数字图象处理

中国科学技术大学
电子工程与信息科学系

课件下载：https://ustc-dip.github.io/

第10章　图像分割

- 图像分割：将图像细分为构成它的子区域或物体
  - 细分的程度取决于要解决的问题
  - 在应用中，当感兴趣的物体或区域已经被检测出来时，就停止分割
  - 异常图像的分割是图像处理中最困难的任务之一
- 本章中的数分割算法均基于灰度值的两个基本性质之一：不连续性和相似性
  - 对于不连续性的灰度，方法是以**灰度突变**为基础分割一幅图像
    - ✓ 比如图像的边缘
  - 对于相似的灰度，主要方法是根据一组预定义的准则把一幅图像分割为相似的区域
    - ✓ 阈值处理、区域生长、区域分裂和区域聚合都是这类方法的例子

第10章 图像分割

10.1 基础知识

10.2 点、线和边缘检测

10.3 阈值处理

10.4 基于区域的分割

10.5 用形态学分水岭的分割

10.6 分割中运动的应用

第10章 图像分割

10.1 基础知识

10.2 点、线和边缘检测

10.3 阈值处理

10.4 基于区域的分割

10.5 用形态学分水岭的分割

10.6 分割中运动的应用

10.1 基础知识

- 令集合R代表整个图像区域，对R的分割可看做将R分成若干个满足下述条件的非空的子集（子区域）$R_1, R_2, \cdots, R_n$：
  - 1、 $\bigcup_{i=1}^{n} R_i = R$；
  - 2、对$i \neq j$，有$R_i \bigcap R_j = \varnothing$；
  - 3、每个子区域$R_1, R_2, \cdots, R_n$是连通的；
  - 4、对于各个子区域，有均匀性测度度量P为真

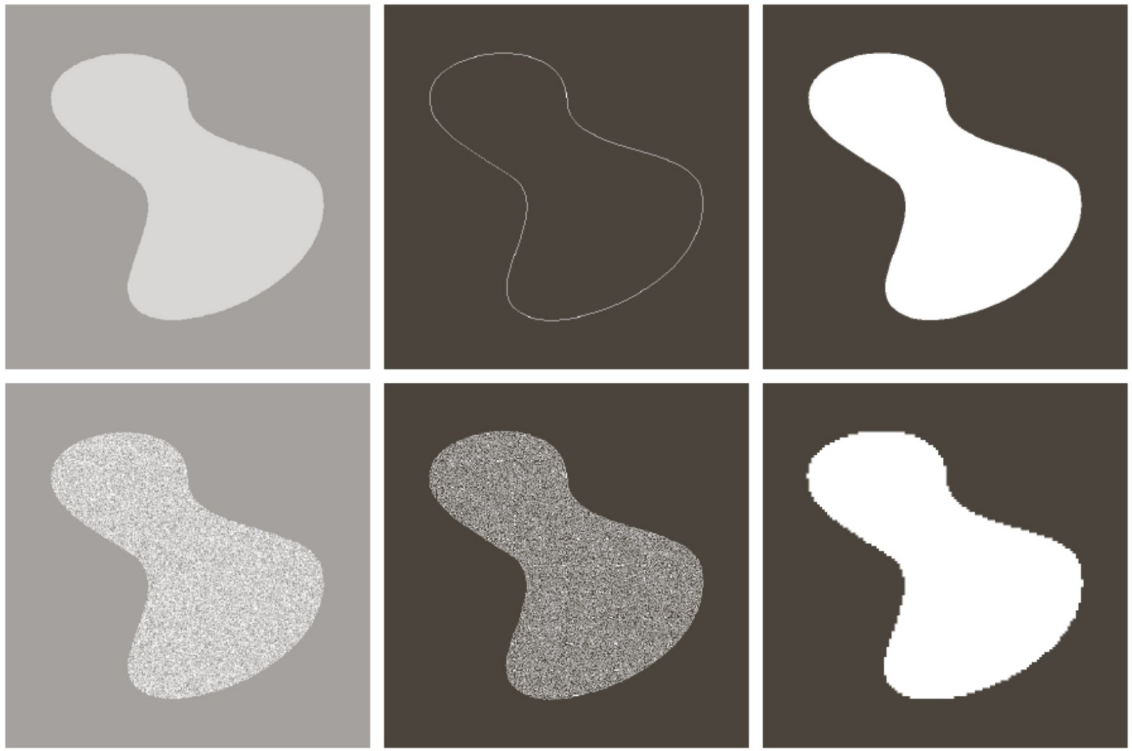
$$P(R_i) = \text{TRUE} \quad ;$$

  - 5、对其中任意两个和两个以上相邻子区域之并，其均匀性测度度量P为假，即：

$$P(R_i \cup R_j) = \text{FALSE}, (i \neq j)$$

10.1 基础知识

- 下面三幅图像说明了基于区域的分割



a b c
d e f

FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

第10章 图像分割

10.1 基础知识

10.2 点、线和边缘检测

10.3 阈值处理

10.4 基于区域的分割

10.5 用形态学分水岭的分割

10.6 分割中运动的应用

10.2 点、线和边缘检测

- 本节介绍以**灰度局部剧烈变化的检测**为基础的分割方法。
- 我们感兴趣的三种图像特征是孤立点、线和边缘
- 小节
  - 10.2.1 背景知识
  - 10.2.2 孤立点的检测
  - 10.2.3 线检测
  - 10.2.4 边缘模型
  - 10.2.5 基本边缘检测
  - 10.2.6 更先进的边缘检测技术
  - 10.2.7 边缘连接和边界检测

10.2.1 背景知识

- 就像局部平均平滑一幅图像那样，假设**平均处理**类似于**积分**，对于**灰度的突变**，**局部变化**可以用**微分**来检测。
- 由于变化非常短促，因此一阶微分和二阶微分特别适合

- 数字函数的导数可用差分定义
  - 一维函数$f(x)$在点x处的导数的近似：数字差分

$$\frac{\partial f}{\partial x} = f'(x) \approx f(x+1) - f(x)$$

  - 关于点x处的二阶导数的近似：

$$\frac{\partial^2 f}{\partial x^2} = f''(x) \approx f(x+1) + f(x-1) - 2f(x)$$

- 计算图像中每个像素位置处的一阶导数和二阶导数的另一种方法是使用空间滤波器：

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{k=1}^{9} w_k z_k$$

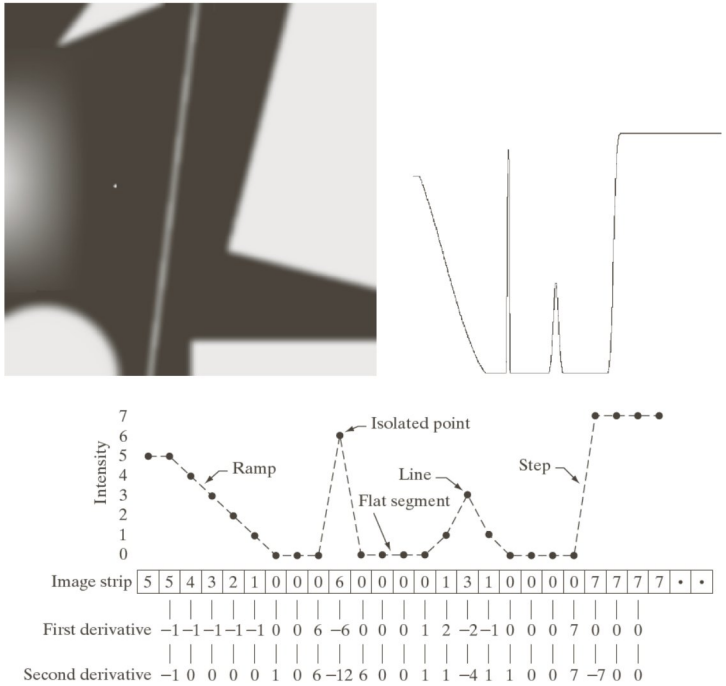
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

FIGURE 10.3
A general 3 × 3 spatial filter mask.

10.2.1 背景知识

- 基于一阶导数和二阶导数检测图像包含有不同的实心物体、一条线和单个噪声点（右图）

- 结论：
  - （1）一阶导数通常在图像中产生较粗的边缘；
  - （2）二阶导数对精细细节，如细线、孤立点和噪声有较强的响应；
  - （3）二阶导数在灰度斜坡和灰度台阶过渡处会产生双边缘响应；
  - （4）二阶导数的符号可用于确定边缘的过渡是从亮到暗还是从暗到亮



a b
c

FIGURE 10.2 (a) Image. (b) Horizontal intensity profile through the center of the image, including the isolated noise point. (c) Simplified profile (the points are joined by dashes for clarity). The image strip corresponds to the intensity profile, and the numbers in the boxes are the intensity values of the dots shown in the profile. The derivatives were obtained using Eqs. (10.2-1) and (10.2-2).

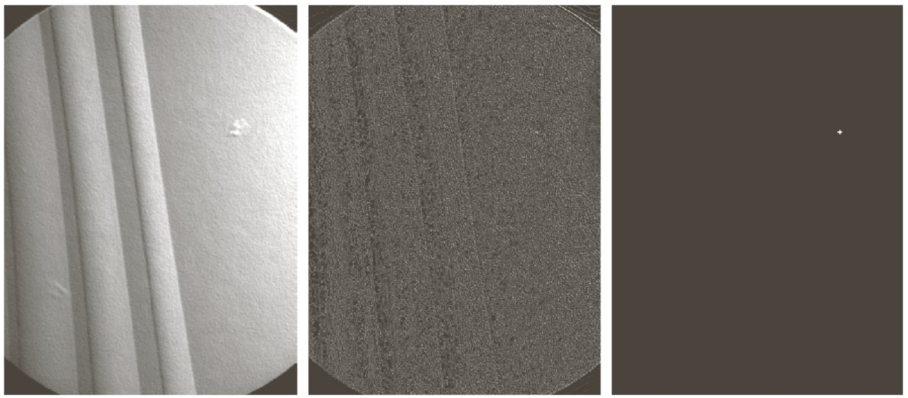
10.2.2 孤立点的检测

- 点的检测应以二阶导数为基础，即使用拉普拉斯算子：

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

- 同样，该算子可用模板来实现
- 如果在某个点处，该模板的**响应的绝对值超过了指定的阈值**，则在(x,y)处的点已被检测到。在输出图像中，该点标为1，否则为0

1	1	1
1	−8	1
1	1	1

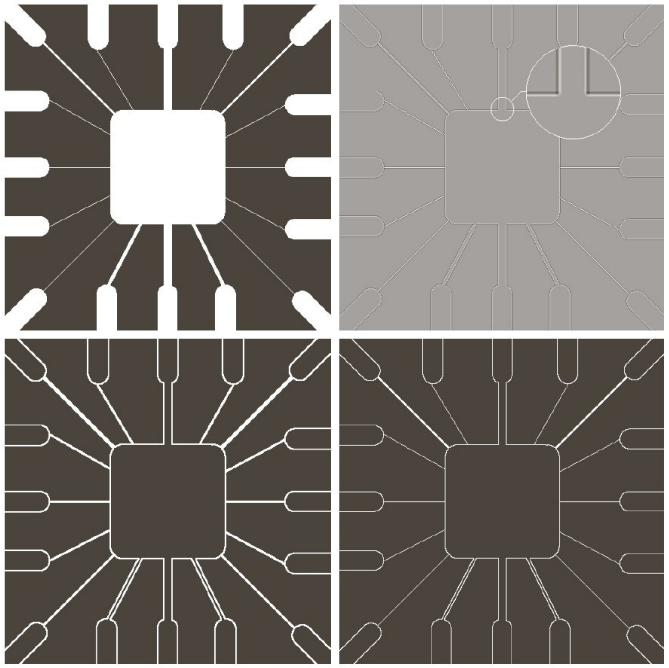


a
b c d

FIGURE 10.4 (a) Point detection (Laplacian) mask. (b) X-ray image of turbine blade with a porosity. The porosity contains a single black pixel. (c) Result of convolving the mask with the image. (d) Result of using Eq. (10.2-8) showing a single point (the point was enlarged to make it easier to see). (Original image courtesy of X-TEK Systems, Ltd.)

10.2.3 线检测

- 对于线检测，二阶导数将导致更强的响应，产生比一阶导数更细的线。我们可以使用拉普拉斯模板，但要处理双线效应
- 例：



a b
c d

FIGURE 10.5
(a) Original image. (b) Laplacian image; the magnified section shows the positive/negative double-line effect characteristic of the Laplacian. (c) Absolute value of the Laplacian. (d) Positive values of the Laplacian.

- 当把3×3滤波器居中放在一条宽为5个像素的恒定灰度线上时，其响应将为零，这样就产生了**双线效应**

10.2.3 线检测

- 点检测的拉普拉斯算子是**各向同性**的，因此**其响应与方向无关**（相对于该3×3拉普拉斯目标的4个方向：垂直方向、水平方向和两个对角方向）。
- 通常，我们的兴趣在于检测**特定方向的线**。
  - 第一个模板对于水平线有最佳响应；
  - 第二个模板对于45度方向的线有最佳响应；
  - 第三个模板对于垂直线有最佳响应；
  - 第四个模板对于-45度方向的线有最佳响应

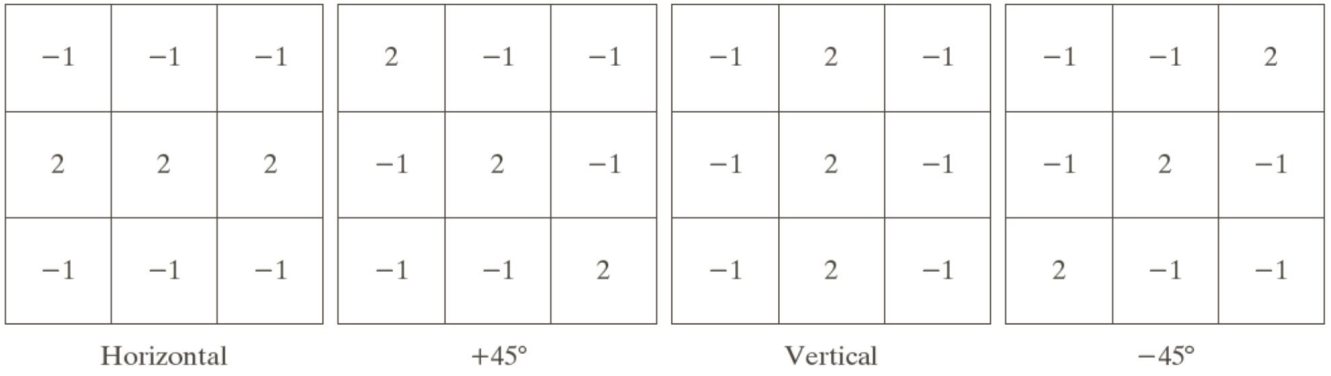
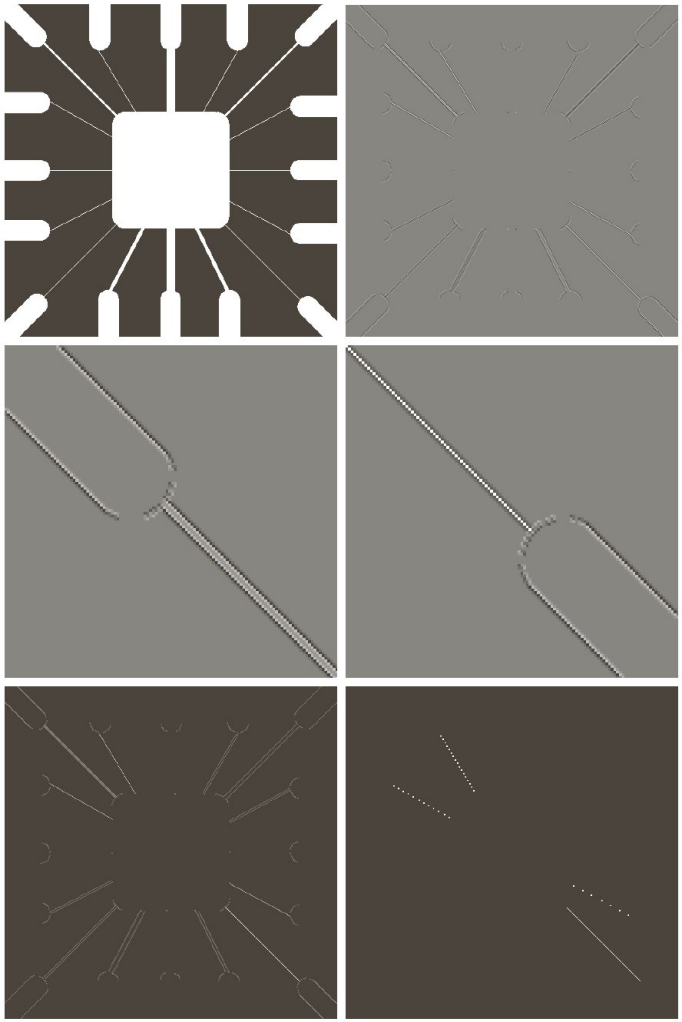


FIGURE 10.6 Line detection masks. Angles are with respect to the axis system in Fig. 2.18(b).

- 每个模板的系数之和为0，这表明**恒定灰度区域中的响应为0**

10.2.3 线检测

- 特定方向线的检测

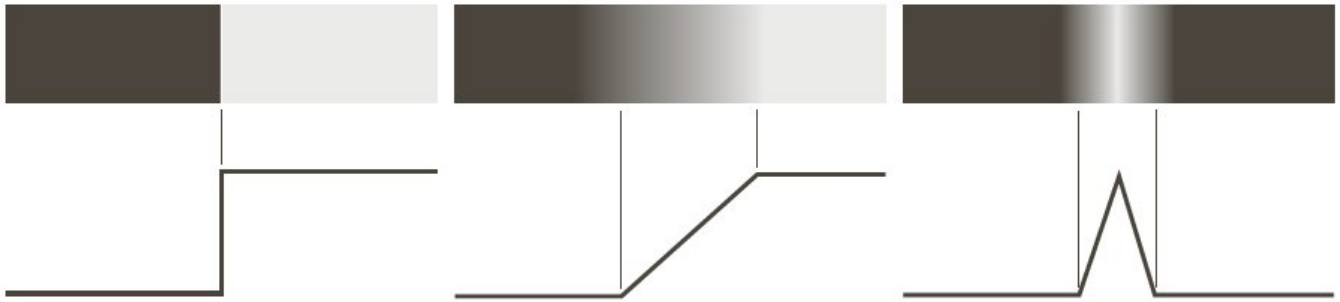


a b
c d
e f

FIGURE 10.7
(a) Image of a wire-bond template. (b) Result of processing with the +45° line detector mask in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

10.2.4 边缘模型

- 边缘检测是基于**灰度突变**来分割图像的常用方法
- 边缘模型根据它们的灰度剖面来分类。有台阶边缘、斜坡边缘和“屋顶”边缘等
- 下图显示了这些边缘模型和相应的灰度剖面

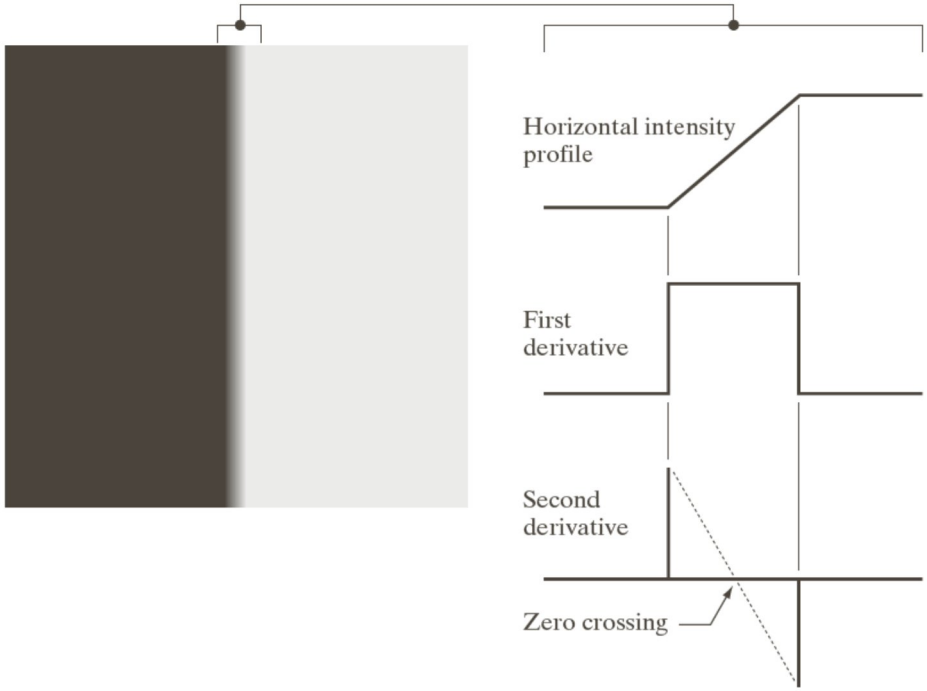


a b c

FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

10.2.4 边缘模型

- 下图结论
  - 一阶导数的**幅度**可用于检测图像中的某个点处是否存在一个边缘
  - 二阶导数的**符号**用于确定一个边缘像素是位于该边缘的暗侧还是位于该边缘的亮侧
- 围绕一条边缘的二阶导数的两个附加性质：
  - 对图像中的每条边缘，二阶导数生成两个值（一个不希望的特点）
  - 二阶导数的**零交叉点**可用于定位粗边缘的中心



a b

FIGURE 10.10 (a) Two regions of constant intensity separated by an ideal vertical ramp edge. (b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

10.2.4 边缘模型

- 微弱的可见噪声严重影响检测边缘所用的一阶导数和二阶导数

- 因此执行边缘检测的三个基本步骤是：
  - （1）为降噪图像进行平滑处理
  - （2）边缘点的检测。这是一个局部操作，从一幅图像中提取所有边缘点的潜在候选者
  - （3）边缘定位。这一步的目的是从候选边缘点中选择组成边缘点集合的真实成员

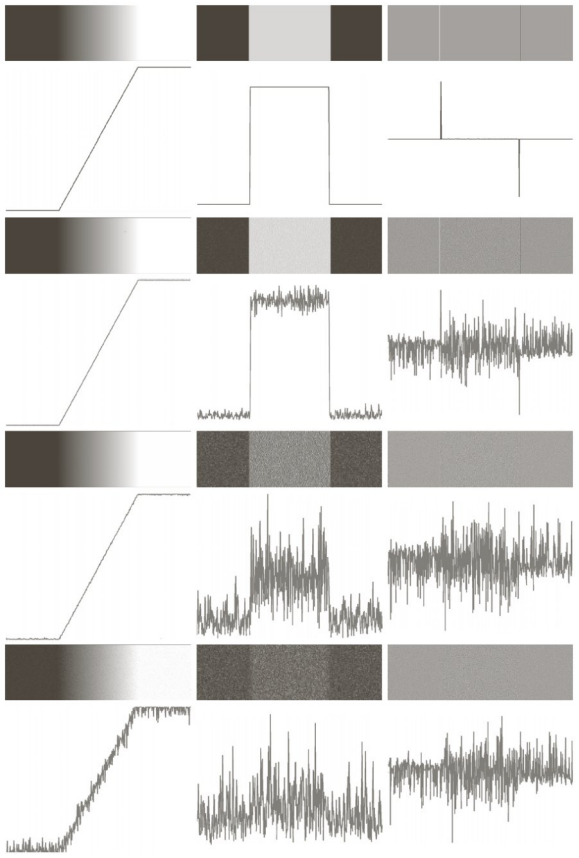


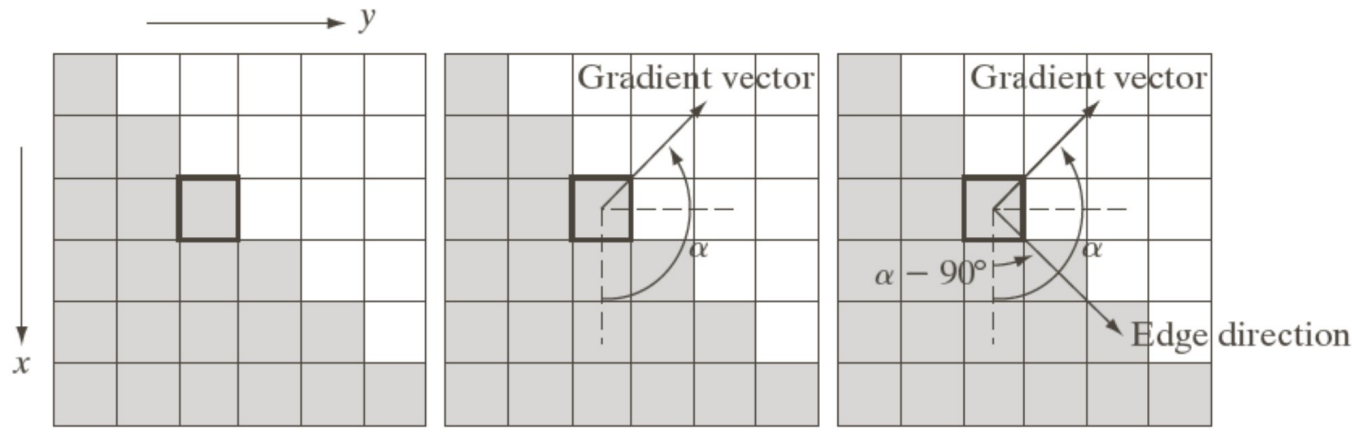
FIGURE 10.11 First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images and intensity profiles.

10.2.5 基本边缘检测

- 图像梯度及其性质
  - 梯度向量、幅度和方向角

$$\nabla f(x,y) = \begin{bmatrix} G_x & G_y \end{bmatrix}^T = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^T \qquad \text{mag}(\nabla f) = \left[G_x^2 + G_y^2 \right]^{1/2} \qquad \alpha(x,y) = \arctan\left(G_y / G_x \right)$$

  - 图像边缘方向与边缘上点的梯度向量正交

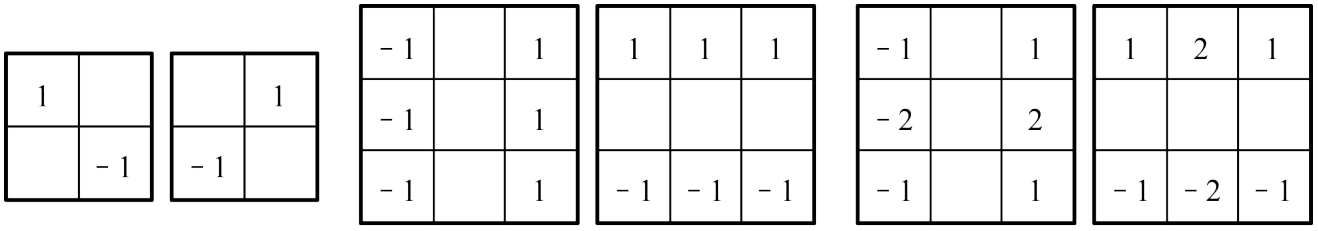


a b c

FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

10.2.5 基本边缘检测

- 梯度算子
  - 导数用模板对图像进行滤波来实现



1	
	−1

	1
−1	

(a) Roberts

−1		1
−1		1
−1		1

1	1	1
−1	−1	−1

(b) Prewitt

−1		1
−2		2
−1		1

1	2	1
−1	−2	−1

(c) Sobel

  - 检测对角边缘的模板

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

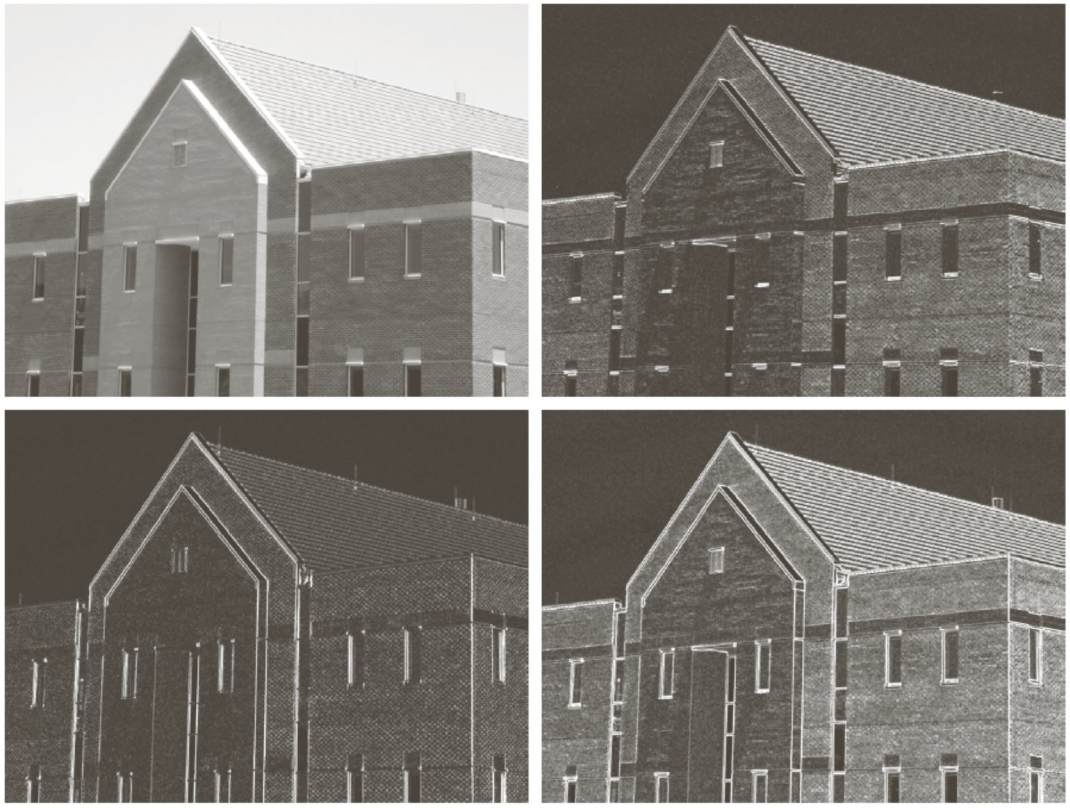
0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

10.2.5 基本边缘检测

- 梯度算子



a b
c d

FIGURE 10.16
(a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

10.2.5 基本边缘检测

- 梯度算子
  - **精细的细节在边缘检测中通常是不符合需要的**，因为它往往表现为噪声，使主要边缘的检测变得复杂。
  - 减少精细细节的一个方法是对图像进行平滑处理，如下图



a b
c d

FIGURE 10.18
Same sequence as in Fig. 10.16, but with the original image smoothed using a 5 × 5 averaging filter prior to edge detection.

10.2.5 基本边缘检测

- 与阈值处理相结合的梯度
  - 在计算梯度前对图像进行平滑处理，边缘检测可做更多的选择。
  - 实现相同基本目标的另一种方法是，对**梯度图像进行阈值处理**。
    - ✓ 阈值处理后的图像边缘更少，并且图像中的边缘要清晰得多。
    - ✓ 另一方面，许多边缘被断开了
  - 实践中通常既做平滑处理也做阈值处理
    - ✓ 突出主要边缘，并尽可能保持连接



a b

FIGURE 10.20 (a) Thresholded version of the image in Fig. 10.16(d), with the threshold selected as 33% of the highest value in the image; this threshold was just high enough to eliminate most of the brick edges in the gradient image. (b) Thresholded version of the image in Fig. 10.18(d), obtained using a threshold equal to 33% of the highest value in that image.

10.2.6 更先进的边缘检测

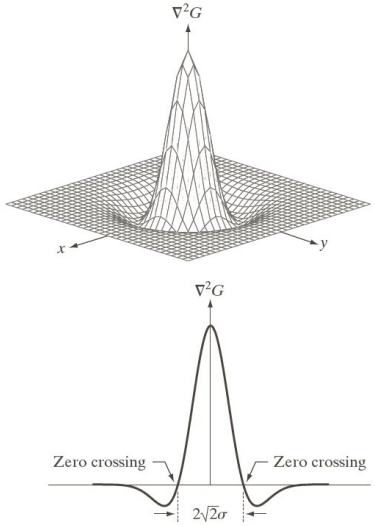
- 用于边缘检测的算子应有两个特点
  - 能计算图像中每一个点处的一阶导数或二阶导数的数字近似的微分算子
  - 能被“调整”以便在**任何期望的尺寸**上起作用，因此大算子也可用于检测模糊边缘，小算子可用于检测锐度集中的精细细节
- Marr-Hildreth边缘检测器（**马尔算子**）
  - 满足上述条件的算子是$\nabla^2 h$：高斯拉普拉斯（LoG）
  - 下图显示了一个LoG的负函数的三维图、图像和剖面

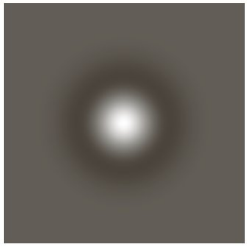
$$h(x,y) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \qquad g(x,y) = h(x,y) \star f(x,y)$$

$$\nabla^2 g = \nabla^2[h(x,y) \star f(x,y)] = \nabla^2 h(x,y) \star f(x,y)$$

$$= \left(\frac{r^2-\sigma^2}{\sigma^4}\right)\exp\left(-\frac{r^2}{2\sigma^2}\right) \star f(x,y)$$

$$\nabla^2 h = h_{LOG}(r) = \left(\frac{r^2-\sigma^2}{\sigma^4}\right)\exp\left(-\frac{r^2}{2\sigma^2}\right) \qquad (LOG)$$





0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

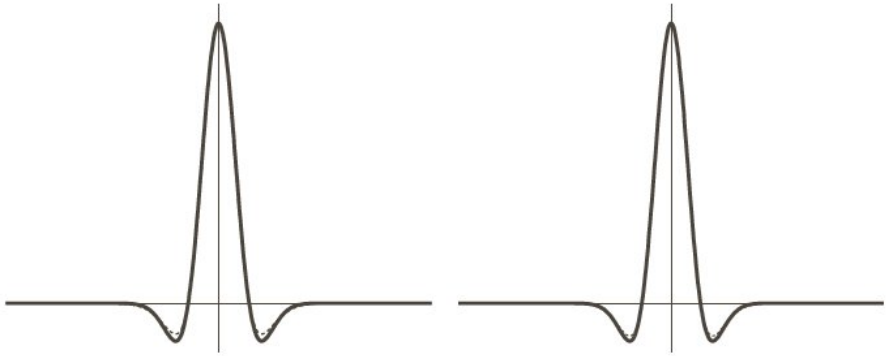
a b
c d

FIGURE 10.21 (a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5 × 5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

10.2.6 更先进的边缘检测

- Marr-Hildreth边缘检测器（马尔算子）
  - （1）算子的高斯部分会模糊图像
  - （2）关于∇^2：尽管一阶导数可用于检测灰度突变，但它们是有方向的算子。另一方面，拉普拉斯有**各向同性**的优点，符合人的视觉系统特性
- Marr-Hildreth边缘检测算法可小结如下：
  - 用一个2-D的高斯平滑模板与源图象卷积
  - 计算卷积后图象的拉普拉斯值
  - 检测拉普拉斯图象中的**过零点**作为边缘点
- 使用高斯差分（DoG）来近似LoG滤波器

$$\mathrm{DoG}(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}$$

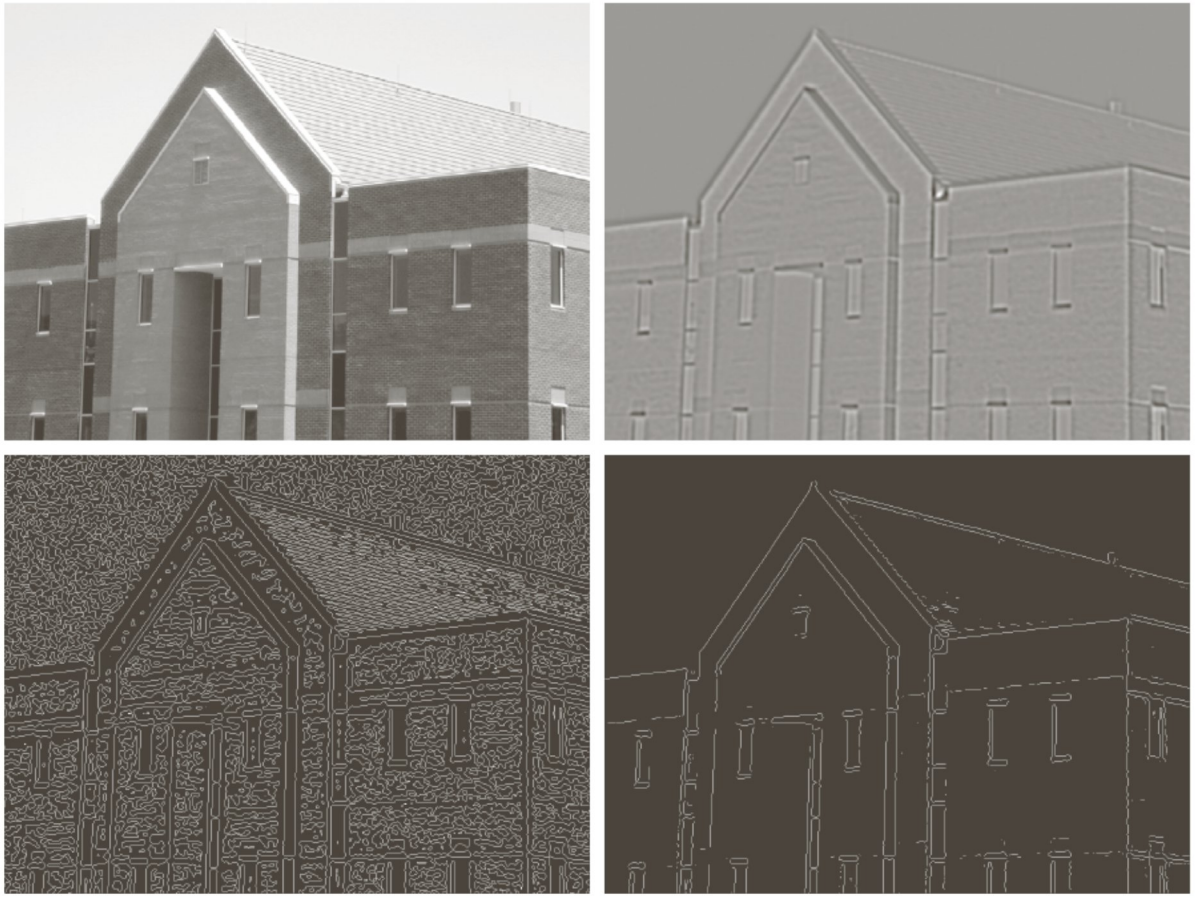


a b

FIGURE 10.23 (a) Negatives of the LoG (solid) and DoG (dotted) profiles using a standard deviation ratio of 1.75:1. (b) Profiles obtained using a ratio of 1.6:1.

10.2.6 更先进的边缘检测

- Marr-Hildreth边缘检测算法说明



a b
c d

FIGURE 10.22
(a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.

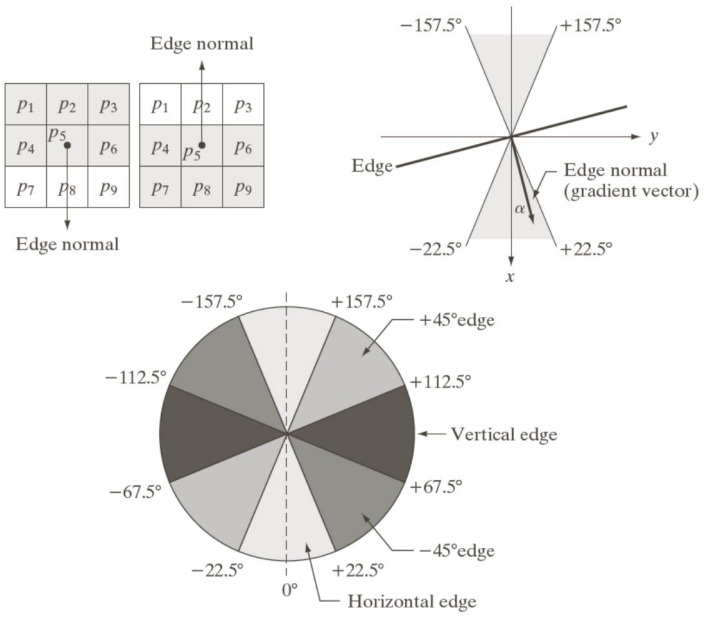
10.2.6 更先进的边缘检测

- 坎尼边缘检测器
  - 最优边缘检测算子，坎尼方法基于三个基本目标：
  - （1）低错误率。既要少将真正的边缘丢失也要少将非边缘判为边缘
  - （2）边缘点应被很好地定位。检测出的边缘应在真正的边界上
  - （3）单一的边缘响应。得到的边界为单象素宽

- 坎尼边缘检测算法的基本步骤总结如下：
  - （1）用一个高斯滤波器平滑输入图像
  - （2）计算梯度幅值图像和角度图像
  - （3）对梯度幅值图像应用非最大抑制
  - （4）用双阈值处理和连接分析来检测并连接边缘

10.2.6 更先进的边缘检测

- 坎尼边缘检测器（高斯滤波平滑）
  - 令$f(x,y)$表示输入图像，$G(x,y)$表示高斯函数：$G(x,y)=e^{-\frac{x^2+y^2}{2\sigma^2}}$
  - 用G和f的卷积形成一幅平滑后的图像$f_s(x,y)=G(x,y)\star f(x,y)$
  - 计算梯度幅度$M(x,y)=\sqrt{g_x^2+g_y^2}$和方向$\alpha(x,y)=\arctan[\frac{g_y}{g_x}]$
- 坎尼边缘检测器（非最大抑制）
  - 把边缘方向量化为四个方向：水平、垂直、+45°和-45°



a b
c

FIGURE 10.24
(a) Two possible orientations of a horizontal edge (in gray) in a 3 × 3 neighborhood. (b) Range of values (in gray) of α, the direction angle of the *edge normal*, for a horizontal edge. (c) The angle ranges of the edge normals for the four types of edge directions in a 3 × 3 neighborhood. Each edge direction has two ranges, shown in corresponding shades of gray.

10.2.6 更先进的边缘检测

- 坎尼边缘检测器（非最大抑制）
  - 令d_1，d_2，d_3和d_4表示刚才讨论的3×3区域的四个基本边缘方向。对于$\alpha(x,y)$中以每点(x,y)为中心的3×3区域，有如下非最大抑制方案：
    - ✓ 寻找最接近$\alpha(x,y)$的方向d_k
    - ✓ 若$M(x,y)$的值至少小于沿d_k的两个邻居之一，则令$g_N(x,y)=0$（抑制）；否则，令$g_N(x,y)=M(x,y)$，这里$g_N(x,y)$为非最大抑制后的图像。
- 坎尼边缘检测器（双阈值处理）
  - 对$g_N(x,y)$进行阈值处理，以便减少伪边缘点
  - 两个阈值：一个低阈值T_L和一个高阈值T_H。
    - ✓ 一般，高、低阈值比率为2:1或3:1
  - 将阈值操作想象为创建两幅附加的图像

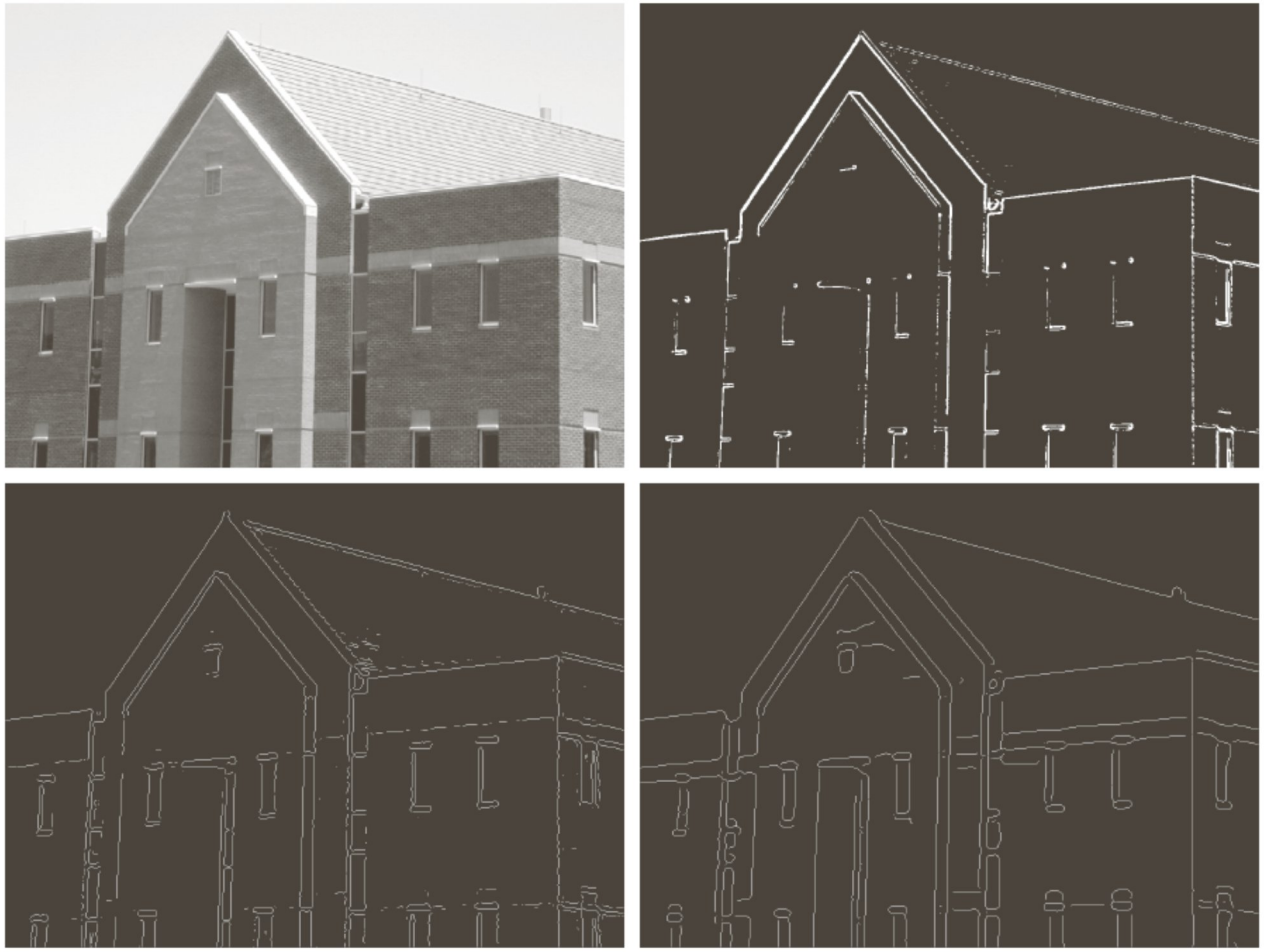
    $g_{NH}(x,y)=g_N(x,y)\geq T_H$　　$g_{NL}(x,y)=g_N(x,y)\geq T_L$
  - 通过令$g_{NL}(x,y)=g_{NL}(x,y)-g_{NH}(x,y)$，我们从$g_{NL}(x,y)$中删除所有来自$g_{NH}(x,y)$的非零像素。$g_{NH}(x,y)$和$g_{NL}(x,y)$中的非零像素可分别视为“强”和“弱”边缘像素
  - 阈值处理后，$g_{NH}(x,y)$中的所有像素均被假设为有效的边缘像素，并被立即标记。取决于T_H的值，$g_{NH}(x,y)$中的边缘通常会存在缝隙

10.2.6 更先进的边缘检测

- 坎尼边缘检测器（双阈值处理）
  - 较长的边缘用下列步骤形成：
  - （a）在$g_{NH}(x,y)$中定位下一个未被访问的边缘像素p
  - （b）在$g_{NL}(x,y)$中将所有弱像素标记为有效边缘像素，用8连通的连接方法连接到p
  - （c）若$g_{NH}(x,y)$中的所有非零像素已被访问，则调到步骤（d），否则返回步骤（a）
  - （d）将$g_{NL}(x,y)$中未标记为有效边缘像素的所有像素置零
- 最后，将来自$g_{NL}(x,y)$的所有非零像素附加到$g_{NH}(x,y)$，用坎尼算子形成最终的输出图像
- **尽管非最大抑制后的边缘比原来的梯度边缘细，但仍保持了比1像素粗的边缘。为得到1像素粗的边缘，通常要在最后一步执行一次边缘细化算法**
- 坎尼算法改进性能所付出的代价是，实现起来**更复杂，同时执行时间更长**。在某些应用中，成本和速度需求通常要求我们使用更简单的技术，主要是阈值梯度方法。
- 当关注的主要是边缘质量是，通常会使用Marr-Hildreth算法和坎尼算法，特别是坎尼算法

10.2.6 更先进的边缘检测

- 坎尼边缘检测方法说明



a b
c d

FIGURE 10.25
(a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1].
(b) Thresholded gradient of smoothed image.
(c) Image obtained using the Marr-Hildreth algorithm.
(d) Image obtained using the Canny algorithm. Note the significant improvement of the Canny image compared to the other two.

10.2.7 边缘连接和边界检测

- 理想情况下，边缘检测应该仅产生位于边缘上的像素集合。
- 实际上，由于噪声、不均匀照明引起的**边缘间断**，以及其他引入灰度值虚假的不连续的影响，这些像素并不能完全描述边缘特性。
- 因此，一般是在边缘检测后紧跟连接算法，将边缘像素组合成有意义的边缘或区域边界。
- 本节讨论三种基本的边缘连接方法
  - 第一种方法需要有关局部区域中的边缘点（如一个3×3邻域）的知识；
  - 第二种方法要求区域边界上的点已知；
  - 第三种方法是处理整个边缘图像的全局方法。

10.2.7 边缘连接和边界检测

局部处理

- 在每个点(x,y)处的一个小邻域内分析像素的特点，根据预定的准则，将所有的相似点连接起来，以形成满足**相同特性**像素的一条边缘
- 一种特别适合于实时应用的简化方法由如下步骤组成：

（1）计算输入图像$f(x,y)$的梯度幅度阵列$\mathrm{M}(\mathrm{x},\mathrm{y})$和角度阵列$\alpha(\mathrm{x},\mathrm{y})$

（2）形成一幅二值图像g，任何坐标对(x,y)处的值由下式给出：

$$g(x,y)=\begin{cases}1, & M(x,y)>T_M \text{且} \alpha(x,y)=A\pm T_A \\ 0, & \text{其他}\end{cases}$$

式中，T_M是一个阈值，A是一个指定的角度方向，$\pm T_A$定义了一个关于A的可接受方向“带宽”

（3）**扫描g的行**，并在不超过指定长度K的每一行中填充（置1）所有缝隙（0的集合）。注意，按照定义，缝隙一定要限制在一个1或多个1的两端。分别处理各行，它们之间没有记忆

（4）在任何其他方向θ上检测缝隙，以该角度选择g，并应用步骤3中的水平扫描过程。然后，将结果以-θ旋转回来

10.2.7 边缘连接和边界检测

- 使用局部处理的边缘连接



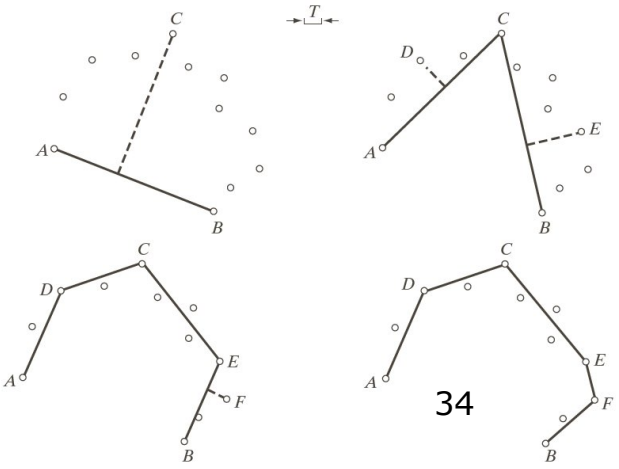
a b c
d e f

FIGURE 10.27 (a) A 534 × 566 image of the rear of a vehicle. (b) Gradient magnitude image. (c) Horizontally connected edge pixels. (d) Vertically connected edge pixels. (e) The logical OR of the two preceding images. (f) Final result obtained using morphological thinning. (Original image courtesy of Perceptics Corporation.)

10.2.7 边缘连接和边界检测

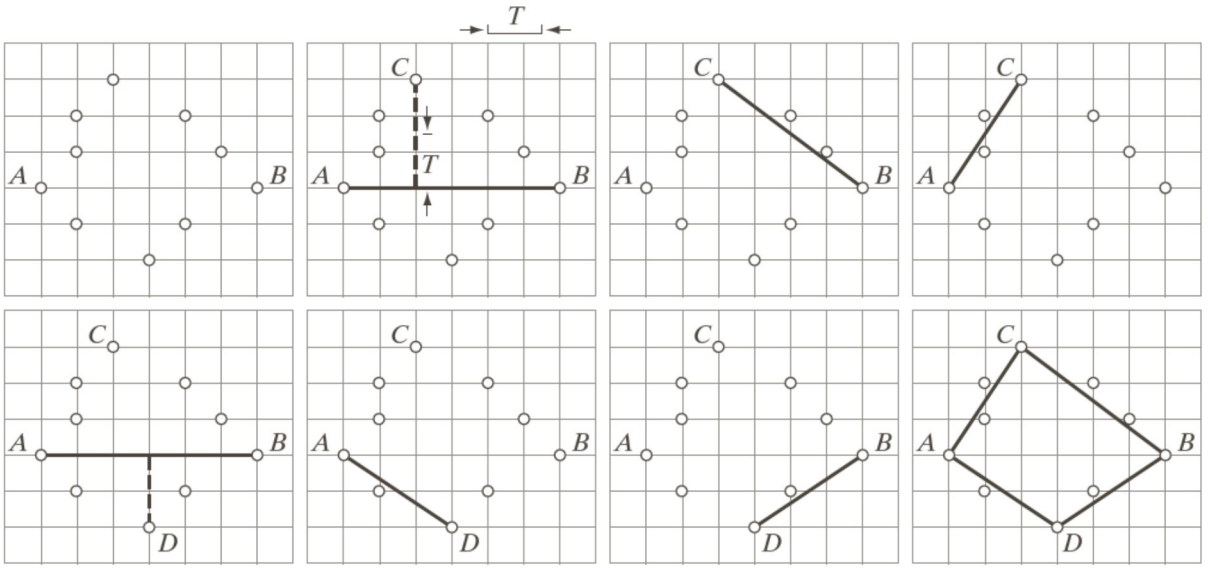
区域处理

- 对这类处理的一种方法是函数近似，我们对已知点拟合一条二维曲线。我们的兴趣通常在于快速执行的技术，多边形近似尤其有吸引力
- 寻找一个多边形拟合开放和闭合曲线的算法描述如下：
  - ✓ 1、令P是一个已排序序列，显然，这些点是一幅二值图像中的1值点。指定两个起始点A和B。他们是多边形的两个起始顶点
  - ✓ 2、指定一个阈值T，以及两个空堆栈“开”（OPEN）和“闭”（CLOSED）
  - ✓ 3、若P中的点对应于一条闭合曲线，则把A放到“开”中，并把B放到“开”和“闭”中。若点对应于一条开放曲线，则把A放到“开”中，而把B放到“闭”中
  - ✓ 4、计算从“闭”中最后一个顶点到“开”中最后一个顶点的线的参数
  - ✓ 5、计算步骤4所得的直线至P中所有点的距离，序列把它们放到步骤4所得的两个顶点之间。选择具有最大距离D_{max}的点V_{max}（解决任意性问题）
  - ✓ 若$D_{max} > T$，则把V_{max}作为一个新顶点放在“开”的末尾。转到步骤4
  - ✓ 否则，从“开”中移除最后一个顶点，并把它作为“闭”的最后一个顶点插入
  - ✓ 若“开”非空，转到步骤4
  - ✓ 否则，退出。“闭”中顶点就是拟合P中的点的多边形的顶点



10.2.7 边缘连接和边界检测

- 用多边形近似连接边缘



a b c d
e f g h

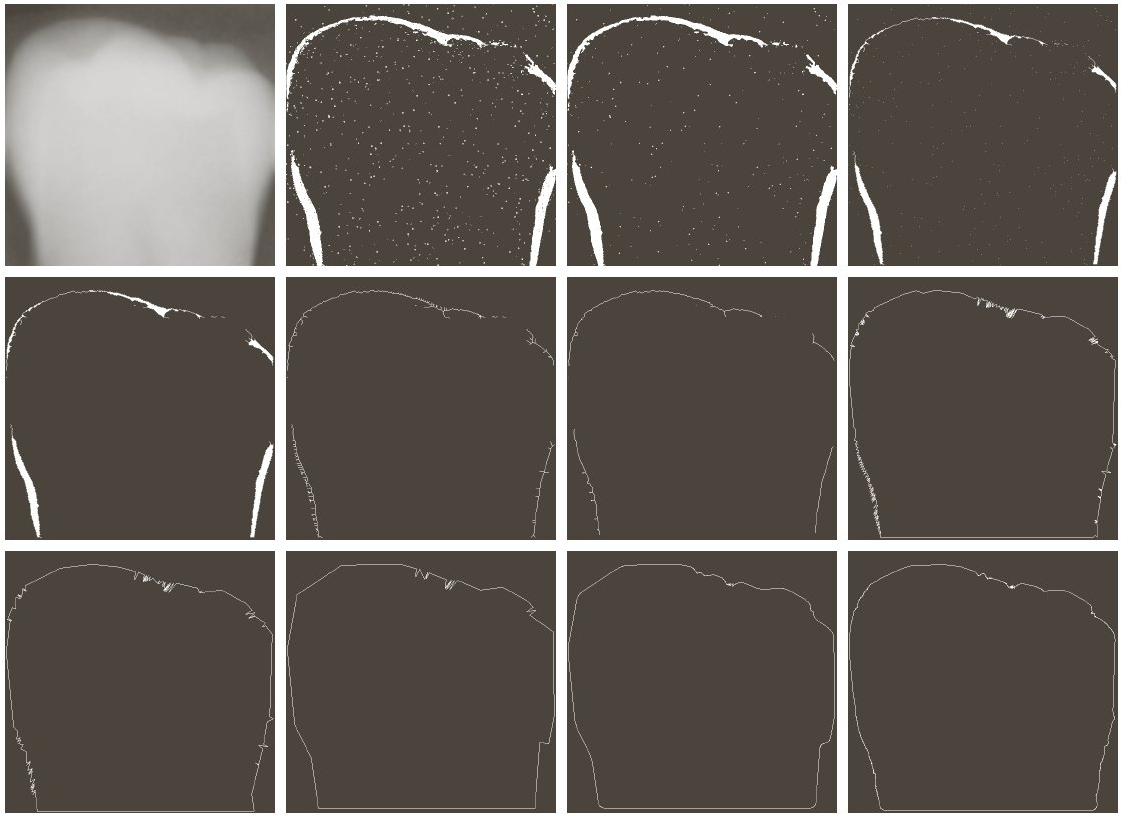
FIGURE 10.29 (a) A set of points in a clockwise path (the points labeled *A* and *B* were chosen as the starting vertices). (b) The distance from point *C* to the line passing through *A* and *B* is the largest of all the points between *A* and *B* and also passed the threshold test, so *C* is a new vertex. (d)–(g) Various stages of the algorithm. (h) The final vertices, shown connected with straight lines to form a polygon. Table 10.1 shows step-by-step details.

CLOSED	OPEN	Curve segment processed	Vertex generated
B	*B*, *A*	—	*A*, *B*
B	*B*, *A*	(*BA*)	*C*
B	*B*, *A*, *C*	(*BC*)	—
B, *C*	*B*, *A*	(*CA*)	—
B, *C*, *A*	*B*	(*AB*)	*D*
B, *C*, *A*	*B*, *D*	(*AD*)	—
B, *C*, *A*, *D*	*B*	(*DB*)	—
B, *C*, *A*, *D*, *B*	Empty	—	—

TABLE 10.1
Step-by-step details of the mechanics in Example 10.11.

10.2.7 边缘连接和边界检测

- 一幅图像的边界的多边形拟合



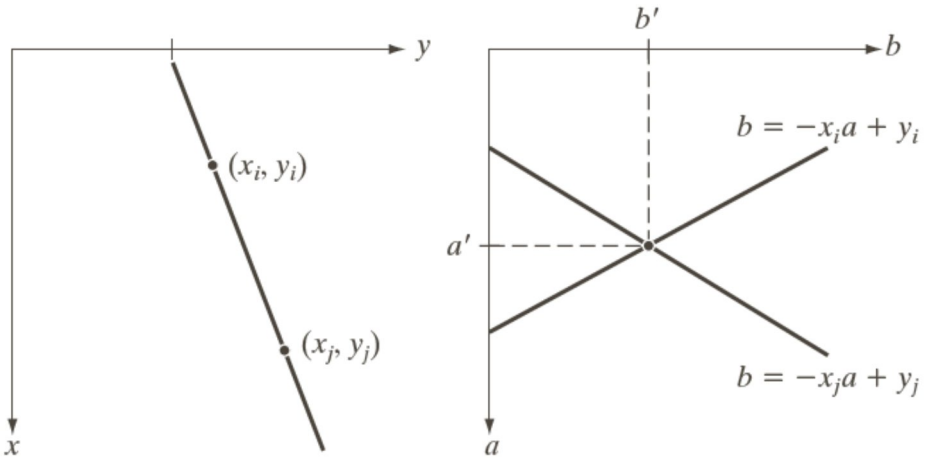
a b c d
e f g h
i j k l

FIGURE 10.30 (a) A 550 × 566 X-ray image of a human tooth. (b) Gradient image. (c) Result of majority filtering. (d) Result of morphological shrinking. (e) Result of morphological cleaning. (f) Skeleton. (g) Spur reduction. (h)–(j) Polygonal fit using thresholds of approximately 0.5%, 1%, and 2% of image width ($T = 3$, 6, and 12). (k) Boundary in (j) smoothed with a 1-D averaging filter of size 1 × 31 (approximately 5% of image width). (l) Boundary in (h) smoothed with the same filter.

10.2.7 边缘连接和边界检测

使用霍夫变换的全局处理

- 检测像素是否位于指定形状的曲线上。一旦检测到，这些曲线就会形成边缘或感兴趣的区域边界
- 例如：给定一幅图像中的n个点，假设我们希望找到这些点中一个位于直线上的子集。
  - ✓ 一种可行的解决方法是：先找到所有由每对点确定的直线，然后寻找靠近特定直线的点的所有子集。这种方法涉及寻找$\frac{n(n-1)}{2}\sim n^2$条直线，然后对每个点，与所有直线执行$\frac{(n)(n(n-1))}{2}\sim n^3$次比较。这种方法计算量太大而没有应用价值
- 霍夫变换是一种替代方法：从图像的xy平面到参数空间

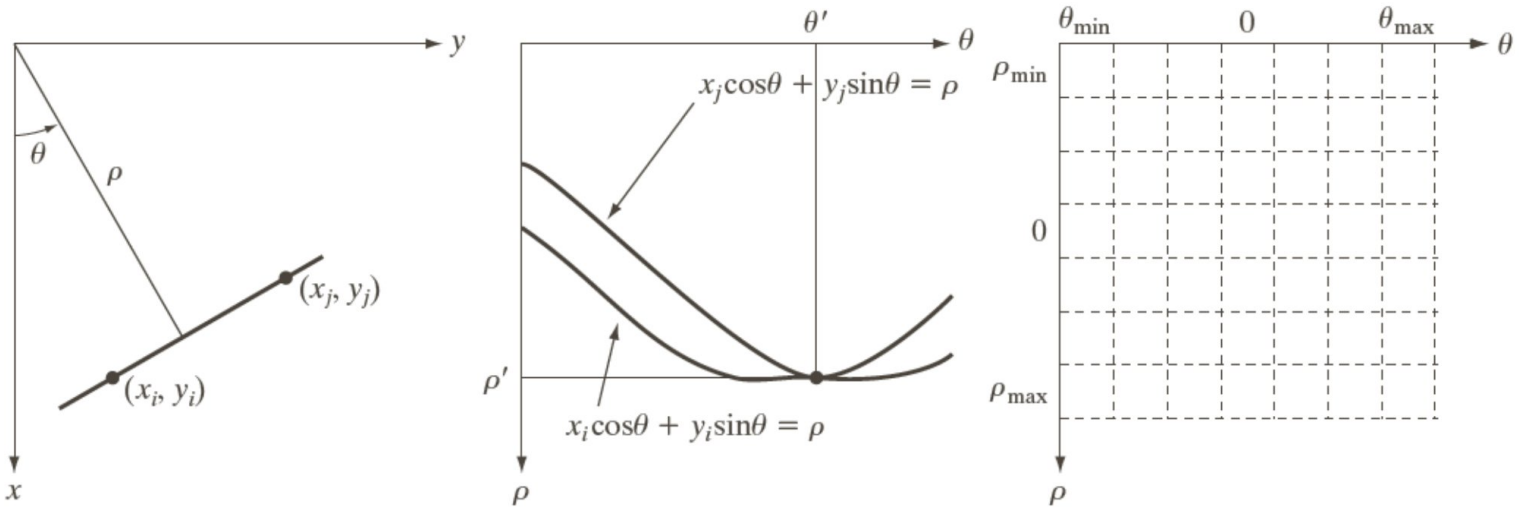


a b

FIGURE 10.31 (a) *xy*-plane. (b) Parameter space.

10.2.7 边缘连接和边界检测

- 使用霍夫变换的全局处理
  - 当直线逼近垂直方向时，a（直线的斜率）会趋于无限大。
  - 解决该问题的方法之一是，使用一条直线的法线表示：$x\cos\theta + y\sin\theta = \rho$



a b c

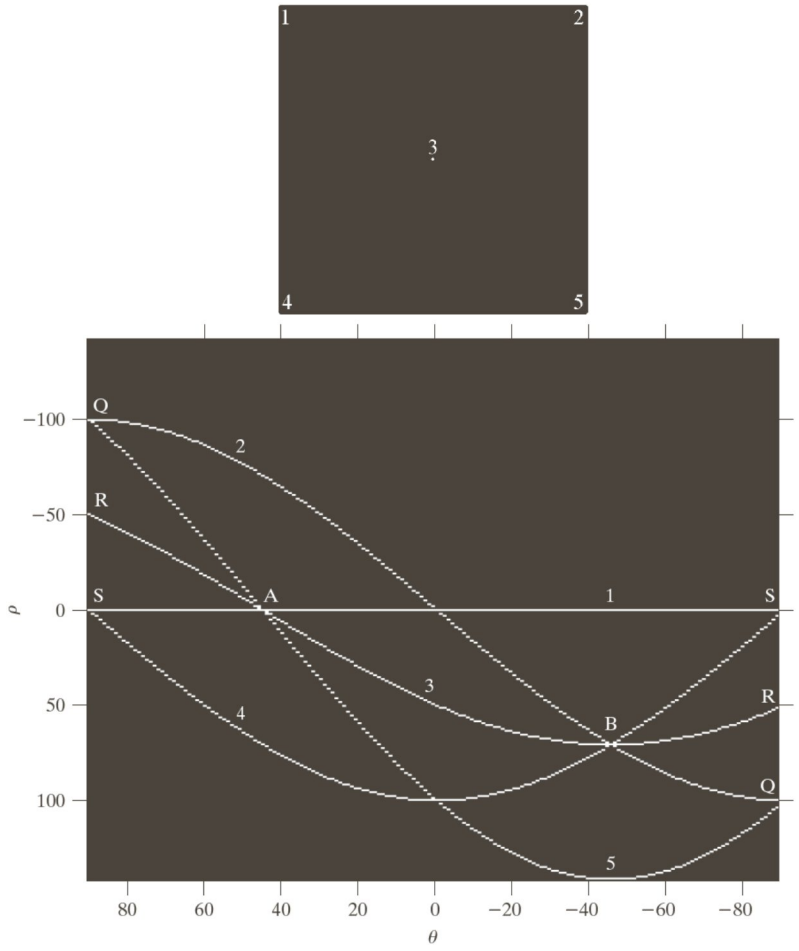
FIGURE 10.32 (a) (ρ, θ) parameterization of line in the xy-plane. (b) Sinusoidal curves in the $\rho\theta$-plane; the point of intersection (ρ', θ') corresponds to the line passing through points (x_i, y_i) and (x_j, y_j) in the xy-plane. (c) Division of the $\rho\theta$-plane into accumulator cells.

10.2.7 边缘连接和边界检测

- 使用霍夫变换的全局处理
  - 霍夫变换**将ρ−θ参数空间划分为所谓的累加单元**，坐标(i,j)处的单元具有累加值A(i,j)，它对应于与参数空间坐标(ρ_i,θ_i)相关联的正方形。
  - 最初，这些单元置为零。然后，对于xy平面中的每个非背景点(x_k,y_k)，令θ等于θ轴上每个允许的细分值，同时使用方程$\rho = x_k cos\theta + y_k sin\theta$解出对于的$\rho$。对得到的$\rho$值进行四舍五入得到沿$\rho$轴最接近的允许单元值。
  - 若选择的一个θ_p值得到解ρ_q，则令$A(p,q) = A(p,q) + 1$。在这一过程结束后，A(i,j)中的值P将意味着xy平面中有P个点位于直线$x\cos\theta_j + y\sin\theta_j = \rho_i$上。ρ−θ平面中的细分数量决定了这些点的共线精度。
  - 可以证明，刚刚讨论的这种方法的计算次数与xy平面中非背景点的数量n呈线性关系
- 霍夫变换也适用于形如$g(v,c) = 0$的任何函数，例如圆、椭圆
- 现在回到边缘连接问题。基于霍夫变换的一种连接方法如下：
  - （1）使用本节讨论的任何技术得到一幅二值图像
  - （2）指定ρ−θ平面中的细分
  - （3）对像素高度集中的地方检验其累加单元的数量
  - （4）检验选中单元中像素间的关系

10.2.7 边缘连接和边界检测

- 基本霍夫变换性质的说明

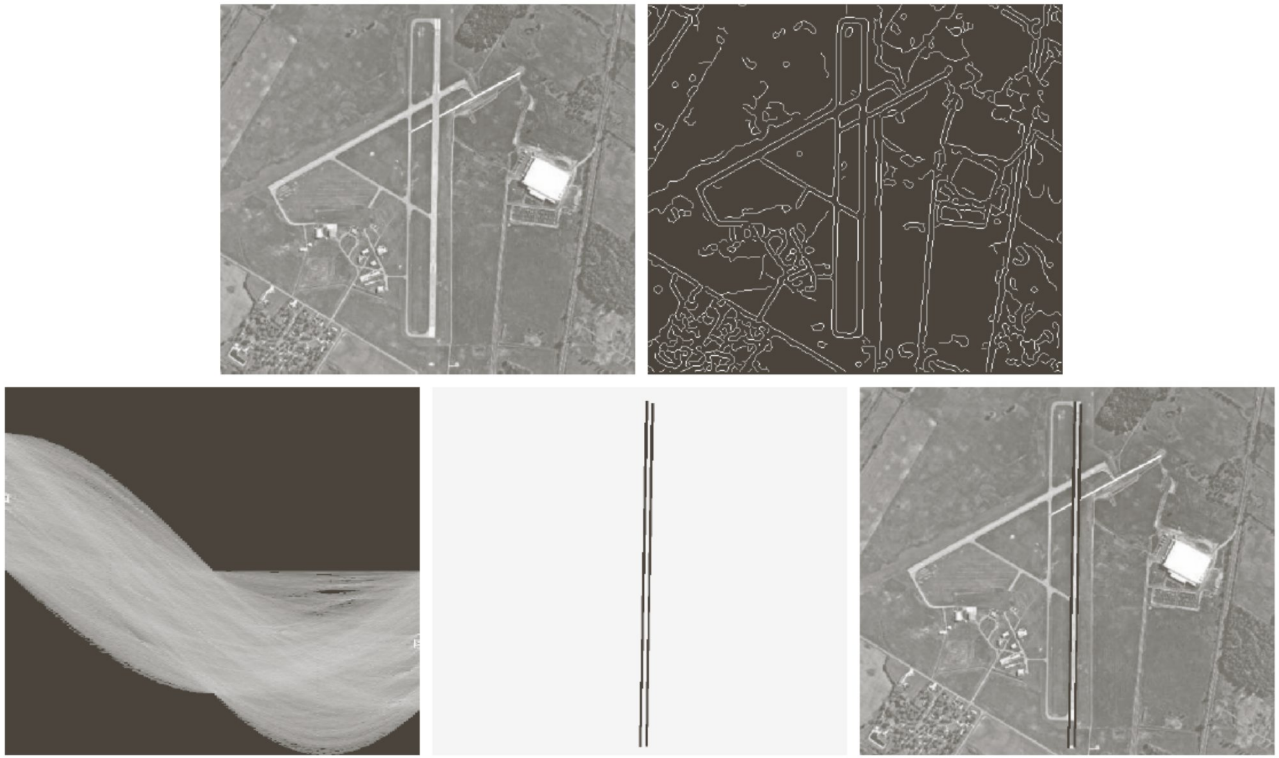


a
b

FIGURE 10.33
(a) Image of size 101 × 101 pixels, containing five points.
(b) Corresponding parameter space.
(The points in (a) were enlarged to make them easier to see.)

10.2.7 边缘连接和边界检测

- 使用霍夫变换连接边缘



a b
c d e

FIGURE 10.34 (a) A 502 × 564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes). (e) Lines superimposed on the original image.

第10章 图像分割

10.1 基础知识

10.2 点、线和边缘检测

10.3 阈值处理

10.4 基于区域的分割

10.5 用形态学分水岭的分割

10.6 分割中运动的应用

10.3 阈值处理

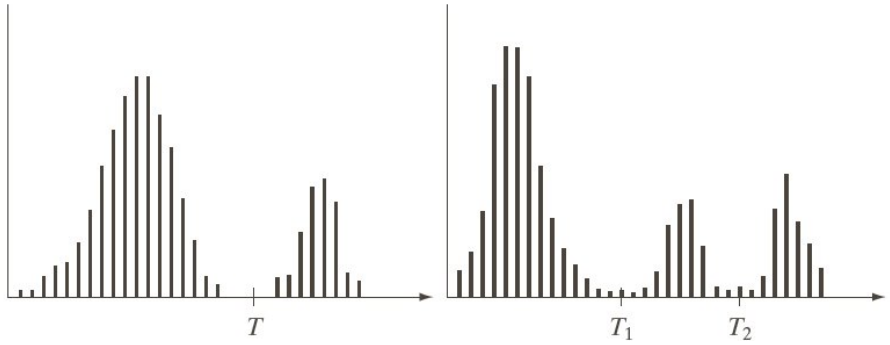
- 由于阈值处理直观、实现简单且计算速度快，因此图像阈值处理在图像分割应用中处于核心地位。
- 在前面的章节中，我们采用首先寻找边缘线段，然后将这些线段连接为边界的方法来识别区域。本节讨论基于灰度值或灰度值的特性来将图像直接划分为区域的技术
- 小节
  - 10.3.1 基础知识
  - 10.3.2 基本的全局阈值处理
  - 10.3.3 用Otsu方法的最佳全局阈值处理
  - 10.3.4 用图像平滑改善全局阈值处理
  - 10.3.5 利用边缘改进全局阈值处理
  - 10.3.6 多阈值处理
  - 10.3.7 可变阈值处理
  - 10.3.8 多变量阈值处理

10.3.1 基础知识

- 灰度阈值处理基础
  - 左图中的灰度直方图对应于图像$f(x,y)$，该图像由暗色背景上的较亮物体组成。
  - 从背景中提取物体：选择一个将这些模式分开的阈值T；然后，$f(x,y) > T$的任何点(x,y)称为一个对象点；否则该点称为背景点。
  - 换句话说，分割后图像$g(x,y)$由下式给出：

$$g(x,y) = \begin{cases} 1 & \text{如} \quad f(x,y) > T \\ 0 & \text{如} \quad f(x,y) \le T \end{cases}$$

  - 当T是一个适用于整个图像的常数时，称为全局阈值处理。当T在一幅图像上改



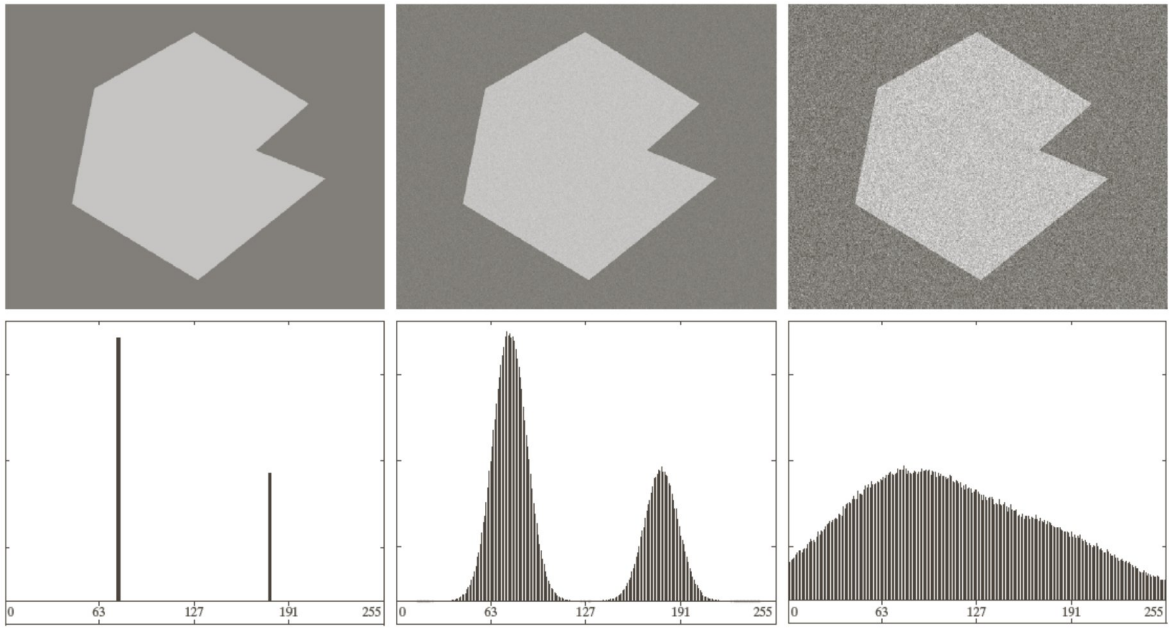
a b

FIGURE 10.35 Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds.

  - 右图显示了一个更为困难的阈值处理问题，它包含有三个支配模式的直方图。我们可以采用双阈值处理的方法

10.3.1 基础知识

- 图像阈值处理中噪声的作用



a b c
d e f

FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

10.3.1 基础知识

- 光照和反射的作用
  - 可变的灰度函数表示非均匀光照

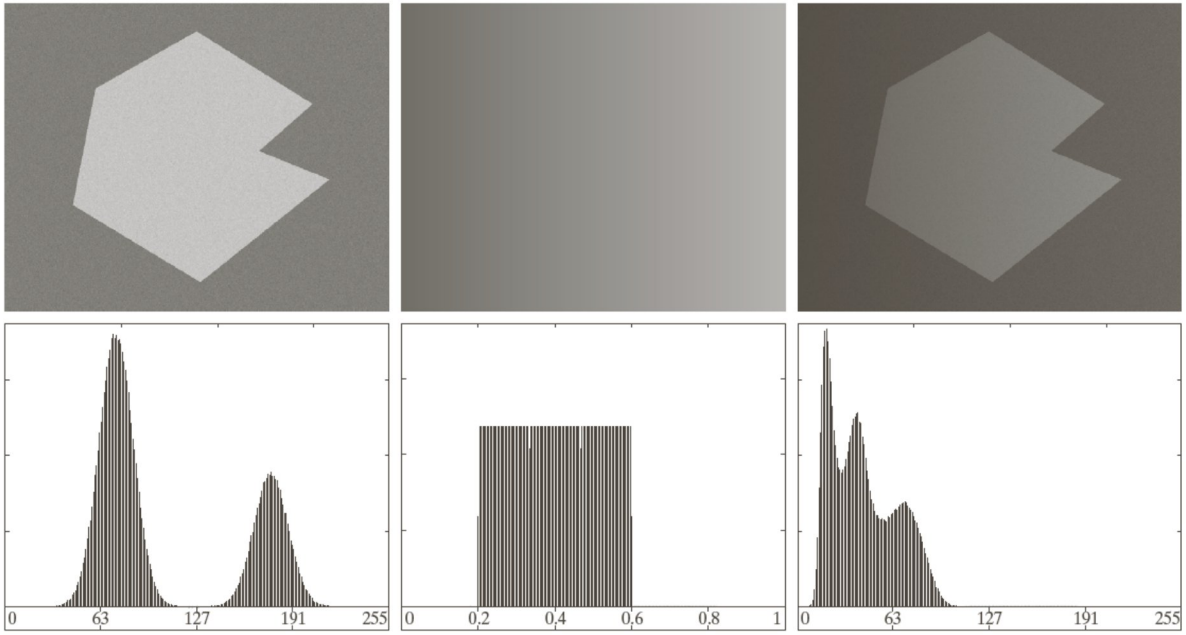


FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

  - 解决办法
    - ✓ 直接校正这种阴影模式
    - ✓ 通过**顶帽变换**处理来尝试校正全局阴影模式
    - ✓ 使用**可变阈值**近似处理非均匀性

10.3.2 基本的全局阈值处理

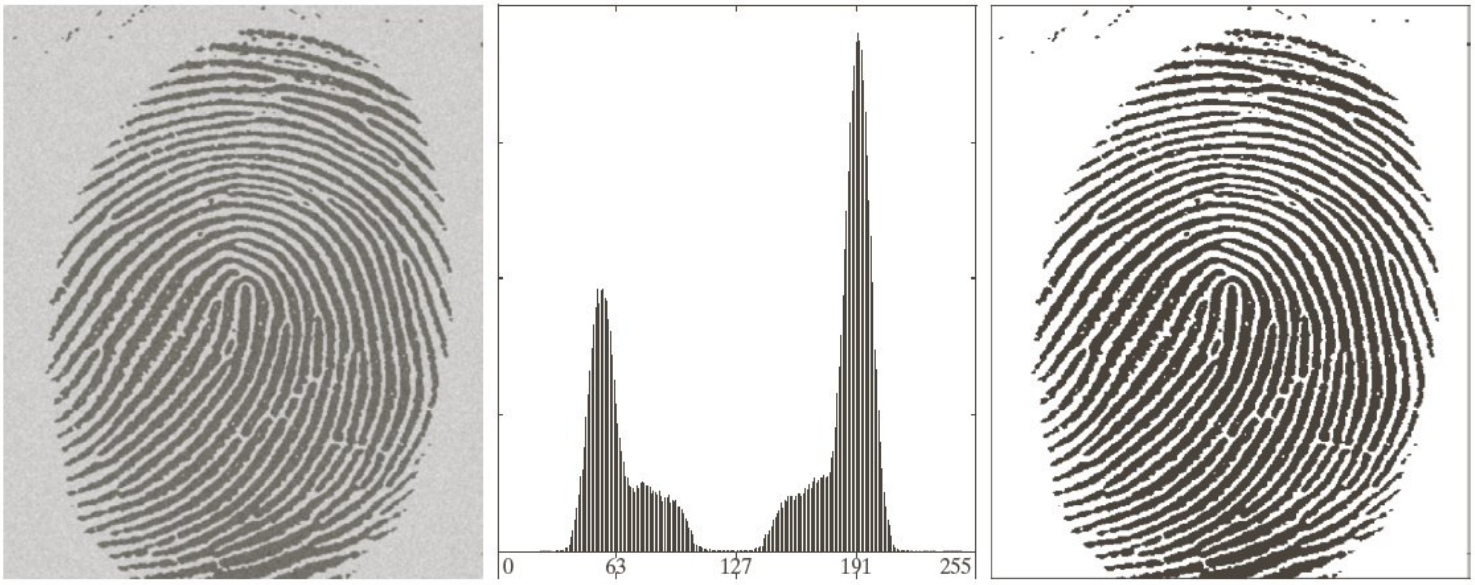
- 当物体和背景像素的灰度分布十分明显时，可以用适用于整个图像的单个（全局）阈值。
- 在大多数应用中，通常图像之间有较大变化，即使全局阈值是一种合适的方法，也需要有能对每幅图像**自动估计阈值**的算法。
- 下面的迭代算法可用于这一目的(**本质是k-means聚类**，k=2)
  - （1）为全局阈值T选择一个初始估计值
  - （2）用单个阈值T分割该图像。这将产生两组像素：G_1由灰度值大于T的所有像素组成，G_2由所有小于等于T的像素组成
  - （3）对G_1和G_2的像素分别计算平均灰度值（均值） m_1和m_2
  - （4）计算一个新的阈值：

$$T = \frac{1}{2}(m_1 + m_2)$$

  - （5）重复步骤（2）到步骤（4），直到连续迭代中的T值间的差小于一个预定义的参数ΔT为止
- 参数ΔT用于控制迭代的次数。通常， ΔT越大，则算法执行的迭代次数少。图像的平均灰度对于T来说是较好的初始选择

10.3.2 基本的全局阈值处理

- 全局阈值处理



a b c

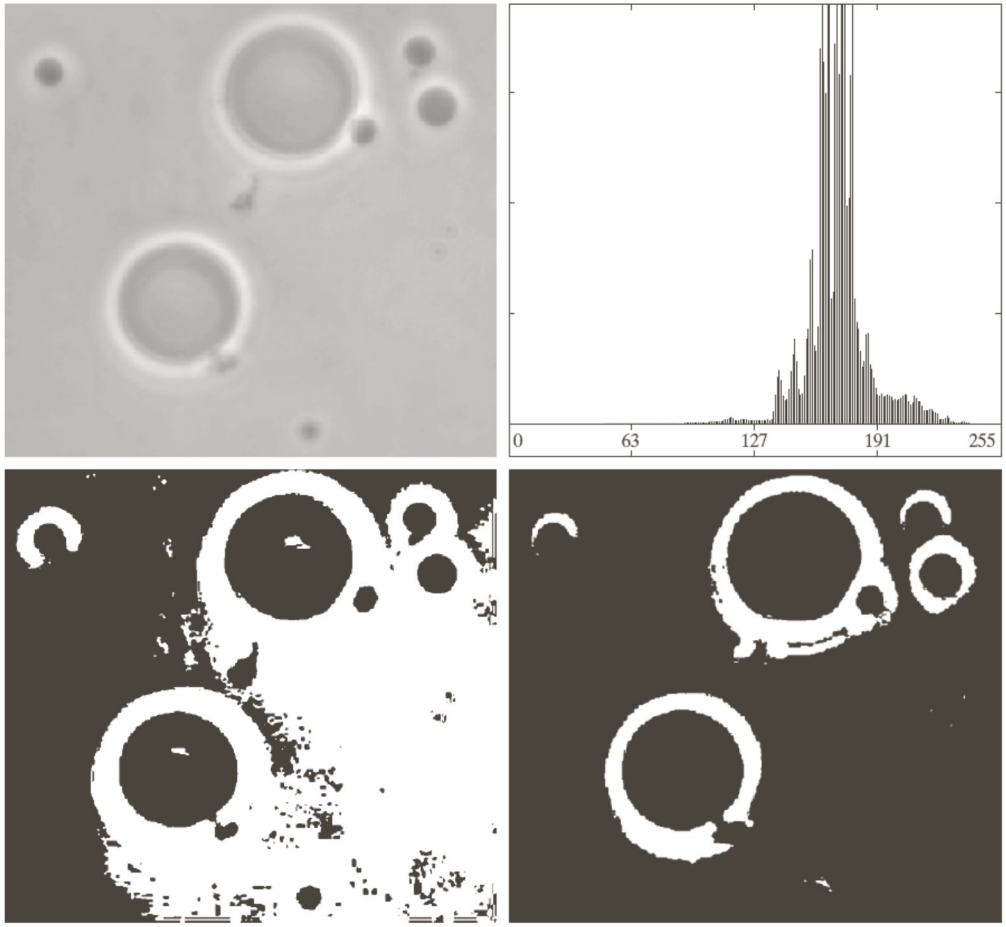
FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

10.3.3 Otsu方法：最佳全局阈值处理

- 该方法在**类间方差最大**的情况下是最佳的；完全以在一幅图像的直方图上执行计算为基础。
- Otsu算法小节如下：
  - 计算输入图像的**归一化**直方图。使用$p_i, i = 0,1,2,\cdots,L-1$表示该直方图的各个分量
  - 计算累积和：$P_1(k) = \sum_{i=0}^{k} p_i$，$(k = 0,1,2,\cdots,L-1)$，$P_2(k) = 1 - P_1(k)$
  - 计算累积均值：$m(k) = \sum_{i=0}^{k} ip_i$，对于$k = 0,1,2,\cdots,L-1$，
  - 计算全局灰度均值：$m_G = \sum_{i=0}^{L-1} ip_i$
  - 用$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$，对于$k = 0,1,2,\cdots,L-1$，计算类间方差$\sigma_B^2(k)$
  - 得到Otsu阈值k^*，即使得$\sigma_B^2(k)$最大的k值。如果最大值不唯一，用相应检测到的各个最大值的k的平均得到k^*
  - 在$k = k^*$处计算$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2}$，得到可分性测度$\eta^*$

10.3.3 用Otsu方法的最佳全局阈值处理

- 使用Otsu方法的最佳全局阈值处理

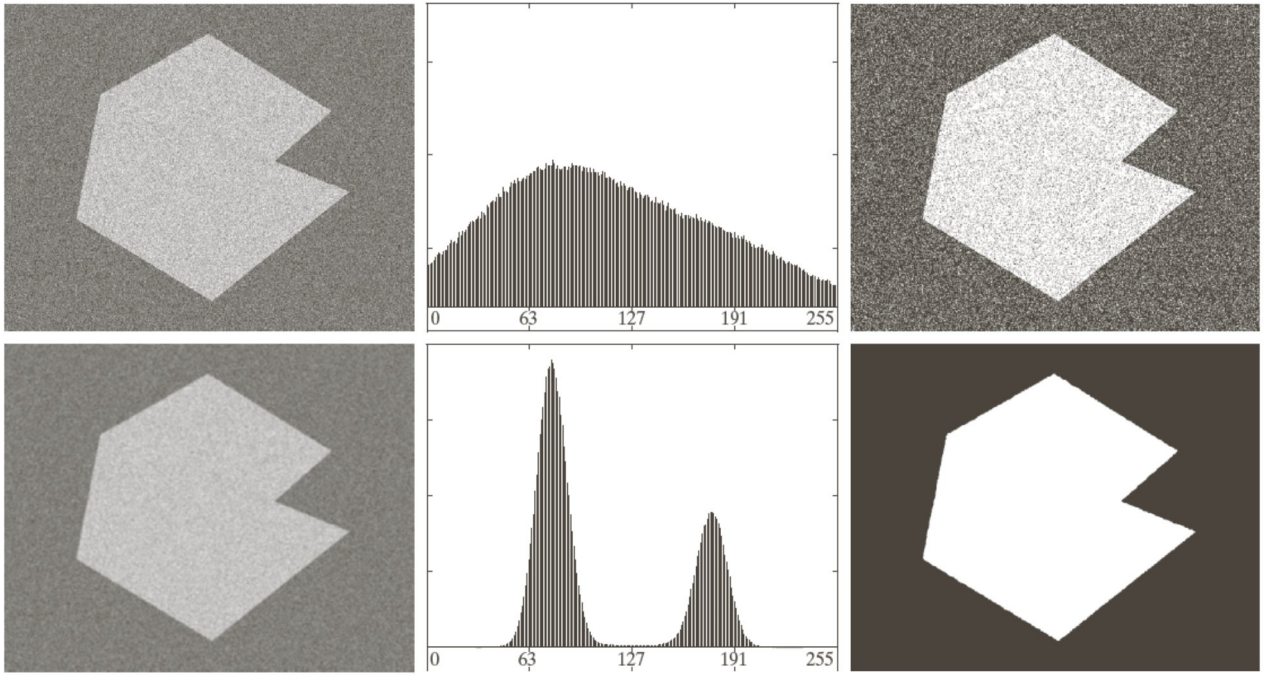


a b
c d

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.)

10.3.4 用图像平滑改善全局阈值处理

- 噪声会将简单的阈值处理问题变为不可解决的问题
  - 解决方法：在阈值处理之前平滑图像
- 例：

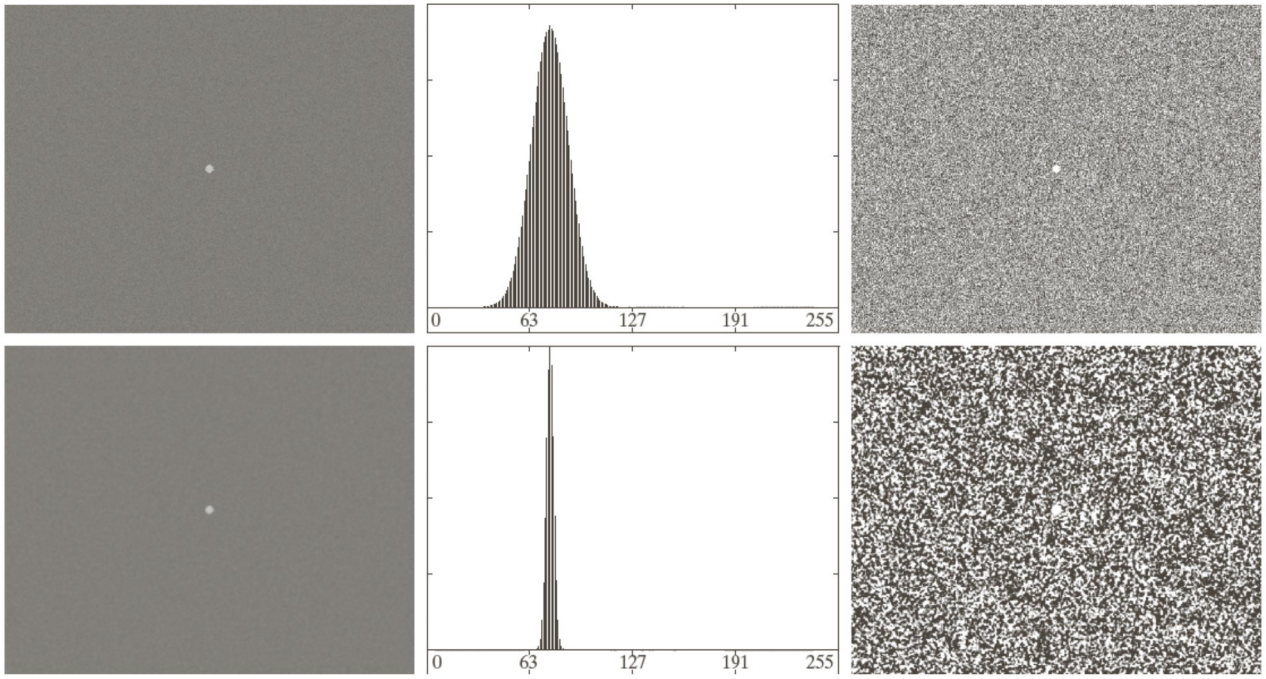


a b c
d e f

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.

10.3.4 用图像平滑改善全局阈值处理

- 考虑降低图中相对于背景的区域大小所带来的影响
  - 失败的原因可归于这样一个事实，即区域太小，以至于该区域对直方图的贡献与由噪声引起的灰度扩散相比无足轻重。



a b c
d e f

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.

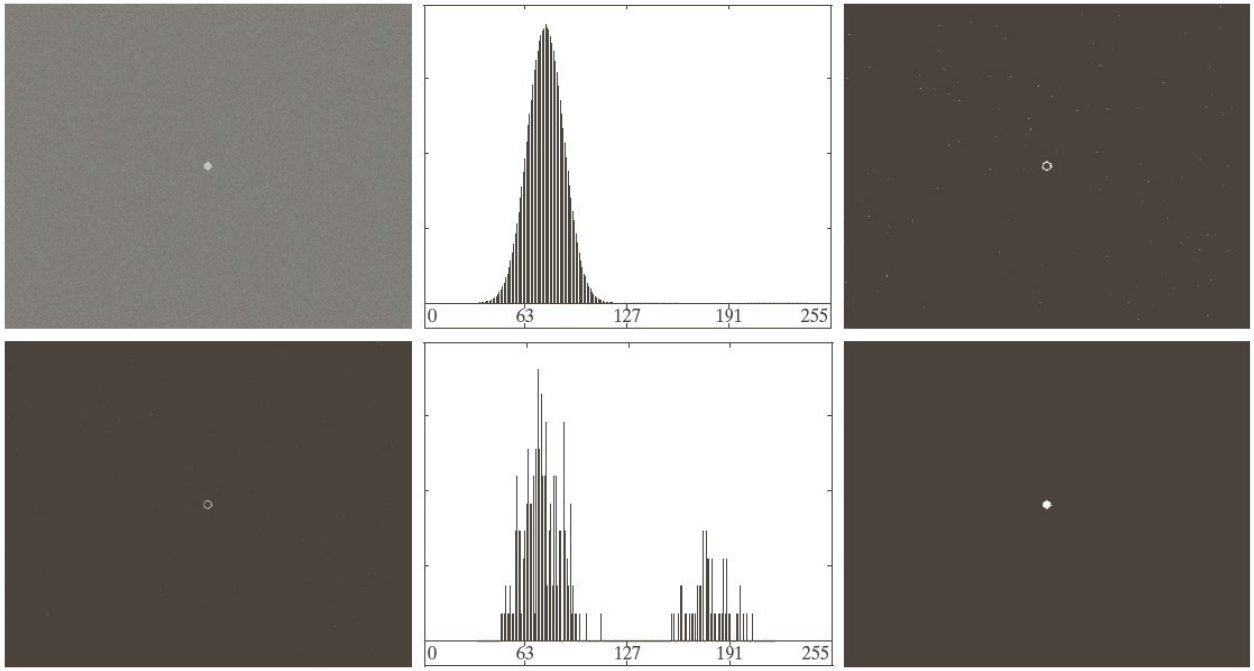
10.3.5 利用边缘改进全局阈值处理

- 如果直方图的波峰是高、窄、对称的，且被深波谷分开，则选取一个“较好”阈值的机会相当大。**改进直方图形状的一种方法是，仅考虑那些位于或靠近物体和背景间的边缘的像素**。该算法可小结如下，其中$f(x,y)$是输入图像：
  - 1、采用10.2节讨论的任何一种方法来计算一幅**边缘图像**，无论是$f(x,y)$梯度的幅度还是拉普拉斯的绝对值均可
  - 2、指定一个阈值T
  - 3、用步骤2中的阈值对步骤1中的图像进行阈值处理，产生一幅二值图像$g_T(x,y)$。在从$f(x,y)$中选取对应于“强”边缘像素的下一步中，该图像用做一幅模板图像
  - 4、仅用$f(x,y)$中对应于$g_T(x,y)$中像素值为1的位置的像素计算直方图
  - 5、用步骤4中直方图全局地分割$f(x,y)$，例如使用Otsu方法

  - 若将T设为小于边缘图像的最小值的任何值，$g_T(x,y)$将由全1组成，这意味着$f(x,y)$的所有像素将用于计算图像的直方图。习惯以百分比的形式来指定T值，通常较高（90%），以便在计算中用梯度/拉普拉斯图像中的较少像素

10.3.5 利用边缘改进全局阈值处理

- 用以梯度为基础的边缘信息改进全局阈值处理

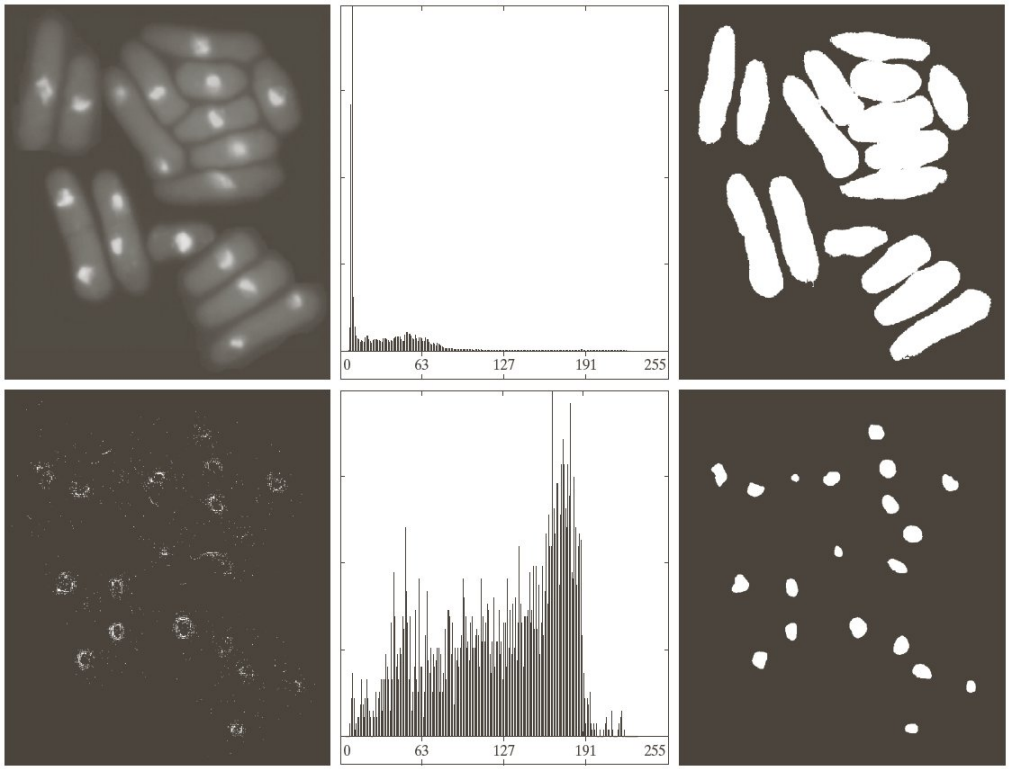


a b c
d e f

FIGURE 10.42 (a) Noisy image from Fig. 10.41(a) and (b) its histogram. (c) 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.

10.3.5 利用边缘改进全局阈值处理

- 用拉普拉斯为基础的边缘信息改进全局阈值处理



a b c
d e f

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.)

10.3.6 多阈值处理

- 迄今为止，我们关注的是用单个全局阈值对图像进行分割。我们可将10.3.3节介绍的阈值处理方法扩展到任意数量的阈值，因此以这种方法为基础的可分性测度也可以扩展到任意数量的分类
  - 在K个类$C_1, C_2, \cdots, C_K$的情况下，类间方差可归纳为下面的表达式：

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

式中，$P_k = \sum_{i \in C_k} P_i$，$m_k = \frac{1}{P_k} \sum_{i \in C_k} iP_i$，$m_G = \sum_{i=0}^{L-1} ip_i$

K类有K-1个阈值来分离，这些值$k_1^*, k_2^*, \cdots, k_{K-1}^*$是上式的最大值：

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

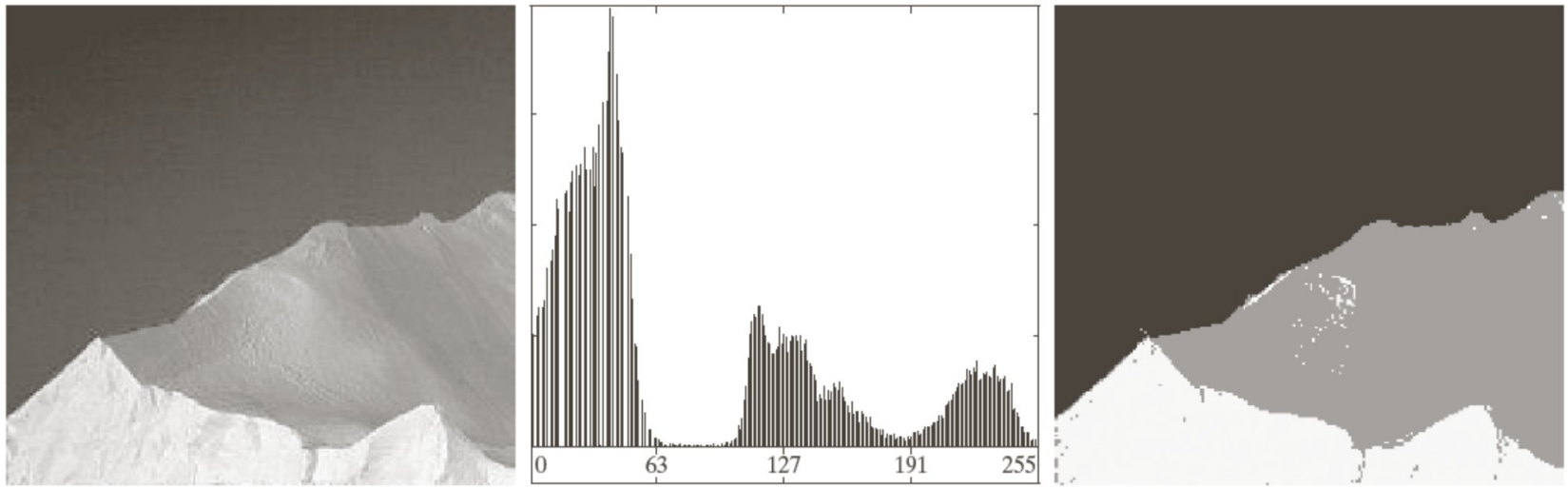
  - 对于由三个灰度间隔组成的三个类（两个阈值），最佳阈值是：

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

该过程由选择第一个k_1值开始（为1）。接着，k_2的所有值在大于k_1和小于L-1的范围内增加。然后，将k_1增大到其下一个值，k_2的所有值再次在大于k_1的所有值范围内增加。重复该过程，直到$k_1 = L-3$为止。该结果是一个二维阵列$\sigma_B^2(k_1, k_2)$，最后一步是在阵列中寻找最大值。对应于最大值的k_1值和k_2值就是最佳阈值k_1^*和k_2^*。

10.3.6 多阈值处理

- 多个全局阈值处理

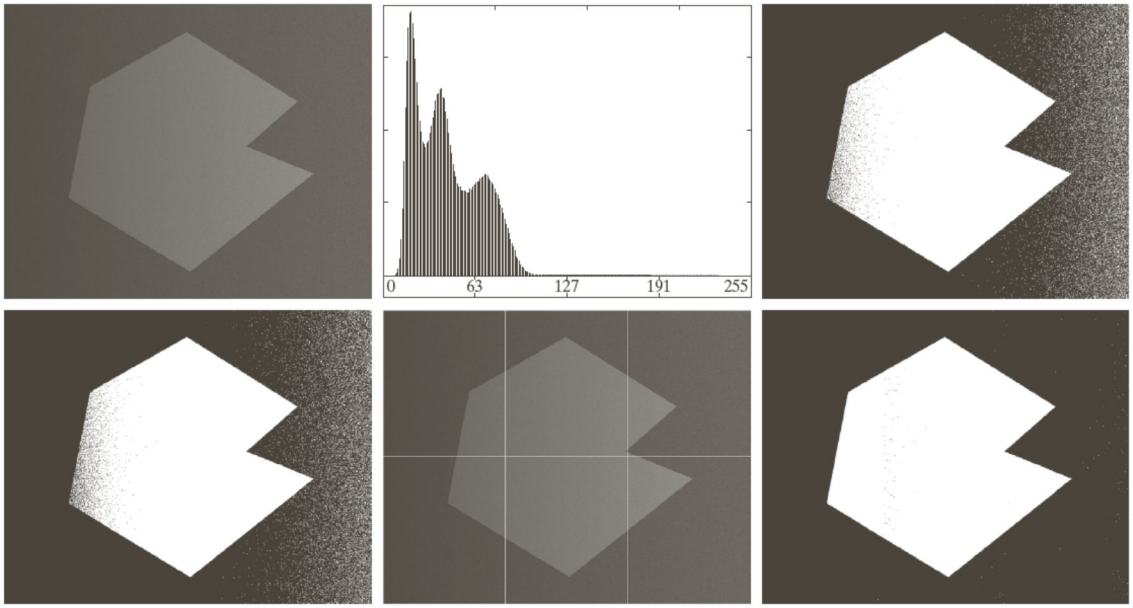


a b c

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

10.3.7 可变阈值处理

- 图像分块
  - 可变阈值处理最简单的方法之一是，把一幅图像分成不重叠的矩形。
  - 这种方法用于补偿光照和/或反射的不均匀性。选择的矩形要足够小，以便每个矩形的光照都近似是均匀的
  - 例：



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.

10.3.7 可变阈值处理

- 基于局部图像特性的可变阈值处理
  - 更为一般的方法是在一幅图像中的每点(x,y)计算阈值，该阈值以一个或多个在(x,y)邻域计算的特性为基础
  - 令σ_{xy}和m_{xy}表示一幅图像中以坐标(x,y)为中心的邻域S_{xy}所包含的像素集合的标准差和均值。下面是可变局部阈值的通用形式：

$$T_{xy} = a\sigma_{xy} + bm_{xy}$$

式中，a和b是非负常数，且$T_{xy} = a\sigma_{xy} + bm_G$。分割后的图像计算如下：

$$g(x,y) = \begin{cases} 1, & f(x,y) > T_{xy} \\ 0, & f(x,y) \le T_{xy} \end{cases}$$

该式对图像中的所有像素位置进行求值，并在每个点(x,y)处使用邻域S_{xy}中的像素计算不同的阈值

  - 使用以(x,y)处的邻域计算出的参数为基础的属性，有效的权值可显著增强局部阈值处理：

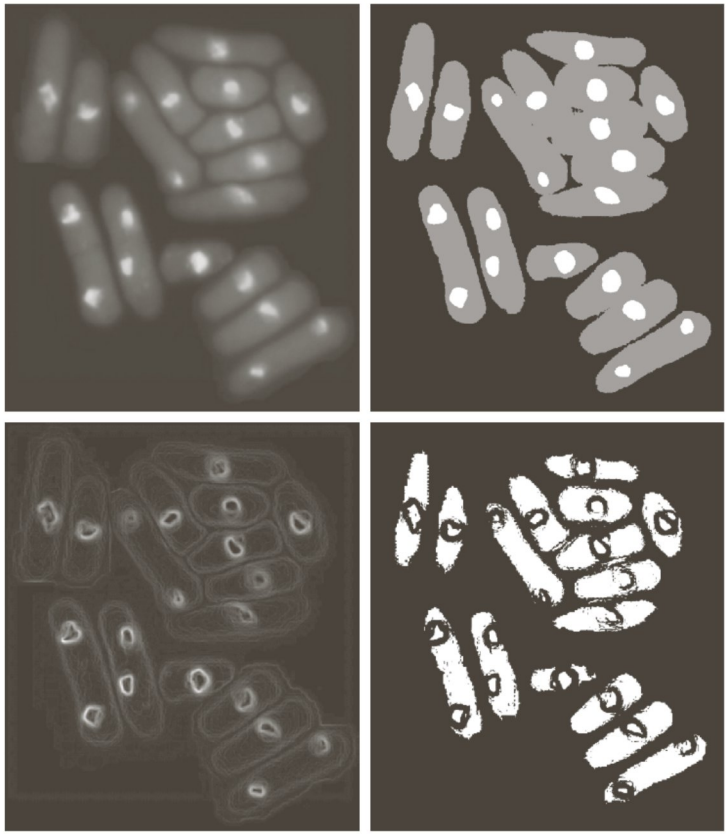
$$g(x,y) = \begin{cases} 1, & Q(\text{局部参数})\text{为真} \\ 0, & Q(\text{局部参数})\text{为假} \end{cases}$$

式中，Q是以邻域S_{xy}中像素计算的参数为基础的一个属性。

10.3.7 可变阈值处理

- 基于局部图像特性的可变阈值处理

$$Q(\sigma_{xy}, m_{xy}) = \begin{cases} \text{真}, & f(x,y) > a\sigma_{xy} \ AND f(x,y) > bm_{xy} \\ \text{假}, & \text{其他} \end{cases}$$



a b
c d

FIGURE 10.48
(a) Image from Fig. 10.43. (b) Image segmented using the dual thresholding approach discussed in Section 10.3.6. (c) Image of local standard deviations. (d) Result obtained using local thresholding.

10.3.7 可变阈值处理

- 使用移动平均
  - 刚才讨论的局部阈值处理方法的一种特殊情形，是以一幅图像的**扫描行**计算**移动平均**为基础的。
  - 通常，为减少光照偏差，扫描是以**Z字形**模式逐行执行的。令z_{k+1}表示步骤$k+1$中扫描序列遇到的点的灰度。这个新点的移动平均由下式给出：

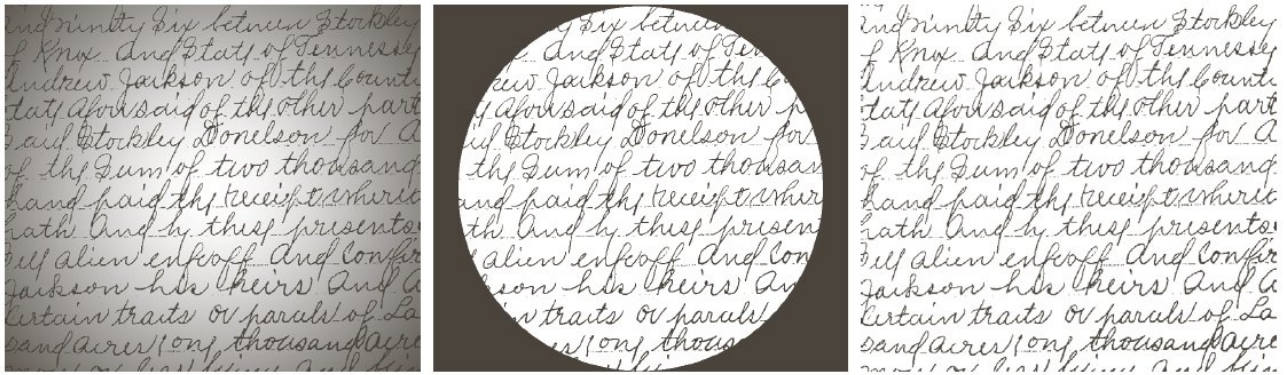
$$m(k+1)=\frac{1}{n}\sum_{i=k+2-n}^{k+1}z_i=m(k)+\frac{1}{n}(z_{k+1}-z_{k+1-n})$$

  - 式中，n是用于计算平均的点数，$m(1)=z_1/n$。

  - 该算法仅初始化一次，而不在每行都初始化。因为对图像中的每个点都计算移动平均，因此用$g(x,y)=\begin{cases}1, & f(x,y)>T_{xy}\\ 0, & f(x,y)\le T_{xy}\end{cases}$实现分割
    - ✓ 其中$T_{xy}=bm_{xy}$，b是常数，m_{xy}是在输入图像中的点(x,y)处得到的移动平均

10.3.7 可变阈值处理

- 用移动平均的文档阈值处理



a b c

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.



a b c

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.

10.3.8 多变量阈值处理

- 迄今为止是基于单个变量（灰度）的阈值处理。
- 某些情况下，传感器可产生多个可利用的变量来表征图像中的每个像素，这样，就允许进行多变量阈值处理。例如，对彩色图像的阈值处理

- 多变量阈值处理可视为一种距离计算。
  - 假设我们想要从一幅彩色图像中提取具有指定颜色的所有区域。令a表示，计算任意彩色点z和平均彩色a间的距离测度$D(z,a)$。然后，按下式分割图像：

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

  - 其中，T是一个阈值。通常，如果z是一个n维向量，则距离可用欧式距离或马氏距离度量

第10章 图像分割

10.1 基础知识

10.2 点、线和边缘检测

10.3 阈值处理

10.4 基于区域的分割

10.5 用形态学分水岭的分割

10.6 分割中运动的应用

10.4 基于区域的分割

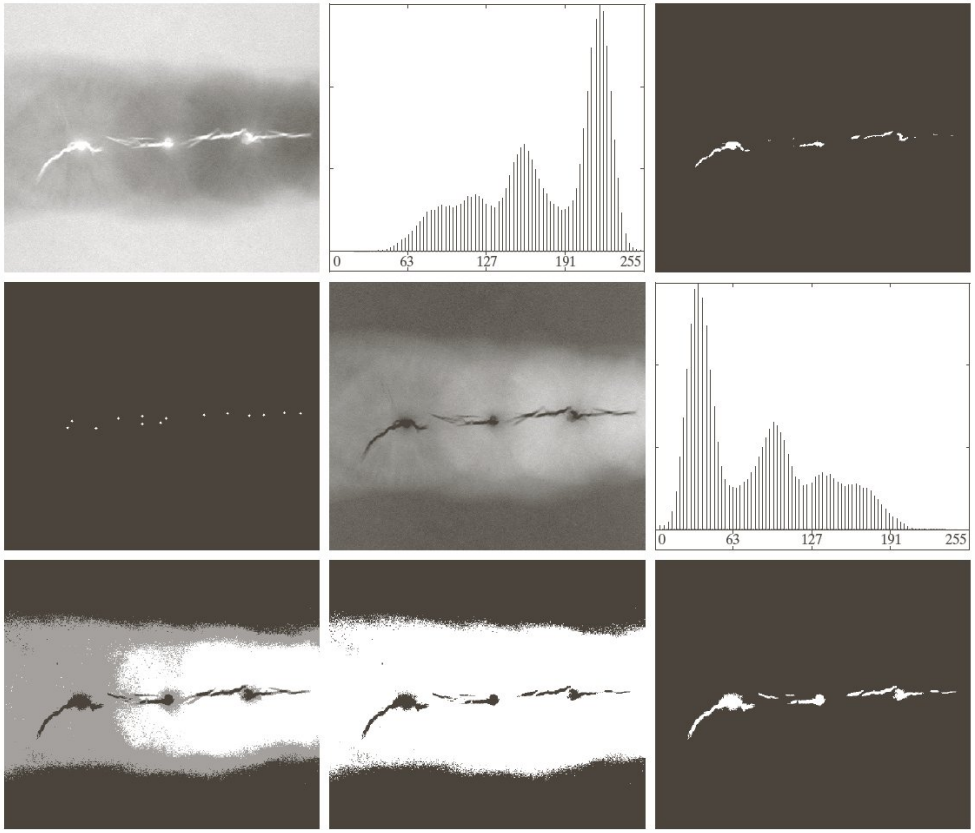
- 如10.1节讨论那样，分割的目的是将一幅图像划分为多个区域。
- 在10.2节中，我们基于灰度级的不连续性尝试寻找区域间的边界来解决这一问题。
- 而在10.3节中，分割是通过以像素特性分布为基础的阈值处理来完成的，如灰度值或彩色。
- 本节讨论以直接寻找区域为基础的分割技术
- 小节
  - 10.4.1 区域生长
  - 10.4.2 区域分裂与聚合

10.4.1 区域生长

- 区域生长是根据预先定义的生长准则，将像素或子区域组合为更大区域的过程。
- 基本方法是**从一组“种子”点开始**，将与种子预先定义的**性质相似**的那些邻域像素添加到每个种子上，来形成这些生长区域。
  - **连通属性，终止规则**
- 令$f(x,y)$表示一个输入图像；$S(x,y)$表示一个种子阵列，阵列中种子点位置处为1，其他为0；Q表示在每个位置(x,y)处的属性。
- 假设阵列f和S的尺寸相同。基于8连接的一个基本区域生长算法：
  - 1、在$S(x,y)$中寻找所有连通分量，并把每个连通分量腐蚀为一个像素；把找到的所有这种像素标记为1，把S中的所有其他像素标记为0
  - 2、在坐标对(x,y)处形成图像f_Q：**若输入图像在该坐标处满足给定的属性Q**，则令$f_Q(x,y)=1$，否则令$f_Q(x,y)=0$
  - 3、令g是这样形成的图像：即把f_Q中为8连通种子点的所有1值点，添加到S中的每个种子点
  - 4、用不同的区域标记（如$1,2,3,\cdots$）标记g中的每个连通分量。这就是由区域生长得到的分割图像

10.4.1 区域生长

- 使用区域生长的分割

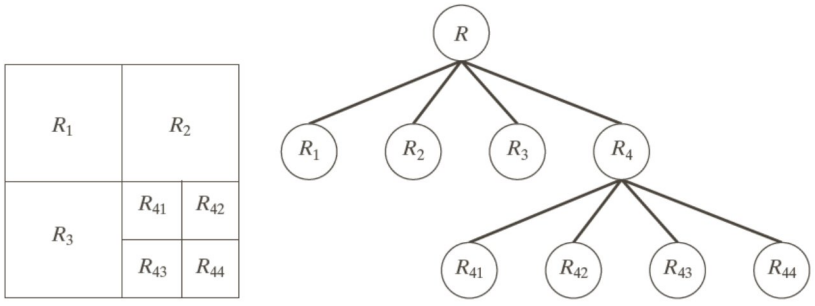


a b c
d e f
g h i

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.)

10.4.2 区域分裂与聚合

- 基本思想：首先将一幅图像细分为一组任意的不想交的区域，然后聚合和/或分裂这些区域
- 令R表示整幅图像区域，并选择一个**属性Q**。则区域分裂与聚合的算法过程如下：
  - 把满足$Q(R_i) = FALSE$的任何区域R_i分裂为4个不相交的象限区域
  - 不可能进一步分裂时，对满足条件$Q\left(R_j \cup R_k\right) = TRUE$的任意两个邻接区域$R_j$和$R_k$进行聚合
  - 无法进一步聚合时，停止操作
- 一般要规定一个不能再进一步执行分裂的最小四象限区的尺寸

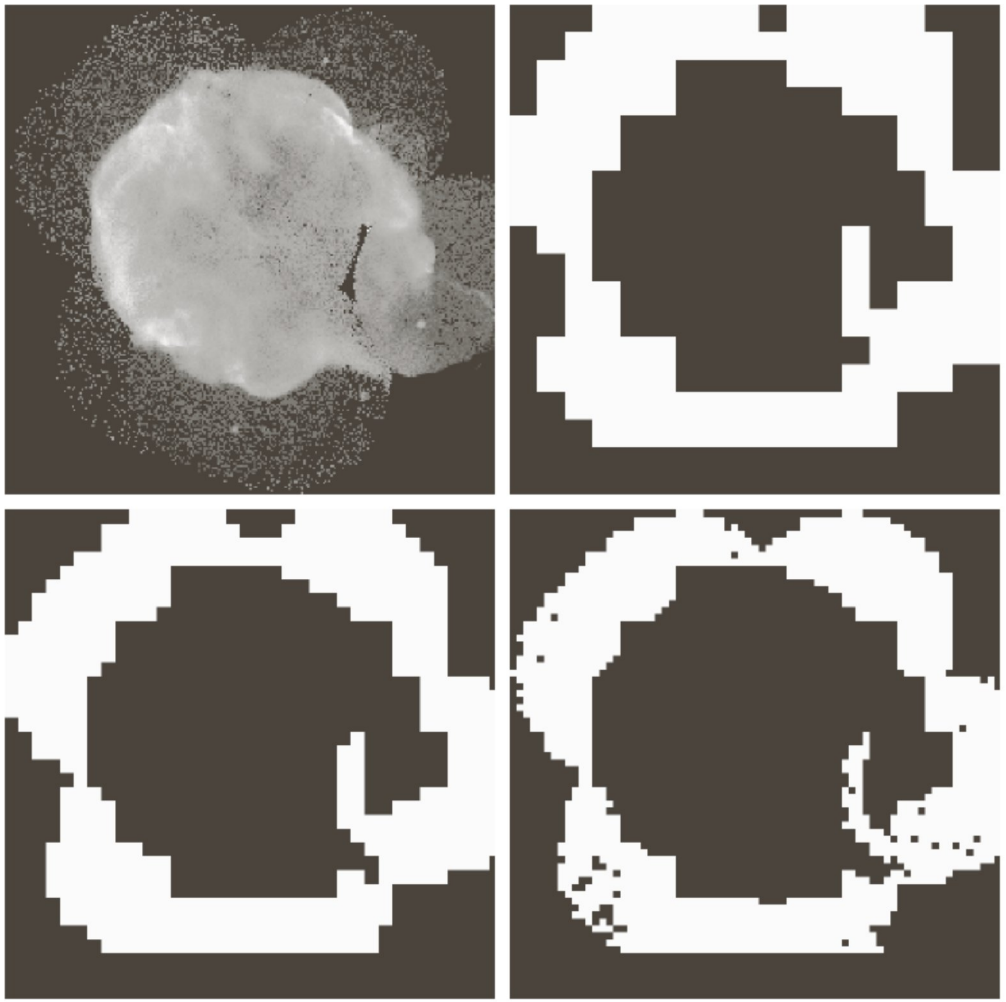


a b

FIGURE 10.52 (a) Partitioned image. (b) Corresponding quadtree. R represents the entire image region.

10.4.2 区域分裂与聚合

- 用区域分裂和聚合进行分割



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.)

第10章 图像分割

10.1 基础知识

10.2 点、线和边缘检测

10.3 阈值处理

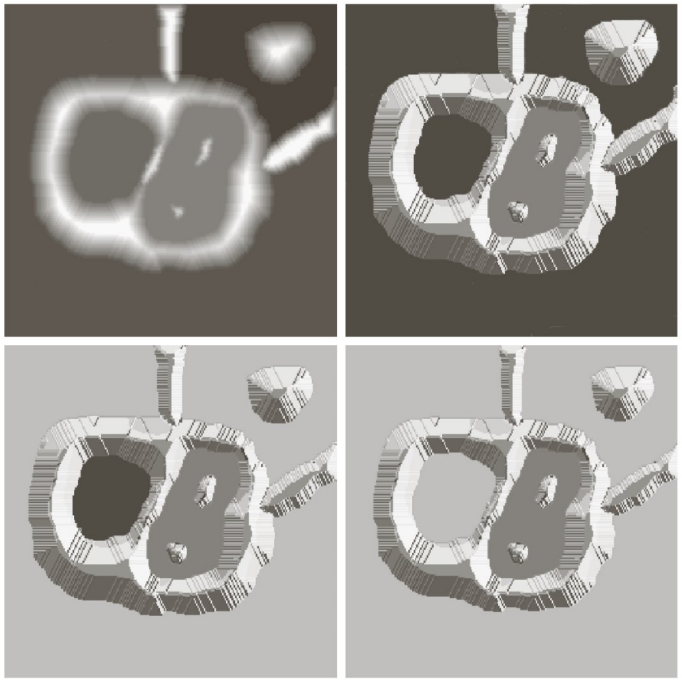
10.4 基于区域的分割

10.5 用形态学分水岭的分割

10.6 分割中运动的应用

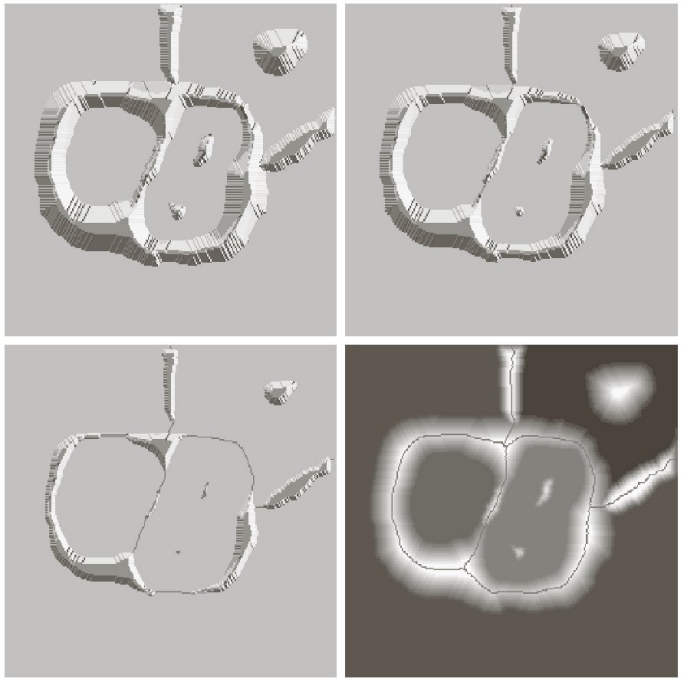
10.5.1 背景知识

- 分水岭也称分水线/水线，将图像的梯度图看成三维地形表示
- 基本思想：
  - 假设在每个区域的最小值上打一个洞，并且让水通过洞以均匀的速率上升，从低到高淹没整个地形。当不同汇水盆地中上升的水聚集时，修建一个水坝来阻止这种聚合。
  - 水将达到在水线上只能见到各个水坝的顶部的程度。这些大坝的边界对应于分水岭的分割线



a b
c d

FIGURE 10.54
(a) Original image. (b) Topographic view. (c)–(d) Two stages of flooding.



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.)

10.5.2 水坝构建

- 使用形态学膨胀来构建水坝：
  - 令C[n−1]表示a图两个集合的并集
  - 令q表示b图中这个连通分量
- 假设a图中的每个连通分量被c图所示结构元膨胀满足两个条件：
  - 膨胀必须受到q约束（膨胀过程中，结构元的中心只能位于q中的点处）
  - 不能对会引起正被膨胀集合聚合（变成单个连通分量）的那些点执行膨胀
- 很明显，q中只有满足上述两个条件的点，才能描绘出d图中由叉线所示的1像素宽的连通路径

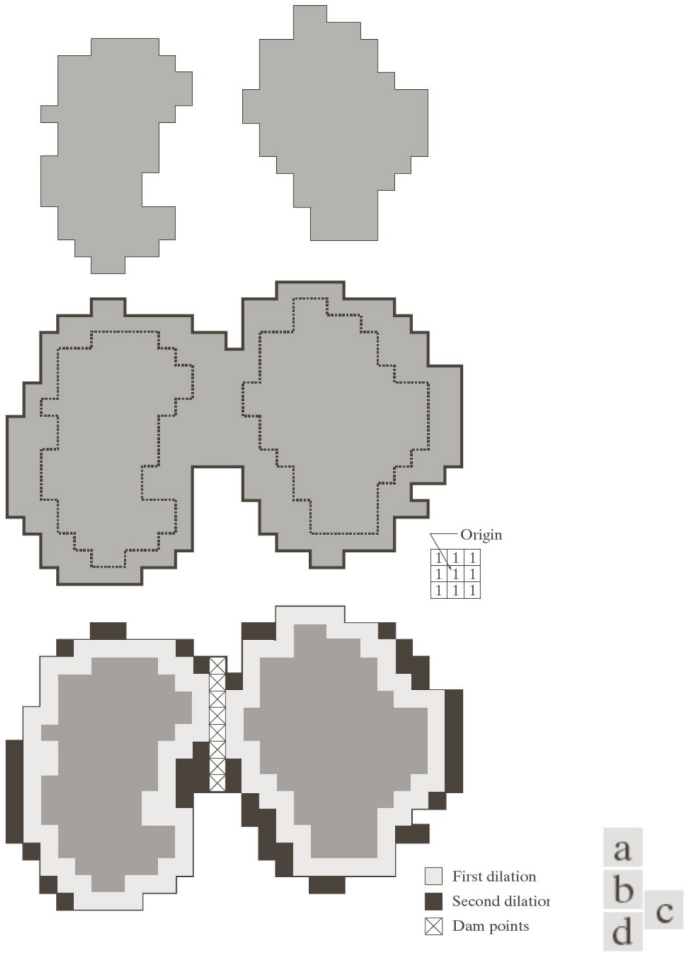


FIGURE 10.55 (a) Two partially flooded catchment basins at stage $n-1$ of flooding. (b) Flooding at stage n, showing that water has spilled between basins. (c) Structuring element used for dilation. (d) Result of dilation and dam construction.

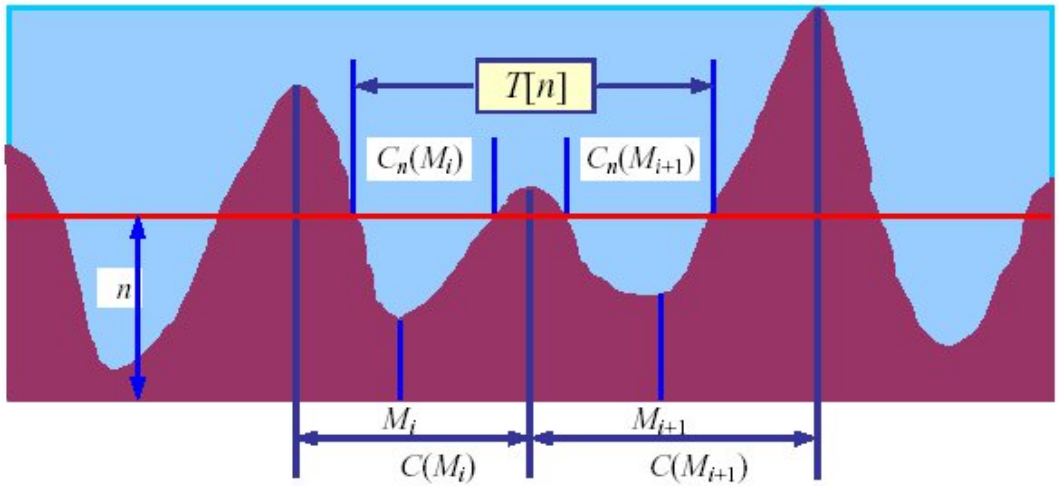
10.5.3 分水岭分割算法

- 计算步骤：
  - 待分割图象$f(x,y)$，其梯度图象为$g(x,y)$
  - 用M_1，M_2，…，M_R表示$g(x,y)$中各局部极小值的象素位置，$C(M_i)$为与M_i对应的区域中的象素坐标集合。
  - 用n表示当前阈值，$T[n]$代表记为：

$$T[n]=\{(u,v)\,|\,g(u,v)<n\}$$

  - 对M_i所在的区域，其中满足条件的坐标集合$C_n(M_i)$可看作一幅二值图象

$$C_n(M_i)=C(M_i)\cap T[n]$$



10.5.3 分水岭分割算法

- 计算步骤：
  - 用$C[n]$代表在灰度阈值为n时图象中所有满足条件的象素

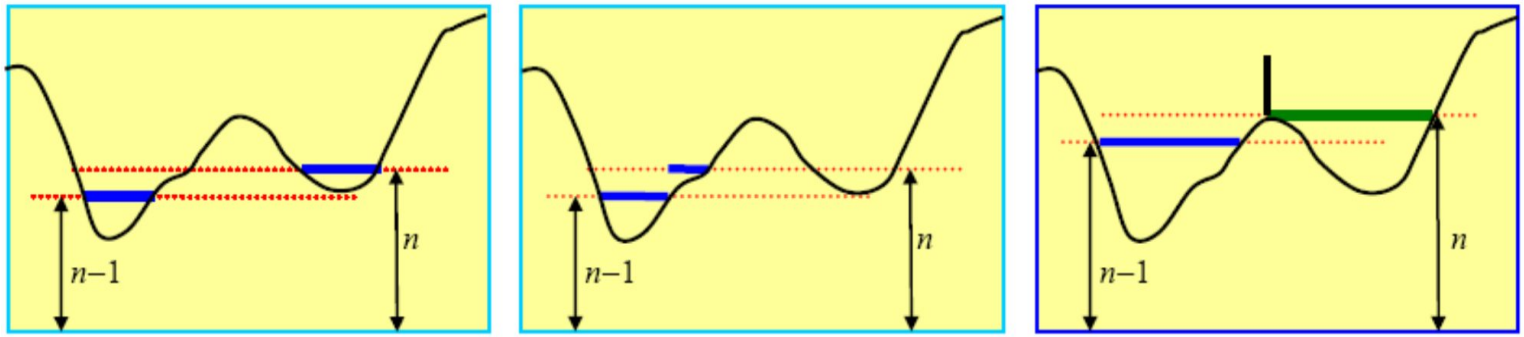
$$C[n] = \bigcup_{i=1}^{R} C_n(M_i)$$

  - $C[\max + 1]$将是所有区域的并集：$C[max+1] = \bigcup_{i=1}^{R} C_{max+1}(M_i)$
  - $C[n-1]$是$C[n]$的子集，$C[n]$等价于$T[n]$，所以$C[n-1]$又是$T[n]$的子集
  - 令 S 代表$T[n]$中的连通组元集合，对每个连通组元$s \in S[n]$，有3种可能性：

    1、$s \cap C[n-1]$是1个空集，则$C[n]$可由把连通组元 s 加到$C[n-1]$中得到

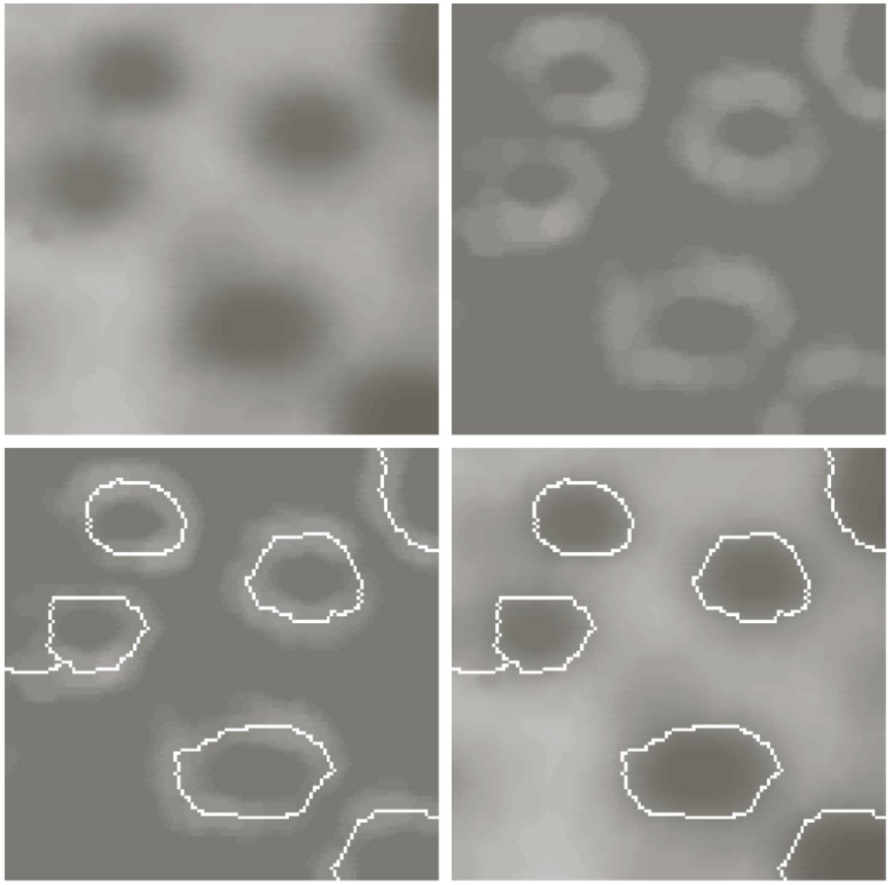
    2、$s \cap C[n-1]$里包含$C[n-1]$中的一个连通组元，则与上同

    3、$s \cap C[n-1]$里包含$C[n-1]$中一个以上的连通组元，则需要在 s 中建分水岭



10.5.3 分水岭分割算法

- 分水岭分割算法的说明

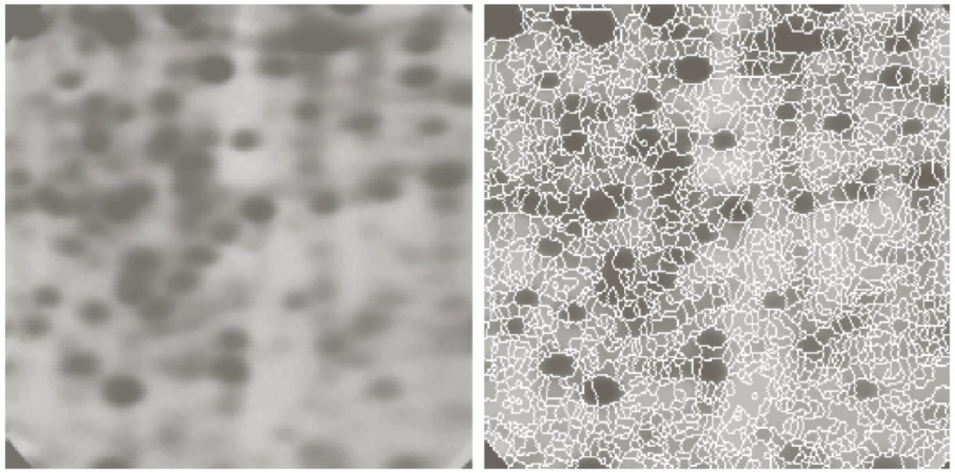


a b
c d

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.)

10.5.4 标记的使用

- 分水岭算法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，如图所示
  - 解决方案是，加入预处理步骤来限制允许存在的区域的数量



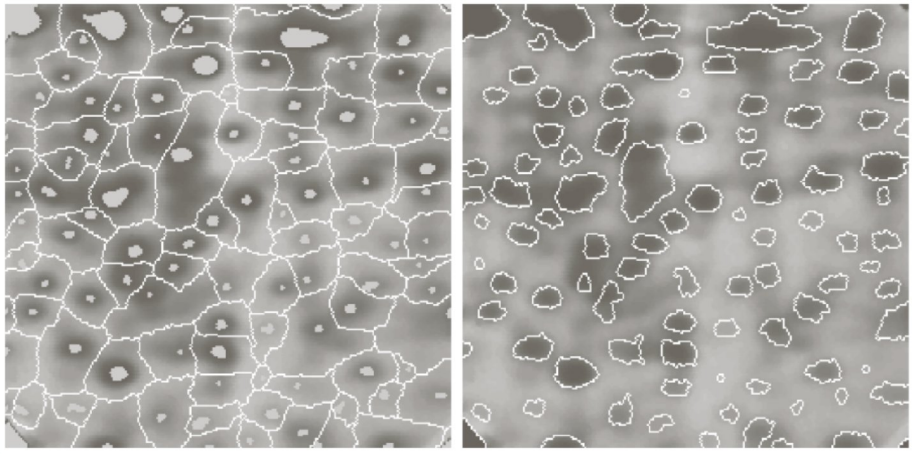
a b

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.)

- 用于控制过度分割的一种方法基于标记。标记是指属于一幅图像的**连通分量**。
  - 与感兴趣物体联系的标记称为**内部标记**，与背景关联的标记称为**外部标记**。
- 选择标记的典型过程由两个主要步骤组成
  - （1）预处理，比如平滑滤波；（2）定义标记必须满足的一个准则集合

10.5.4 标记的使用

- 假设我们将内部标记定义为：
  - （1）被高“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）连通分量中所有的点有相同的灰度值
- 图像经过平滑后，内部标记在a图中以浅灰色、斑点区域显示。分水线定义为外部标记。外部标记有效地将图像分割成了不同的区域，问题简化为将每个区域划分为两部分：单个物体及其背景
- 将算法限制在只对在该特殊区域中包含这一标记的单一分水岭进行操作



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.)

第10章 图像分割

10.1 基础知识

10.2 点、线和边缘检测

10.3 阈值处理

10.4 基于区域的分割

10.5 用形态学分水岭的分割

10.6 分割中运动的应用

10.6.1 空间域技术

- 基本方法
  - 分别检测在时刻t_i和t_j获取的两帧图像$f(x,y,t_i)$和$f(x,y,t_j)$之间的变化的最简方法之一是，逐像素比较这两幅图像。形成的一幅差值图像定义为

$$d_{ij}(x,y)=\begin{cases}1, & \left|f(x,y,t_i)-f(x,y,t_j)\right|>T\\ 0, & \text{其他}\end{cases}$$

  - 实际上，$d_{ij}(x,y)$中的1值项可能由噪声造成，去除它们的简单办法是，在$d_{ij}(x,y)$中构成由1组成的4连通或8连通区域，然后忽略1值项少于预定元素数量的任何区域
- 累积差值
  - 考虑一个图像帧序列$f(x,y,t_1),f(x,y,t_2),\cdots,f(x,y,t_n)$，令$f(x,y,t_1)$为参考图像。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。
  - 考虑如下累积差值图像的三种类型：绝对ADI、正ADI和负ADI。令$R(x,y)$表示参考图像，对于任何k>1，且记住ADI的值是计数，我们对(x,y)的所有相关值定义如下：

$$A_k(x,y)=\begin{cases}A_{k-1}(x,y)+1, & \left|R(x,y)-f(x,y,k)\right|>T\\ A_{k-1}(x,y), & \text{其他}\end{cases}$$

10.6.1 空间域技术

- 累积差值

$$P_k(x,y)=\begin{cases}P_{k-1}(x,y)+1, & \mathrm{R(x,y)}-f(x,y,k)>T\\ P_{k-1}(x,y), & \text{其他}\end{cases}$$

$$N_k(x,y)=\begin{cases}N_{k-1}(x,y)+1, & \mathrm{R(x,y)}-f(x,y,k)<-T\\ N_{k-1}(x,y), & \text{其他}\end{cases}$$

- 式中，$A_k(x,y)$，$P_k(x,y)$和$N_k(x,y)$分别为遇到序列中的第k幅图像后的绝对ADI、正ADI和负ADI
- 绝对、正和负的累积差异图像的计算



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.

10.6.1 空间域技术

- 建立参考图像
  - 前两节的技术成功的关键是，找到一幅可以与后续图像进行比较的**参考图像**
  - 实际上，仅用固定元素得到一幅参考图像并不总是可行的，所以有必要根据一组包含一个或多个运动物体的图像来构建一幅参考图像。
  - 生成参考图像的过程：考虑将图像序列中的第一幅图像作为参考图像；当一个非固定成分完全移出其参考帧的位置时，当前帧中对应背景可复制到最初被参考帧中物体所占据的位置。
  - 当所有的移动物体均已完全移出原来的位置时，就创建了一幅仅包含固定成分的参考图像。
  - 通过监视正ADI的变化，可确定物体的位移
  - 构造一幅参考图像



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.)