

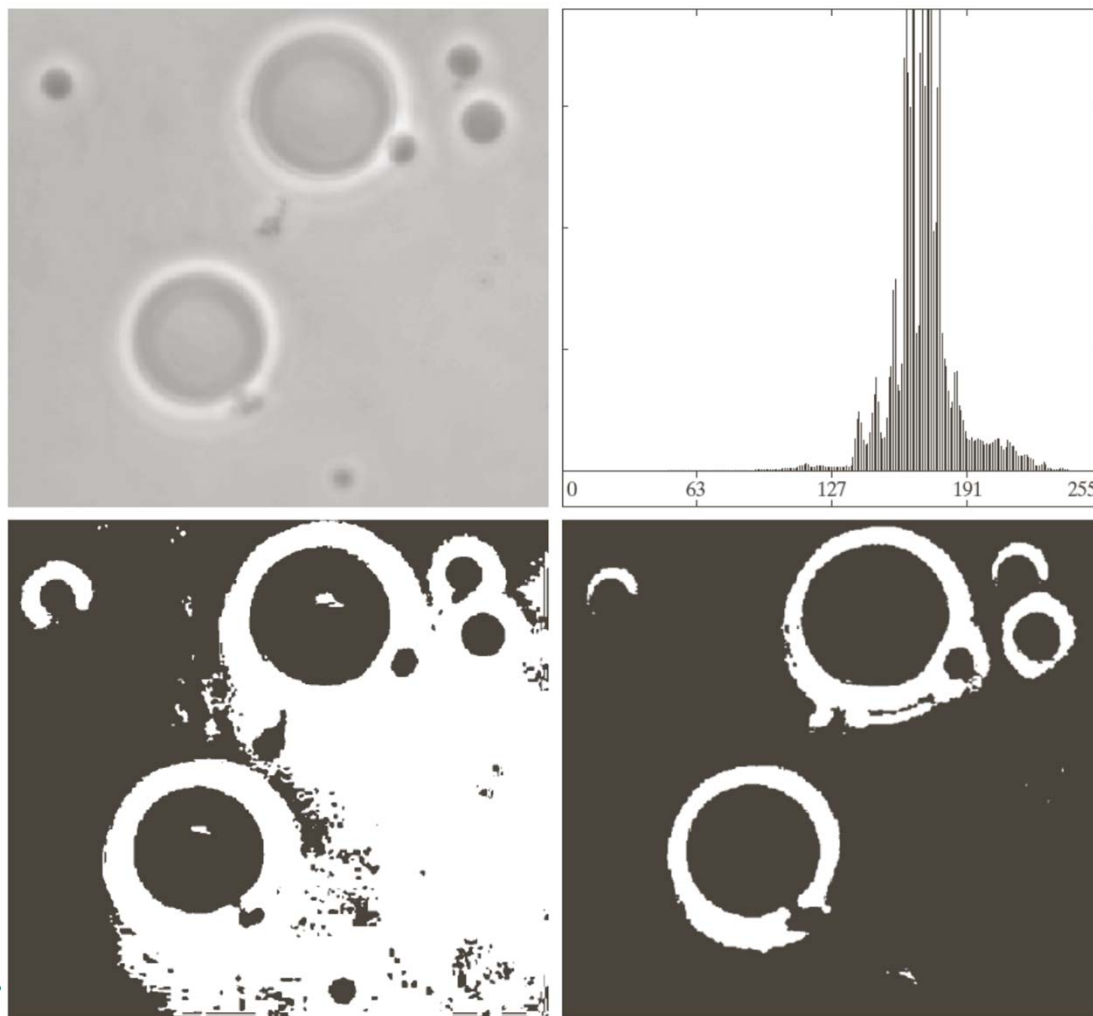
使用Otsu方法的全局最优阈值处理

- ❑ 假设图像中只有目标和背景，设置一个门限 T
 - 使分割后目标内部和背景内部总方差（称为类内方差）较小
 - 而目标与背景的灰度平均值之差又达较大（用类间方差表示）
- ❑ 对两类：
 - 像素频度分别为 P_1 和 P_2 ， $P_1+P_2=1$
 - 平均灰度值分别为 μ_1 和 μ_2
 - 方差分别为 σ_1^2 和 σ_2^2
- ❑ 整幅图的平均灰度为 μ_t ，则类内方差为

$$\sigma_w^2 = P_1\sigma_1^2 + P_2\sigma_2^2$$

- 类间方差为 $\sigma_B^2 = P_1(\mu_1 - \mu_t)^2 + P_2(\mu_2 - \mu_t)^2$
 - 因 $\mu_t = P_1\mu_1 + P_2\mu_2$ 代入
 - 有 $\sigma_B^2 = P_1P_2(\mu_1 - \mu_2)^2 = \frac{P_1}{(1-P_1)} \cdot (\mu_t - \mu_1)^2$
 - 取使类间方差最大(即 $\frac{\sigma_B^2}{\sigma_w^2}$ 最大)的 \mathbf{T} 作为最佳阈值
 - 为计算方便, 可用使 σ_B^2 为最大来近似代替 $\frac{\sigma_B^2}{\sigma_w^2}$ 最大
 - 先计算 $P_1 = \sum_{i=0}^T p(i)$ $\mu_1 = \sum_{i=0}^T i \cdot p(i) / P_1$, 再计算 σ_B^2
 - 使用穷举法, 找到使 σ_B^2 最大的 \mathbf{T} (最大值不唯一, 可取平均)
 - 最佳阈值处的归一化度量 $\eta(k^*) = \sigma_B^2(k^*) / \sigma_G^2$ 作为类别可分性的定量估计;
 - 已有一些迭代算法, 用于快速找到使 σ_B^2 最大的阈值 \mathbf{T}

基本全局阈
值处理法：
T=169



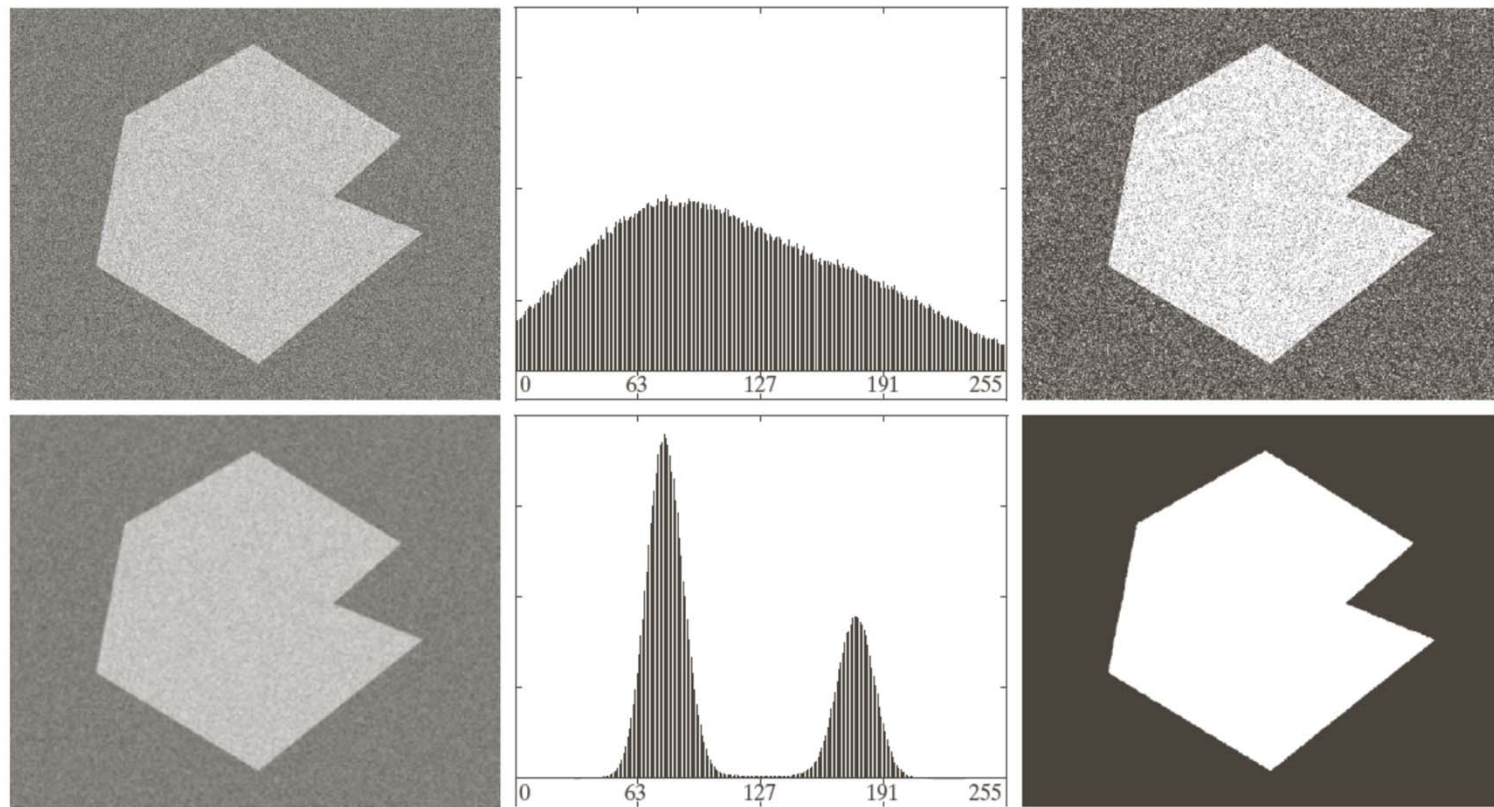
a b
c d

FIGURE 10.39

(a) Original image.
(b) Histogram (high peaks were clipped to highlight details in the lower values).
(c) Segmentation result using the basic global algorithm from Section 10.3.2.
(d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

Otsu法: T=181,
可分性度量: $\eta = 0.467$

用图像平滑改善全局阈值处理



平滑带来的
巨大改进!

a	b	c
d	e	f

FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

前景相对背景大小对分割的影响

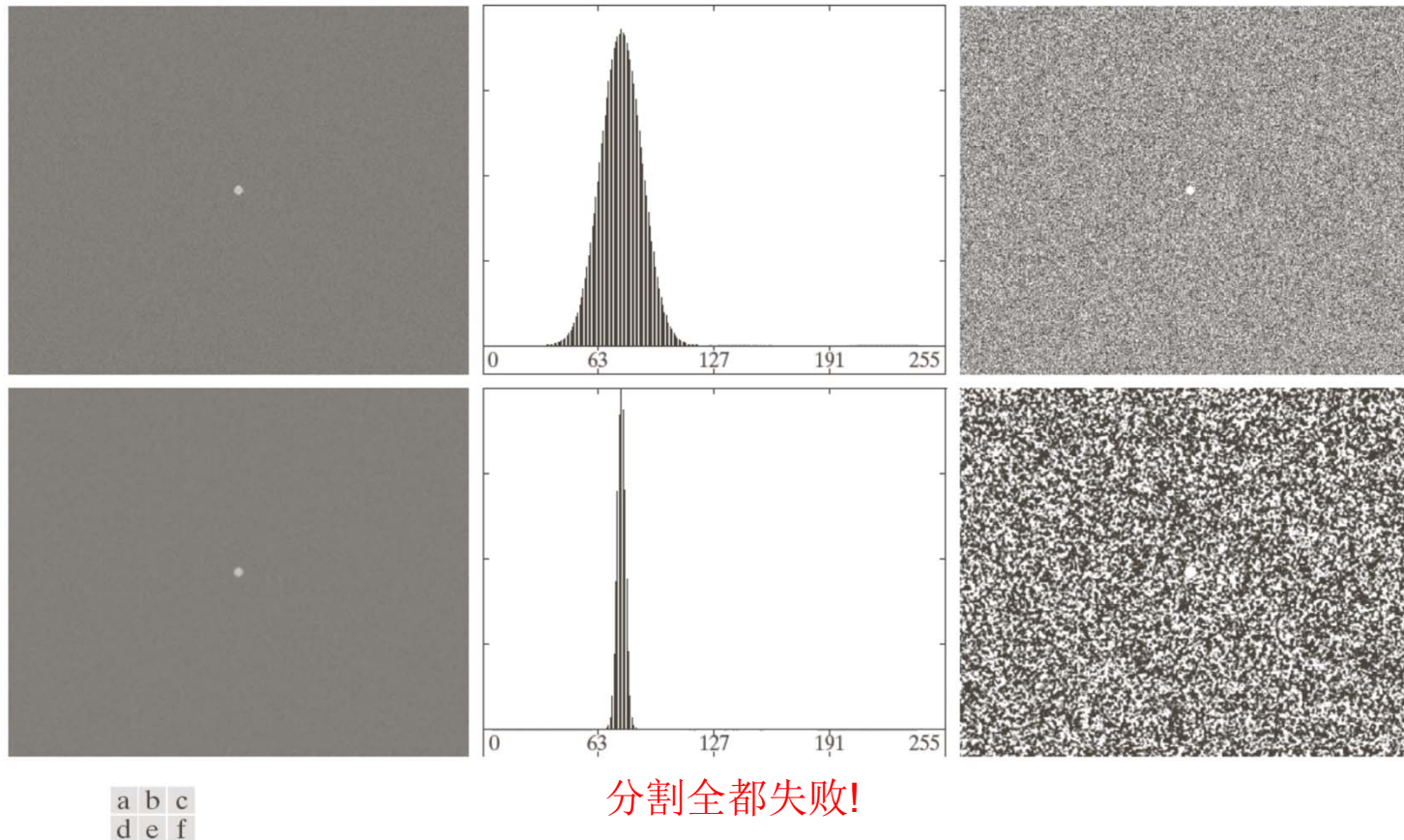


FIGURE 10.41 (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.

利用边缘改进全局阈值处理

- ❑ **观察：** 如果直方图的两个波峰高、窄、对称，波谷深，则容易找到好的全局阈值。
- ❑ **思想：**
 - 1. 仅用位于或者接近边缘的像素，则得到的直方图将有几个高度近似相同的波峰。
 - 2. 像素位于物体和背景上的概率近似相等，改善了直方图模式的对称性；
 - 3. 用满足基于梯度和拉普拉斯算子的简单度量的像素有加深直方图波峰间的波谷的倾向。
- ❑ **算法：**
 - 1. 计算原图的边缘图；
 - 2. 指定阈值 T ，并对边缘图进行二值处理，获取强边缘点，作为模板图像；
 - 3. 仅用模板图像对应的**原图像素**计算直方图；
 - 4. 利用获得的边缘像素直方图，做全局分割（如**Otsu**方法等）。

利用梯度为主的边缘信息改善分割

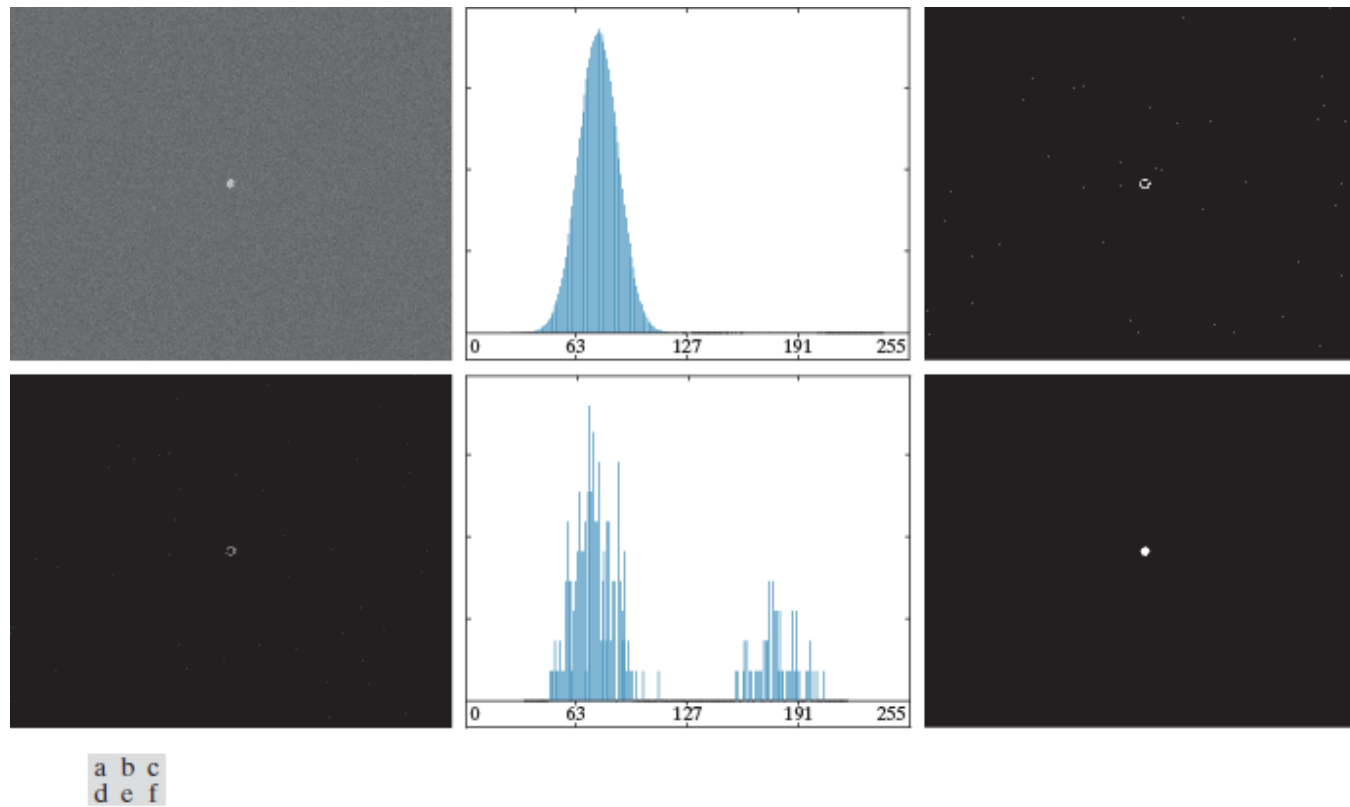


FIGURE 10.39 (a) Noisy image from Fig. 10.38(a) and (b) its histogram. (c) Mask image formed as the gradient magnitude image thresholded at the 99.7 percentile. (d) Image formed as the product of (a) and (c). (e) Histogram of the nonzero pixels in the image in (d). (f) Result of segmenting image (a) with the Otsu threshold based on the histogram in (e). The threshold was 134, which is approximately midway between the peaks in this histogram.

利用拉普拉斯为基础的边缘信息改进全局阈值处理

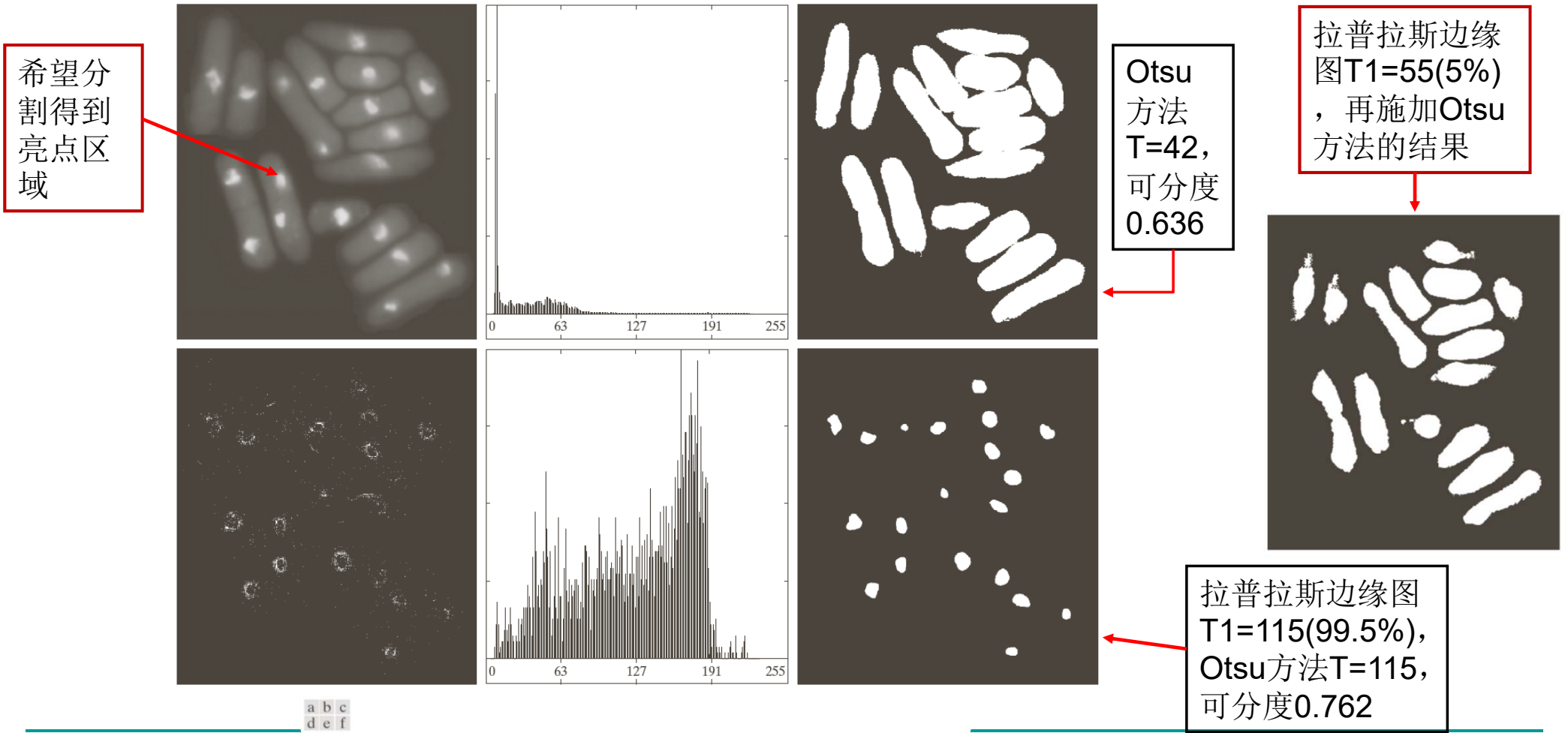


FIGURE 10.43 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the histogram in (b). (d) Thresholded absolute Laplacian. (e) Histogram of the nonzero pixels in the product of (a) and (d). (f) Original image thresholded using Otsu's method based on the histogram in (e). (Original image courtesy of Professor Susan L. Forsburg, University of Southern California.)

多阈值处理（多类的Otsu方法）

- 在k个类 c_1, c_2, \dots, c_k 的情况下，类间方差可归纳为

$$\sigma_B^2 = \sum_{k=1}^K P_k (m_k - m_G)^2$$

- 其中 $P_k = \sum_{i \in C_k} p_i$, $m_k = \frac{1}{P_k} \sum_{i \in C_k} ip_i$

- K类由K-1个阈值来分离，即使下式成立：

$$\sigma_B^2(k_1^*, k_2^*, \dots, k_{K-1}^*) = \max_{0 < k_1 < k_2 < \dots < L-1} \sigma_B^2(k_1, k_2, \dots, k_{K-1})$$

同样采用全局遍历方法可以求解（对多个 k 设置嵌套for 循环）

可分性度量： $\eta(k_1^*, k_2^*) = \frac{\sigma_B^2(k_1^*, k_2^*)}{\sigma_G^2}$

三个类的分割例子

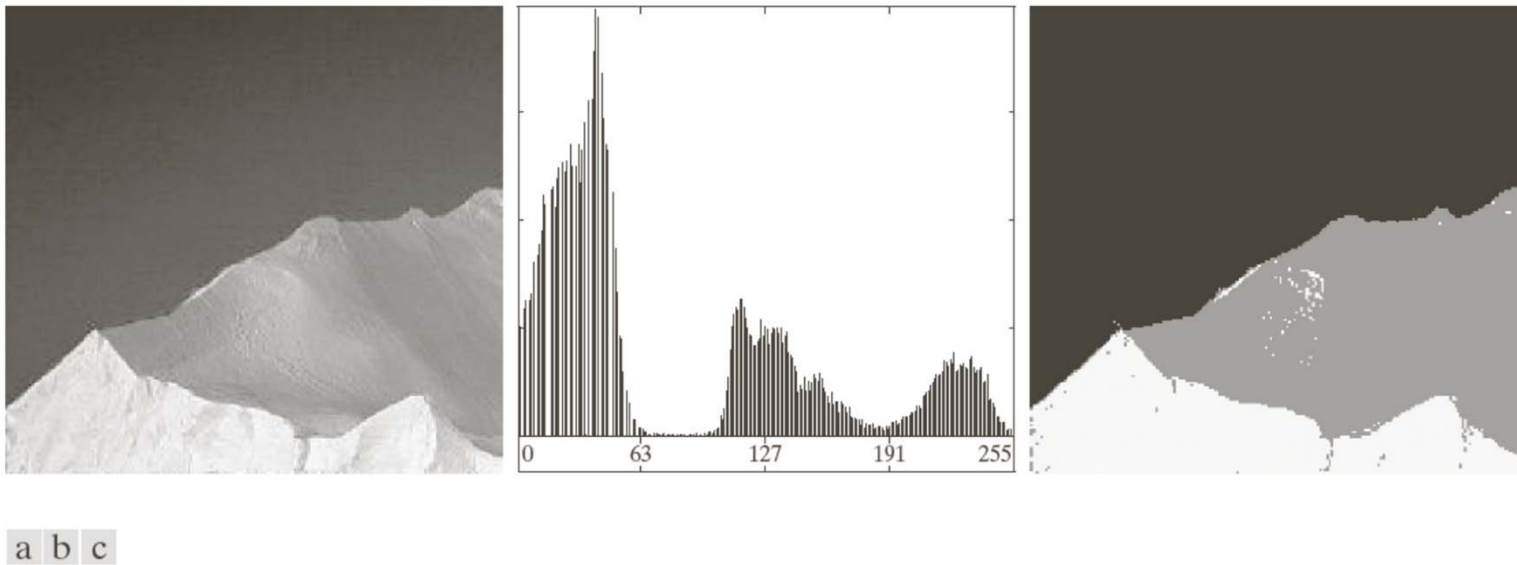


FIGURE 10.45 (a) Image of iceberg. (b) Histogram. (c) Image segmented into three regions using dual Otsu thresholds. (Original image courtesy of NOAA.)

可变阈值处理

- 局部阈值法

- **问题:**光照不均匀或透射不均匀时，整幅图像的分割没有合适的单一门限。

- 可采用方法:

- **1.阴影校正**

- 先采集一幅没有目标的背景图像，与有目标的图相减，从而消除光照不均匀现象

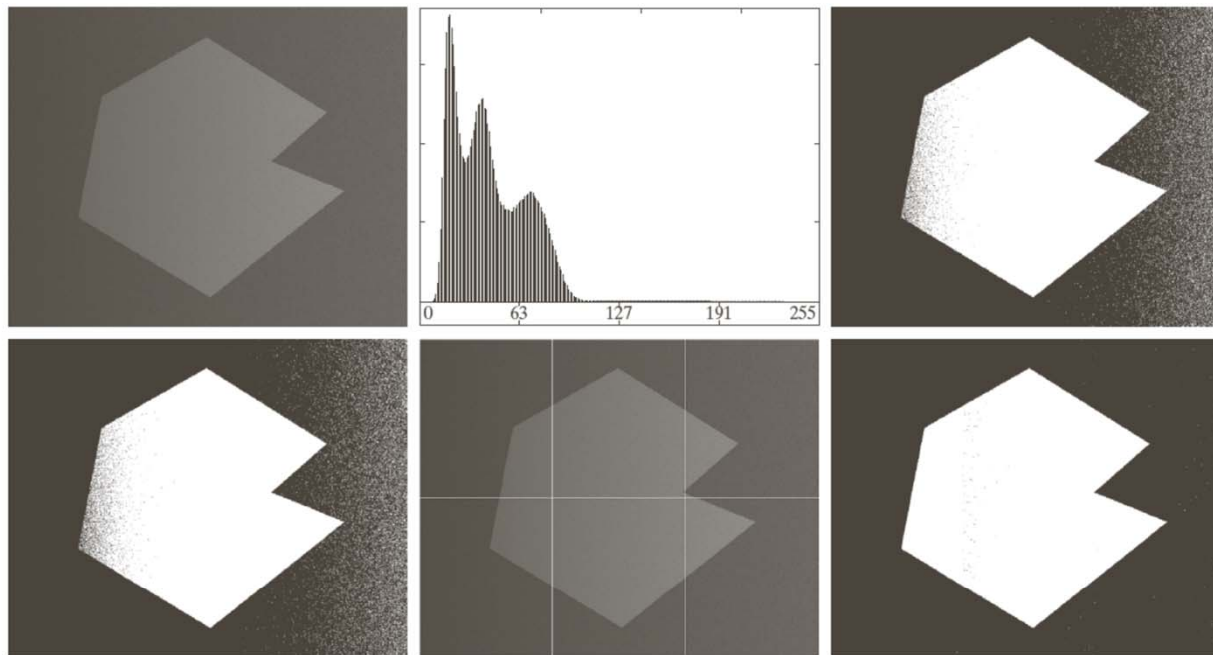
- **2.局部阈值法 (无法得到背景图像时)**

- 把图像分块，对每块选取一局部门限进行分割

- 如果某图像块内有目标和背景，则直方图呈双峰，据此可确定局部门限

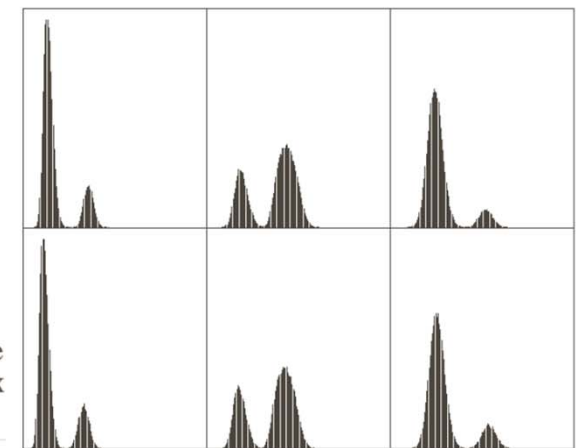
- 如果图像块内只有目标或只有背景，则直方图不呈双峰，找不出合适的局部门限值，但可根据邻近的双峰的块得到局部门限，通过内插给该图像块规定一门限，从而进行分割

通过图像分块的可变阈值分割



a b c
d e f

FIGURE 10.46 (a) Noisy, shaded image and (b) its histogram. (c) Segmentation of (a) using the iterative global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (e) Image subdivided into six subimages. (f) Result of applying Otsu's method to each subimage individually.



基于局部图像特性的可变阈值处理

- 为每个像素点计算一个阈值（自适应）：
 - 在每个像素点的一个邻域内的像素的**标准差**和**均值**，它们是局部对比度和平均灰度的描述子。
 - 使用以 (\mathbf{x}, \mathbf{y}) 的参数为基础的属性：

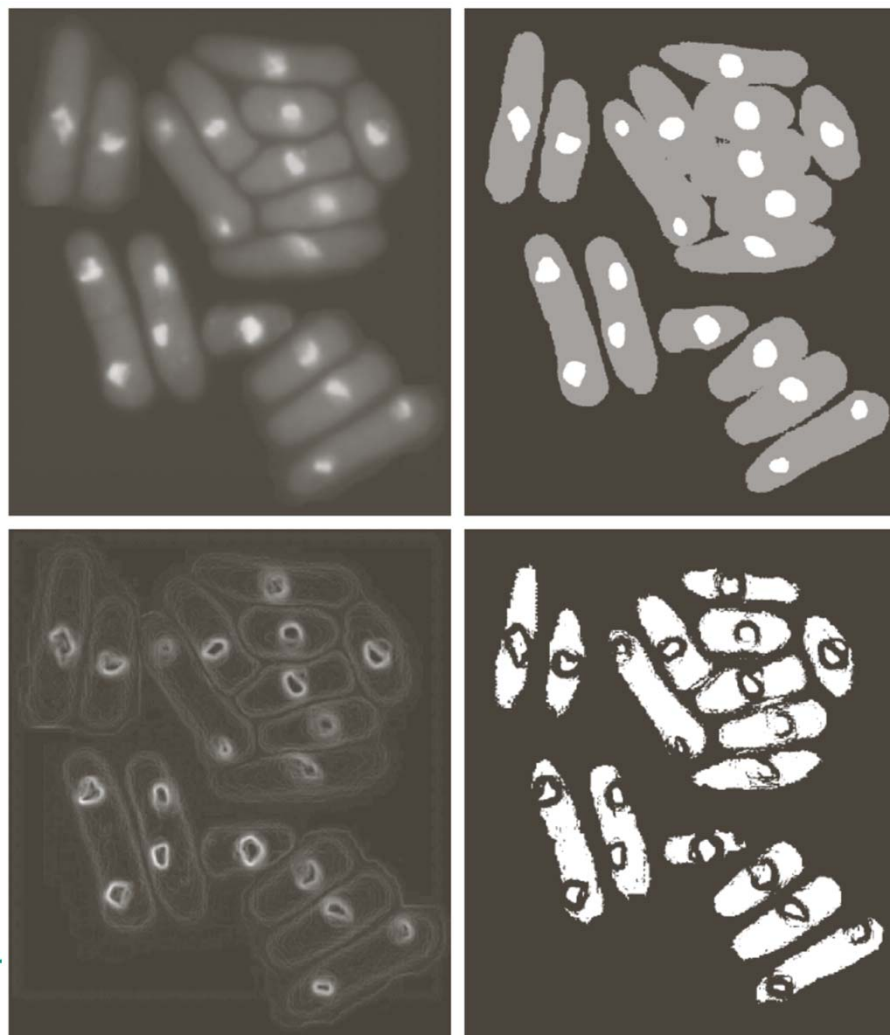
- $$g(x, y) = \begin{cases} 1, & Q = True \\ 0, & Q = False \end{cases} \quad \text{其中:}$$

$$Q(\sigma_{xy}, m_{xy}) = \begin{cases} True, & f(x, y) > a\sigma_{xy} \text{ AND } f(x, y) > bm_{xy} \text{ (or } bm_G) \\ False, & \text{others} \end{cases}$$

排除边缘像素，因此能
把物体边缘更容易分开。
通常边缘的方差较大，
该条件不易满足

背景近似恒定时，获得
前景像素

基于局部图像特性的可变阈值处理



a b
c d

FIGURE 10.43

(a) Image from Fig. 10.40.

(b) Image segmented using the dual thresholding approach given by Eq. (10-76).

(c) Image of local standard deviations.

(d) Result obtained using local thresholding.

3*3邻域内计算
均值和方差

$a=30$, $b=1.5$

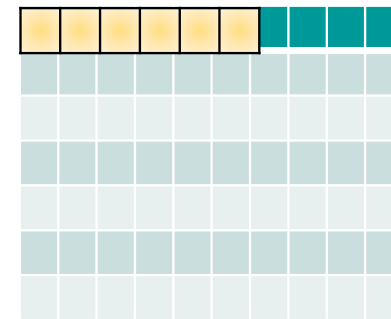
使用移动平均的可变阈值处理

- 对文档处理特别适合
- 对图像中的每个像素点计算移动平均：

$$m_{xy}(k+1) = \frac{1}{n} \sum_{i=k+2-n}^{k+1} z_i = m_{xy}(k) + \frac{1}{n} (z_{k+1} - z_{k-n})$$

- 用如下式子计算分割

$$g(x, y) = \begin{cases} 1, & f(x, y) > T_{xy} \\ 0, & f(x, y) \leq T_{xy} \end{cases}, \text{ where } T_{xy} = b m_{xy}$$



- 适用于感兴趣的物体与图像尺寸相比较小（较细） 场合。相当于滑动分块分割。

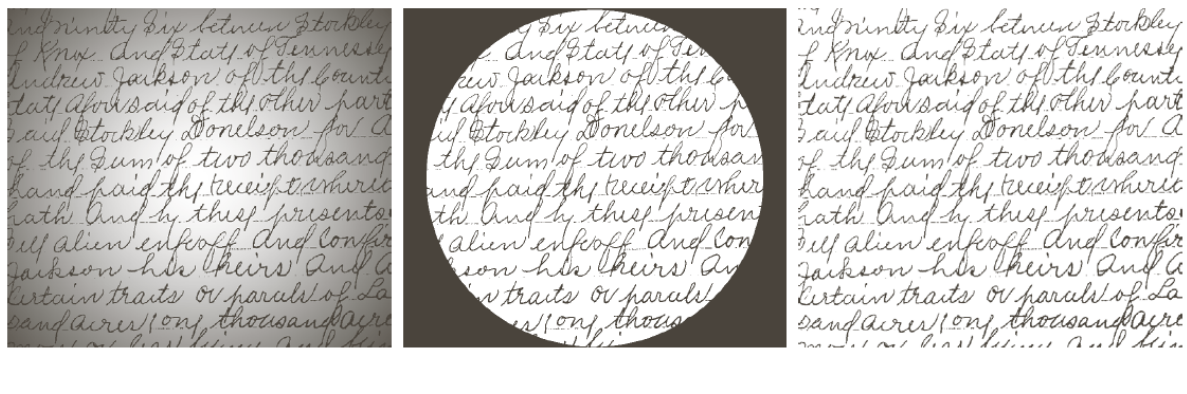


FIGURE 10.49 (a) Text image corrupted by spot shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

窗口宽度 n 取平均笔画宽度的5倍，此例中 $n=20$, $b=0.5$ 。

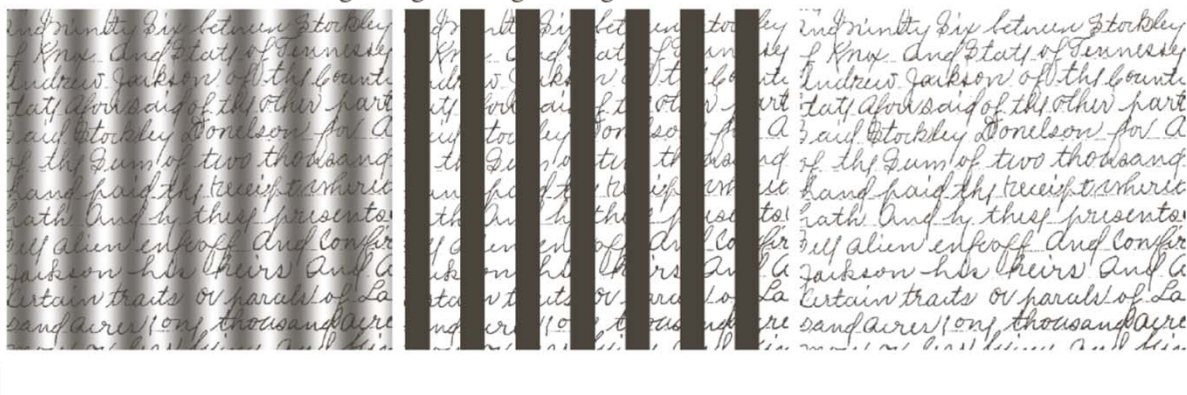
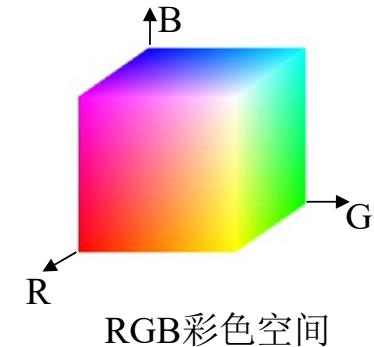


FIGURE 10.50 (a) Text image corrupted by sinusoidal shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

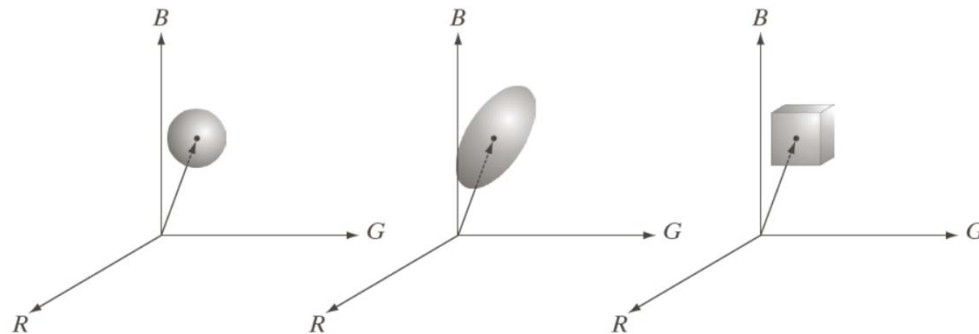
多变量阈值处理（彩色图像）

- 阈值分割模型（矢量阈值分割）

$$g(x, y) = \begin{cases} 1, & D(\mathbf{z}, \mathbf{a}) < T \\ 0, & \text{Others} \end{cases}$$



- 其中： $D(\mathbf{z}, \mathbf{a}) = \|\mathbf{z} - \mathbf{a}\| = \left[(\mathbf{z} - \mathbf{a})^T (\mathbf{z} - \mathbf{a}) \right]^{1/2}$ （欧氏距离）
- 或者： $D(\mathbf{z}, \mathbf{a}) = \|\mathbf{z} - \mathbf{a}\| = \left[(\mathbf{z} - \mathbf{a})^T \mathbf{C}^{-1} (\mathbf{z} - \mathbf{a}) \right]^{1/2}$ （马氏距离）

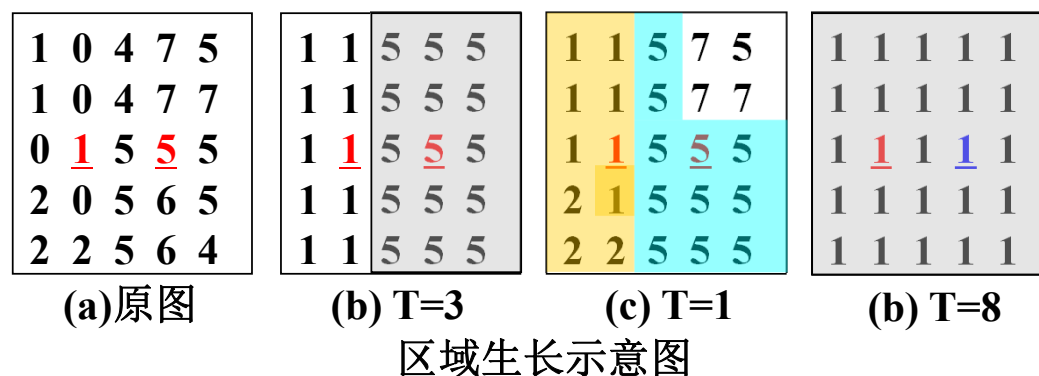


8.4 基于区域的分割

区域生长

- 基本思想:
 - 将具有相似性质的像素聚集到同一区域
- 方法:
 - 先对每个需要分割的区域找一个种子像素作为生长的起点
 - 然后将种子像素周围与种子像素有相同或相似性质的像素，加入到种子像素所在的区域中（注意生长过程中要保持区域内的连通性）
 - 再将这些新加入的像素做为新的种子，继续进行上面的过程，直到不再有像素加入区域

- 在实际应用区域生长法时需要解决三个问题：
 - 选定一组能正确代表所需区域的种子像素；
 - 确定生长过程中将相邻像素加入进来的准则；
 - 设定过程停止的条件。



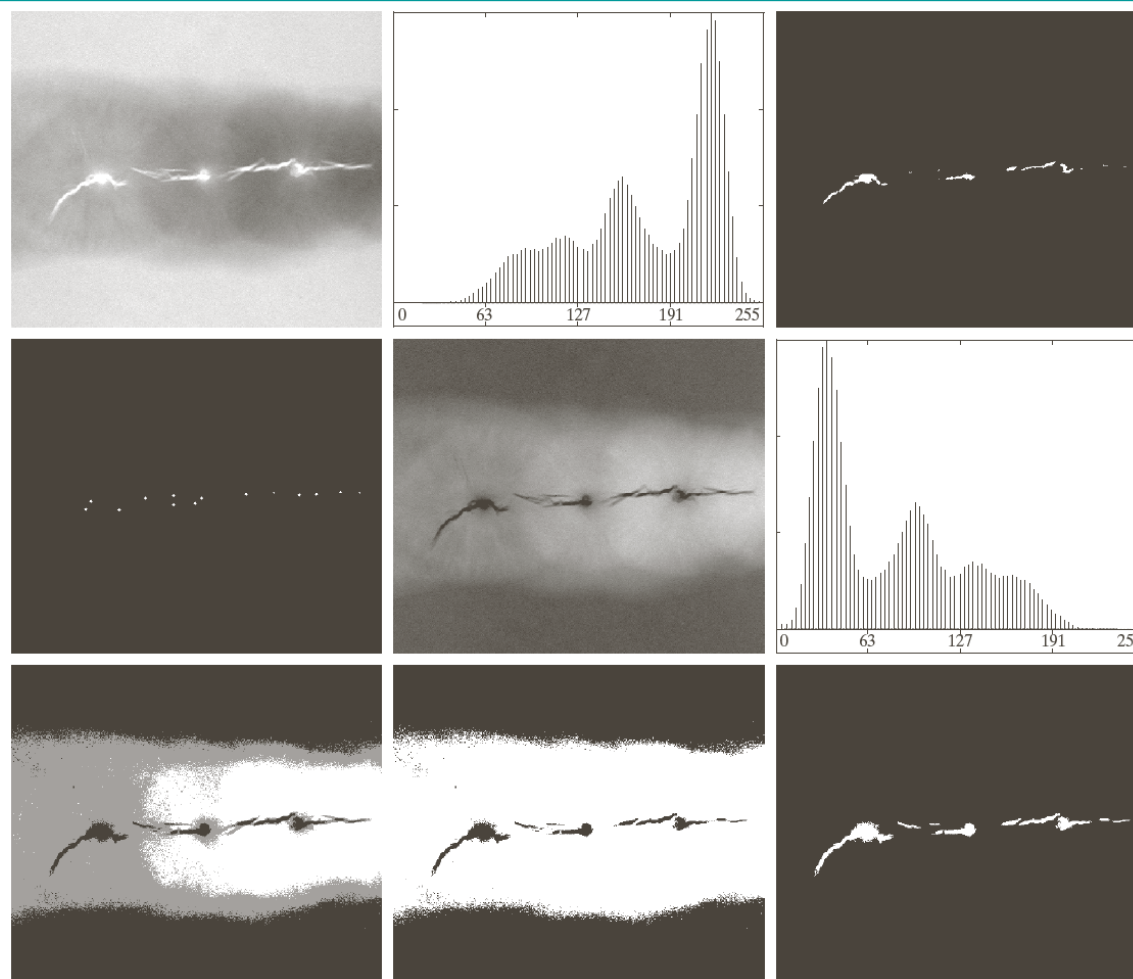
这里T是与选定种子像素的差值的阈值

腐蚀后的
种子点

双阈值分割法
作用于差值图
像结果

T1= 68;

T2=126



阈值为254
(99.5%)

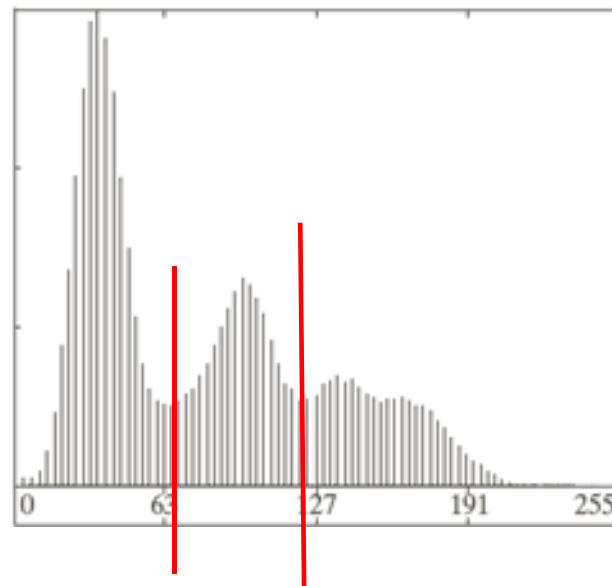
原图与种子
区域的差值
(相似性)
直方图

区域生长
最终结果

a b c
d e f
g h i

T=68

FIGURE 10.51 (a) X-ray image of a defective weld. (b) Histogram. (c) Initial seed image. (d) Final seed image (the points were enlarged for clarity). (e) Absolute value of the difference between (a) and (c). (f) Histogram of (e). (g) Difference image thresholded using dual thresholds. (h) Difference image thresholded with the smallest of the dual thresholds. (i) Segmentation result obtained by region growing. (Original image courtesy of X-TEK Systems, Ltd.)

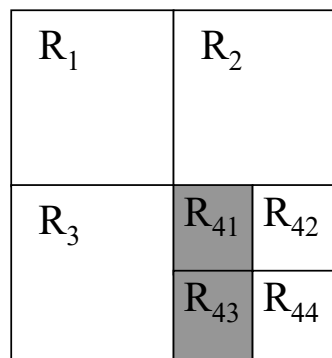


例,准则：（1）与种子像素间的灰度差绝对值小于65，这个数值是通过分析直方图获得（双阈值otsu分割法）；（2）加入像素须与已生长区域八连通。如果某个像素与多个生长区域连通，则将这些区域合并。

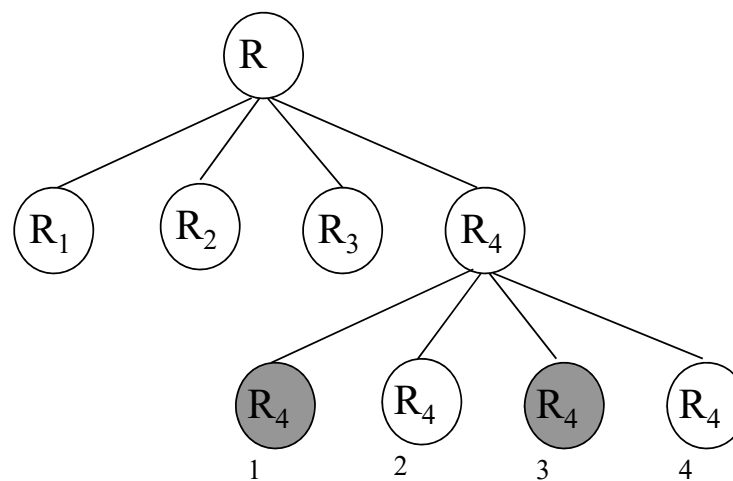
区域分裂与聚合

- 将一幅图像细分为一组任意的不相交区域，然后聚合或分裂这些区域，并使其满足分割条件。
 - 通常将图像按4象限方式分裂成4个区域，可用4叉树形式表示；
 - 对每个区域，检查内部相似性
 - 如果区域内部像素性质不一致，则对该区域进一步分裂
 - 如果相邻区域有相似的性质，则将它们合并
 - 被合并的区域可能有不同形状和大小

■ 分裂合并法 与 四叉树



(a) 被分割的图像



(b) 对应的四叉树

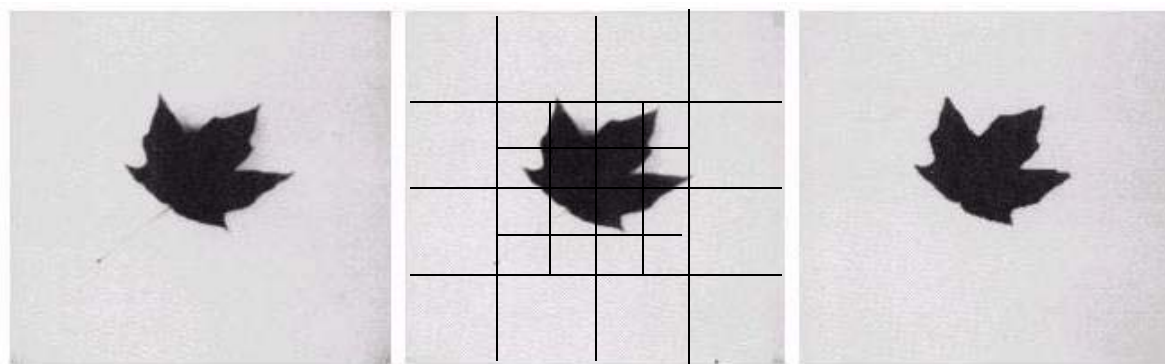
区域的分裂合并

- 采用区域分裂合并方法的例子

- 相似性准则:

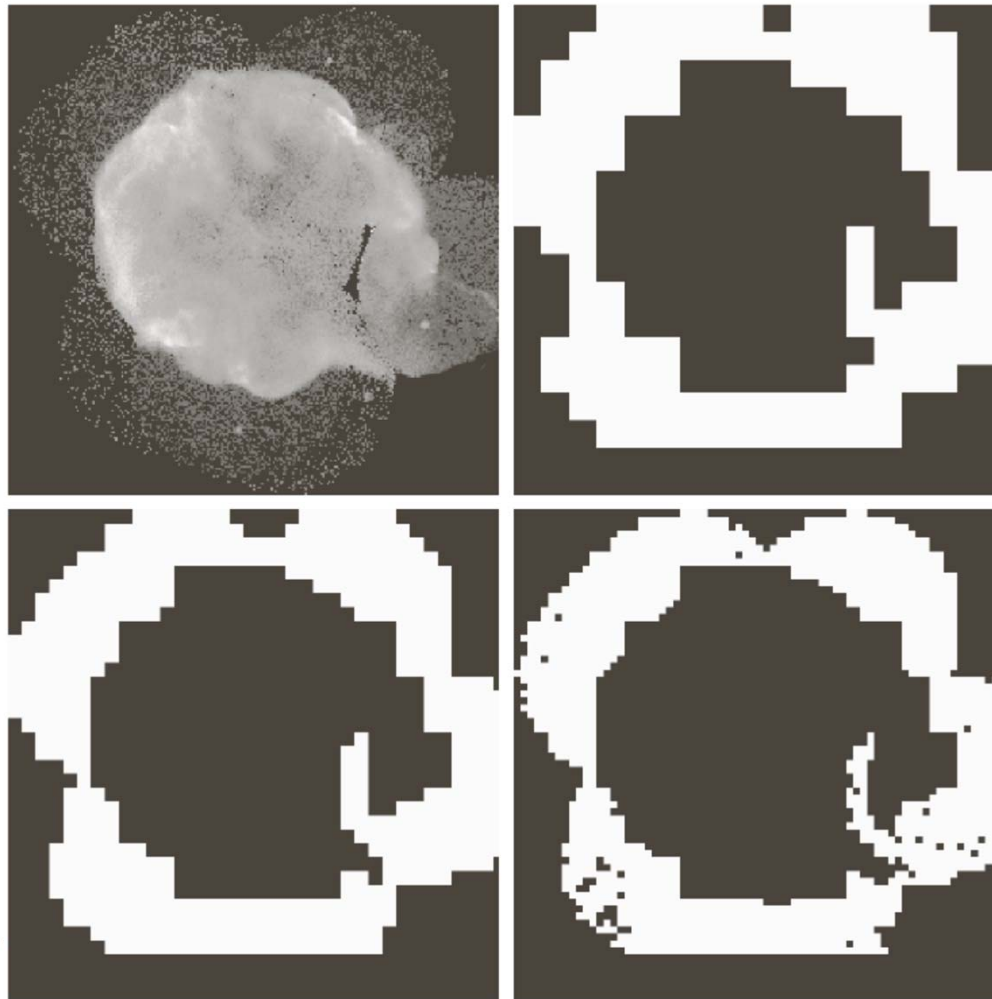
- 区域内至少有**80%**像素灰度值集中在其均值附近

$$|x_{ij} - m_i| \leq 2\sigma_i$$



区域的分裂合并示意图

利用分裂聚合法对星座环分割



相似性准则:

$$Q = \begin{cases} \text{True}, & \sigma > a \quad \text{and} \quad 0 < m < b \\ \text{False}, & \text{Others} \end{cases}$$

a	b
c	d

FIGURE 10.53

(a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32×32 , 16×16 , and 8×8 pixels, respectively. (Original image courtesy of NASA.)

使用k均值聚类的区域分割

- 基本思想：将向量观测集合 \mathbf{Q} 划分为规定数量 k 的聚类。在聚类过程中，每个观测值都被分配给具有最近均值的类别。

- 令 $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}$ ，k-means 方法的优化目标为

$$\arg \min_C \left(\sum_{i=1}^k \sum_{\mathbf{z} \in C_i} \|\mathbf{z} - \mathbf{m}_i\|^2 \right)$$

K-means具体算法

- (1) 初始化算法：规定一组初始均值 $\mathbf{m}_i(1), i = 1, 2, \dots, k$.
- (2) 将样本分配给聚类：将每个样本分配的均值最接近的类别集合，即：
$$\mathbf{z}_q \rightarrow C_i \text{ if } \|\mathbf{z}_q - \mathbf{m}_i\|^2 < \|\mathbf{z}_q - \mathbf{m}_j\|^2 \quad j = 1, 2, \dots, k (j \neq i); \quad q = 1, 2, \dots, Q$$
- (3) 更新聚类中心（均值）：
$$\mathbf{m}_i = \frac{1}{|C_i|} \sum_{\mathbf{z} \in C_i} \mathbf{z} \quad i = 1, 2, \dots, k$$
- (4) 完备性验证：计算当前步骤和前几步平均向量之间的差的欧氏范数。若该残差小于预设阈值，则停止；否则返回步骤（2）。

K-means 例子

a b

FIGURE 10.49

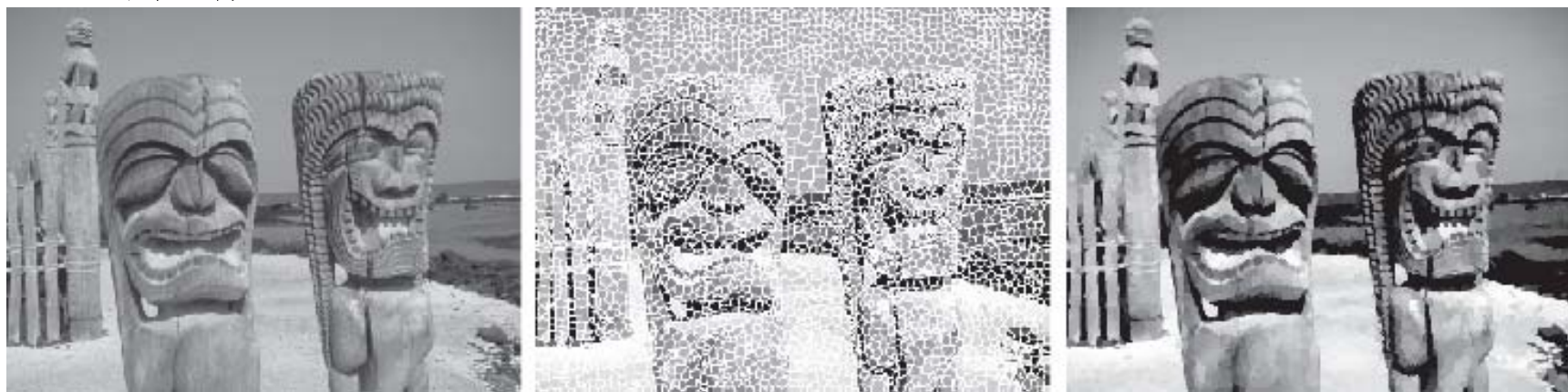
(a) Image of size 688×688 pixels.

(b) Image segmented using the k -means algorithm with $k = 3$.



使用超像素的区域分割

- 主要思想是通过将像素组合到比各个像素更有感知意义的原始区域来取代标准像素网格。
- 目的是减少计算开销，并通过减少不相关的细节来提高分割算法的性能。



a b c

FIGURE 10.50 (a) Image of size 600×480 (480,000) pixels. (b) Image composed of 4,000 superpixels (the boundaries between superpixels (in white) are superimposed on the superpixel image for reference—the boundaries are not part of the data). (c) Superpixel image. (Original image courtesy of the U.S. National Park Services.).

超像素数目的限制

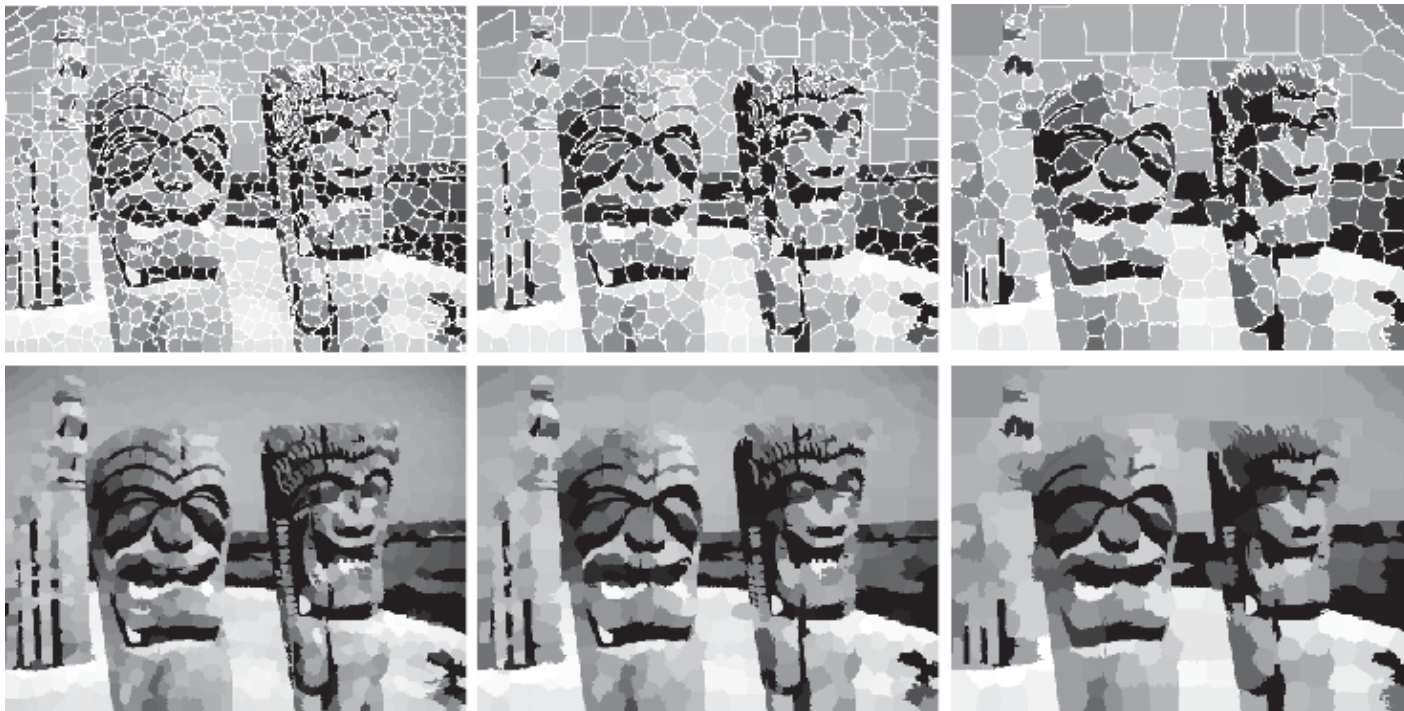


FIGURE 10.51 Top row: Results of using 1,000, 500, and 250 superpixels in the representation of Fig. 10.50(a). As before, the boundaries between superpixels are superimposed on the images for reference. Bottom row: Superpixel images.

SLIC超像素算法

- 通常使用包含三个颜色分量和两个空间坐标的五维向量

;

$$\mathbf{z} = \begin{bmatrix} r \\ g \\ b \\ x \\ y \end{bmatrix}$$

- 算法步骤见右侧。
- 与k-means相比:
 - (1) 距离D不规定为欧氏范数
 - ;
 - (2) 距离只针对 $2s \times 2s$ 区域范围内计算, 减少计算时间。

1. **Initialize the algorithm:** Compute the initial superpixel cluster centers,

$$\mathbf{m}_i = [r_i \ g_i \ b_i \ x_i \ y_i]^T, \ i = 1, 2, \dots, n_{sp}$$

by sampling the image at regular grid steps, s . Move the cluster centers to the lowest gradient position in a 3×3 neighborhood. For each pixel location, p , in the image, set a label $L(p) = -1$ and a distance $d(p) = \infty$.

2. **Assign samples to cluster centers:** For each cluster center \mathbf{m}_i , $i = 1, 2, \dots, n_{sp}$, compute the distance, $D_i(p)$ between \mathbf{m}_i and *each* pixel p in a $2s \times 2s$ neighborhood about \mathbf{m}_i . Then, for each p and $i = 1, 2, \dots, n_{sp}$, if $D_i < d(p)$, let $d(p) = D_i$ and $L(p) = i$.
3. **Update the cluster centers:** Let C_i denote the set of pixels in the image with label $L(p) = i$. Update \mathbf{m}_i :

$$\mathbf{m}_i = \frac{1}{|C_i|} \sum_{\mathbf{z} \in C_i} \mathbf{z} \quad i = 1, 2, \dots, n_{sp}$$

where $|C_i|$ is the number of pixels in set C_i , and the \mathbf{z} 's are given by Eq. (10-86).

4. **Test for convergence:** Compute the Euclidean norms of the differences between the mean vectors in the current and previous steps. Compute the residual error, E , as the sum of the n_{sp} norms. If $E < T$, where T a specified nonnegative threshold, go to Step 5. Else, go back to Step 2.
5. **Post-process the superpixel regions:** Replace all the superpixels in each region, C_i , by their average value, \mathbf{m}_i .

SLIC中的距离测度

- 颜色和空间坐标的尺度不同，且不相关，考虑分开处理，如：

$$d_c = [(r_j - r_i)^2 + (g_j - g_i)^2 + (b_j - b_i)^2]^{1/2} \quad d_s = [(x_j - x_i)^2 + (y_j - y_i)^2]^{1/2}$$

$$D = \left[\left(\frac{d_c}{d_{cm}} \right)^2 + \left(\frac{d_s}{d_{sm}} \right)^2 \right]^{1/2}$$

- 也可以令 $d_{cm} = c$ ，则有

$$D = \left[\left(\frac{d_c}{c} \right)^2 + \left(\frac{d_s}{s} \right)^2 \right]^{1/2} \quad \text{或者} \quad D = \left[d_c^2 + \left(\frac{d_s}{s} \right)^2 c^2 \right]^{1/2}$$

常数 c 用来加权颜色相似性和空间接近性之间的相对重要性。

使用超像素分割图像示例

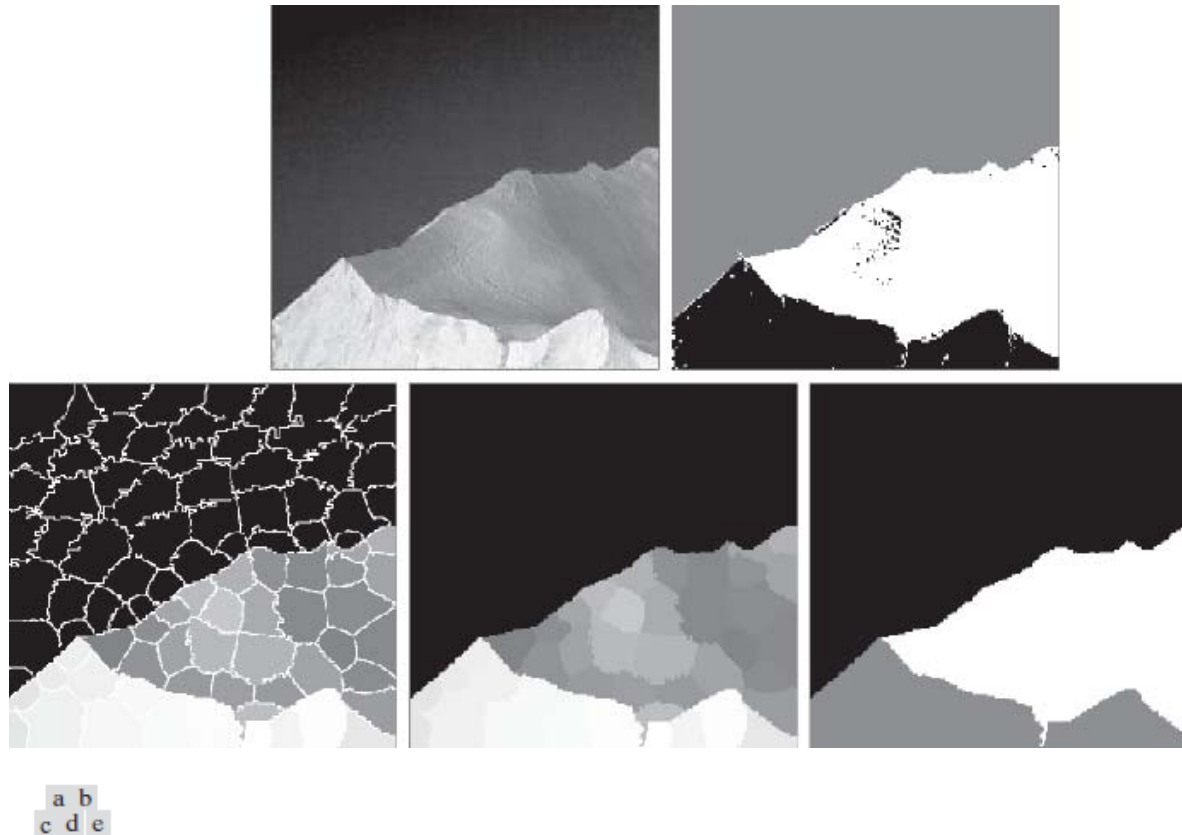


FIGURE 10.52 (a) Image of size 533×566 (301,678) pixels. (b) Image segmented using the k -means algorithm. (c) 100-element superpixel image showing boundaries for reference. (d) Same image without boundaries. (e) Superpixel image (d) segmented using the k -means algorithm. (Original image courtesy of NOAA.)

使用图割分割区域

■ 原理

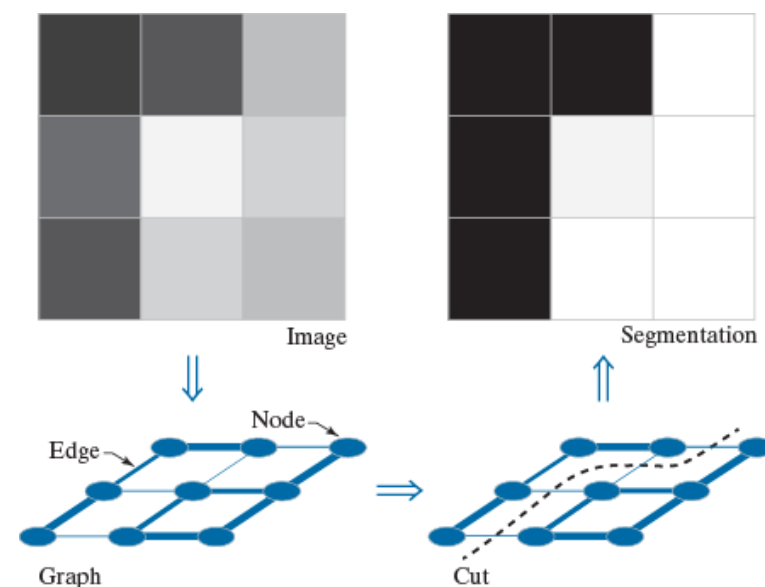
- ❑ 将图像像素表示为图的各个节点，求图到节点组的最优分割；

■ 最优准则

- ❑ 某个组内的成员的值较高，不同组中的成员的值较低

■ 特点

- ❑ 效果较优，计算复杂性较高

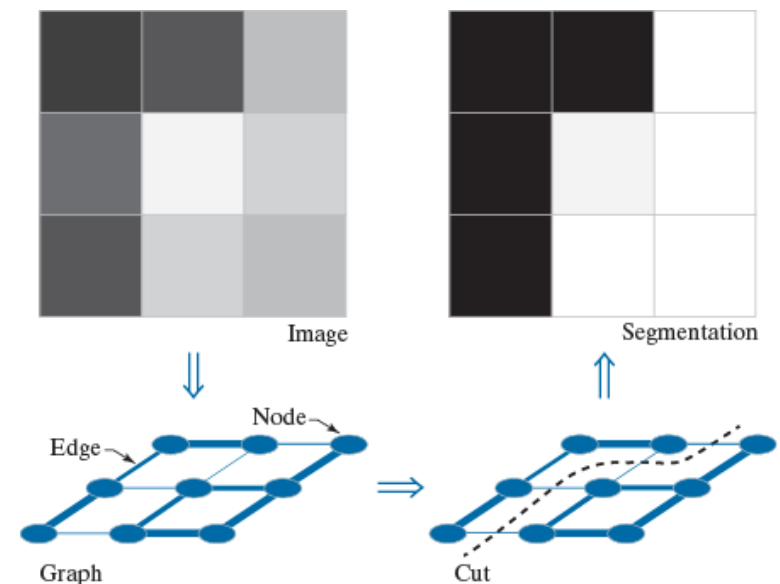


数学模型描述

- 图 G 示由节点集合 V 和连接这些顶点的边的结合 E 组成的数学结构：

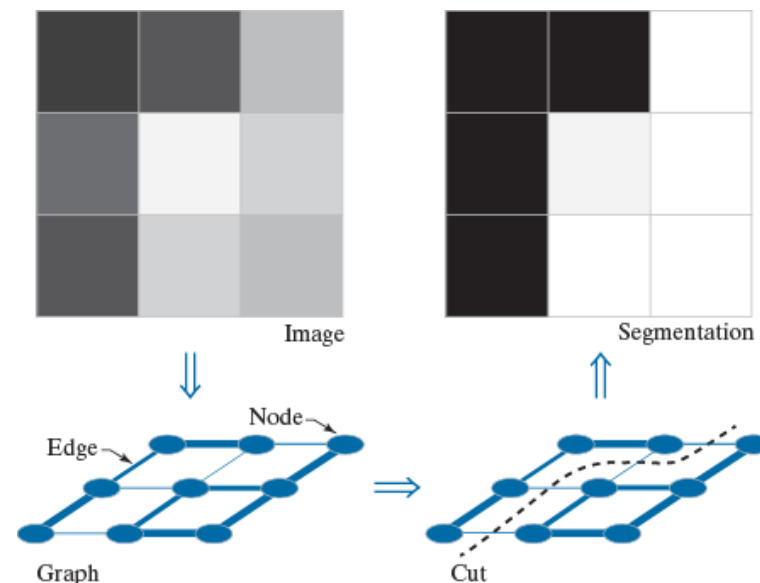
$$G = (V, E)$$

- V 是一个集合， $E \subseteq V \times V$ 是 V 中有序元素对的集合。
- 有向图与无向图。在分割中，我们感兴趣的主要是无向图。
- 边由矩阵 W 表征，矩阵元素 $w(i, j) = w(j, i)$ ，是连接节点 i 和 j 相关联的权重。
- W 是对称矩阵，权重与节点对之间的相似度成正比。边与权相关联的图称为加权图



图像分割问题

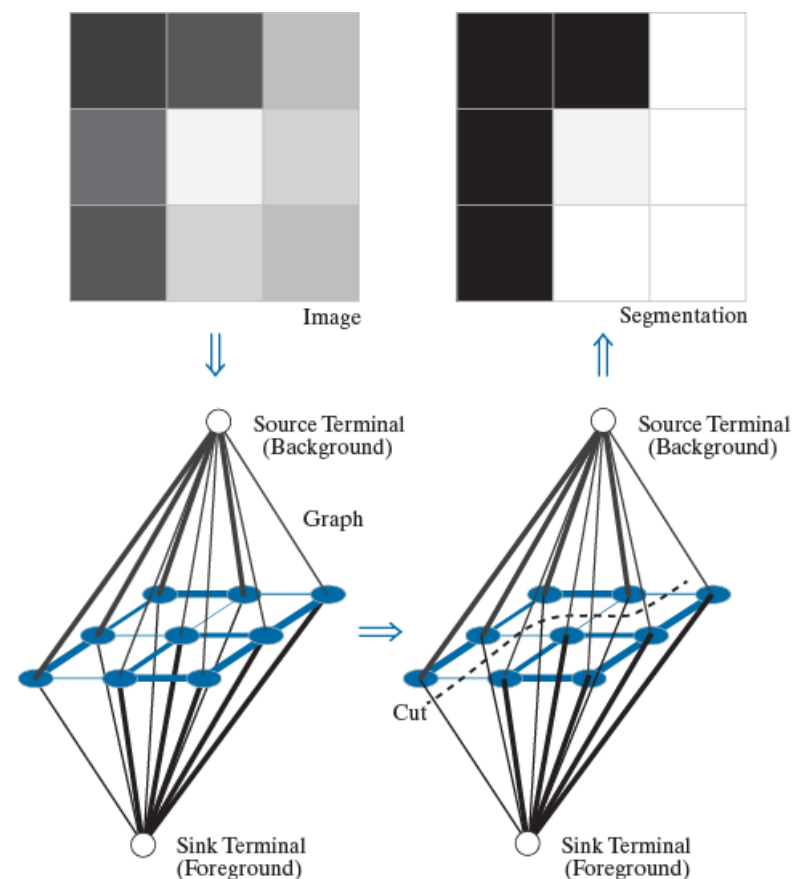
- 图的节点是像素（或者超像素）。通过将图 V 划分为多个不相交的子集。
- 关键问题：
 - 如何关联图和图像？
 - 边可以仅考虑4连通像素。
 - 边的权重通常由空间关系和灰度测度（如纹理和颜色）形成。
 - 如何切割图？
 - 图割方法



$$w(i, j) = 1 / (|I(n_i) - I(n_j)| + c)$$

图的另一种通用表示

- 增加了两个额外的节点：
 - ❑ 源终端节点与聚终端节点
 - ❑ 与之相连的边称为 **t** 边
 - ❑ 作用是将每个像素与是一个背景像素或者前景像素的概率关联起来，**t** 链的权重就是概率。



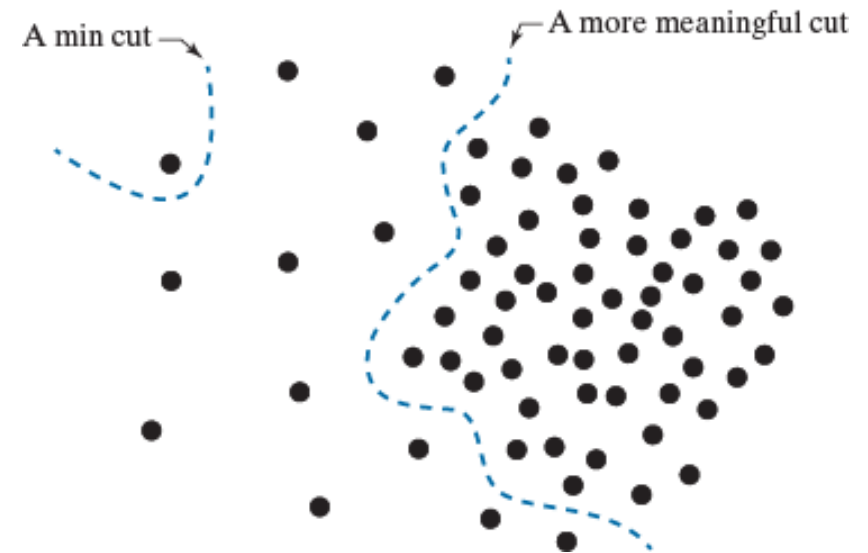
最小图割

■ 最大流最小割定理：

- 在流网络中，从源终端节点传递到汇聚终端节点的最大流量等于最小割。
- 最小割定义为边的最小权重集合，若删除它则会断开汇聚终端节点与源终端节点，其为

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

- ## ■ 直接应用该定理容易导致不正确的过小分割。



归一化割

- 采用“分离”测度来计算图中连接到所有节点总边的部分代价，定义归一化割为：

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

- 其中： $assoc(A, V) = \sum_{u \in A, z \in V} w(u, z)$

$$assoc(B, V) = \sum_{v \in B, z \in V} w(v, z)$$

- 无论 $cut(A, B)$ 为多少， $Ncut(A, B)$ 总是大于等于1。

- 定义：

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

- 则有

$$Ncut(A, B) = 2 - Nassoc(A, B)$$

- 说明最小化归一化割的时候会最大化集内节点到全图的权重比。

计算最小图割

- NP hard 问题
- Shi 和Malik [2000]通过把最小化表示为表示为一个广义的本征值问题，求得最小化Ncut的一个离散近似解。



a b c

FIGURE 10.56 (a) Image of size 600×600 pixels. (b) Image smoothed with a 25×25 box kernel. (c) Graph cut segmentation obtained by specifying two regions.

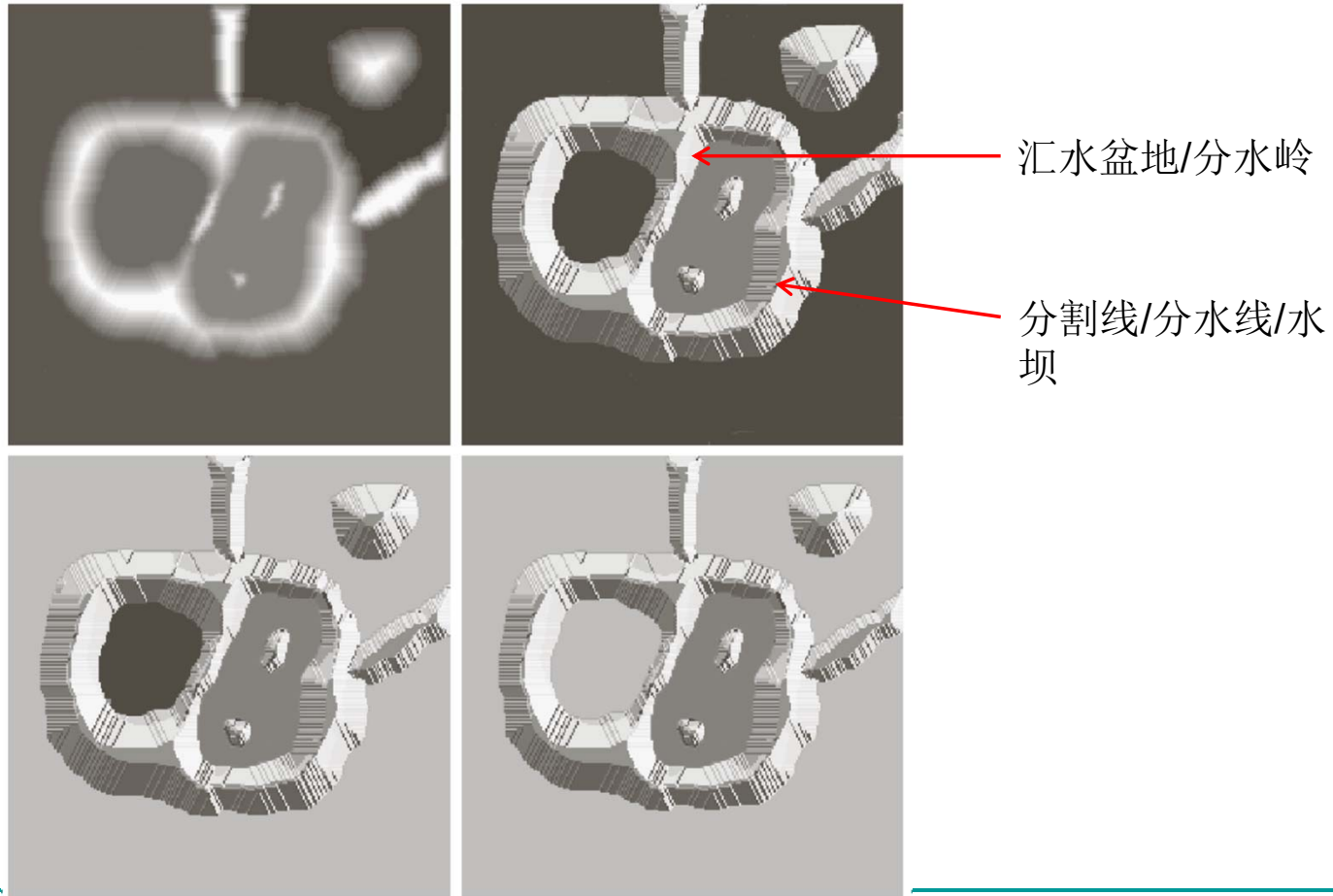
$$w(i, j) = \begin{cases} e^{-\frac{[I(n_i) - I(n_j)]^2}{\sigma_I^2}} e^{-\frac{\text{dist}(n_i, n_j)}{\sigma_d^2}} & \text{if } \text{dist}(n_i, n_j) < r \\ 0 & \text{otherwise} \end{cases}$$

用形态学分水岭的分割

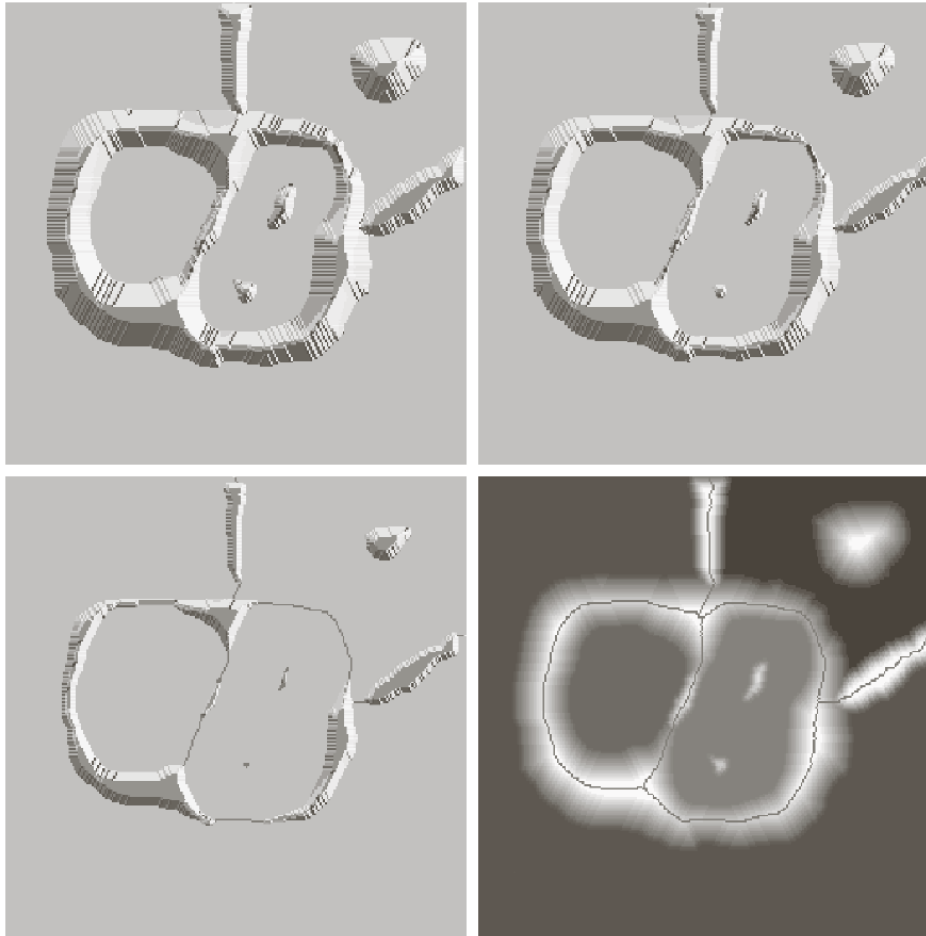
- 三种主要概念的分割：边缘检测（需要后处理连接）、阈值处理（速度快）、区域生长（连续性好），各有优缺点。
- 形态学分水岭分割将以上三种方法中的很多概念进行了具体化，能产生更稳定的分割结果。包括连接的分割边界。为在分割中结合**基于知识的约束**提供了一个简单的框架。
- 主要应用之一是从背景中提取近乎一致的前景，**经常应用于一幅图像的梯度图**，而不是图像本身。汇水盆地的**区域最小值**就可以对应于**感兴趣物体的较小梯度值**。

概念示意

地形俯视图



分水岭算法概念示意（ctd）



e f
g h

FIGURE 10.54
(Continued)
(e) Result of further flooding.
(f) Beginning of merging of water from two catchment basins (a short dam was built between them). (g) Longer dams. (h) Final watershed (segmentation) lines.
(Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

结果的一条重要性质是：
分水线组成一条连通的路径。

水坝构建

- 水坝的构造以二值图像为基础。构建分离二元点集的水坝的最简单方法是使用形态学膨胀。
- 所有水坝的高度通常设置为1加上图像中允许的最大灰度值。

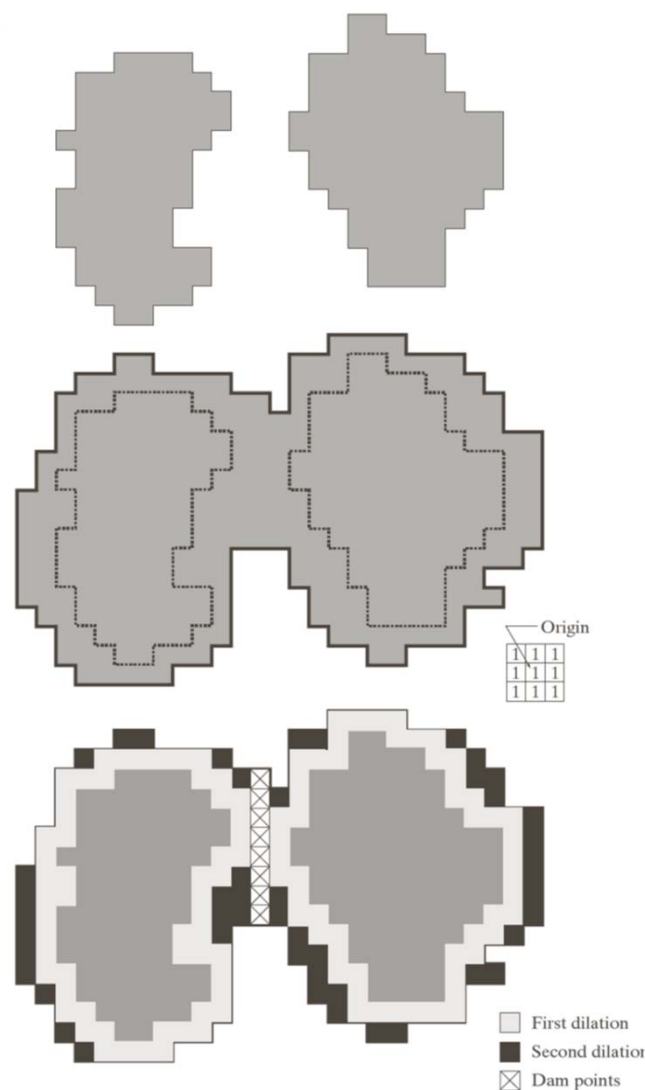


图 10.55 (a) 第 $n-1$ 步淹没处，两个部分淹没的汇水盆地；(b) 在淹没的第 n 步处，显示水已在两个盆地间溢出；(c) 用于膨胀的结构元；(d) 膨胀结果和水坝构建

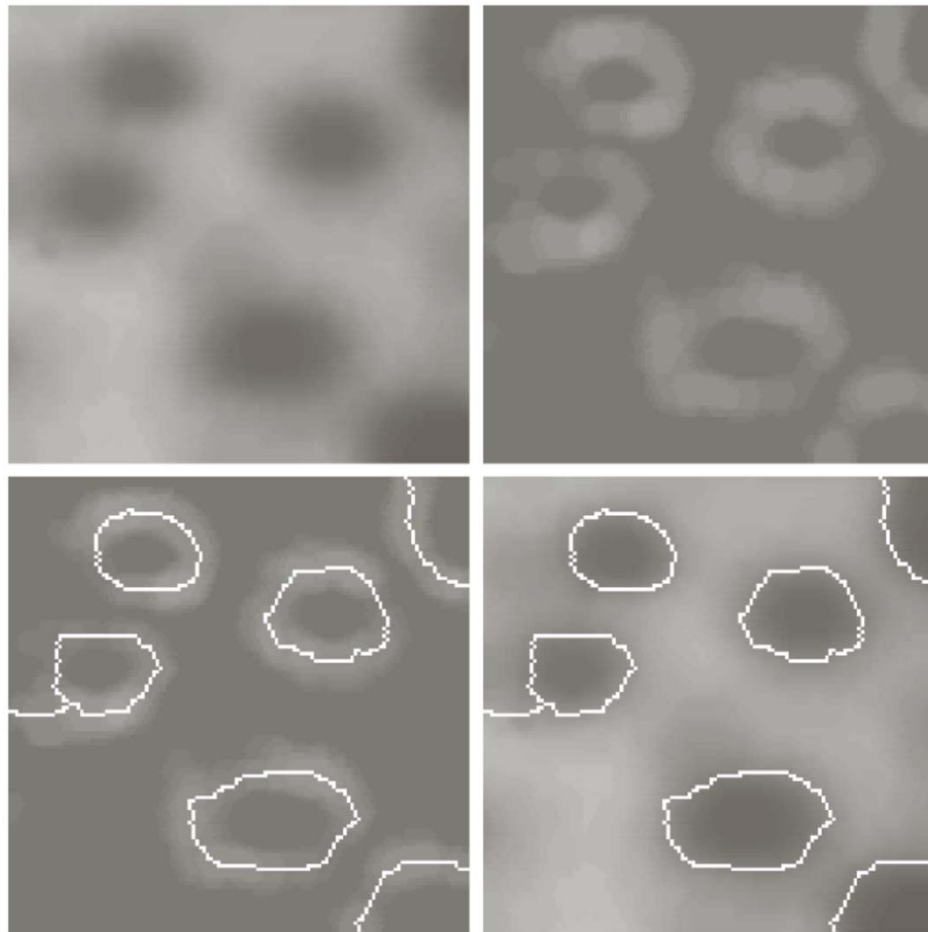
分水岭分割算法 (I)

- 令 M_1, M_2, \dots, M_R 是表示图像 $g(x, y)$ 的区域最小值点的集合, $C(M_i)$ 是与区域最小值 M_i 相联系的汇水盆地中的点坐标集。 $T(n)$ 表示满足 $g(s, t) < n$ 的坐标 (s, t) 的集合, 即: $T(n) = \{(s, t) | g(s, t) < n\}$
- 令 $C_n(M_i) = C(M_i) \cap T[n]$, 则 $C[n] = \bigcup_{i=1}^R C_n(M_i)$, 可以证明: $C[n-1]$ 中的每个连通分量都包含在 **T[n]** 的连通分量中。
- 算法使用 $C[\min+1] = T[\min+1]$ 来初始化, 然后进行递归处理, 由 $C[n-1]$ 计算 $C[n]$ 。

分水岭分割算法（II）

- 由 $C[n-1]$ 计算 $C[n]$ 的过程：
 - 令 Q 表示 $T[n]$ 中的连通分量的集合，对于每个连通分量 $q \in Q[n]$ ，有如下三种可能：
 - 1. $q \cap C[n-1]$ 为空集；（遇到一个新的最小值（盆地））
 - 2. $q \cap C[n-1]$ 包含 $C[n-1]$ 的一个连通分量；（ q 位于某些局部最小值的汇水盆地）
 - 3. $q \cap C[n-1]$ 包含 $C[n-1]$ 的一个以上连通分量。（遇到了分隔山脊线，必须构筑水坝），采用元素为1，大小为 3×3 的结构元来膨胀 $q \cap C[n-1]$ 且膨胀被限制到 q 时（测地膨胀），可以构建一座1个像素宽的水坝。
 - 根据 $g(x, y)$ 的直方图，可以确定算法中的 **min** 和 **max** 等参数，提高计算效率。

分水岭分割算法示例



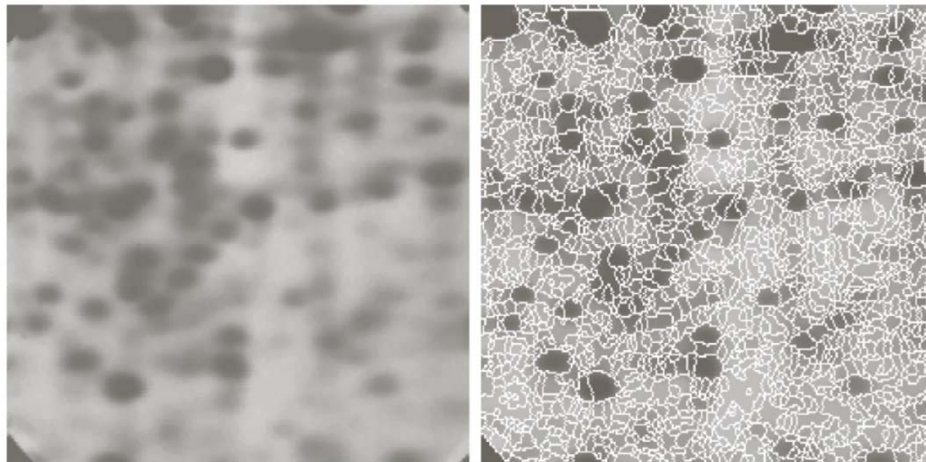
a	b
c	d

FIGURE 10.56

(a) Image of blobs.
(b) Image gradient.
(c) Watershed lines.
(d) Watershed lines superimposed on original image.
(Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

分水岭分割算法的局限性

- 通常会由于噪声和梯度的其他局部不规则性造成过度分割。



a b

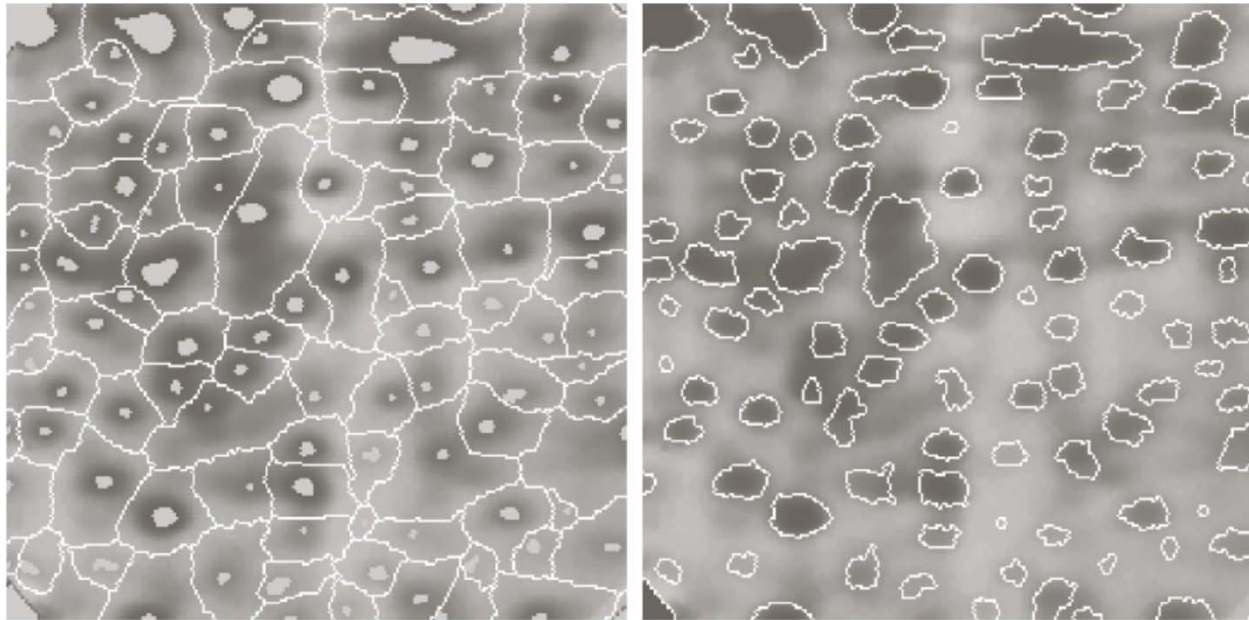
FIGURE 10.57
(a) Electrophoresis image. (b) Result of applying the watershed segmentation algorithm to the gradient image. Oversegmentation is evident. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

- 一个原因：存在着大量潜在的最小值。可以用平滑滤波器对图像进行过滤。
- 用标记约束感兴趣区域

标记的使用

- 与感兴趣物体相联系的标记称为**内部标记**，与背景相关联的标记称为**外部标记**。
- 选择标记的典型过程分为两步：（1）预处理（平滑滤波器）；（2）定义标记必须满足的准则集合。
- 一种定义内部标记的定义为（**同时满足**）：（1）被更高“海拔”点包围的区域；（2）区域中形成一个连通分量的那些点；（3）连通分量中所有点有相同灰度值。
 - （**不限于这些定义，可以用尺寸、形状、位置等与分割有关的先验知识来做定义，人类先验知识的指导**）
- 对内部标记，在允许的区域最小值限制下对**平滑后**的图像应用分水岭算法，得到的分水线定位为**外部标记**。
- 将外部标记分割的每个区域再划分为两个部分：单个物体及其背景。

标记以及基于标记的分割



a b

FIGURE 10.58 (a) Image showing internal markers (light gray regions) and external markers (watershed lines). (b) Result of segmentation. Note the improvement over Fig. 10.47(b). (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

分割中运动的应用

- 运动是人类和其他任何动物从无关细节背景中提取感兴趣物体或区域的强有力的提示。
- 在成像应用中，运动来自传感器和正被观看的场景间的相对位移。

- 空间域技术

- 差值图
$$d_{ij}(x, y) = \begin{cases} 1, & |f(x, y, t_i) - f(x, y, t_j)| > T \\ 0, & \text{Others.} \end{cases}$$

- 为1的像素通常是由物体运动造成的，也可能由噪声造成。

- 累积差值图

累积差值图

- 一幅累积差值图像（**ADI**）是通过将参考图像与序列中每幅后续图像进行比较形成的。
- 当参考图像和序列中一幅图像之间在某个像素位置上出现一次差异时，累积图像中每个像素位置的计数器计数一次。
- 有三类累积差值图，绝对**ADI**、正**ADI**和负**ADI**。**R**为参考图像。

$$A_k(x, y) = \begin{cases} A_{k-1}(x, y) + 1, & |R(x, y) - f(x, y, k)| > T \\ A_{k-1}(x, y), & \text{Others} \end{cases}$$
$$P_k(x, y) = \begin{cases} P_{k-1}(x, y) + 1, & [R(x, y) - f(x, y, k)] > T \\ P_{k-1}(x, y), & \text{Others} \end{cases}$$
$$N_k(x, y) = \begin{cases} N_{k-1}(x, y) + 1, & [R(x, y) - f(x, y, k)] < -T \\ N_{k-1}(x, y), & \text{Others} \end{cases}$$

其中： $R(x, y) = f(x, y, 1)$

ADI示例

当运动物体灰度大于背景时：

- 正**ADI**的非零区域等于运动物体的大小；
- 正**ADI**的位置对应于参考帧中运动物体的位置；
- 当运动物体完全移出参考帧物体时，正**ADI**的面积不再变化；
- 物体的运动方向和速度可以由绝对**ADI**和负**ADI**中的值决定。



a b c

FIGURE 10.59 ADIs of a rectangular object moving in a southeasterly direction. (a) Absolute ADI. (b) Positive ADI. (c) Negative ADI.

用ADI建立参考图像示例

- 很多时候需要根据一组包含一个或者多个运动物体的图像来构建一幅参考图像；
- 通过监视**ADI**的变化，当一个物体完全移出其参考帧的位置时，可以由当前帧中的背景复制到最初被参考帧中物体占据的位置，创建一幅仅包含静止分量的参考图像。



a b c

FIGURE 10.60 Building a static reference image. (a) and (b) Two frames in a sequence. (c) Eastbound automobile subtracted from (a) and the background restored from the corresponding area in (b). (Jain and Jain.)

作业

- 书面作业：
- **10.3, 10.6, 10.25, 10.31, 10.34, 10.49**
- 上机作业（二选1）：
 - 1.利用霍夫变换实现一幅图像中的直线检测；
 - 2.使用**otsu**方法实现一幅图像的最佳全局阈值分割。