第9章　形态学图像处理

- 形态学
  - 通常表示生物学的一个分支，研究动植物的形态和结构
  - 本章使用同一词语表示数学形态学的内容
- 数字形态学的作用
  - 从图像中提取表达和描绘区域形状的有用图像分量，如边界、骨架和凸壳等。
  - 我们对预处理或后处理的形态学技术也感兴趣，比如形态学过滤、细化和修剪等。
- 本章介绍的许多概念可在n维欧式空间E^n中用公式表达
  - 本章首先在Z^2空间，讨论二值图像
  - 在9.6节，我们的讨论将扩展到灰度图像。

第9章　形态学图像处理

9.1　预备知识

9.2　腐蚀和膨胀

9.3　开操作与闭操作

9.4　击中与击不中变换

9.5　一些基本的形态学算法

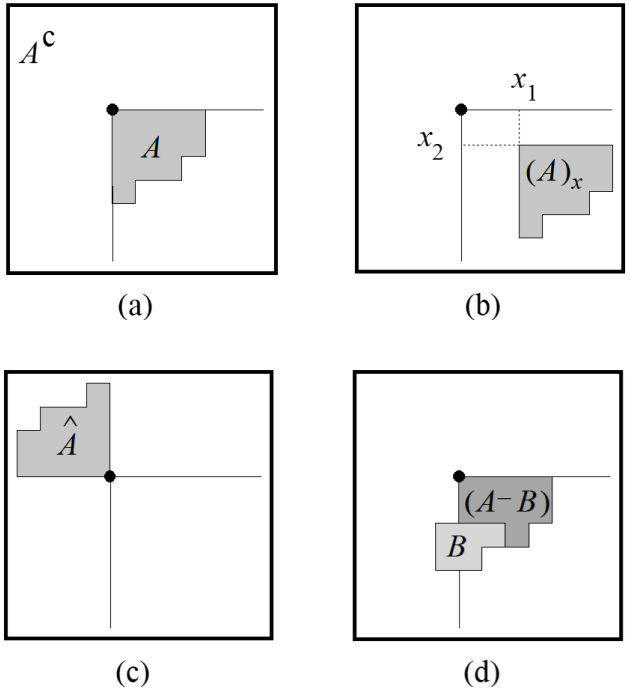
9.6　灰度级形态学

第9章　形态学图像处理

9.1　预备知识
9.2　腐蚀和膨胀
9.3　开操作与闭操作
9.4　击中与击不中变换
9.5　一些基本的形态学算法
9.6　灰度级形态学

9.1 预备知识

- 数学形态学的语言是集合论
  - 数学形态学中的集合表示图像中的对象。
  - 例如，在二值图像中，所有**白色像素**的集合是该图像的一个完整的形态学描述。问题集合是二维整数空间Z^2的元素。
- 基本集合定义：
  - 集合：用大写字母表示，空集记为∅
  - 元素：用小写字母表示
  - 子集： $A \subseteq B$
  - 并集： $C = A \cup B$
  - 交集： $D = A \cap B$
  - 补集： $A^c = \{x | x \notin A\}$
  - **平移**： $(A)_x = \{y | y = a + x, a \in A\}$
  - **映像**： $\hat{A} = \{x | x = -a, a \in A\}$
  - 差集： $A - B = \{x | x \in A, x \notin B\} = A \cap B^c$



9.1 预备知识

- 形态学中集合的映像和平移广泛用来表达基于结构元（SE）的操作：研究一幅图像中感兴趣特性所用的小集合或子图像。
- 结构元：
  - 成员：每个涂阴影的方块
  - **原点**：由一个黑点指出；当SE对称且未显示原点时，假定原点位于对称中心处
  - 矩形阵列：通过添加最小可能数量的背景元素形成一个矩形阵列来实现。

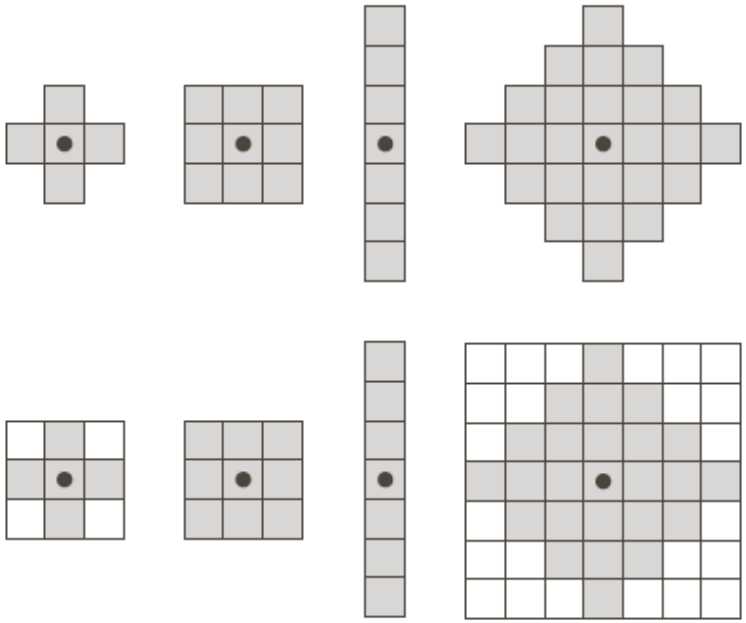


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

9.1 预备知识

- 如何使用结构元?
  - 用添加背景元素的办法把待处理的集合A也转换为一个矩形阵列。
  - 当结构元原点位于原始集合的边界上时，背景边界要大到足以容纳整个结构元。
  - 边界腐蚀操作：
    - ✓ 通过让B在A上滑动，以便B的原点访问A的每个元素，来创建一个新集合。
    - ✓ 在B的每个原点位置，如果B完全被A包含，则将该位置标记为新集合的一个成员；否则，将该位置标记为非新集合的成员。

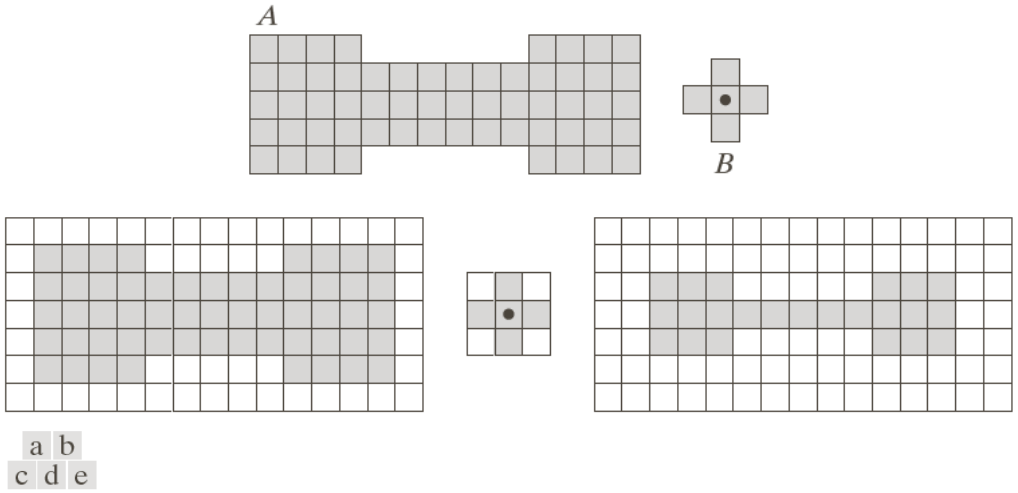


FIGURE 9.3 (a) A set (each shaded square is a member of the set). (b) A structuring element. (c) The set padded with background elements to form a rectangular array and provide a background border. (d) Structuring element as a rectangular array. (e) Set processed by the structuring element.

第9章　形态学图像处理

9.1　预备知识

9.2　腐蚀和膨胀

9.2.1　腐蚀

9.2.2　膨胀

9.2.3　对偶性

9.3　开操作与闭操作

9.4　击中与击不中变换

9.5　一些基本的形态学算法

9.6　灰度级形态学

9.2.1 腐蚀

- 作为Z^2中的集合A和B，表示为$A \ominus B$的B对A的腐蚀定义为

$$A \ominus B = \{z|(B)_z \subseteq A\}$$

  - 将B平移，当其原点位于z时，其包含在A中，则z为一个有效的位置
  - 所有有效的z构成了腐蚀之后的结果
- 因为B必须包含在A中这一陈述等价于B不与背景共享任何公共元素，故可以将腐蚀表达为如下的等价形式：

$$A \ominus B = \{z|(B)_z \cap A^c = \emptyset\}$$

- 腐蚀的一个例子：
  - B对A的腐蚀

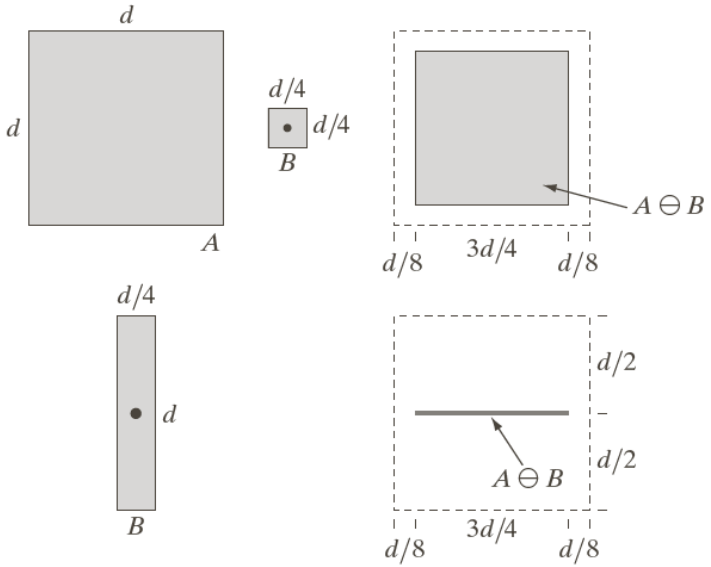
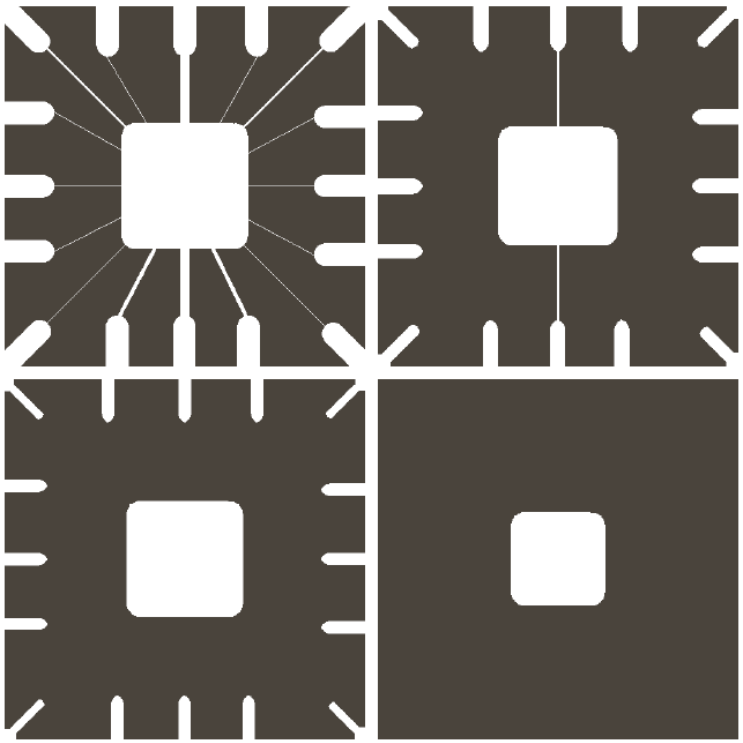


FIGURE 9.4 (a) Set A. (b) Square structuring element, B. (c) Erosion of A by B, shown shaded. (d) Elongated structuring element. (e) Erosion of A by B using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference.

9.2.1 腐蚀

- 腐蚀：去除图像的某些部分
  - 假设我们希望去掉图中连接中心区域到边界焊接点的线
- 腐蚀缩小或细化了二值图像中的物体。
- 事实上，我们可以将腐蚀视为形态学滤波操作，这种操作把比结构元小的图像细节从图像中滤除



a b
c d

FIGURE 9.5 Using erosion to remove image components. (a) A 486 × 486 binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes 11 × 11, 15 × 15, and 45 × 45, respectively. The elements of the SEs were all 1s.

第9章 形态学图像处理

9.1 预备知识

9.2 腐蚀和膨胀

9.2.1 腐蚀

9.2.2 膨胀

9.2.3 对偶性

9.3 开操作与闭操作

9.4 击中与击不中变换

9.5 一些基本的形态学算法

9.6 灰度级形态学

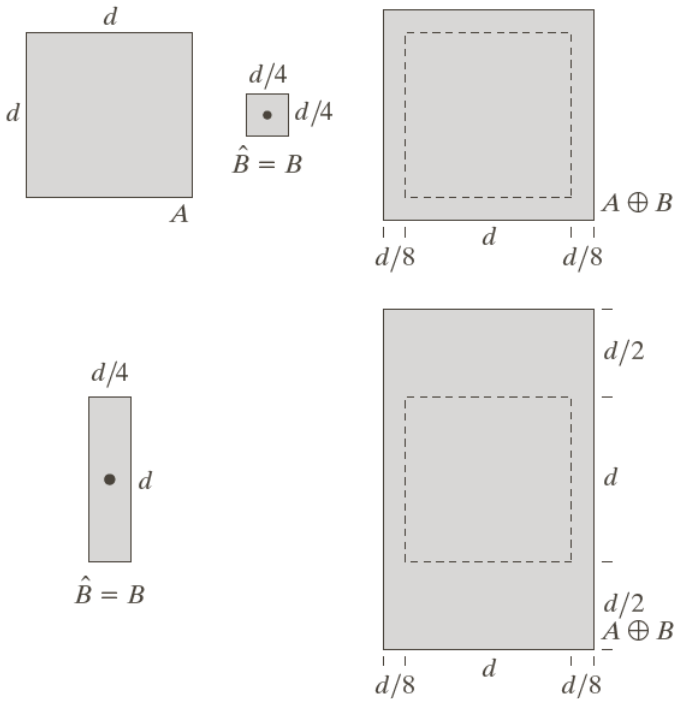
9.2.2 膨胀

- A和B是Z^2中的集合，表示为A ⊕ B的B对A的膨胀定义为

$$A \oplus B = \{z|(\hat{B})_z \cap A \neq \emptyset\}$$

  - 将B关于原点映像之后，再平移，当其原点平移到z时，如果其与A有重叠元素，则z为一个有效的位置
  - 所有有效的z构成了膨胀之后的结果

- 腐蚀是一种收缩或细化操作，
- 而膨胀则会“增长”或“粗化”二值图像中的物体。

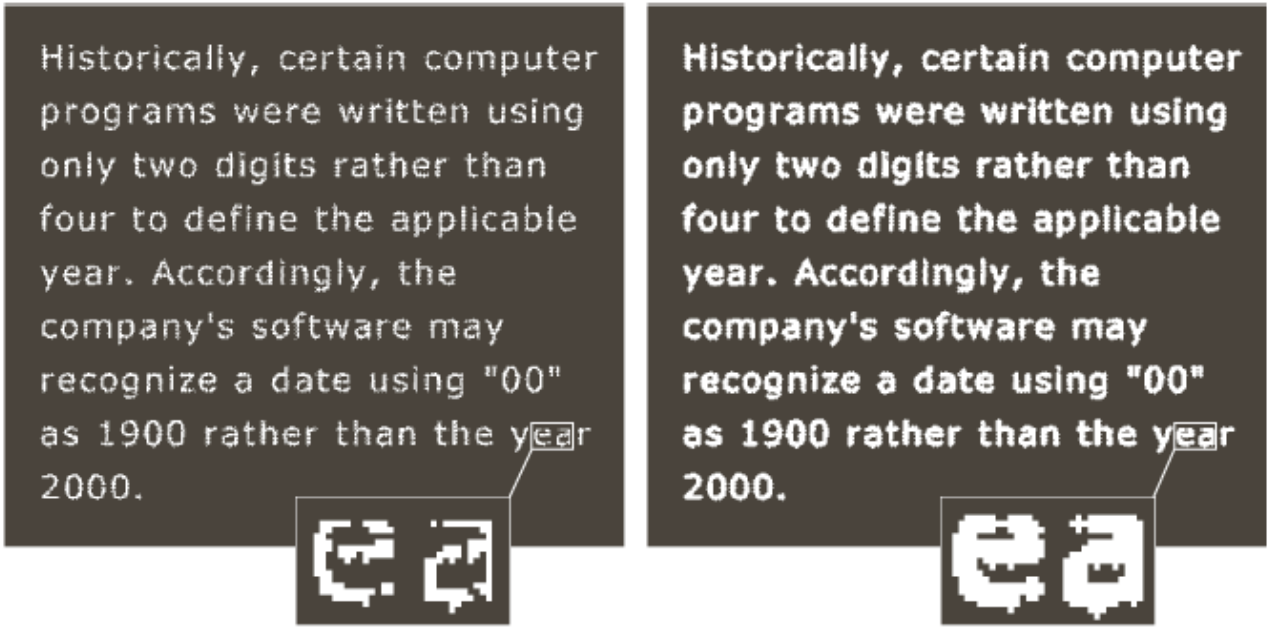


a b c
d e

FIGURE 9.6
(a) Set *A*. (b) Square structuring element (the dot denotes the origin). (c) Dilation of *A* by *B*, shown shaded. (d) Elongated structuring element. (e) Dilation of *A* using this element. The dotted border in (c) and (e) is the boundary of set *A*, shown only for reference

9.2.2 膨胀

- 膨胀应用：连接裂缝
  - **低通滤波方法**则从一幅二值图像开始，生成一幅灰度图像，它需要用一个阈值函数将灰度图像转换为二值图像。
  - **形态学方法**一个直接优点是，形态学方法可在一幅二值图像中直接得到结果。



a c
b

FIGURE 9.7
(a) Sample text of poor resolution with broken characters (see magnified view). (b) Structuring element. (c) Dilation of (a) by (b). Broken segments were joined.

0	1	0
1	1	1
0	1	0

第9章 形态学图像处理

9.1 预备知识

9.2 腐蚀和膨胀

9.2.1 腐蚀

9.2.2 膨胀

9.2.3 对偶性

9.3 开操作与闭操作

9.4 击中与击不中变换

9.5 一些基本的形态学算法

9.6 灰度级形态学

9.2.3 对偶性

- 膨胀和腐蚀彼此关于集合求补运算和反射运算是对偶的，即

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

和

$$(A \oplus B)^c = A^c \ominus \hat{B}$$

- B对A的腐蚀是$\hat{B}$对A^c的膨胀的补集，反之亦然。
- 证明：$(\hat{\hat{B}} = B)$
  - $(A \ominus B)^c = \{z|(B)_z \subseteq \mathrm{A}\}^c = \{z|(B)_z \cap A^c = \emptyset\}^c = \{z|(B)_z \cap A^c \neq \emptyset\} = A^c \oplus \hat{B}$
  - $(A \oplus B)^c = \left\{z\middle|(\hat{B})_z \cap A \neq \emptyset\right\}^c = \left\{z\middle|(\hat{B})_z \cap A = \emptyset\right\} = \left\{z\middle|(\hat{B})_z \subseteq A^c\right\} = A^c \ominus \hat{B}$

- 当结构元关于其原点对称时（**通常如此**），即$\hat{B} = B$，可以用相同的结构元简单地使用B膨胀图像的背景（即膨胀A^c），对该结果求补即可得到B对该幅图像的腐蚀。

第9章 形态学图像处理

9.1 预备知识

9.2 腐蚀和膨胀

9.3 开操作与闭操作

9.4 击中与击不中变换

9.5 一些基本的形态学算法

9.6 灰度级形态学

9.3 开操作与闭操作

- 如我们所见，膨胀会扩大一幅图像的组成部分，而腐蚀则会缩小一幅图像中的组成部分
- 结构元B对集合A的**开操作**，表示为$A \circ B$，定义如下：

$$A \circ B = (A \ominus B) \oplus \mathrm{B}$$

因此，B对A的开操作就是B对A的腐蚀，紧接着用B对结果进行膨胀

- 类似地，用结构元B对集合A的**闭操作**，表示为$\mathrm{A} \cdot \mathrm{B}$，定义如下：

$$\mathrm{A} \cdot \mathrm{B} = (\mathrm{A} \oplus \mathrm{B}) \ominus B$$

因此，B对集合A的闭操作就是用B对A膨胀，再用B对结果腐蚀

- **开操作**一般会平滑物体的轮廓、断开较窄的狭颈并消除较细的突出物。
- **闭操作**同样也会平滑轮廓的一部分，但与开操作相反，它通常会弥合较窄的间断和细长的沟壑，消除较小的孔洞，填补轮廓线中的断裂

9.3 开操作与闭操作：几何解释

开操作

- 假设我们把结构元B视为一个“转球”。
- 然后， $A \circ B$的边界由B中的点建立：当B在A的**边界内侧**滚动时，B所能到达的A的边界的最远点。
- 这种几何拟合特性可表达为一个集合论公式，即B对A的开操作是通过**拟合**到A的B的所有平移的并集得到的。

$$A \circ B = \bigcup \{(B)_z | (B)_z \subseteq A\}$$

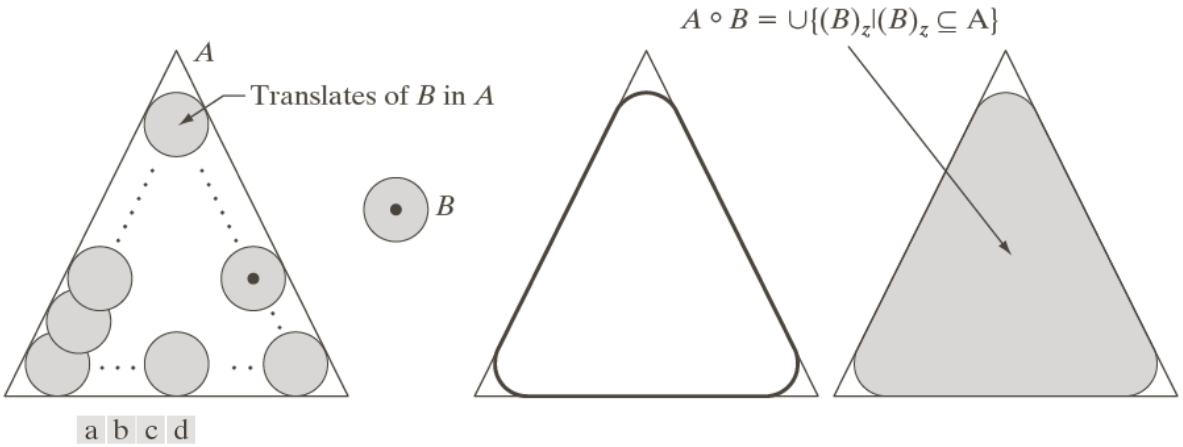
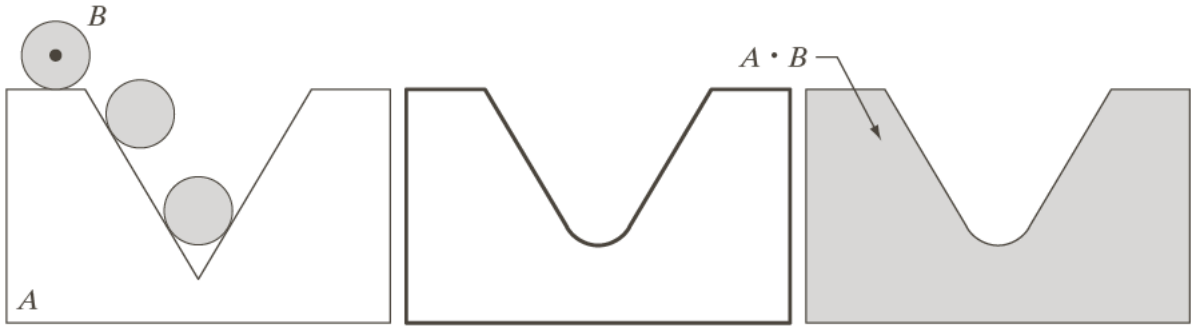


FIGURE 9.8 (a) Structuring element B “rolling” along the inner boundary of A (the dot indicates the origin of B). (b) Structuring element. (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded). We did not shade A in (a) for clarity.

9.3 开操作与闭操作：几何解释

- **闭操作**有类似的集合解释。
  - 开操作与闭操作彼此对偶，所以闭操作在**边界外侧**滚动球体是意料之中的事情。
  - B在A的**边界外侧**外切滚动一圈， 所有$(B)_z$的并集构成的内包络，即为$A \cdot B$。

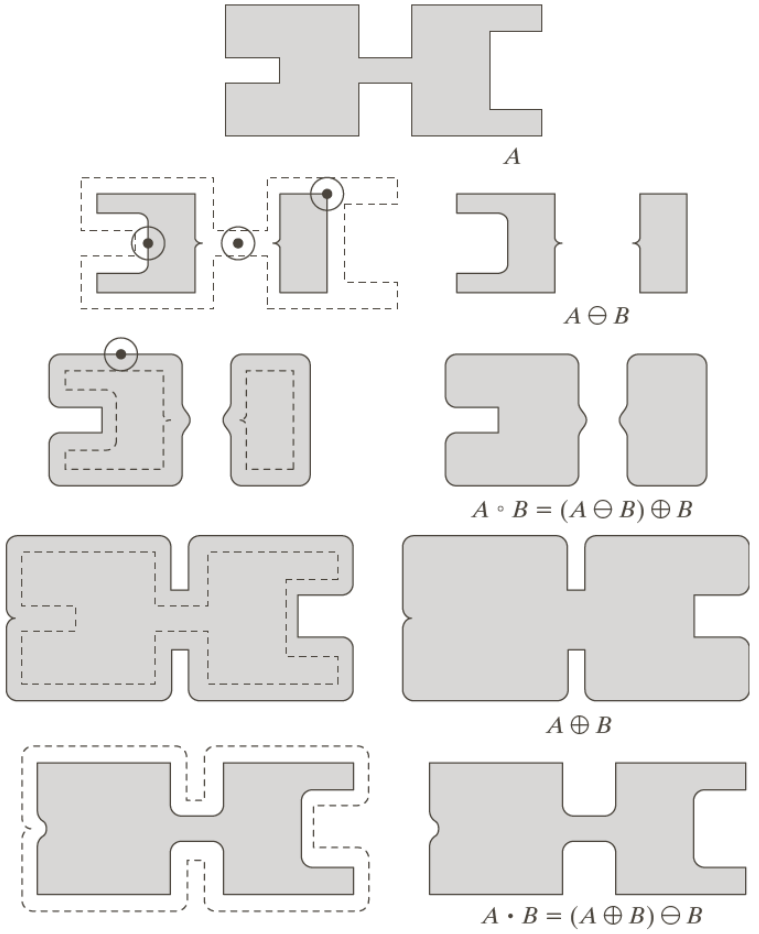


a b c

FIGURE 9.9 (a) Structuring element B "rolling" on the outer boundary of set A. (b) The heavy line is the outer boundary of the closing. (c) Complete closing (shaded). We did not shade A in (a) for clarity.

9.3 开操作与闭操作

- 开操作与闭操作的简单示例
  - 图c中两个主要部分之间的连接消失。连接部分的宽度与结构元的直径相比要细
  - 图e中方向向外的角变圆，而方向向内的角则未受影响
  - 图i中方向向内的角变圆，而方向向外的角则保持不变。在A的边界上，最左边的突入部分的尺寸明显减小，因为在这个位置上圆盘无法拟合。
  - 还要注意使用圆盘形结构元对集合A进行开操作和闭操作所得到的物体的各个部分都变得平滑了



a
b c
d e
f g
h i

FIGURE 9.10 Morphological opening and closing. The structuring element is the small circle shown in various positions in (b). The SE was not shaded here for clarity. The dark dot is the center of the structuring element.

9.3 开操作与闭操作

- 类似于膨胀和腐蚀，开操作和闭操作彼此关于集合求补和反射也是**对偶**的，即

$$(\mathrm{A}\cdot\mathrm{B})^c=(A^c\circ\hat{B})$$

和

$$(\mathrm{A}\circ\mathrm{B})^c=(A^c\cdot\hat{B})$$

- 开操作满足下列性质：
  - $\mathrm{A}\circ\mathrm{B}$是A的一个子集（子图像）
  - 如果C是D的一个子集，则$\mathrm{C}\circ\mathrm{B}$是$D\circ\mathrm{B}$的一个子集
  - $(\mathrm{A}\circ\mathrm{B})\circ\mathrm{B}=\mathrm{A}\circ\mathrm{B}$
- 类似地，闭操作满足下列性质：
  - A是$\mathrm{A}\cdot\mathrm{B}$的一个子集（子图像）
  - 如果C是D的一个子集，则$\mathrm{C}\cdot\mathrm{B}$是$D\cdot\mathrm{B}$的一个子集
  - $(\mathrm{A}\cdot\mathrm{B})\cdot\mathrm{B}=\mathrm{A}\cdot\mathrm{B}$
- **注意：算子应用一次后，一个集合的多次开操作或闭操作没有影响**

9.3 开操作与闭操作

- 针对形态学滤波使用开操作和闭操作



a b
d c
e f

FIGURE 9.11
(a) Noisy image.
(b) Structuring element.
(c) Eroded image.
(d) Opening of A.
(e) Dilation of the opening.
(f) Closing of the opening.
(Original image courtesy of the National Institute of Standards and Technology.)

第9章 形态学图像处理

9.1 预备知识

9.2 腐蚀和膨胀

9.3 开操作与闭操作

9.4 击中与击不中变换

9.5 一些基本的形态学算法

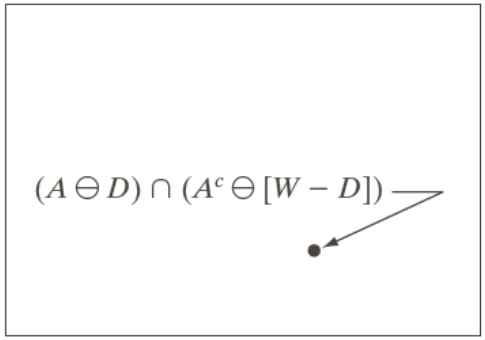
9.6 灰度级形态学

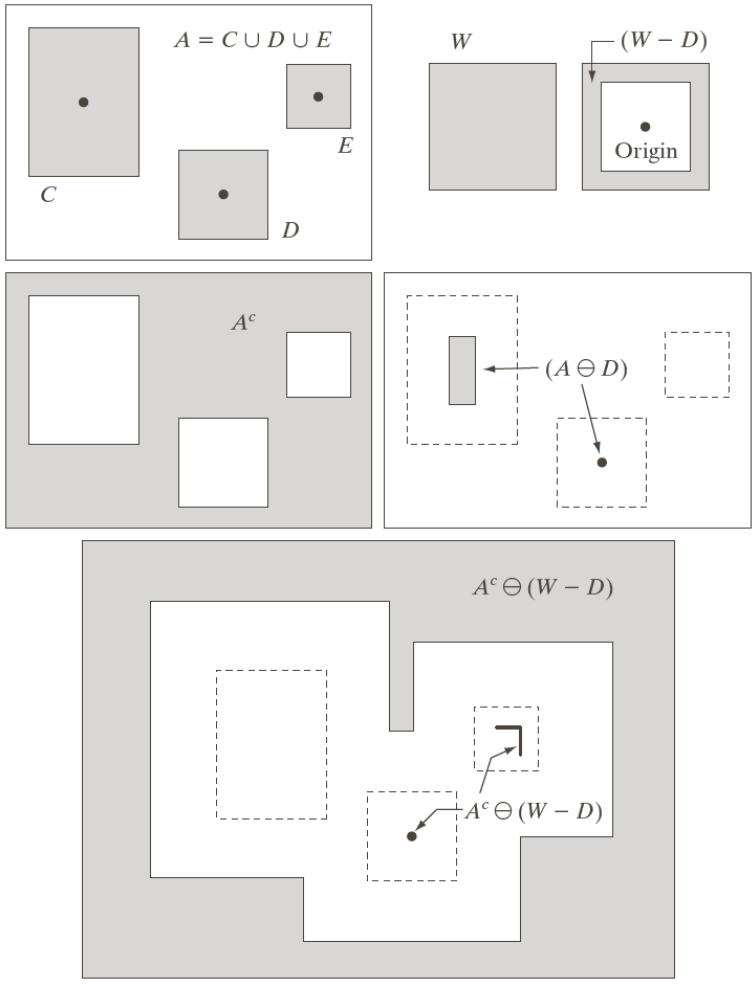
9.4 击中或击不中变换

- 形态学击中或击不中变换是形状检测的基本工具。

- 如果B表示由D及其背景组成的集合，则B在A中的匹配，表示为A ⊛ B，是

$$A \circledast B = (A \ominus D) \cap [A^c \ominus (W - D)]$$





a b
c d
e
f

FIGURE 9.12 (a) Set A. (b) A window, W, and the local background of D with respect to W, $(W - D)$. (c) Complement of A. (d) Erosion of A by D. (e) Erosion of A^c by $(W - D)$. (f) Intersection of (d) and (e), showing the location of the origin of D, as desired. The dots indicate the origins of C, D, and E.

9.4 击中或击不中变换

- 令$B=(B_1,B_2)$，其中B_1是由与一个目标相联系的B的元素构成的集合，B_2是由与相应背景相联系的B的元素构成的集合。用这种表示方法，击中或击不中变换为：

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

- 使用腐蚀与膨胀的对偶关系，可将上式写为

$$A \circledast B = (A \ominus B_1) \cap (A \oplus \widehat{B_2})^c$$

- 使用与物体有关的结构元B_1和与背景有关的结构元B_2的原因，基于一个假设的定义——仅当两个或多个物体形成相脱离的集合时，这些物体才是可分的。

- 在某些应用中，我们可能对检测某个集合内由1和0组成的某些模式感兴趣，在这种情况下不需要背景。因此，击中或击不中就简化为简单的腐蚀。

第9章　形态学图像处理

9.1　预备知识

9.2　腐蚀和膨胀

9.3　开操作与闭操作

9.4　击中与击不中变换

9.5　一些基本的形态学算法

9.6　灰度级形态学

9.5 一些基本的形态学算法

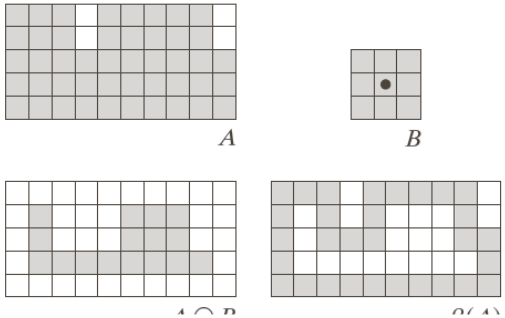
- 9.5.1 边界提取
- 9.5.2 孔洞填充
- 9.5.3 连通分量的提取
- 9.5.4 凸壳
- 9.5.5 细化
- 9.5.6 粗化
- 9.5.7 骨架
- 9.5.8 裁剪
- 9.5.9 形态学重建

9.5.1 边界提取

- 表示为β(A)的集合A的边界可以通过先用B对A腐蚀，而后执行A和腐蚀的结果之间的差集得到，即

$$\beta(A) = A - (A \ominus B)$$

- 下图说明了边界提取的机理。这幅图像显示了一个简单的二值物体、一个结构元B和使用上式得到的结果



a b
c d

FIGURE 9.13 (a) Set A. (b) Structuring element B. (c) A eroded by B. (d) Boundary, given by the set difference between A and its erosion.



a b

FIGURE 9.14 (a) A simple binary image, with 1s represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

9.5.2 孔洞填充

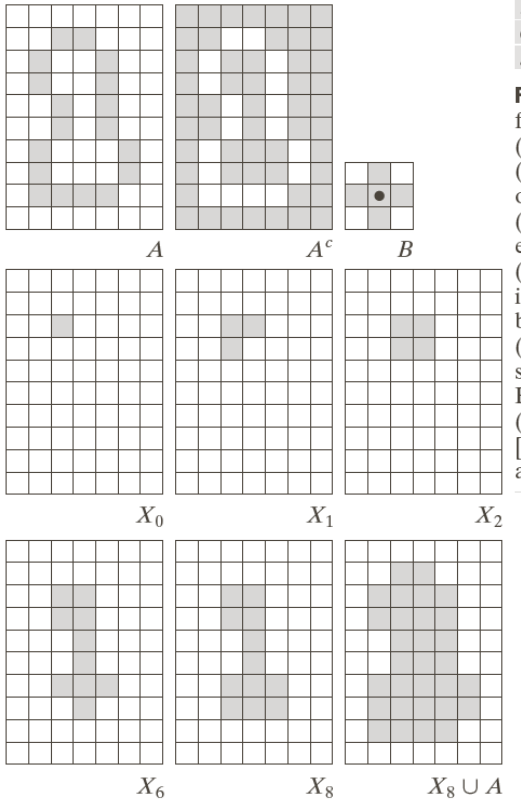
- 孔洞定义:由前景像素相连接的边界所**包围**的背景区域。
- 令A表示一个集合，其元素是8连通的边界，每个边界包围一个背景区域。**当给定每个孔洞中的一个点后**，目的就是用1填充所有的孔洞。
- 给定每个孔洞中对应于X_0中的位置点（这一点已经置为1），我们从形成一个由0组成的阵列X_0开始。然后，如下过程将用1填充所有孔洞：

$$X_k = (X_{k-1} \oplus \mathrm{B} \cap A^c) \quad k = 1,2,3,\cdots$$

- 如果$X_k = X_{k-1}$，则算法在迭代的第k步结束。然后，集合X_k包含所有被填充的孔洞。X_k和A的并集包含所有填充的孔洞及这些孔洞的边界。
- 如果左边不加限制，那么膨胀将填充整个区域。然而，每一步中与A^c的交集操作将把结果限制到感兴趣区域内，称为**条件膨胀**。

9.5.2 孔洞填充

- 下图说明了孔洞填充的机理



a b c
d e f
g h i

FIGURE 9.15 Hole filling. (a) Set A (shown shaded). (b) Complement of A. (c) Structuring element B. (d) Initial point inside the boundary. (e)–(h) Various steps of Eq. (9.5-2). (i) Final result [union of (a) and (h)].

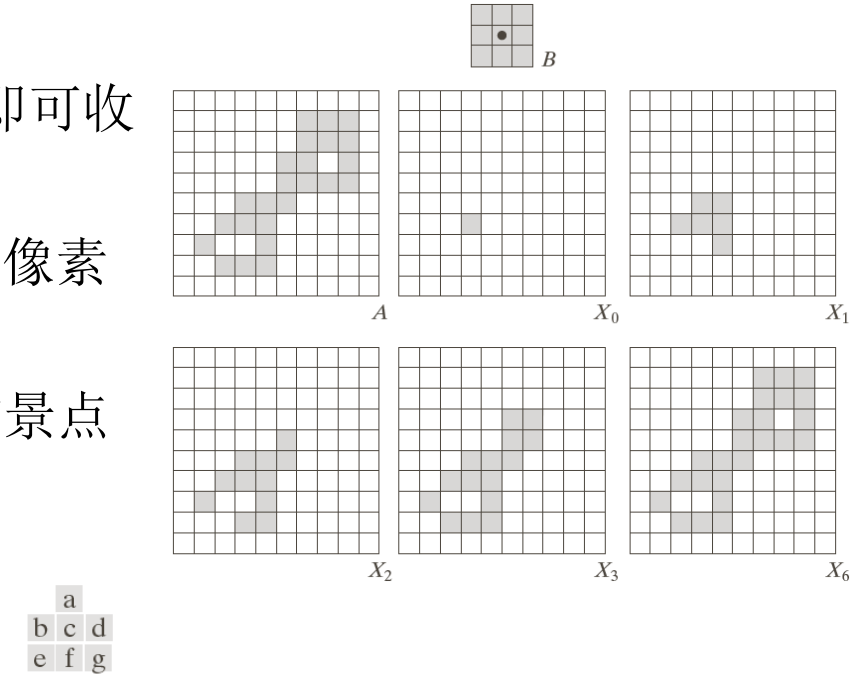


a b c

FIGURE 9.16 (a) Binary image (the white dot inside one of the regions is the starting point for the hole-filling algorithm). (b) Result of filling that region. (c) Result of filling all holes.

9.5.3 连通分量的提取

- 令A是包含一个或多个连通分量的集合；形成一个阵列X_0，除了在对应于A中每个连通分量的一个点的每个已知位置处我们已置为1外，该阵列的所有其他元素均为0。如下迭代过程可完成这一目的：$X_k = (X_{k-1} \oplus \mathrm{B} \cap \mathrm{A}) \quad k = 1,2,3,\cdots$
- 当$X_k = X_{k-1}$时，迭代过程结束，X_k包含输入图像中的所有连通分量。
- 右图说明了此机理，k=6时即可收敛。
- 注意，所用结构元的形状在像素间是基于8连通的
- 本质也是**条件膨胀**，寻找前景点



a
b c d
e f g

FIGURE 9.17 Extracting connected components. (a) Structuring element. (b) Array containing a set with one connected component. (c) Initial array containing a 1 in the region of the connected component. (d)–(g) Various steps in the iteration of Eq. (9.5-3).

9.5.3 连通分量的提取

- 例：使用连通分量检测包装食品中的外来物
  - 这幅图中的最显著的特征是保留下来的点聚集为物体（骨头），而不是彼此孤立的毫无关系的点。我们通过**提取图像中的连通分量**来标记这些物体。
  - 15个连通分量有4个尺寸较大，这足以确定包含在原图像中的重要但不希望的物体



Connected component	No. of pixels in connected comp
01	11
02	9
03	9
04	39
05	133
06	1
07	1
08	743
09	7
10	11
11	11
12	9
13	9
14	674
15	85

a
b
c d

FIGURE 9.18
(a) X-ray image of chicken filet with bone fragments. (b) Thresholded image. (c) Image eroded with a 5×5 structuring element of 1s. (d) Number of pixels in the connected components of (c). (Image courtesy of NTB Elektronische Geraete GmbH, Diepholz, Germany, www.ntbxray.com.)

9.5.4 凸壳

- 如果在集合A内连接任意两个点的直线段都在A的内部，则称集合A是凸形的。
- 任意集合S的凸壳H是包含S的最小凸集。差集H-S称为S的凸缺。下面介绍一种获得集合A的凸壳$C(A)$的算法
- 令$B^i, i = 1,2,3,4$表示下图中的4个结构元。这个过程可通过执行下式实现：

$$X_k^i = \left(X_{k-1} \circledast B^i\right) \cup A \quad i = 1,2,3,4 \text{ 和 } k = 1,2,3,\cdots$$

式中，$X_0^i = A$。当$X_k^i = X_{k-1}^i$时，我们令$D^i = X_k^i$。则A的凸壳为

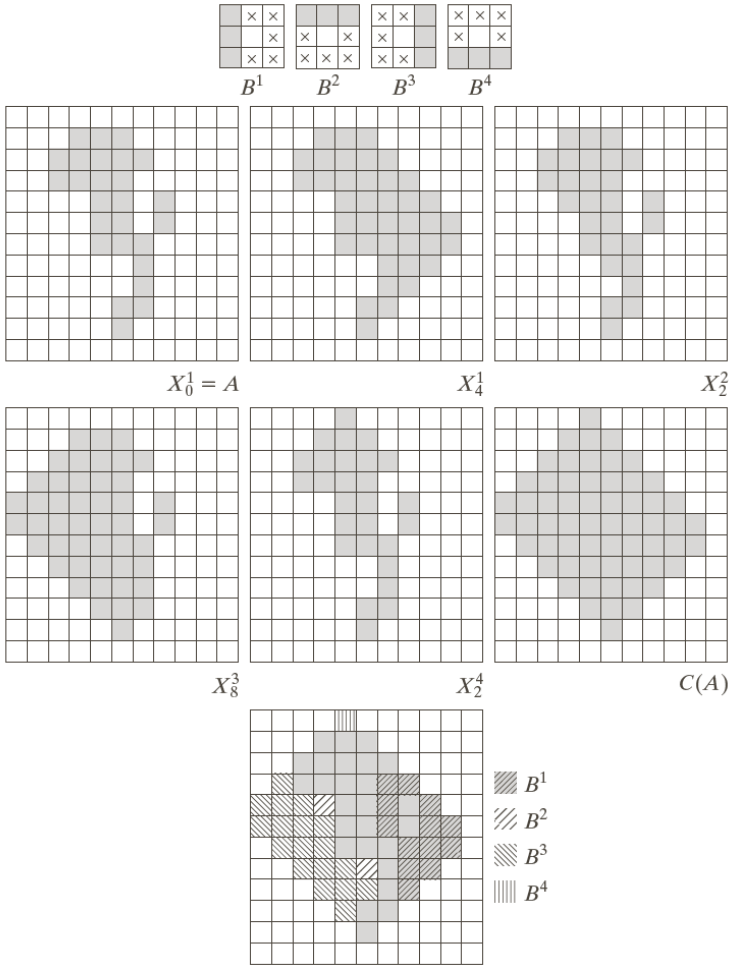
$$C(A) = \bigcup_{i=1}^{4} D^i$$

  - 由反复使用B^1对A做击中或击不中变换组成；不再发生变化时，执行与A的并集运算，结果为D^1。这一过程使用B^2重复（应用于A），直到不发生变化，如此往复。得到的4个D的并集组成了A的凸壳。

- 注意，我们使用**不需要背景匹配**的**击中或击不中**变换的简化实现

9.5.4 凸壳

- 右图说明了凸壳的过程
- 该过程的缺点是，凸壳可能超过确保凸性所需最小尺寸。
- 减少这种影响的一种简单方法是限制生长，以便凸壳不会超过初始点集在水平和垂直方向上的尺寸，如下图：



a
b c d
e f g
h

FIGURE 9.19
(a) Structuring elements. (b) Set A. (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull. (h) Convex hull showing the contribution of each structuring element.

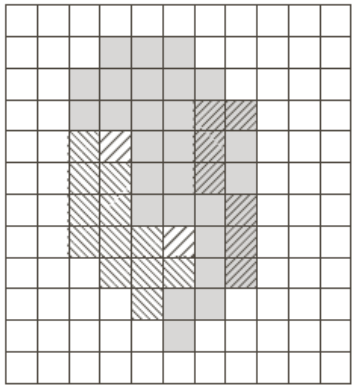


FIGURE 9.20
Result of limiting growth of the convex hull algorithm to the maximum dimensions of the original set of points along the vertical and horizontal directions.

9.5.5 细化

- 结构元B对集合A的细化可表示为$A \otimes B$，它可以根据击中或击不中变换来定义：

$$A \otimes B = A - (A \circledast B)$$

- 同前一节一样，**我们仅对与结构元的模式匹配感兴趣，所以在击中或击不中变换中没有背景运算**。
- 对称地细化A的一种更有用的表达方式是以结构元序列为基础的：

$$\{B\} = \{B^1, B^2, B^3, \cdots, B^n\}$$

式中B^i是B^{i-1}旋转后的形式。

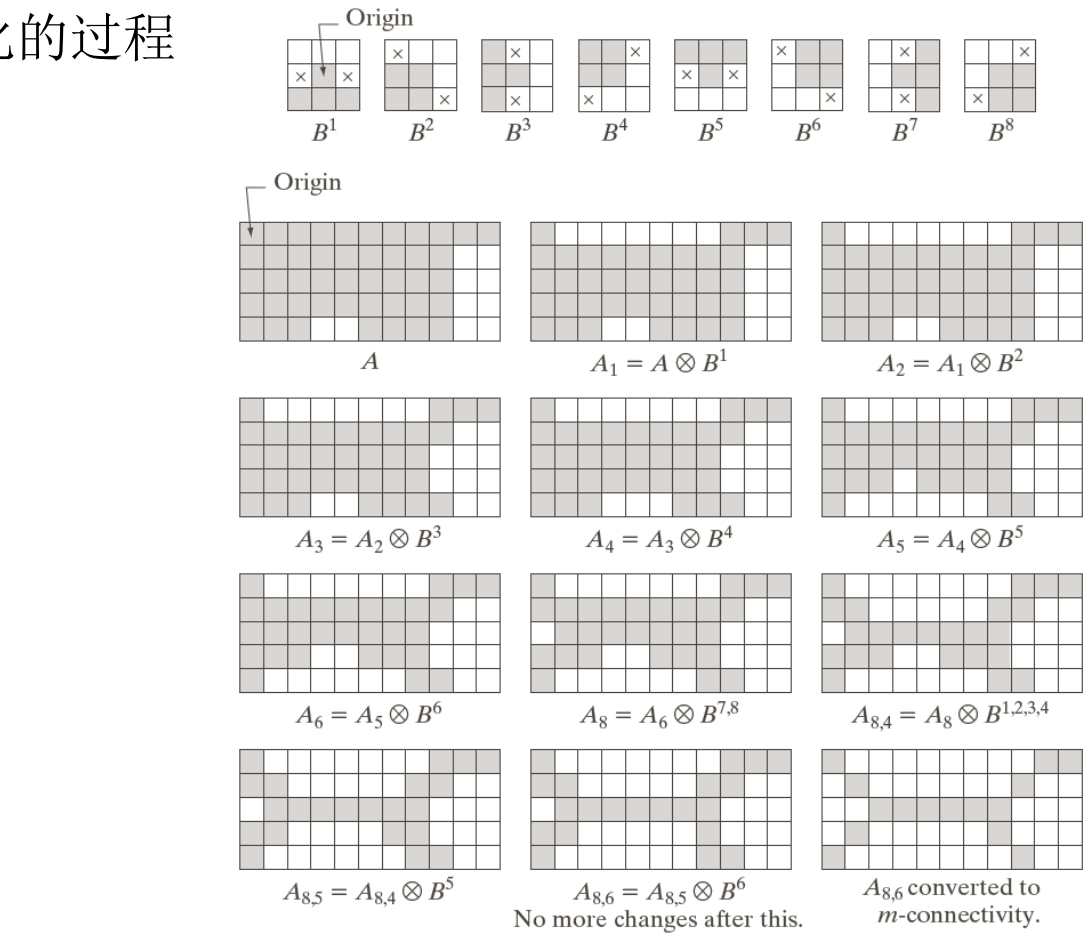
- 使用一个结构元序列将细化定义为

$$A \otimes \{B\} = ((\cdots((A \otimes B^1) \otimes B^2)\cdots) \otimes B^n)$$

这种处理是A被B^1细化一次，然后，得到的结果被B^2细化一次，如此进行下去，直到A被B^n细化一次。整个过程不断重复，直到得到的结果不再发生变化

9.5.5 细化

- 下图说明了细化的过程



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

FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first four elements again. (l) Result after convergence. (m) Conversion to m-connectivity.

9.5.6 粗化

- 粗化是细化的形态学对偶，定义如下：

$$A \odot B = A \cup (A \circledast B)$$

式中B是结构元，与细化一样，粗化处理也可以定义为一系列操作：

$$A \odot \{B\} = ((\cdots((A \odot B^1) \odot B^2)\cdots) \odot B^n)$$

- **然而，针对粗化的分离算法在实际中很少用到，取而代之的过程是先对问题中集合的背景进行细化，而后对结果求补集。**
- 换句话说，为粗化集合A，我们先形成$C = A^c$，而后细化C，然后再求C^c
- 下图说明了这一过程
  - 由于依赖于A的性质，这个过程可能会产生某些断点。因此，通过这种方法的粗化处理通常会跟随一个消除断点的后处理

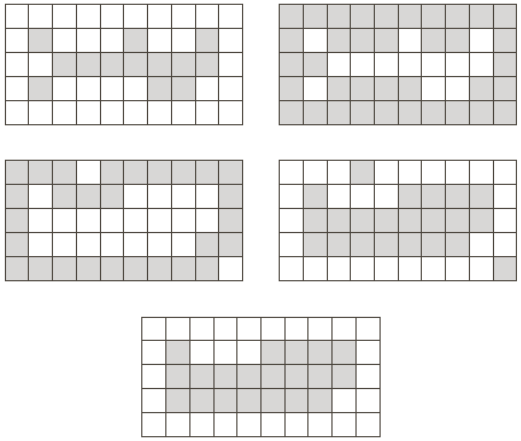
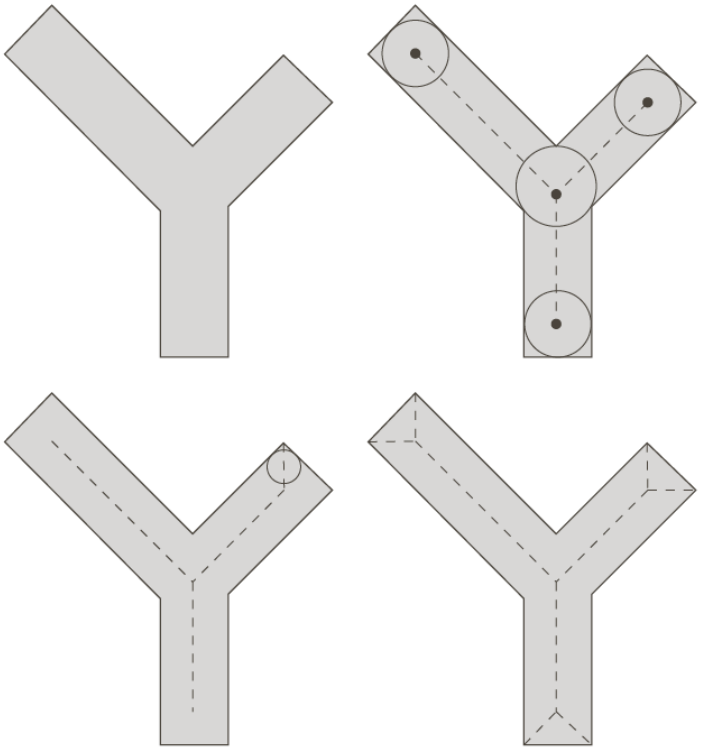


FIGURE 9.22 (a) Set *A*. (b) Complement of *A*. (c) Result of thinning the complement of *A*. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

9.5.7 骨架

- 如图所示，集合A的骨架 S(A) 的概念很简单。由该图我们可以推出：
  - 如果z是S(A)的一个点，并且$(D)_z$是A内以z为中心的最大圆盘，则不存在包含$(D)_z$且位于A内的更大圆盘（不必以z为中心）。圆盘$(D)_z$称为最大圆盘
  - 圆盘$(D)_z$在两个或多个不同的位置与A的边界接触



a b
c d

FIGURE 9.23
(a) Set *A*.
(b) Various positions of maximum disks with centers on the skeleton of *A*.
(c) Another maximum disk on a different segment of the skeleton of *A*.
(d) Complete skeleton.

9.5.7 骨架

- A的骨架可以用腐蚀和开操作来表达，即骨架可以表示为

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

其中，$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$

式中，B是一个结构元，而$A \ominus kB$表示对A的连续k次腐蚀：

$$A \ominus kB = ((\cdots((A \ominus B) \ominus B) \ominus \cdots) \ominus B)$$

K是A被腐蚀为空集前的最后一次迭代步骤。换句话说，

$$K = \max\{k | (A \ominus kB) \neq \emptyset\}$$

- S(A)可以作为骨架子集$S_k(A)$的并集来得到，此外，还可证明使用下式可由这些子集来重建A：

$$A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB)$$

式中$(S_k(A) \oplus kB)$表示对$S_k(A)$的k次连续膨胀，即

$$(S_k(A) \oplus kB) = ((\cdots((S_k(A) \oplus B) \oplus B) \oplus \cdots) \oplus B)$$

9.5.7 骨架

- 计算简单图形的骨架

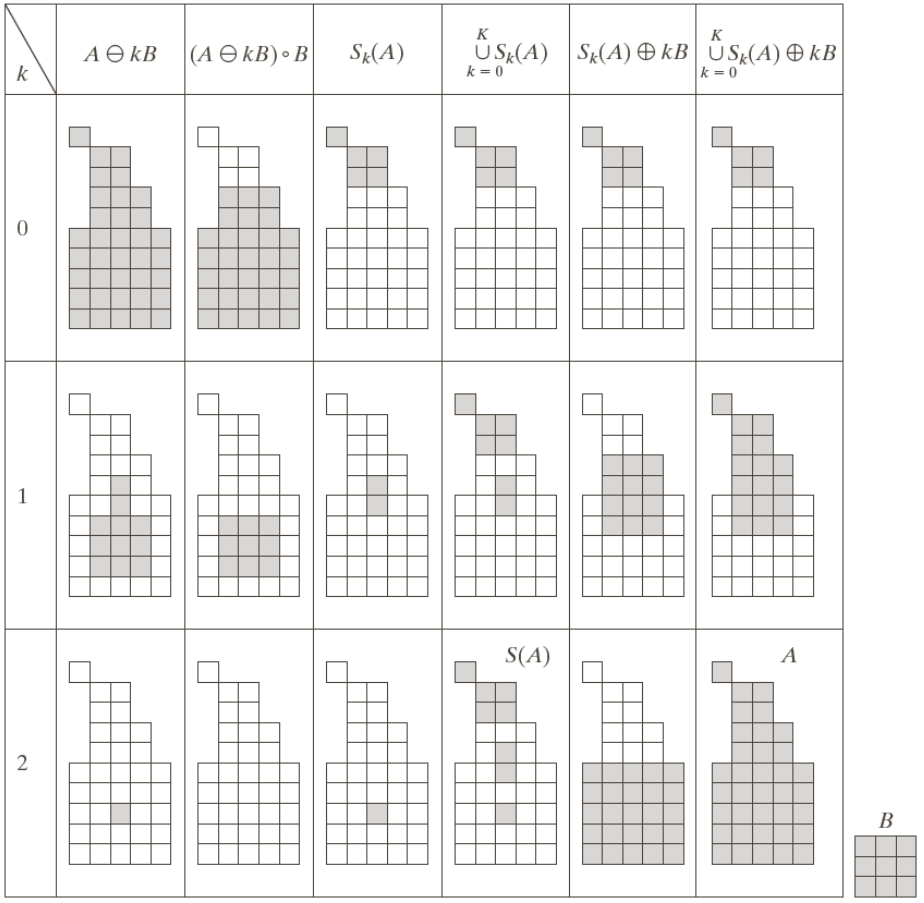
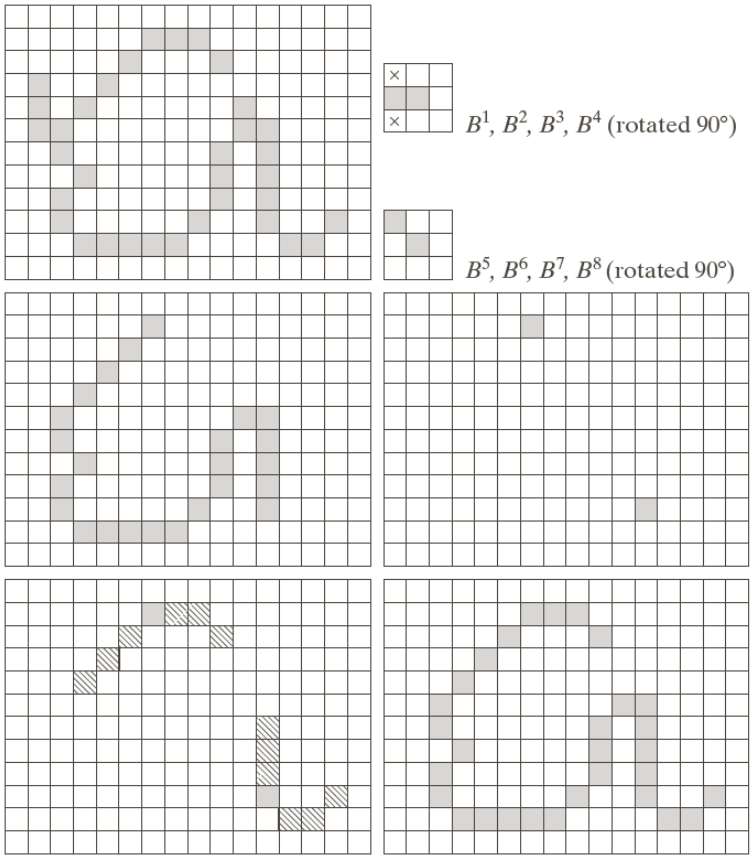


FIGURE 9.24 Implementation of Eqs. (9.5-11) through (9.5-15). The original set is at the top left, and its morphological skeleton is at the bottom of the fourth column. The reconstructed set is at the bottom of the sixth column.

9.5.8 裁剪

- 裁剪方法本质上是对细化和骨架算法的补充，因为这些过程会保留某些**寄生成分**，因而需要用后处理来清除这些寄生成分
- 在手写字符的自动识别中，通常使用的方法是分析每个字符的骨架形状。这些骨架通常带有许多“毛刺”（寄生成分）。



a b
c
d e
f g

FIGURE 9.25
(a) Original image. (b) and (c) Structuring elements used for deleting end points. (d) Result of three cycles of thinning. (e) End points of (d). (f) Dilation of end points conditioned on (a). (g) Pruned image.

9.5.8 裁剪

- 上图显示了手写字符“a”的骨架。该字符最左侧的寄生成分就是我们要消除的部分。使用一系列仅设计用来**检测端点**的结构元，对输入集合A进行**细化**可以得到期望的结果。令

$$X_1 = \mathrm{A} \otimes \{B\}$$

式中，$\{B\}$表示结构元序列。连续对A应用三次，得到集合X_1。

- 下一步是将字符“复原”为原来的形状，但要去掉寄生分支。首先需要求形成一个包含X_1中所有端点的集合X_2：

$$X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k)$$

式中，B^k是相同端点检测子。

- 下一步是用集合A作为限定器，对端点进行三次膨胀（**条件膨胀**）：

$$X_3 = (X_2 \oplus \mathrm{H}) \cap \mathrm{A}$$

最后，X_3和X_1的并集就是我们想要的结果：

$$X_4 = X_1 \cup X_3$$

9.5.9 形态学重建

- 形态学重建涉及两幅图像和一个结构元。
  - 一幅图像是标记，它包含变换的起始点
  - 另一幅图像是模板，它用来约束该变换。
  - 结构元用来定义连续性
- 测地膨胀和腐蚀
  - 形态学重建的核心是测地膨胀和测地腐蚀这两个概念。
- 用腐蚀和膨胀的形态学重建
- 应用实例
  - 重建开操作
  - 填充孔洞
  - 边界清除

9.5.9 形态学重建

- 测地膨胀和腐蚀
  - 令F表示标记图像，G表示模板图像。假定两幅都是二值图像，且**F ⊆ G**。令$D_G^{(1)}(F)$表示大小为1的标记图像关于模板的**测地膨胀**，定义为

$$D_G^{(1)}(F) = (F \oplus \mathrm{B}) \cap \mathrm{G}$$

F关于G的大小为n的测地膨胀定义为

$$D_G^{(n)}(F) = D_G^{(1)}\left[D_G^{(n-1)}(F)\right]$$

式中，$D_G^{(0)}(F) = F$。在这个递推表达式中，集合求交在每一步中都执行。注意，交集算子保证模板G将限制标记F的生长（膨胀）。

  - 下图显示了一个大小为1的测地膨胀的简单例子。

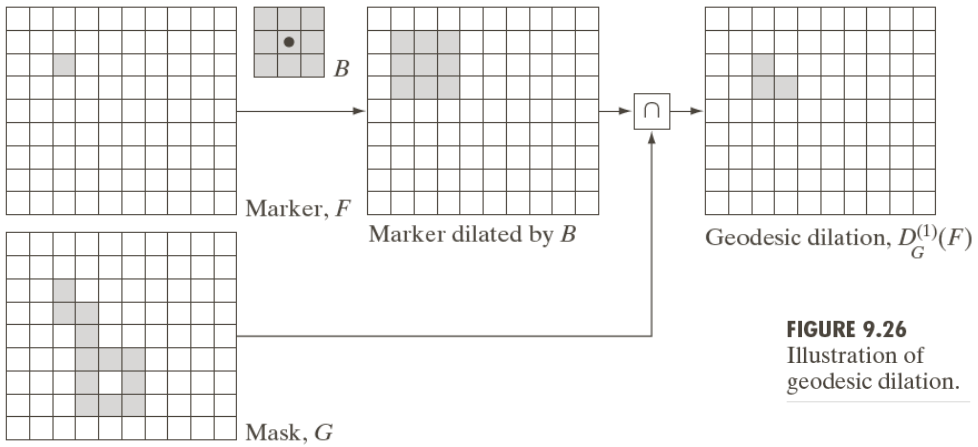


FIGURE 9.26 Illustration of geodesic dilation.

9.5.9 形态学重建

- 测地膨胀和腐蚀
  - 类似地，标记F关于模板G的大小为1的**测地腐蚀**定义为

$$E_G^{(1)}(F) = (F \ominus \mathrm{B})\cup \mathrm{G}$$

F关于G的大小为n的测地腐蚀定义为

$$E_G^{(n)}(F) = E_G^{(1)}\left[E_G^{(n-1)}(F)\right]$$

式中，$E_G^{(0)}(F) = F$。**测地膨胀和测地腐蚀关于集合的补集对偶。**

  - 下图显示了一个大小为1的测地腐蚀的简单例子

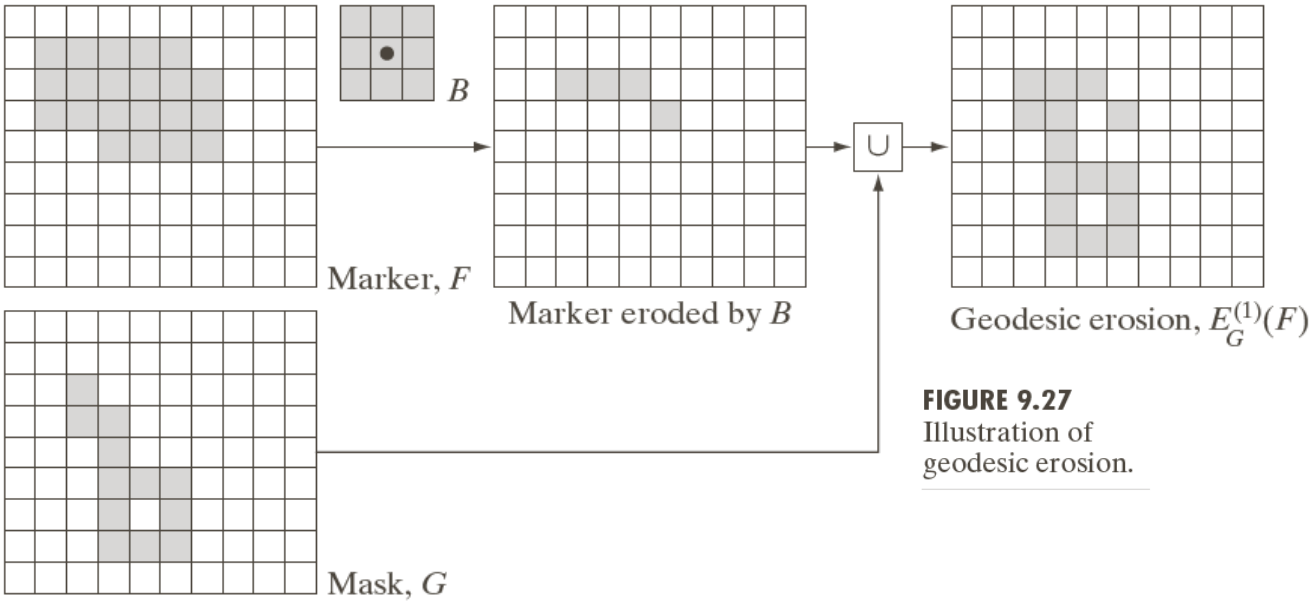


FIGURE 9.27 Illustration of geodesic erosion.

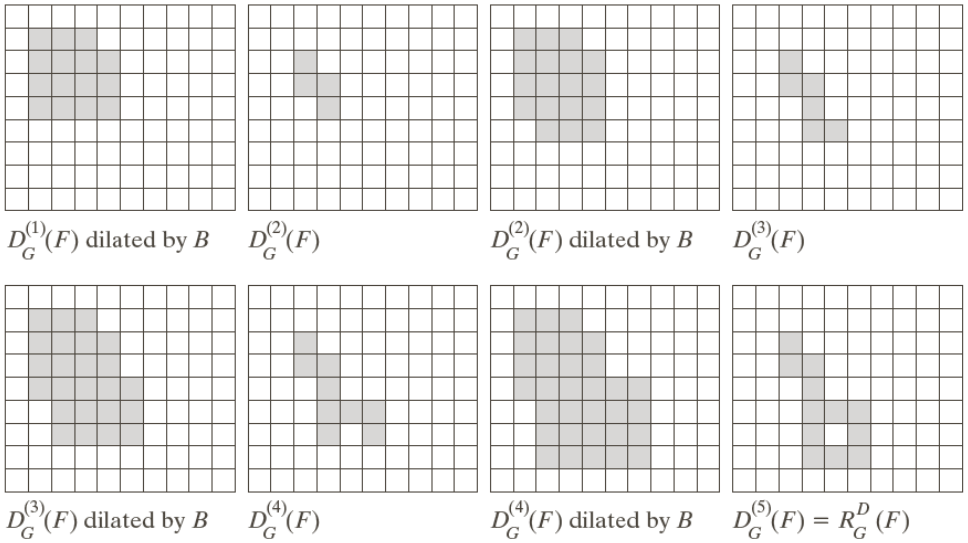
9.5.9 形态学重建

- 用腐蚀和膨胀的形态学重建
  - 来自标记图像F对模板图像G的**膨胀形态学重建**表示为$R_G^D(F)$，它被定义为F关于G的测地膨胀，**反复迭代直至达到稳定状态**；即

$$R_G^D(F) = D_G^{(k)}(F)$$

迭代k次，直至$D_G^{(k)}(F) = D_G^{(k+1)}(F)$。

  - 下图说明了使用膨胀的重建。得到$D_G^{(1)}(F)$后，重建的下一步是膨胀该结果，然后用模板G与其相“与”得到$D_G^{(2)}(F)$，$D_G^{(2)}(F)$的膨胀结果与目标G相“与”得到$D_G^{(3)}(F)$，等等。重复这一过程，**直至达到稳定**



a b c d
e f g h

FIGURE 9.28 Illustration of morphological reconstruction by dilation. F, G, B and $D_G^{(1)}(F)$ are from Fig. 9.26.

9.5.9 形态学重建

- 用腐蚀和膨胀的形态学重建
  - 依照类似的方法，模板图像G对标记图像F的**腐蚀形态学重建**表示为$R_G^E(F)$，它被定义为F关于G的测地腐蚀，反复迭代直至达到稳定状态；即

$$R_G^E(F) = E_G^{(k)}(F)$$

    迭代k次，直至$E_G^{(k)}(F) = E_G^{(k+1)}(F)$。
  - 膨胀和腐蚀形态学重建关于集合补集对偶

9.5.9 形态学重建

- 应用实例
  - **重建开**操作：形态学开操作中，腐蚀会删除小的物体，而后续的膨胀会试图恢复遗留物体的形状。然而这种**恢复的准确性高度依赖于物体的形状和所用结构元的相似性**。
  - 重建开操作可正确地恢复腐蚀后所保留物体的形状。图像F的大小为n的重建开操作，定义为来自F的大小为n的腐蚀的F的膨胀重建； 即

$$O_R^{(n)}(F) = R_F^D[(F \ominus nB)]$$

式中，$(F \ominus nB)$表示B对F的n次腐蚀。

  - 对于**重建闭**操作，可写出类似的式子

右图显示了重建开操作的一个例子，我们的兴趣是提取垂直笔画的长字符

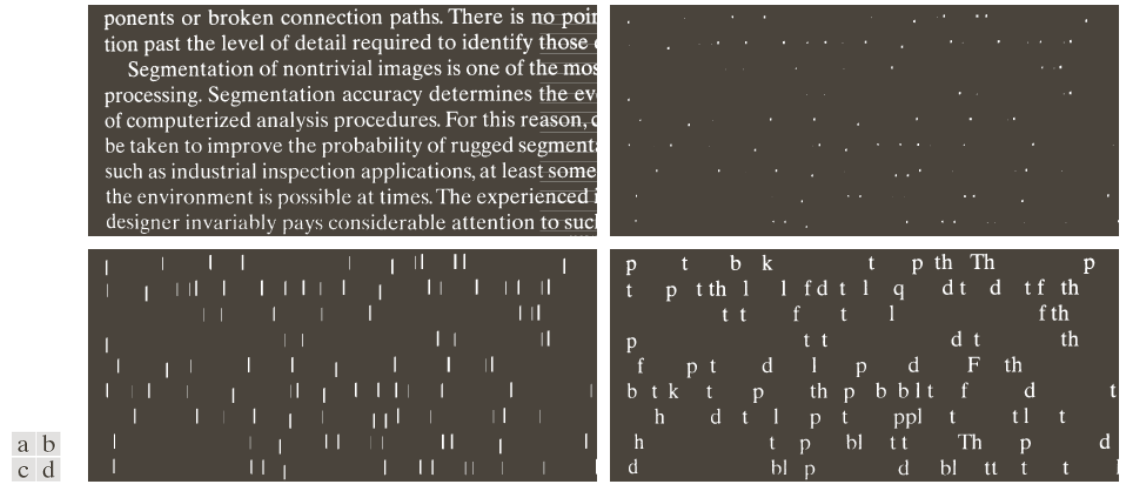


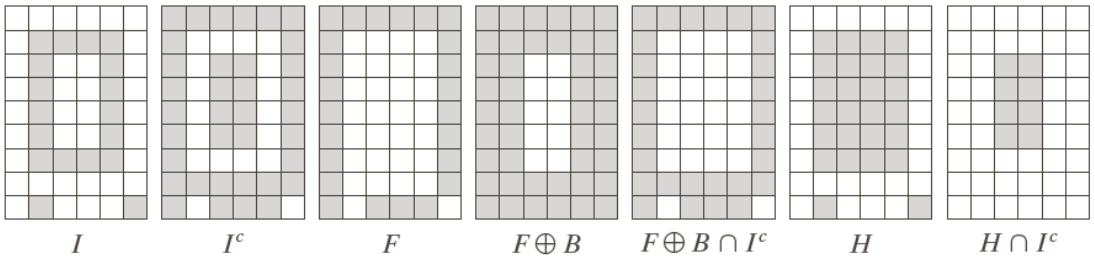
FIGURE 9.29 (a) Text image of size 918 × 2018 pixels. The approximate average height of the tall characters is 50 pixels. (b) Erosion of (a) with a structuring element of size 51 × 1 pixels. (c) Opening of (a) with the same structuring element, shown for reference. (d) Result of opening by reconstruction.

9.5.9 形态学重建

- 应用实例
  - 填充孔洞：一个基于形态学重建的全自动化过程。令 I(x, y) 代表一幅二值图像，并假定我们形成了一幅**标记图像F**，除了在该图像的边界位置为1−I之外，在其他位置均为0，即

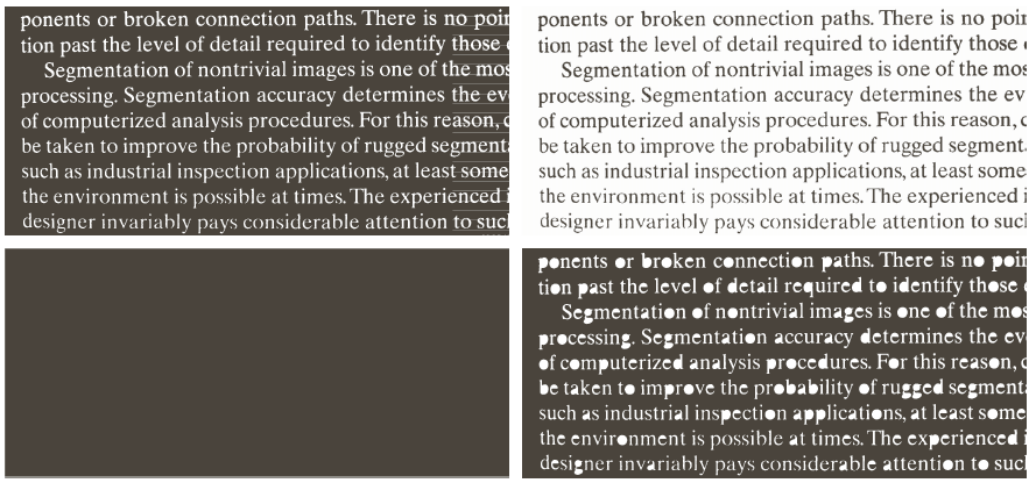
$$F(x,y)=\begin{cases}1-I(x,y), & (x,y)\text{在}I\text{的边界上}\\ 0, & \text{其他}\end{cases}$$

则 $H=[R_{I^c}^{D}(F)]^c$ 是一幅等于I且所有孔洞都被填充的二值图像



a b c d e f g

FIGURE 9.30 Illustration of hole filling on a simple image.



a b c d

FIGURE 9.31 (a) Text image of size 918 × 2018 pixels. (b) Complement of (a) for use as a mask image. (c) Marker image. (d) Result of hole-filling using Eq. (9.5-29).

9.5.9 形态学重建

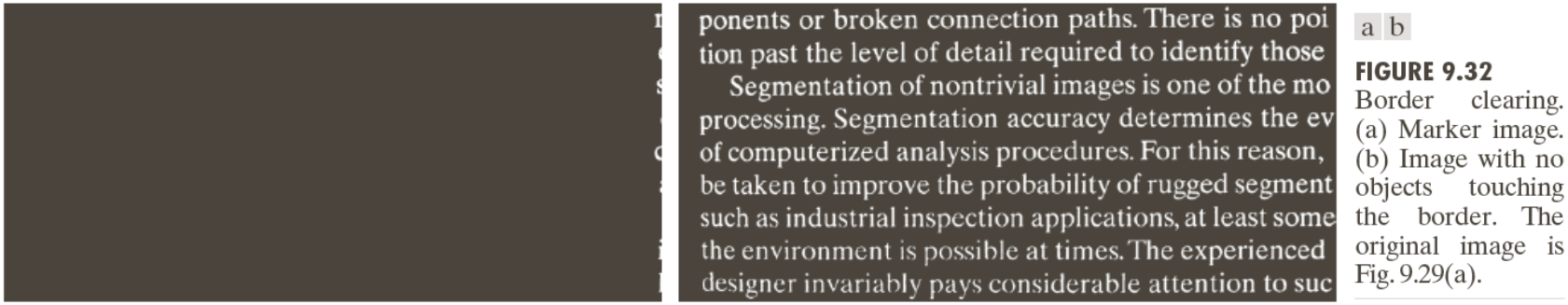
- 应用实例
  - 边界清除：针对后续形状分析的图像物体提取是自动图像处理的基本任务。删除接触边界的物体的算法是一个很有用的工具，因为：

    （1）它可以用于屏蔽图像，以便对进一步处理只保留完整的物体，

    （2）它可用作在视野中存在部分物体的一个信号。
  - 基于形态学重建的边界清除：用原图像作为模板G，并使用下面的标记图像：

$$F(x,y)=\begin{cases} I(x,y), & (x,y)\text{在}I\text{的边界上} \\ 0, & \text{其他} \end{cases}$$

边界清除算法首先计算形态学重建$R_I^D(F)$，然后计算差$X=I-R_I^D(F)$以得到一幅其中没有接触边界的物体的图像X。



a b

FIGURE 9.32 Border clearing. (a) Marker image. (b) Image with no objects touching the border. The original image is Fig. 9.29(a).

第9章　形态学图像处理

9.1　预备知识

9.2　腐蚀和膨胀

9.3　开操作与闭操作

9.4　击中与击不中变换

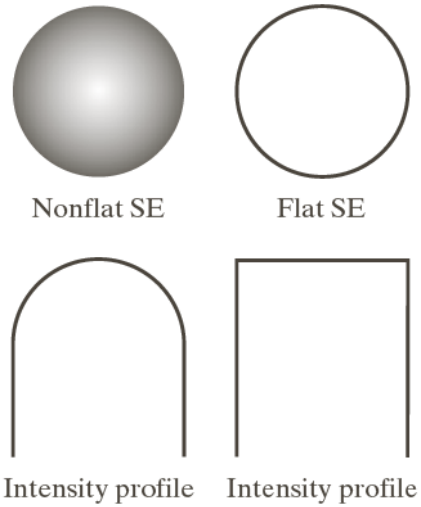
9.5　一些基本的形态学算法

9.6　灰度级形态学

9.6 灰度级形态学

- 将把膨胀、腐蚀、开/闭操作扩展到灰度级图像
  - 处理形如$f(x,y)$和$b(x,y)$的数字图像，其中$f(x,y)$是一幅灰度级图像，而$b(x,y)$是一个结构元。
  - f和b是对每个坐标赋以灰度值（来自实数集合R的一个实数）的函数
- 灰度级形态学中的结构元作为一个“探测器”以明确的特性检验一幅给定的图像。
  - 结构元分为两类：**非平坦**的和**平坦的**。
  - 实际中灰度级结构元并不常用。
- **本节中所有例子都以高度为1、对称的、平坦的结构元为基础，其原点位于中心处。**
- 结构元的映像的定义为

$$\hat{b}(x,y) = b(-x,-y)$$



a b
c d

FIGURE 9.34 Nonflat and flat structuring elements, and corresponding horizontal intensity profiles through their center. All examples in this section are based on flat SEs.

9.6.1 腐蚀和膨胀

- 当b的原点位于(x,y)处时，用一个平坦的结构元b在(x,y)处对图像f的**腐蚀**定义为图像f中与b重合区域的最小值。公式为：

$$[f \ominus \mathrm{b}](\mathrm{x},\mathrm{y}) = \min_{(s,t)\in b}\{f(x+s,y+t)\}$$

- 为寻求b对f的腐蚀，我们把结构元的原点放在图像的每个像素的位置。在任何位置的腐蚀由从包含在与b重合区域中的f的所有值中选取的最小值决定。
- 类似地，当$\hat{b}$的原点位于位置(x,y)处时，平坦结构元b在任何位置(x,y)处对图像f的**膨胀**定义为图像f中与$\hat{b}$重合区域的最大值即

$$[f \oplus \mathrm{b}](\mathrm{x},\mathrm{y}) = \max_{(s,t)\in b}\{f(x-s,y-t)\}$$

- 非平坦结构元具有**随定义域变化的灰度级**， b_N 对图像 f 的腐蚀定义：

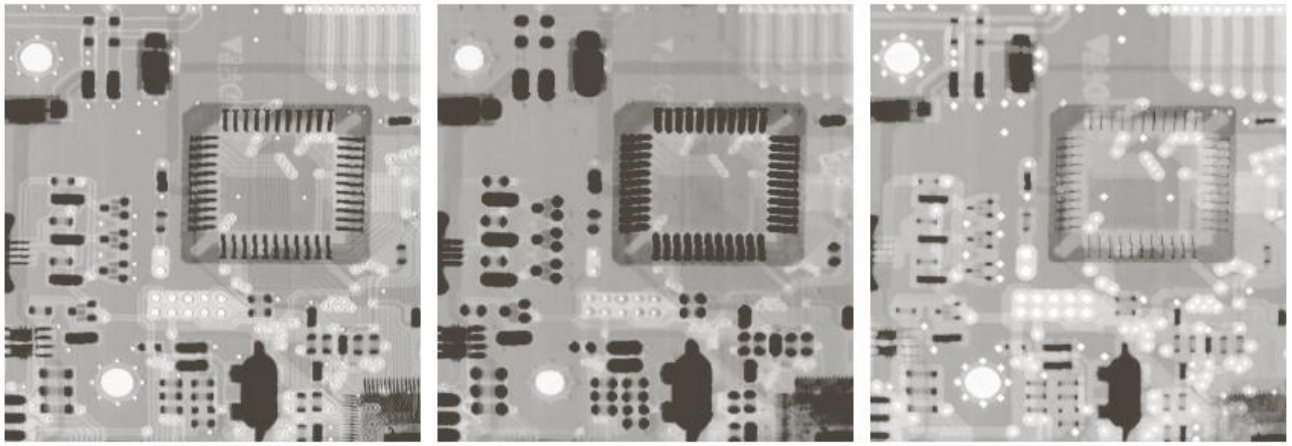
$$[f \ominus b_N](\mathrm{x},\mathrm{y}) = \min_{(s,t)\in b_N}\{f(x+s,y+t) - b_N(s,t)\}$$

- 类似地，使用非平坦结构元的膨胀的定义如下：

$$[f \oplus b_N](\mathrm{x},\mathrm{y}) = \max_{(s,t)\in b_N}\{f(x-s,y-t) + b_N(s,t)\}$$

9.6.1 腐蚀和膨胀

- 灰度级腐蚀和膨胀的说明



a b c

FIGURE 9.35 (a) A gray-scale X-ray image of size 448 × 425 pixels. (b) Erosion using a flat disk SE with a radius of two pixels. (c) Dilation using the same SE. (Original image courtesy of Lixi, Inc.)

- 如二值情况一样，腐蚀和膨胀是关于函数的补集和反射对偶的即

$$(f \ominus \mathrm{b})^c(x,y) = (f^c \oplus \hat{b})(x,y)$$

式中，$f^c = -f(x,y)$。类似地，

$$(f \oplus \mathrm{b})^c = (f^c \ominus \hat{b})$$

9.6.2 开操作和闭操作

- 灰度级图像的开操作和闭操作的表达式，与二值图像的对应操作具有相同的形式。结构元b对图像f的开操作表示为$f \circ b$

$$f \circ b = (f \ominus b) \oplus \mathrm{b}$$

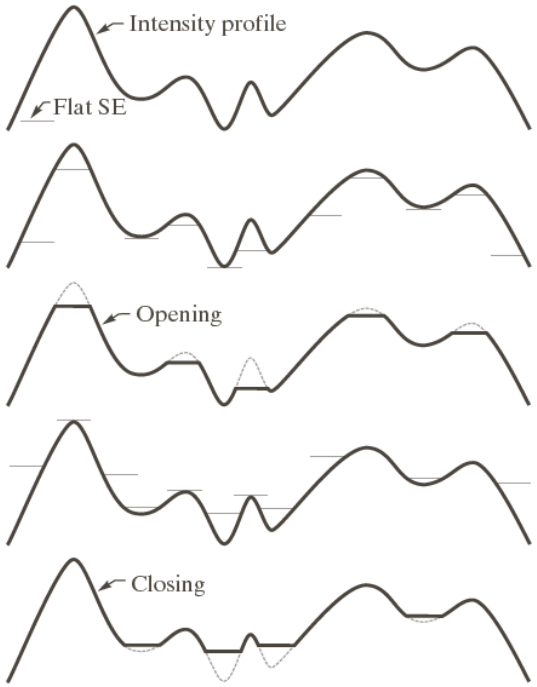
类似地，b对f的闭操作表示为$f \cdot \mathrm{b}$，即$f \cdot \mathrm{b} = (f \oplus \mathrm{b}) \ominus b$

- 开闭操作关于函数的补集和结构元的反射是对偶的：

$$(f \cdot \mathrm{b})^c = (f^c \circ \hat{b}) \text{和} (f \circ \mathrm{b})^c = \left(f^c \cdot \hat{b}\right)$$

- 几何解释

将一个图像函数视为一个三维表面，b对f的开操作可从几何解释为：在b的每个原点位置，开操作是当从f的下表面向上推动结构元时，b的任何部分所达到的最高值。闭操作通过寻找结构元的任何部分所能到达的最低点构建。

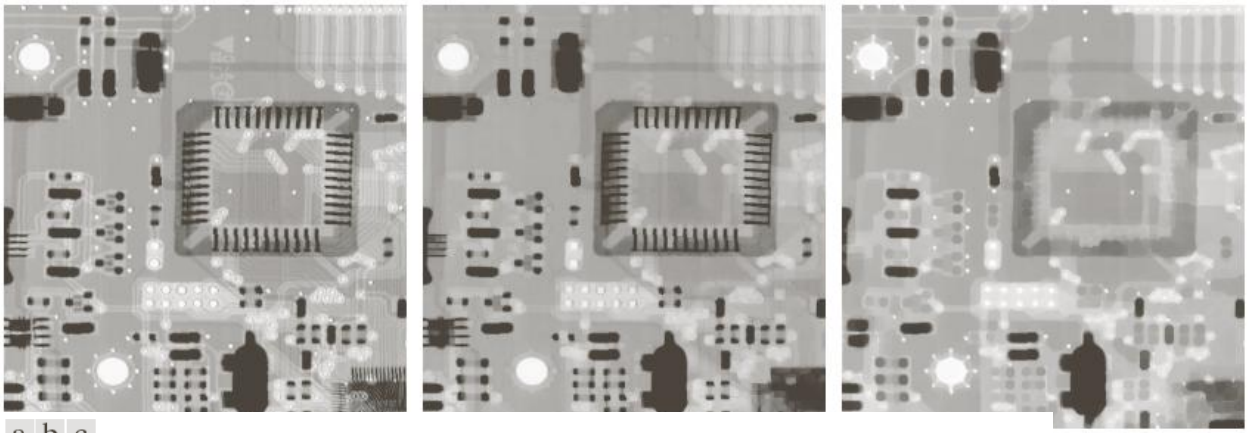


a
b
c
d
e

FIGURE 9.36 Opening and closing in one dimension. (a) Original 1-D signal. (b) Flat structuring element pushed up underneath the signal. (c) Opening. (d) Flat structuring element pushed down along the top of the signal. (e) Closing.

9.6.2 开操作和闭操作

- 开操作用于去除**较小的明亮细节**，而保持**整体灰度级**和**较大的明亮特征**相对不变
- 闭操作抑制暗细节
- 灰度级开操作和闭操作的说明
  - 开操作：所有亮特征的灰度都降低了，降低的程度取决于这些特征相对于结构元的尺寸
  - 与腐蚀的结果不同，开操作对图像的暗特征影响可忽略不计
  - 闭操作：亮细节和背景相对来说未受影响，但削弱了暗特征，削弱的程度取决于这些特征相对于结构元的尺寸

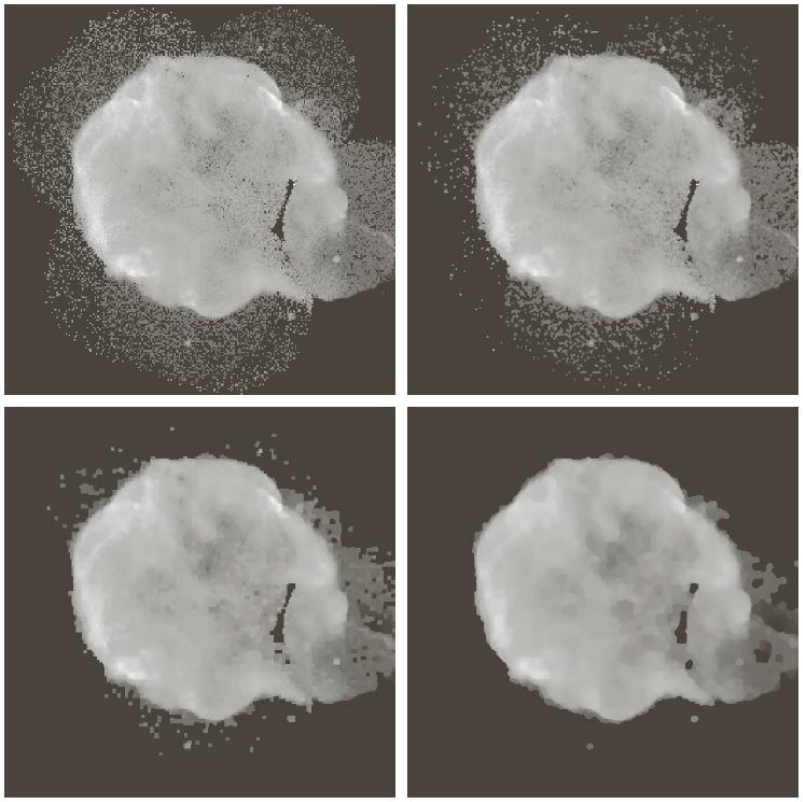


a b c

FIGURE 9.37 (a) A gray-scale X-ray image of size 448 × 425 pixels. (b) Opening using a disk SE with a radius of 3 pixels. (c) Closing using an SE of radius 5.

9.6.3 一些基本的灰度级形态学算法

- 形态学平滑
  - 因为开操作抑制比结构元小的亮细节，而闭操作抑制暗细节，所以它们常常以形态滤波的形式结合起来**平滑图像**和**去除噪声**。
  - 假设下图中心的亮区域是我们感兴趣的目标，而较小的分量是噪声。我们的目的是去除噪声。



a b
c d

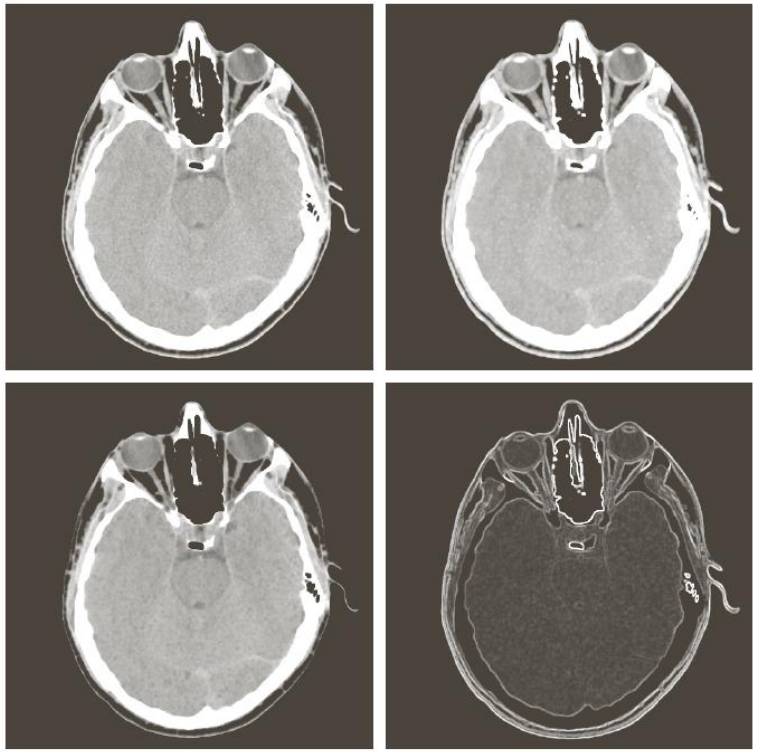
FIGURE 9.38
(a) 566 × 566 image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of performing opening and closing sequences on the original image with disk structuring elements of radii, 1, 3, and 5, respectively. (Original image courtesy of NASA.)

9.6.3 一些基本的灰度级形态学算法

- 形态学梯度
  - 膨胀和腐蚀可与图像相减结合起来得到一幅图像的形态学梯度，由g定义：

$$g = (f \oplus b) - (f \ominus b)$$

  - 膨胀粗化一幅图像中的区域，而腐蚀则细化它们。**膨胀和腐蚀之差强调区域间的边界。**同质区域不受影响，因此相减操作趋于消除同质区域。
  - 最终结果是边缘被增强而同质区域的贡献则被抑制的图像，因此产生了“类似于微分”效果。



a b
c d

FIGURE 9.39
(a) 512 × 512 image of a head CT scan. (b) Dilation. (c) Erosion. (d) Morphological gradient, computed as the difference between (b) and (c). (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

9.6.3 一些基本的灰度级形态学算法

顶帽变换和底帽变换

- 图像相减与开操作和闭操作相结合，可产生所谓的Top-hat（顶帽）变换和Bottom-hat（底帽）变换。
- 灰度级图像f的**顶帽变换**定义为f减去其开操作：

$$T_{hat}(f) = f - (f \circ b)$$

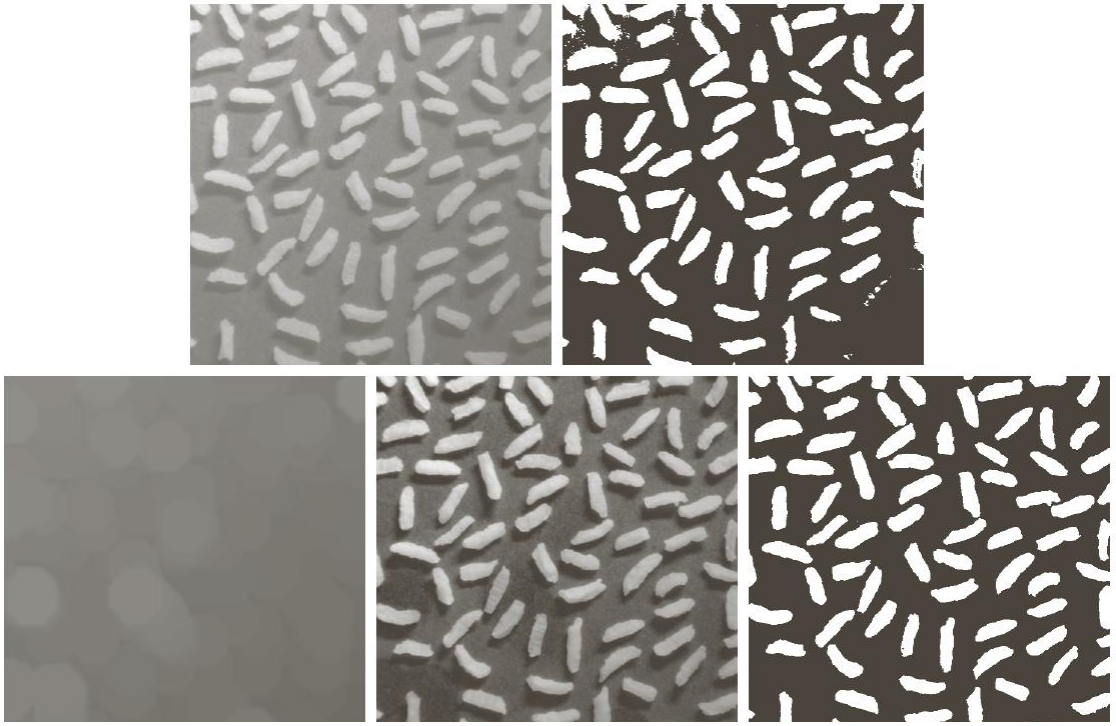
类似地，f的**底帽变换**定义为f的闭操作减去f：

$$B_{hat}(f) = (f \cdot b) - f$$

- 这些变换的主要应用之一是：用一个结构元通过开操作或闭操作从一幅图像中删除物体；然后，差操作得到一幅仅保留已删除分量的图像。
- 顶帽变换用于暗背景上的亮物体，而底帽变换则用于相反的情况。由于这一原因，这些变换又称为白顶帽变换和黑底帽变换
- 顶帽变换的一个重要用途是**校正不均匀光照的影响**
  - ✓ 合适（均匀）的光照便于在从背景中提取目标

9.6.3 一些基本的灰度级形态学算法

- 顶帽变换和底帽变换
  - 下图是在非均匀光照下得到的，如图像底部及最右侧的暗色区域就是明证。非均匀光照的最终结果导致了暗区域的分割错误（一些米粒未从背景中提取出来），且在图像的左上角，背景部分被错误地分类了

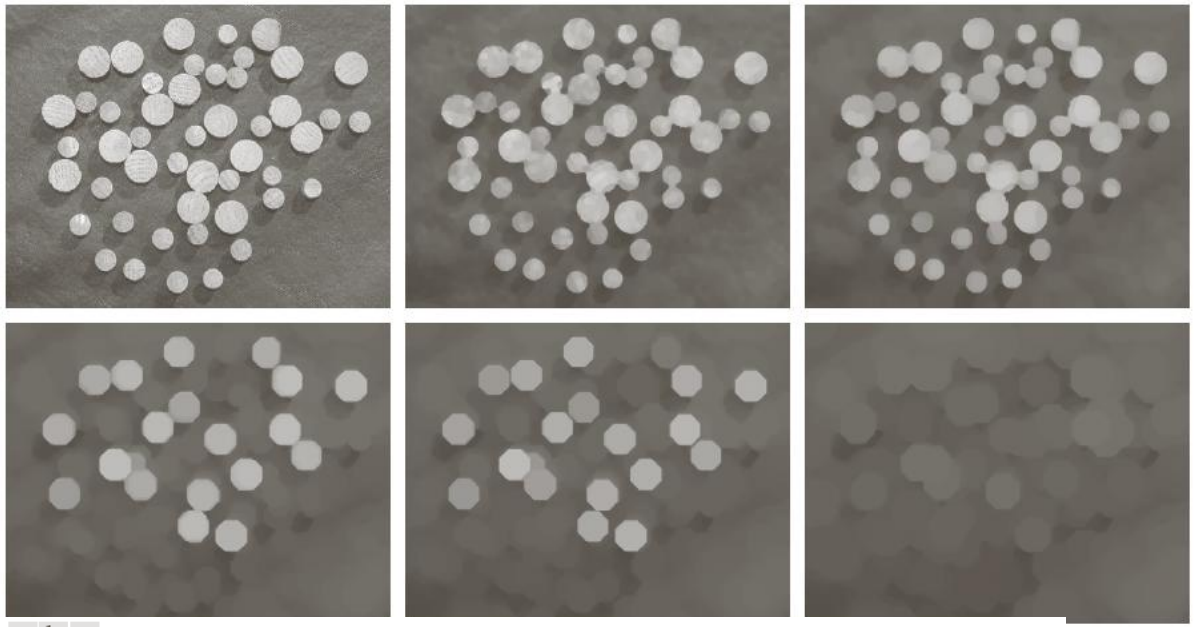


a b
c d e

FIGURE 9.40 Using the top-hat transformation for *shading correction*. (a) Original image of size 600 × 600 pixels. (b) Thresholded image. (c) Image opened using a disk SE of radius 40. (d) Top-hat transformation (the image minus its opening). (e) Thresholded top-hat image.

9.6.3 一些基本的灰度级形态学算法

- 粒度测定
  - 左图是两种不同大小的木钉图像
  - 右图显示了该差值阵列的曲线，图中有两个明显的峰值，表面图像存在两种主要的物体尺寸



a b c
d e f

FIGURE 9.41 (a) 531 × 675 image of wood dowels. (b) Smoothed image. (c)–(f) Openings of (b) with disks of radii equal to 10, 20, 25, and 30 pixels, respectively. (Original image courtesy of Dr. Steve Eddins, The MathWorks, Inc.)

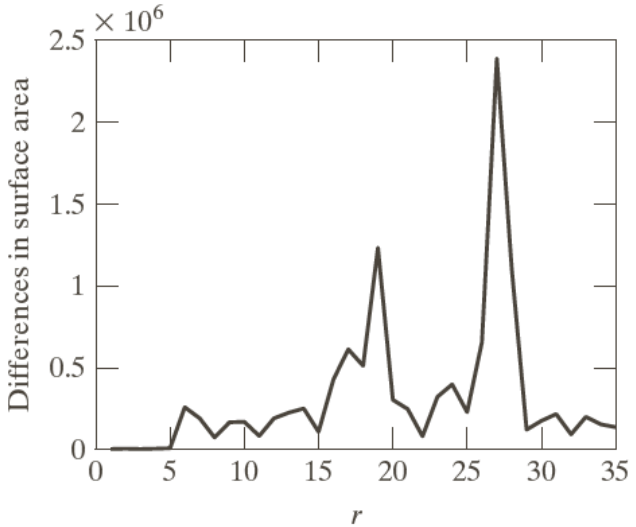
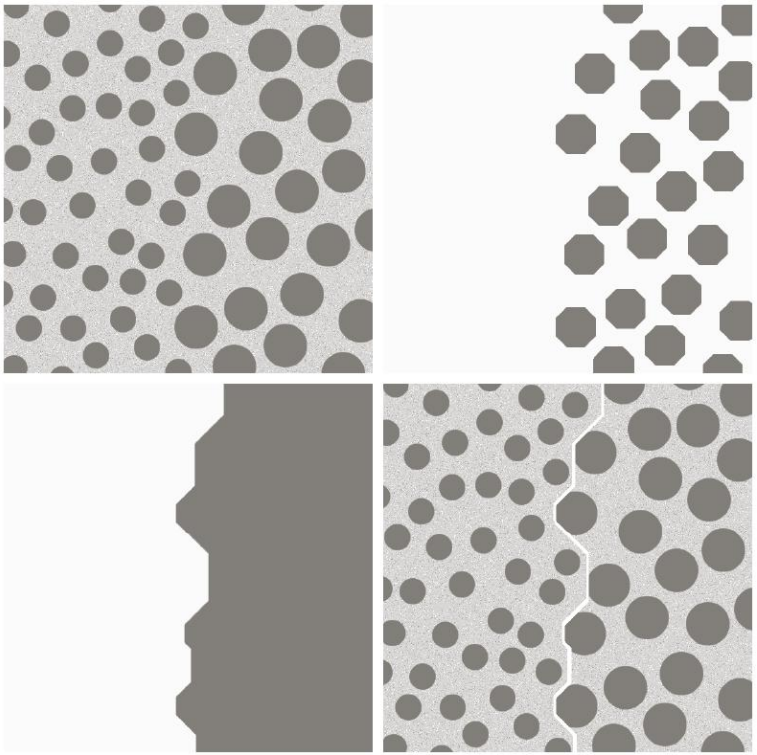


FIGURE 9.42 Differences in surface area as a function of SE disk radius, *r*. The two peaks are indicative of two dominant particle sizes in the image.

9.6.3 一些基本的灰度级形态学算法

- 纹理分割
  - 右图有两个纹理区域：右边大斑点组成的区域和左边小斑点组成的区域。目的是以纹理内容为基础找到两个区域的边界
  - 对图执行闭操作后删除了小斑点
  - 然后执行开操作后删除了大斑点间的亮间隔区域
  - 再对图像执行形态学梯度操作，得到两个区域的边界
  - 将边界叠加到原图像上



a b
c d

FIGURE 9.43 Textural segmentation. (a) A 600 × 600 image consisting of two types of blobs. (b) Image with small blobs removed by closing (a). (c) Image with light patches between large blobs removed by opening (b). (d) Original image with boundary between the two regions in (c) superimposed. The boundary was obtained using a morphological gradient operation.

9.6.4 灰度级形态学重建

- 灰度级形态学重建基本上与二值图像所介绍的相同方法来定义。令f和g分别代表**标记图像**和**模板图像**。我们假设f和g是大小相同的灰度级图像，且$f \leq g$。f关于g的大小为1的**测地膨胀**定义为：

$$D_g^{(1)}(f) = (f \oplus b) \wedge g$$

式中，$\wedge$代表点方式的最小算子。该式指出，大小为1的测地膨胀是先计算b对f的膨胀，然后选择在每个(x,y)点处该结果和g间的最小者。f关于g的大小为n的测地膨胀定义为：

$$D_g^{(n)}(f) = D_g^{(1)}[D_g^{(n-1)}(f)], \quad \text{其中} D_g^{(0)}(f) = f$$

类似地，f关于g的大小为1的测地腐蚀定义为

$$E_g^{(1)}(f) = (f \ominus b) \vee g$$

式中，$\vee$代表点方式的最大算子。f关于g的大小为n的测地腐蚀定义为

$$E_g^{(n)}(f) = E_g^{(1)}[E_g^{(n-1)}(f)], \quad \text{其中} E_g^{(0)}(f) = f$$

9.6.4 灰度级形态学重建

- 灰度级标记图像f对灰度模板图像g的**膨胀形态学重建**，定义为f关于g的测地膨胀反复迭代，直至达到**稳定**；即

$$R_g^D(f) = D_g^{(k)}(f)$$

且k应使$D_g^{(k)}(f) = D_g^{(k+1)}(f)$。

f对g的**腐蚀形态学重建**定义为

$$R_g^E(f) = E_g^{(k)}(f)$$

且k应使$E_g^{(k)}(f) = E_g^{(k+1)}(f)$。

- 如二值情况那样，灰度级图像重建的开操作首先腐蚀输入图像，并用它作为标记图像。图像f的大小为n的**重建开操作**定义为，先对f进行大小为n的腐蚀，再由f的膨胀重建；即

$$O_R^{(n)}(f) = R_f^D[(f \ominus nb)]$$

式中，$(f \ominus nb)$表示b对f的n次腐蚀。

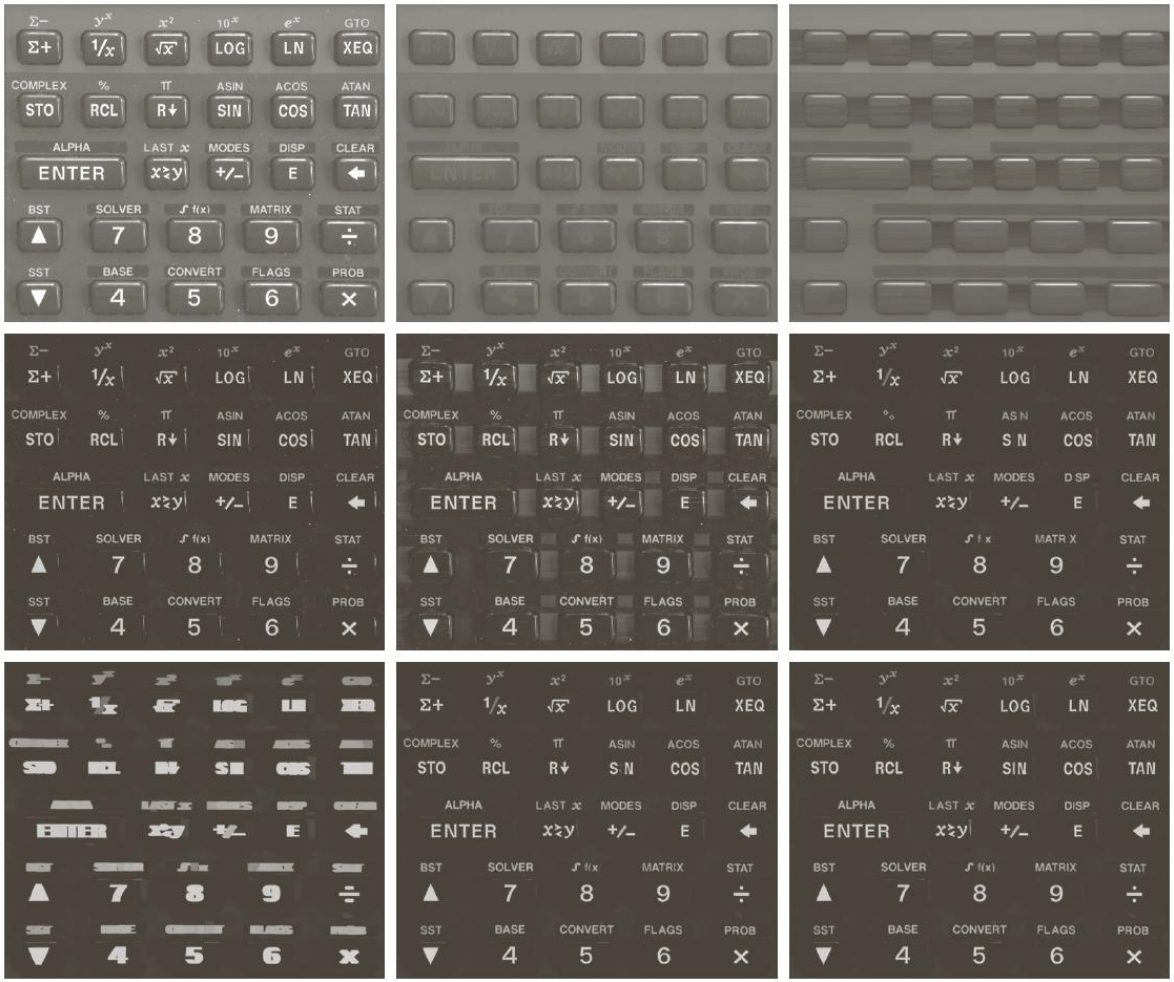
类似地，图像f的大小为n的**重建闭操作**定义为，先对f进行大小为n的膨胀，再由f的腐蚀重建即

$$C_R^{(n)}(f) = R_f^E[(f \oplus nb)]$$

式中，$(f \oplus nb)$表示b对f的n次膨胀。

9.6.4 灰度级形态学重建

- 最后，如下例所示，称为重建顶帽的一种有用的技术是**从一幅图像中减去其重建开操作**
- 使用形态学重建展平复杂的背景
  - 目的是仅留下恒定灰度背景上的正文



a b c
d e f
g h i

FIGURE 9.44 (a) Original image of size 1134 × 1360 pixels. (b) Opening by reconstruction of (a) using a horizontal line 71 pixels long in the erosion. (c) Opening of (a) using the same line. (d) Top-hat by reconstruction. (e) Top-hat. (f) Opening by reconstruction of (d) using a horizontal line 11 pixels long. (g) Dilation of (f) using a horizontal line 21 pixels long. (h) Minimum of (d) and (g). (i) Final reconstruction result. (Images courtesy of Dr. Steve Eddins, The MathWorks, Inc.)