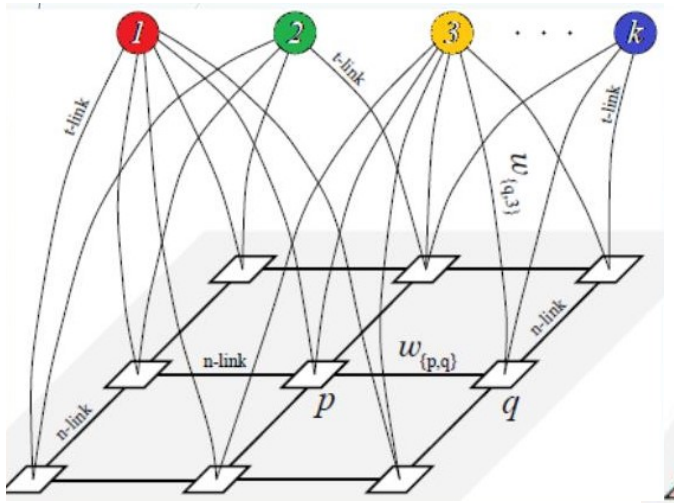
- Optimization solver

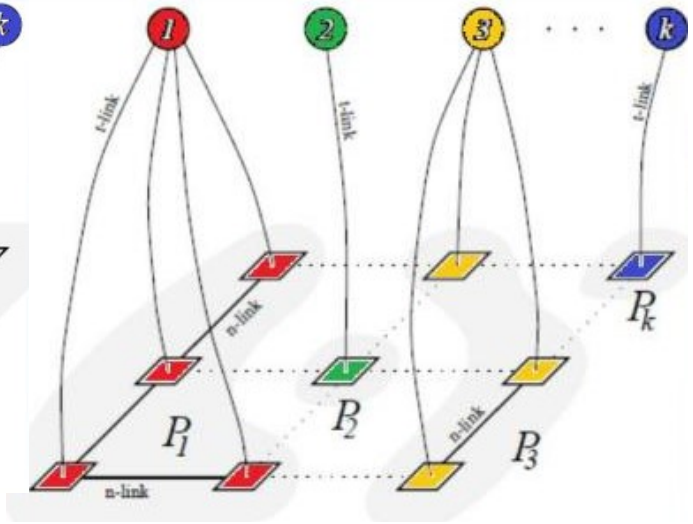
$$\text{minCut} = \min \sum W(i, j)$$



Graph Cuts 图割法 •Extendable to (some) Multi-label Grab Cuts

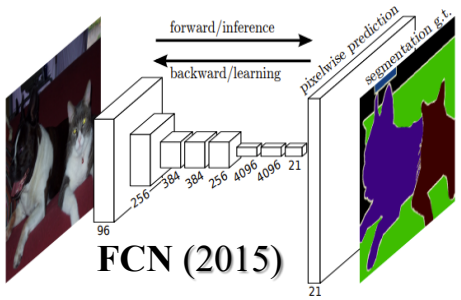




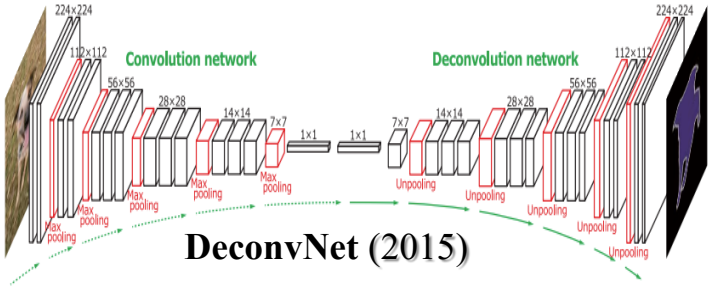


D. Khattab, H. M. Ebied, A. S. Hussein, M. F. Tolba. Multi-label automatic GrabCut for image segmentation. 14th International Conference on Hybrid Intelligent Systems, 2014, 152-157.

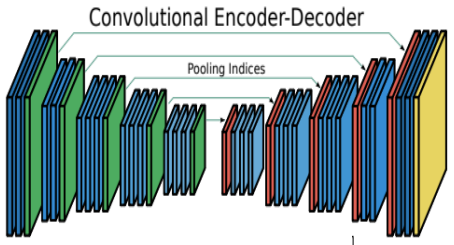
FCN → Segment Anything



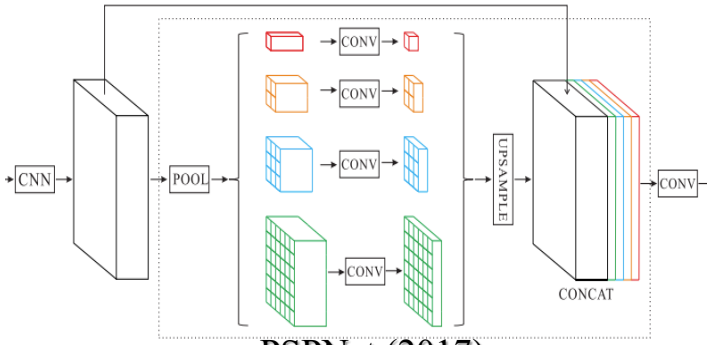
FCN (2015)



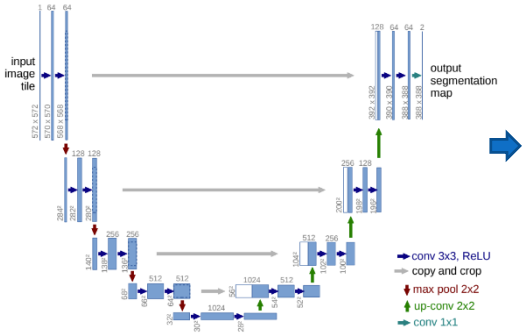
DeconvNet (2015)



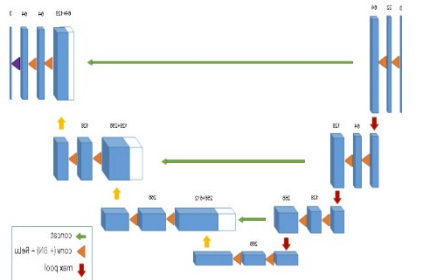
SegNet (2016)



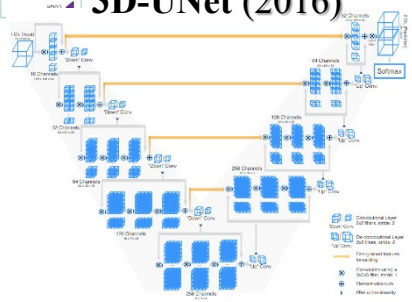
PSPNet (2017)



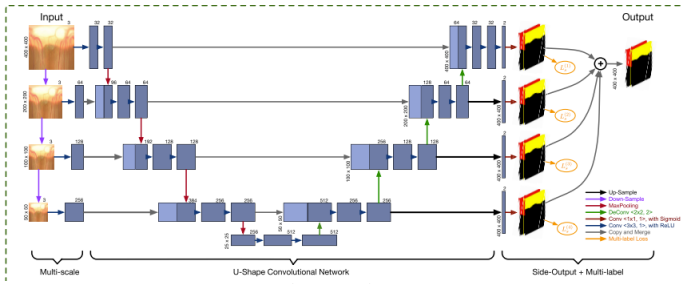
UNet (2015)



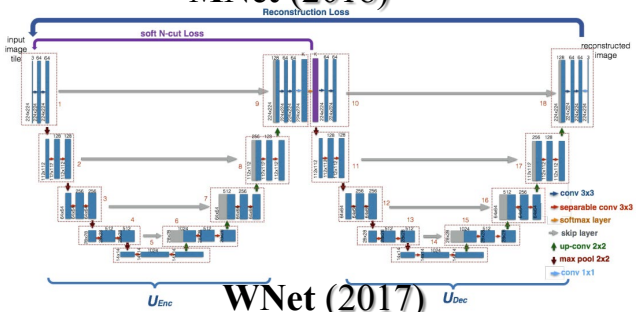
3D-UNet (2016)



VNet (2018)



MNet (2018)

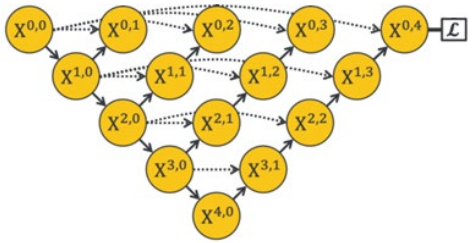


WNet (2017)

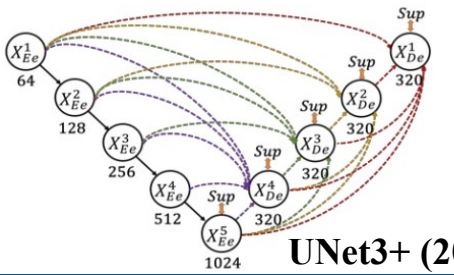
YNet (2018)

ZNet (2022)

nnUNet (2018)

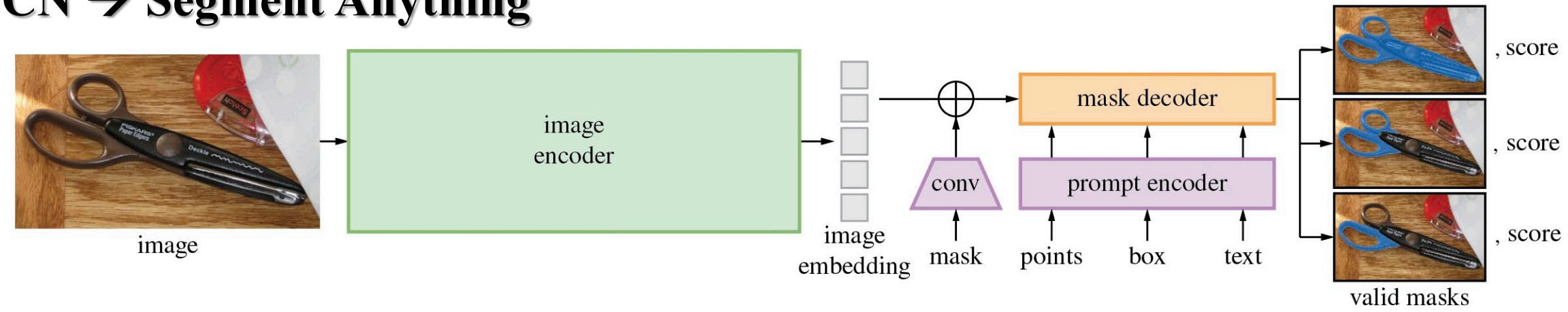


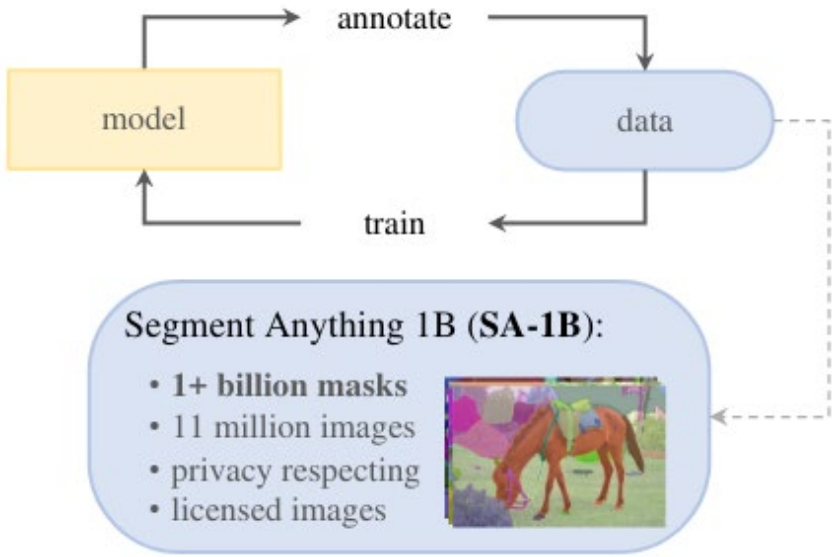
UNet++ (2020)



UNet3+ (2020)

FCN → Segment Anything

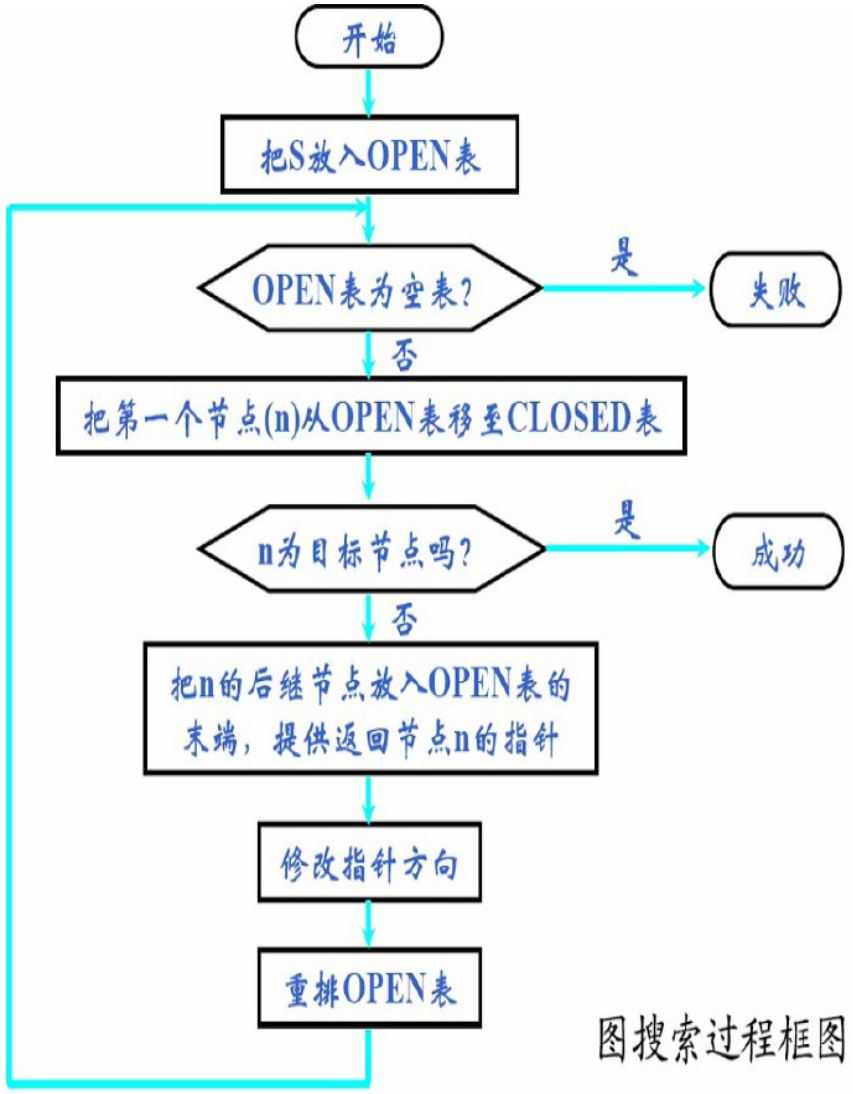




习题

习题6.2 P115

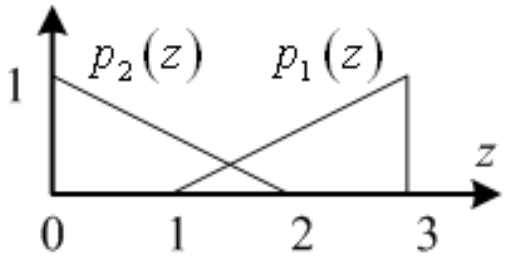
- 查阅有关人工智能的书籍，分析图搜索算法中各步骤地作用



图搜索过程框图

习题6.10 P115

- 设图像具有下图所示的灰度分布，其中$p_1(z)$对应目标，$p_2(z)$对应背景，若目标像素占整幅图像概率为0.5，即α=0.5，求分割目标和背景的最佳阈值。



$$p_1(z) = (z-1)/2$$

$$p_2(z) = 1 - z/2$$

根据最优阈值计算公式：

$$\alpha \cdot p_{obj}(z_k) = (1-\alpha) \cdot p_{bg}(z_k)$$

$$(z-1)/2 = 1 - z/2 \quad \Rightarrow \quad z = 3/2$$

习题6.11 P115

- 一幅图像背景部分均值为20，方差为400，在背景上分布着一些互不重叠的均值为200，方差为400的小目标。设所有目标合起来约占图像总面积的30%，提出一个基于取阈值的分割算法将这些目标分割出。

根据最优阈值计算公式，并假设背景和目标的概率密度函数为高斯模型，且背景和目标的方差相等：

$$z_k = \frac{\mu_{obj} + \mu_{bg}}{2} + \frac{\sigma^2}{\mu_{obj} - \mu_{bg}} \ln\left(\frac{1-\alpha}{\alpha}\right)$$

$$= \frac{20+200}{2} + \frac{400}{200-20} \ln\left(\frac{0.3}{0.7}\right) \approx 110 - 2 = 108$$