



武汉大学
Wuhan University

第九章 形态学图像处理

武汉大学计算机学院



形态学图像处理

本章要点

- 概述
- 集合论基础知识
- 膨胀和腐蚀
- 开操作和闭操作
- 击中或击不中变换
- 形态学的主要应用
- 灰度级图像扩展

概述

- 形态学一般指生物学中研究动物和植物结构的一个分支
- 用数学形态学（也称图像代数）表示以形态为基础对图像进行分析的数学工具
- 基本思想是用具有一定形态的结构元素去度量和提取图像中的对应形状以达到对图像分析和识别的目的
- 形态学图像处理的数学基础和所用语言是集合论

概述

- 形态学图像处理的应用可以简化图像数据，保持它们基本的形状特性，并除去不相干的结构
- 形态学图像处理的基本运算有4个：膨胀、腐蚀、开操作和闭操作

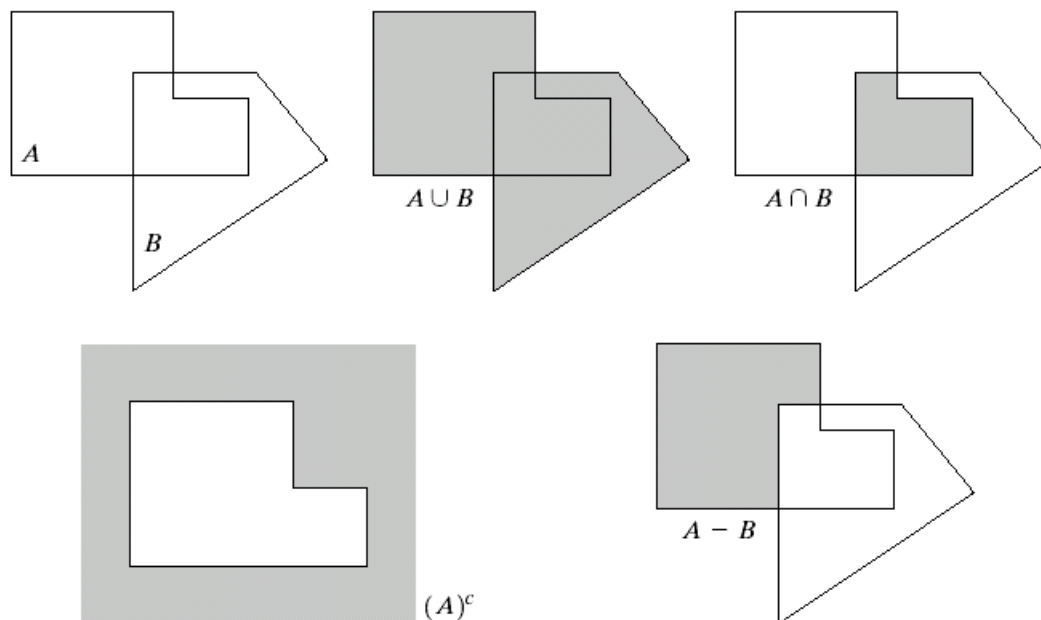
形态学图像处理

本章要点

- 概述
- 集合论基础知识
- 膨胀和腐蚀
- 开操作和闭操作
- 击中或击不中变换
- 形态学的主要应用
- 灰度级图像扩展

集合论基础知识

□ 集合的并、交、补、差



a	b	c
d	e	

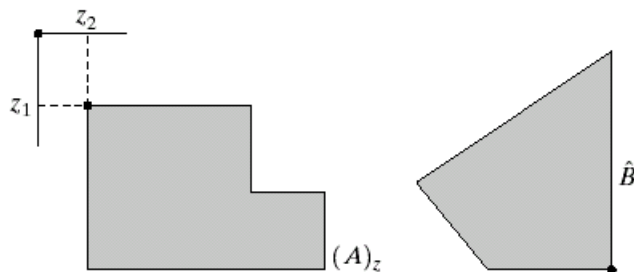
FIGURE 9.1

(a) Two sets A and B . (b) The union of A and B . (c) The intersection of A and B . (d) The complement of A . (e) The difference between A and B .

集合论基础知识

- 集合 B 的反射 \hat{B} , 定义为 $\hat{B} = \{w | w = -b, b \in B\}$ 即关于原集合原点对称
- 集合 A 平移到点 $z=(z_1, z_2)$, 表示为 $(A)_z$, 定义为

$$(A)_z = \{c | c = a + z, a \in A\}$$



a b

FIGURE 9.2

(a) Translation of A by z .

(b) Reflection of B . The sets A and B are from Fig. 9.1.

集合论基础知识

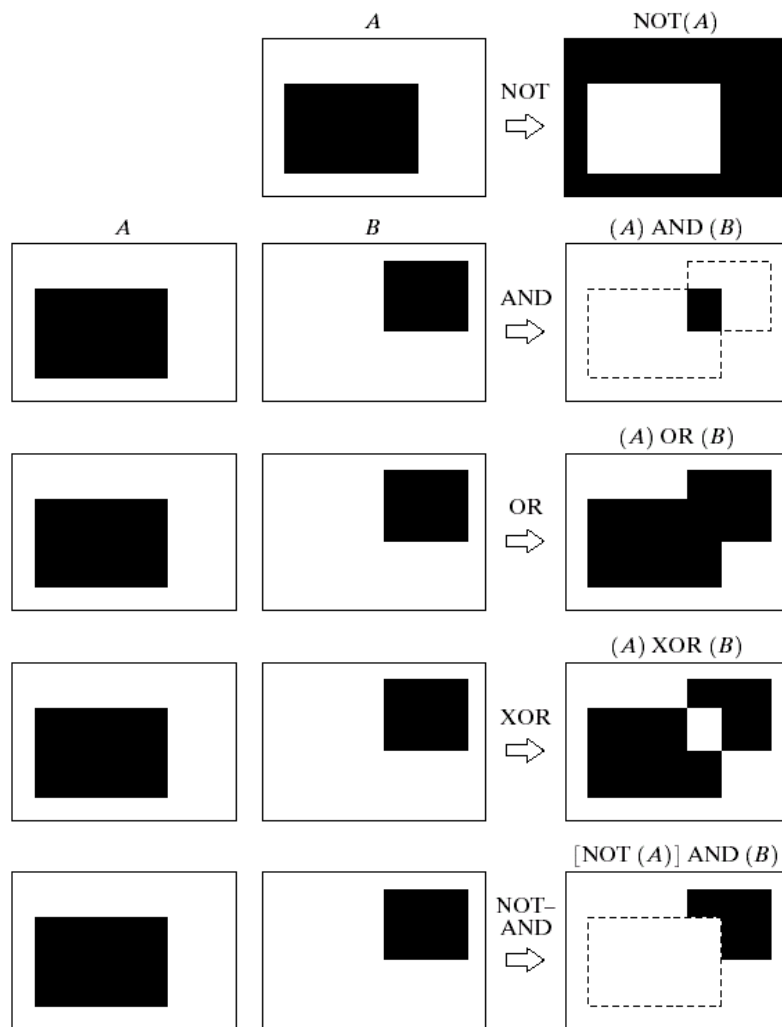


FIGURE 9.3 Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

形态学图像处理

本章要点

- 概述
- 集合论基础知识
- 膨胀和腐蚀
- 开操作和闭操作
- 击中或击不中变换
- 形态学的主要应用
- 灰度级图像扩展

膨胀和腐蚀

- 二值形态学中的运算对象是集合。设 A 为图像集合， S 为结构元为结构元素，数学形态学运算是用 S 对 A 进行操作。
- 需要指出，实际上结构元素本身也是一个图像集合。对每个结构元素可以指定一个原点，它是结构元素参与形态学运算的参考点。
- 应注意，原点可以包含在结构元素中，也可以不包含在结构元素中，但运算的结果常不相同。
- 二值形态学中两个最基本的运算是膨胀与腐蚀

膨胀和腐蚀

□ 膨胀：使图像扩大

A和B是两个集合，A被B膨胀定义为：

$$A \oplus B = \left\{ z \mid \left(\hat{B} \right)_z \cap A \neq \phi \right\}$$

- 上式表示：B的反射进行平移与A的交集不为空

膨胀和腐蚀

□ 膨胀的另一个定义

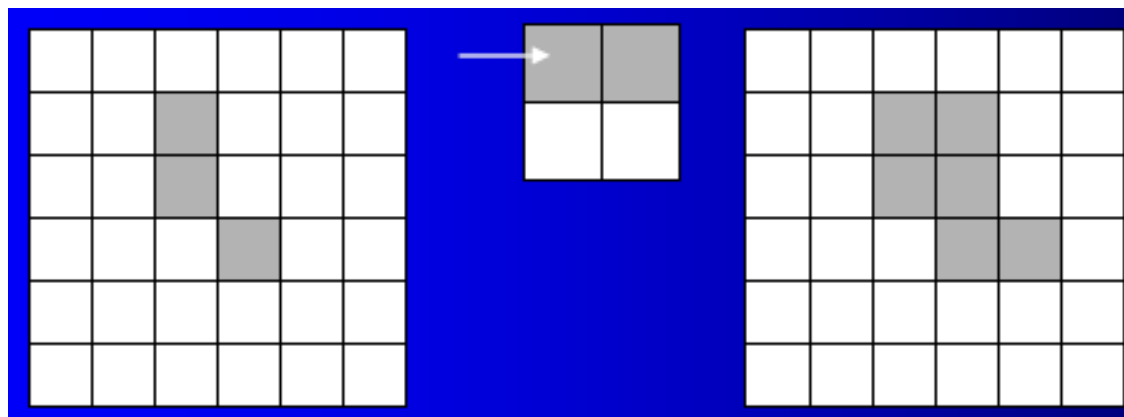
$$A \oplus B = \left\{ z \mid \left(\hat{B} \right)_z \cap A \subseteq A \right\}$$

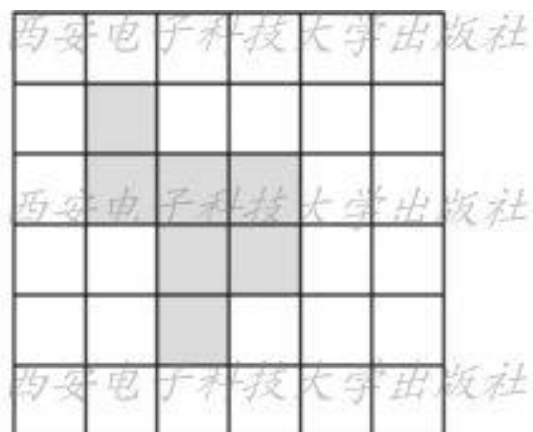
上式表示：B的反射进行平移与A的交集是A的子集

膨胀和腐蚀

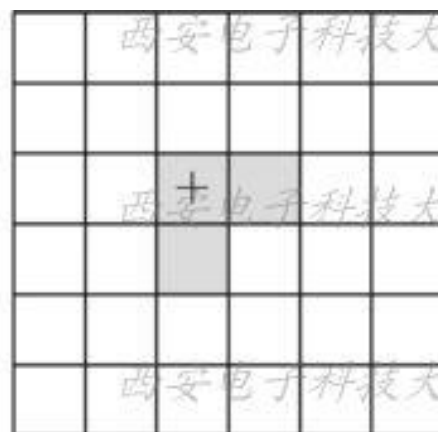
□ 算法实现

- 将结构元素B的原点移至集合A的某一点,
- 将结构元素B中点的坐标与集合A中该点坐标相加,得到对集合中一点的膨胀运算结果.
- 对集合中所有元素重复该过程

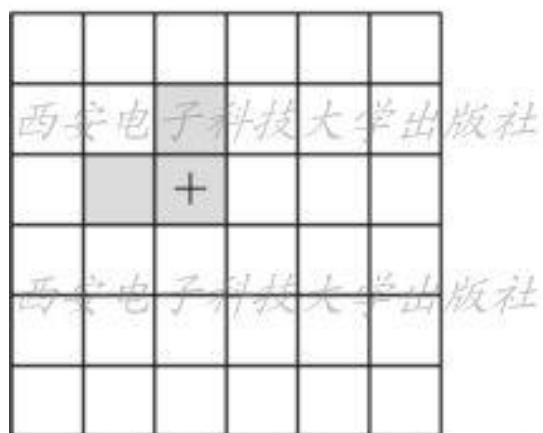




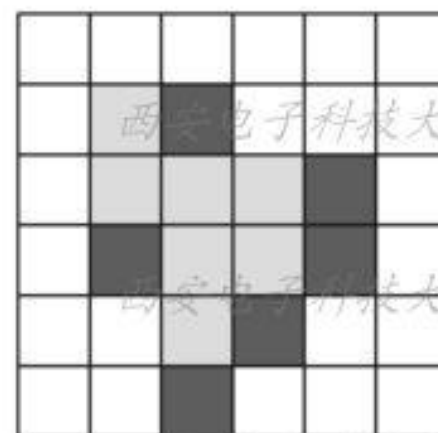
(a) 图像X
西安电子科技大学出版社



(b) 结构元素S
西安电子科技大学



(c) S的反射
西安电子科技大学出版社



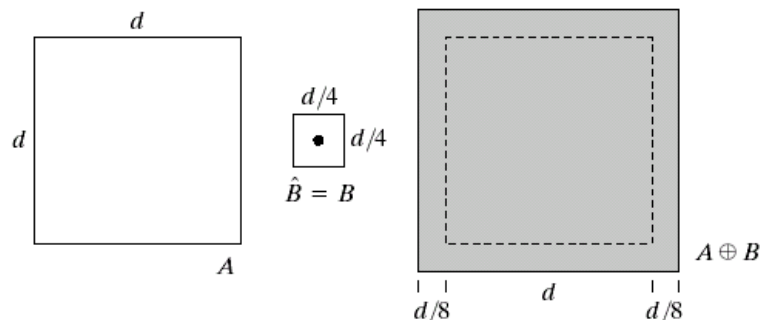
(d) S膨胀X的结果
西安电子科技大学

膨胀和腐蚀

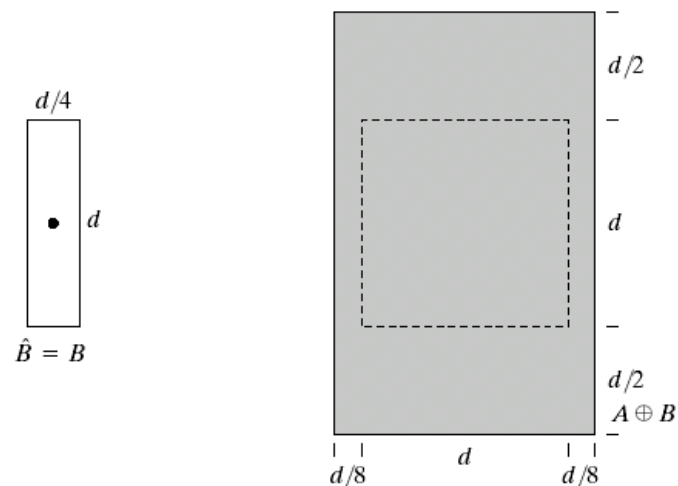
a	b	c
d		e

FIGURE 9.4

- (a) Set A .
- (b) Square structuring element (dot is the center).
- (c) Dilation of A by B , shown shaded.
- (d) Elongated structuring element.
- (e) Dilation of A using this element.



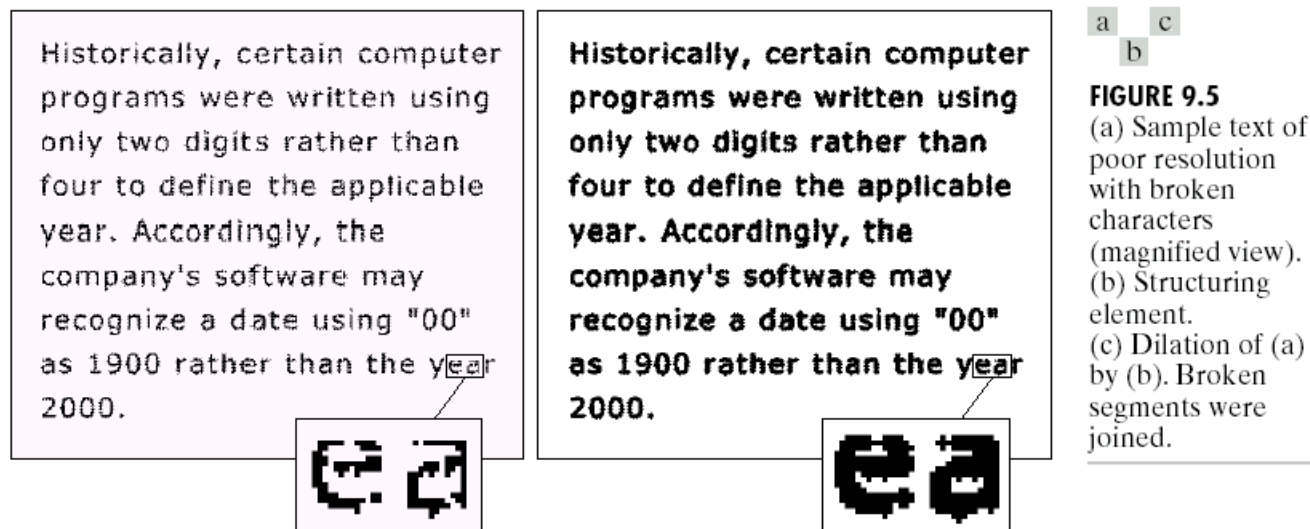
保证B的反射的平移与A的交集不为空



膨胀和腐蚀

□ 桥接文字裂缝

- 优点：在一幅二值图像中直接得到结果，对比低通滤波方法



0	1	0
1	1	1
0	1	0

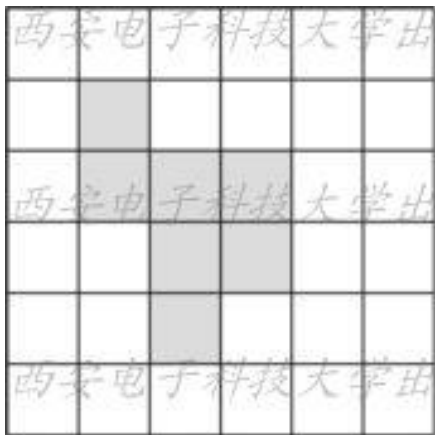
膨胀的结构元素B

膨胀和腐蚀

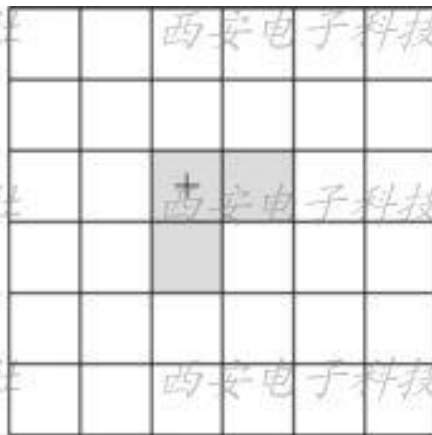
□ 腐蚀：使图像缩小

A和B是两个集合，A被B腐蚀定义为：

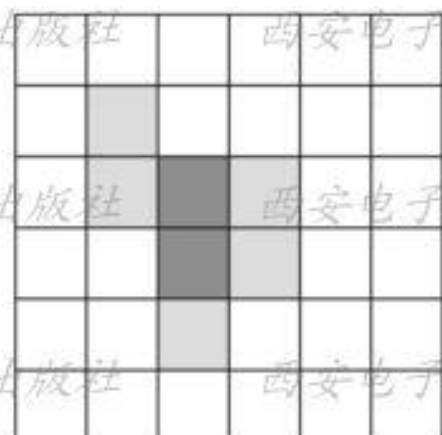
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$



(a) 集合 X



(b) 结构元素 S

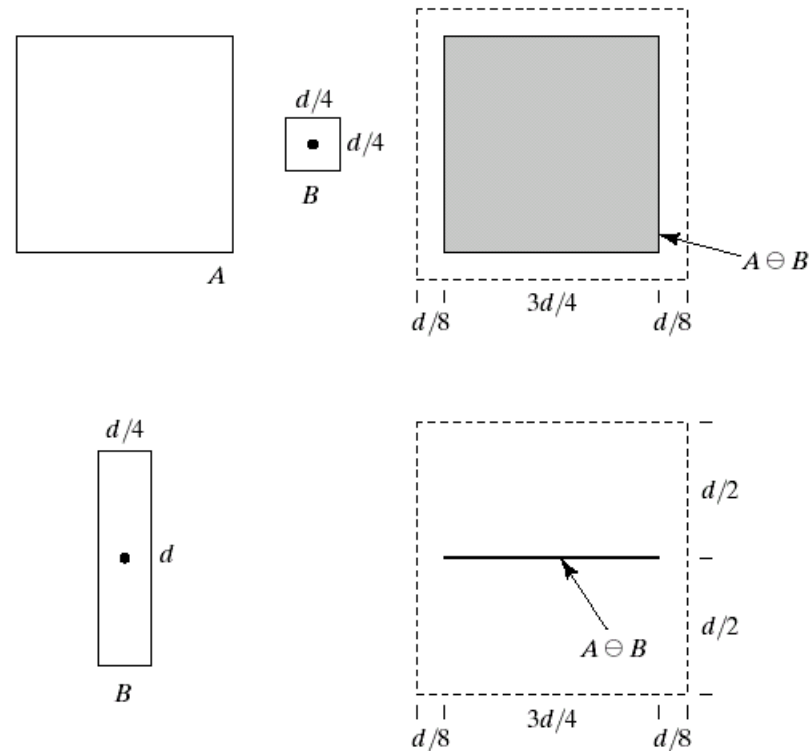


(c) 用 S 腐蚀 X 的结果

腐蚀运算示例

膨胀和腐蚀

A被B腐蚀的结果

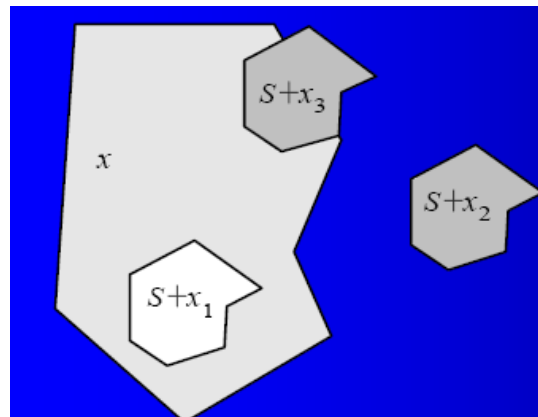


a	b	c
d	e	

FIGURE 9.6 (a) Set A . (b) Square structuring element. (c) Erosion of A by B , shown shaded. (d) Elongated structuring element. (e) Erosion of A using this element.

膨胀和腐蚀

- 对一个给定的目标图像 X 和一个结构元素 S ，将 S 在图像上移动。在每一个当前位置 x ， $S+x$ 只有三种可能的状态
 - 第一种情形说明 $S+x$ 与 X 相关最大，
 - 第二种情形说明 $S+x$ 与 X 不相关，
 - 而第三种情形说明 $S+x$ 与 X 只是部分相关
- X 用 S 腐蚀的结果是所有使 S 平移 x 后仍在 X 中的 x 的集合。换句话说，用 S 来腐蚀 X 得到的集合是 S 完全包括在 X 中时 S 的原点位置的集合。

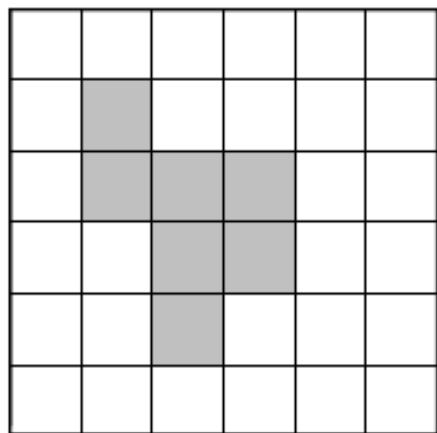


膨胀和腐蚀

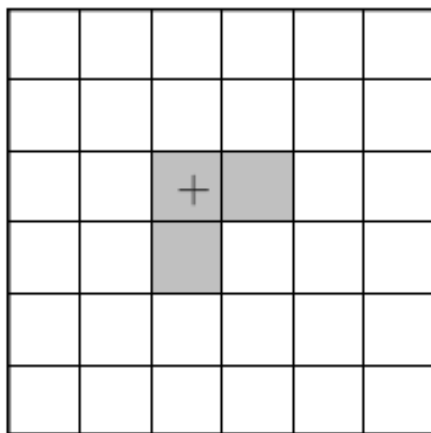
- 腐蚀在数学形态学运算中的作用是消除物体边界点。
- 如果结构元素取 3×3 的像素块，腐蚀将使物体的边界沿周边减少一个像素。
- 腐蚀可以把小于结构元素的物体(毛刺、小凸起)去除，这样选取不同大小的结构元素，就可以在原图像中去掉不同大小的物体。
- 如果两个物体之间有细小的连通，那么当结构元素足够大时，通过腐蚀运算可以将两个物体分开。

膨胀和腐蚀

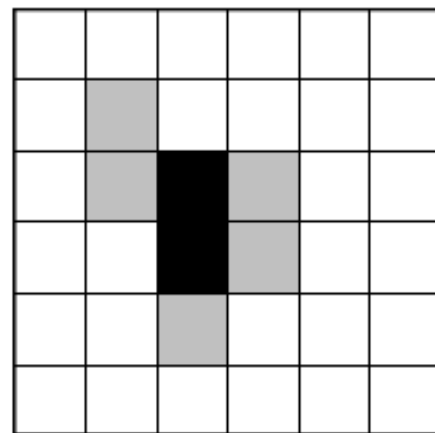
- 图(a)中的阴影部分为集合X，图(b)中的中的阴影部分为结构元素S，而图(c)中黑色部分给出了结果。由图可见，腐蚀将图像（区域）收缩小了。



(a)



(b)

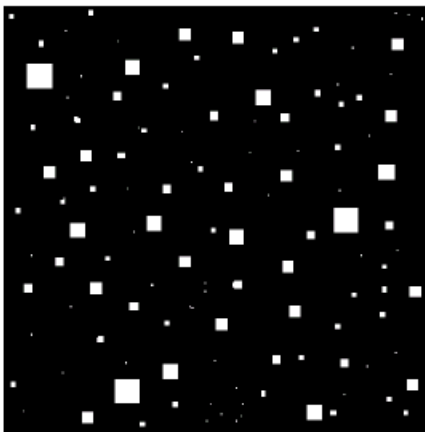


(c)

膨胀和腐蚀

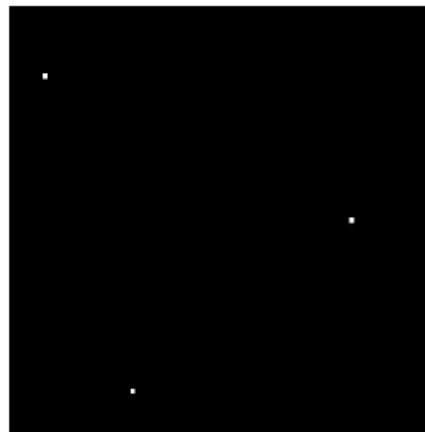
□ 使用腐蚀消除图像的细节部分，产生滤波器的作用

包含边长为
1,3,5,7,9和15像素
正方形的二值图像



a b c

使用 13×13 像素大
小的结构元素腐蚀
原图像的结果



使用 13×13 像素大小的结构
元素膨胀图b，恢复原来
 15×15 尺寸的正方形

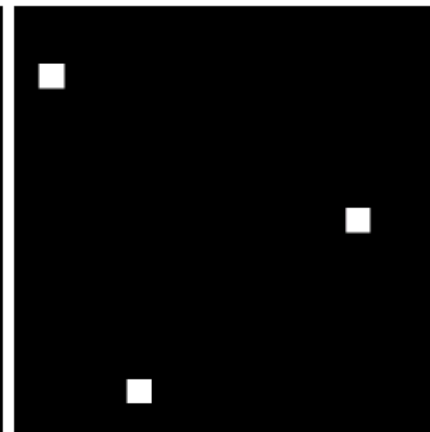


FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

形态学图像处理

本章要点

- 概述
- 集合论基础知识
- 膨胀和腐蚀
- 开操作和闭操作
- 击中或击不中变换
- 形态学的主要应用
- 灰度级图像扩展

开操作和闭操作

□ 开操作：使图像的轮廓变得光滑，断开狭窄的间断和消除细的突出物

■ 使用结构元素B对集合A进行开操作，定义为：

$$A \circ B = (A \ominus B) \oplus B$$

■ 含义：先用B对A腐蚀，然后用B对结果膨胀

■ 另一个定义

$$A \circ B = \cup \{(B)_z \mid (B)_z \subseteq A\}$$

开操作和闭操作

- 闭操作：同样使图像的轮廓变得光滑，但与开操作相反，它能消除狭窄的间断和长细的鸿沟，消除小的孔洞，并填补轮廓线中的裂痕
 - 使用结构元素B对集合A进行闭操作，定义为：
$$A \bullet B = (A \oplus B) \ominus B$$
 - 含义：先用B对A膨胀，然后用B对结果腐蚀

开操作和闭操作

□ 开操作的几何解释

- $A \circ B$ 的边界通过 B 中的点完成
- B 在 A 的边界内转动时, B 中的点所能到达的 A 的边界的的最远点

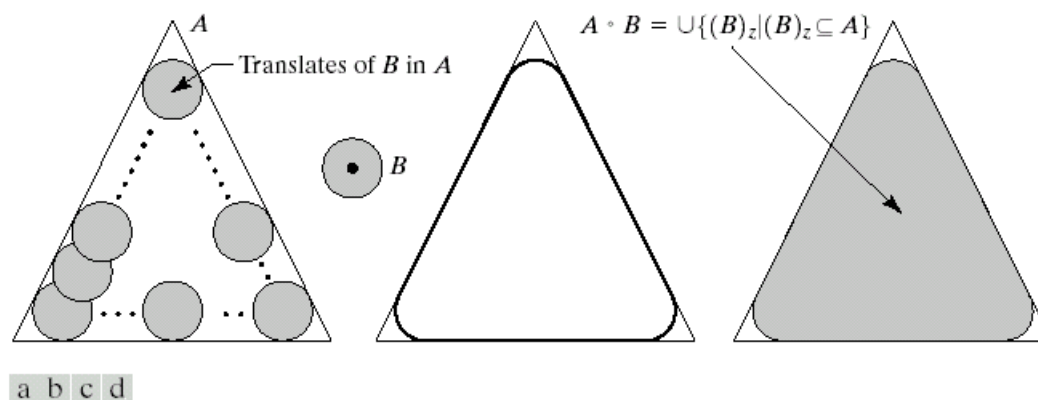


FIGURE 9.8 (a) Structuring element B "rolling" along the inner boundary of A (the dot indicates the origin of B). (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded).

开操作和闭操作

- 闭操作的几何解释
 - $A \bullet B$ 的边界通过 B 中的点完成
 - B 在 A 的边界外部转动

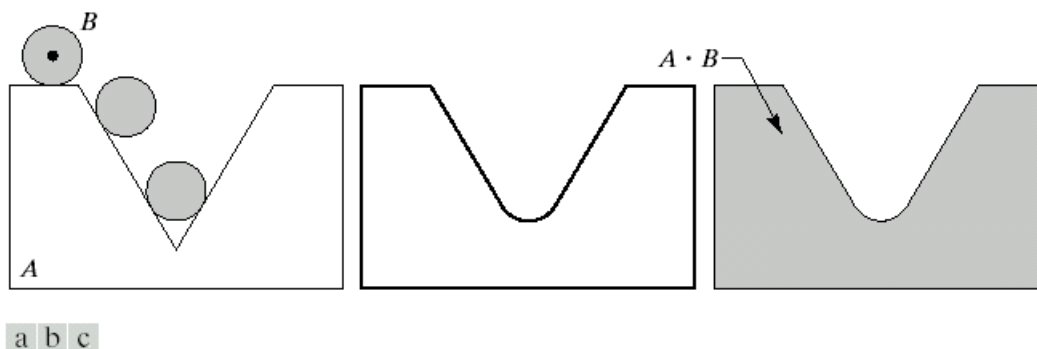


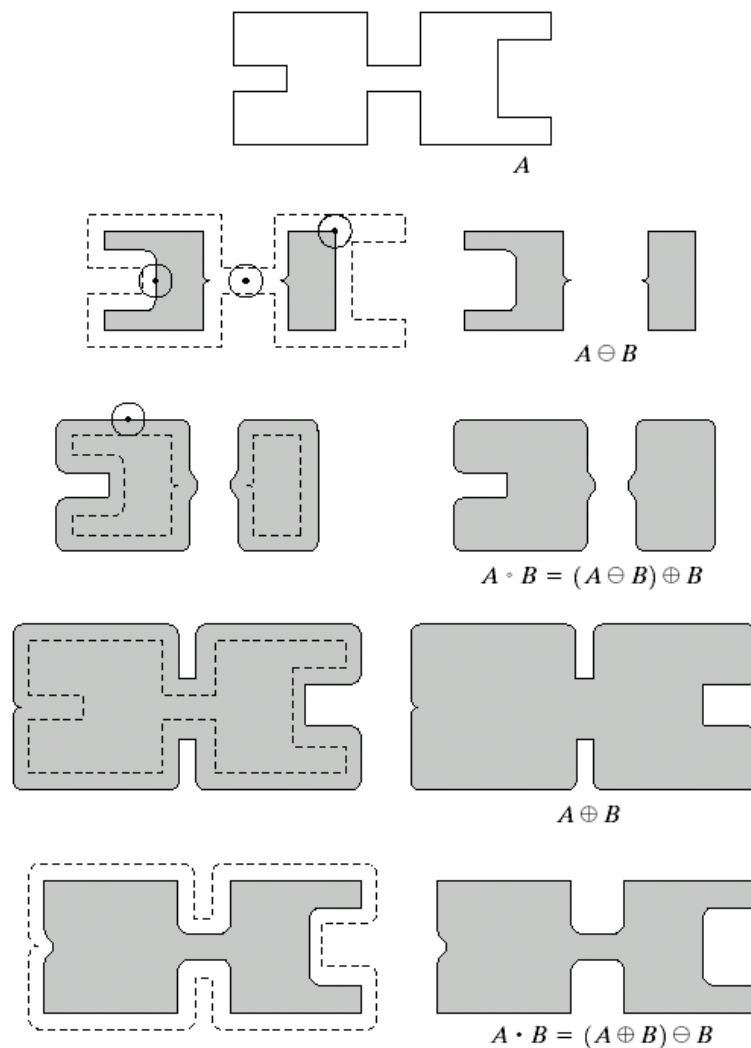
FIGURE 9.9 (a) Structuring element B “rolling” on the outer boundary of set A . (b) Heavy line is the outer boundary of the closing. (c) Complete closing (shaded).

开操作和闭操作

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 dark dot is the center of the structuring element.



开操作和闭操作

□ 开操作的3条性质

- $A^\circ B$ 是 A 的子集合
- 如果 C 是 D 的子集, 则 $C^\circ B$ 是 $D^\circ B$ 的子集
- $(A^\circ B)^\circ B = A^\circ B$

□ 闭操作的3条性质

- A 是 $A \bullet B$ 的子集合
- 如果 C 是 D 的子集, 则 $C \bullet B$ 是 $D \bullet B$ 的子集
- $(A \bullet B) \bullet B = A \bullet B$

开操作和闭操作

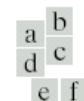
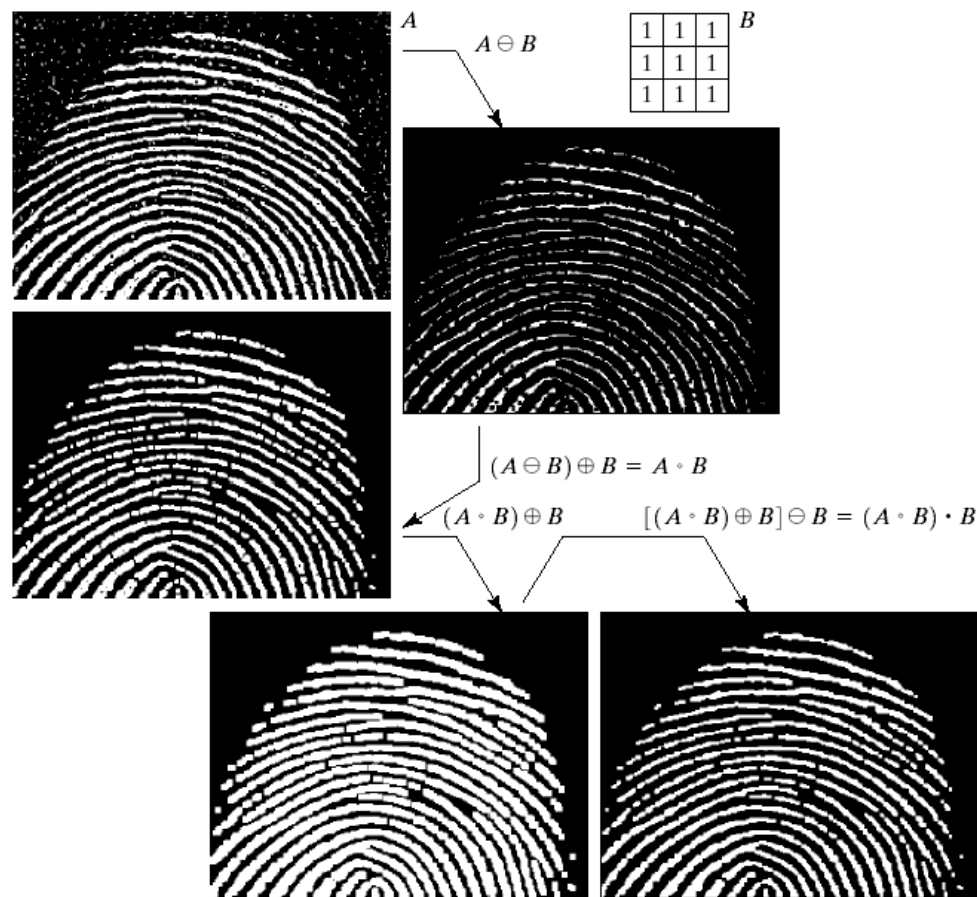


FIGURE 9.11

(a) Noisy image.
 (c) Eroded image.
 (d) Opening of A .
 (d) Dilation of the opening.
 (e) Closing of the opening. (Original image for this example courtesy of the National Institute of Standards and Technology.)

先开操作再闭操作，构成噪声滤波器

开操作和闭操作

- a图是受噪声污染的指纹二值图像，噪声为黑色背景上的亮元素和亮指纹部分的暗元素
- b图是使用的结构元素
- c图是使用结构元素对图a腐蚀的结果：黑色背景噪声消除了，指纹中的噪声尺寸增加
- d图是使用结构元素对图c膨胀的结果：包含于指纹中的噪声分量的尺寸被减小或被完全消除，带来的问题是：在指纹纹路间产生了新的间断

开操作和闭操作

- e图是对图d膨胀的结果，图d的大部分间断被恢复，但指纹的线路变粗了
- f图是对图e腐蚀的结果，即对图d中开操作的闭操作。最后结果消除了噪声斑点
- 缺点：指纹线路还是有缺点，可以通过加入限制性条件解决

形态学图像处理

本章要点

- 概述
- 集合论基础知识
- 膨胀和腐蚀
- 开操作和闭操作
- 击中或击不中变换
- 形态学的主要应用
- 灰度级图像扩展

击中或击不中变换

在8.1.2节中简单地给出了击中与击不中的概念，下面讨论击中与击不中的严格定义及其在数字图像处理中的意义。一般地，一个物体的结构可以由物体内部各种成分之间的关系来确定。为了研究图像的结构，可以逐个地利用各种成分（例如各种结构元素）对其进行检验，判定哪些成分包括在图像之内，哪些在图像之外，从而最终确定图像的结构。击中/击不中变换就是在这个意义上提出的。设 X 是被研究的图像， S 是结构元素，而且 S 由两个不相交的部分 S_1 和 S_2 组成，即 $S = S_1 \cup S_2$ ，且 $S_1 \cap S_2 = \emptyset$ 。

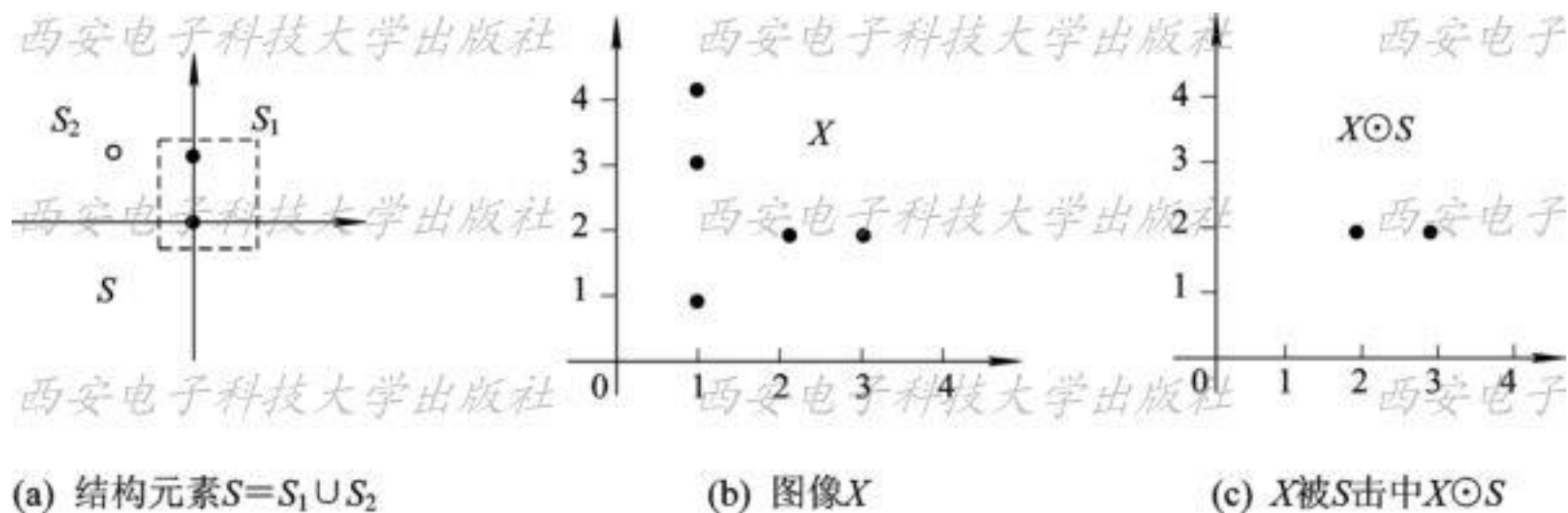
击中或击不中变换

于是， X 被 S “击中” 的结果定义为

$$X \odot S = \{x \mid S_1 + x \subseteq X, \text{ 且 } S_2 + x \subseteq X^c\}$$

从式(8-13)可以看出， X 被 S 击中的结果仍是一个图像，其中每点 x 必须同时满足两个条件： S_1 被 x 平移后包含在 X 内，而且 S_2 被 x 平移后不在 X 内。图8-16给出了一个 X 被 S 击中的例子。

击中或击不中变换



X被S击中示意图

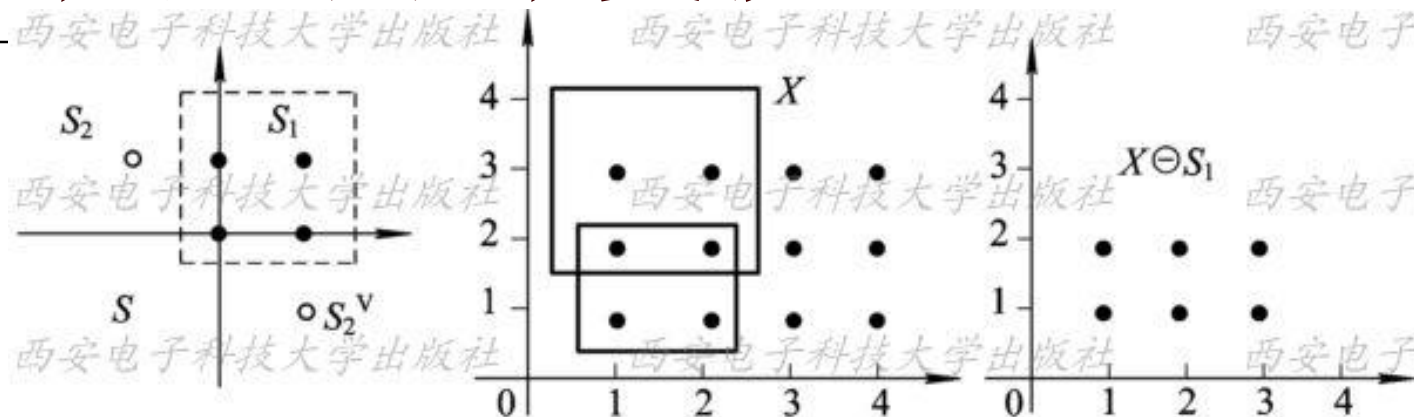
击中或击中不中变换

击中运算还有另外一种表达形式：

$$\begin{aligned} X \odot S &= (X \ominus S_1) \cup (X^c \oplus S_2) \\ &= (X \ominus S_1) \cup (X \oplus S_2^V)^c \\ &= (X \ominus S_1) - (X \oplus S_2^V) \end{aligned}$$

X 被 S 击中的结果相当于 X 被 S_1 腐蚀的结果与 X 被 S_2 的反射集 S_2^V 膨胀的结果之差。由此可见，击中运算也可以借助于腐蚀、膨胀两基本运算来实现。

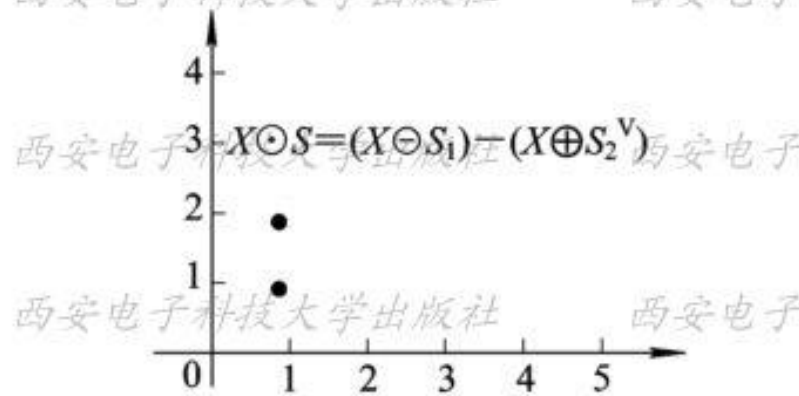
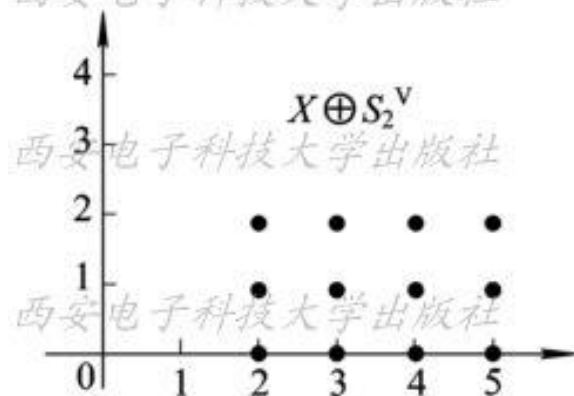
击中或击不中变换



(a) 结构元素 $S = S_1 \cup S_2$

(b) 图像 X

(c) $X \ominus S_1$



(d) $X \oplus S_2^v$

(e) $X \odot S$

X被S击中示意图

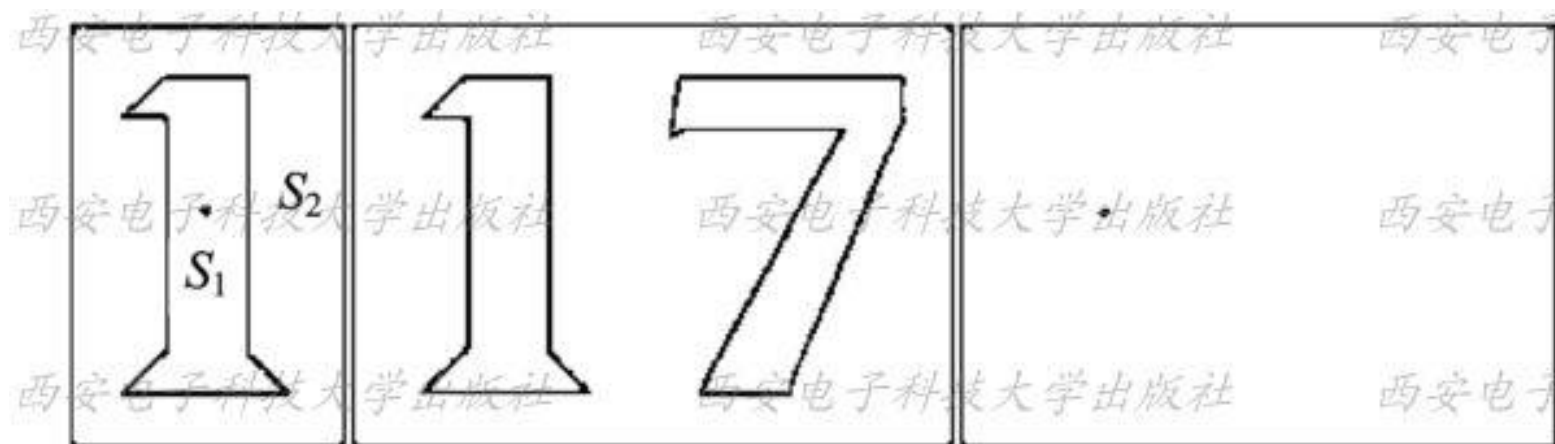
击中或击中不中变换

下面，进一步讨论击中运算的含义。在图8-17中，如果 S 中不包含 S_2 ，那么 $X \odot S$ 与 $X \odot S_1$ 相同，共包括6个点。这表明 X 被 S 腐蚀后还剩6个点，就是说， X 中共包含6个形如 S_1 的结构元素，它们的原点位置构成 $X \odot S$ 。但将 S_2 加入 S 后，相当于对 $X \odot S$ 增加了一个条件：不仅从 X 中找出那些形如 S_1 的部分，而且要去掉那些在左边有一个邻点的部分。这样一来，在 X 中只剩下两部分(在方框内)满足要求的结构元素。它们的原点位置构成了最终的 $X \odot S$ 。

击中或击不中变换

由此可见，击中运算相当于一种条件比较严格的模板匹配，它不仅指出被匹配点所应满足的性质即模板的形状，同时也指出这些点所不应满足的性质，即对周围环境背景的要求。击中/击不中变换可以用于保持拓扑结构的形状细化，以及形状识别和定位。设有一个模板形状(集合) A ，对给定的图像 X ，假定 X 中有包括 A 在内的多个不同物体。我们的目的是识别和定位其中的 A 物体。此时，取一个比 A 稍大的集合 S 作为结构元素并且使得 A 不与 S 的边缘相交，令 $S_1=A$ 且 $S_2=S-A$ (S_2 为“包围” A 或 S_1 的外接边框，此时称 S_1 、 S_2 为一个结构元素对，简称结构对，记为 (S_1, S_2))，那么 $X \odot (S_1, S_2)$ 将给出且仅给出所有 X 中与 A 误差在设定范围内物体的位置。图8-18描述了一个字符识别的示例。

击中或击不中变换



(a) 结构元素 $S = S_1 \cup S_2$

(b) 图像 X

(c) $X \odot (S_1, S_2)$

用击中/击不中变换识别字符

形态学图像处理

本章要点

- 概述
- 集合论基础知识
- 膨胀和腐蚀
- 开操作和闭操作
- 击中或击不中变换
- 形态学的主要应用
- 灰度级图像扩展

形态学的主要应用

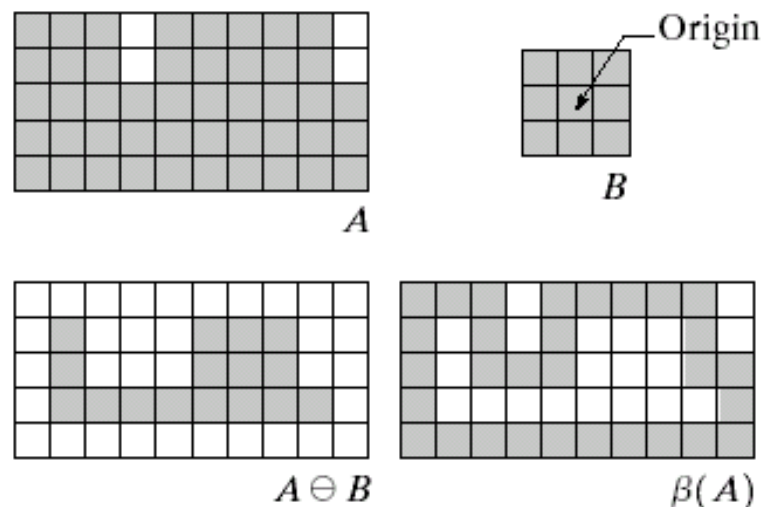
□ 边界提取

定义: $\beta(A) = A - (A \ominus B)$

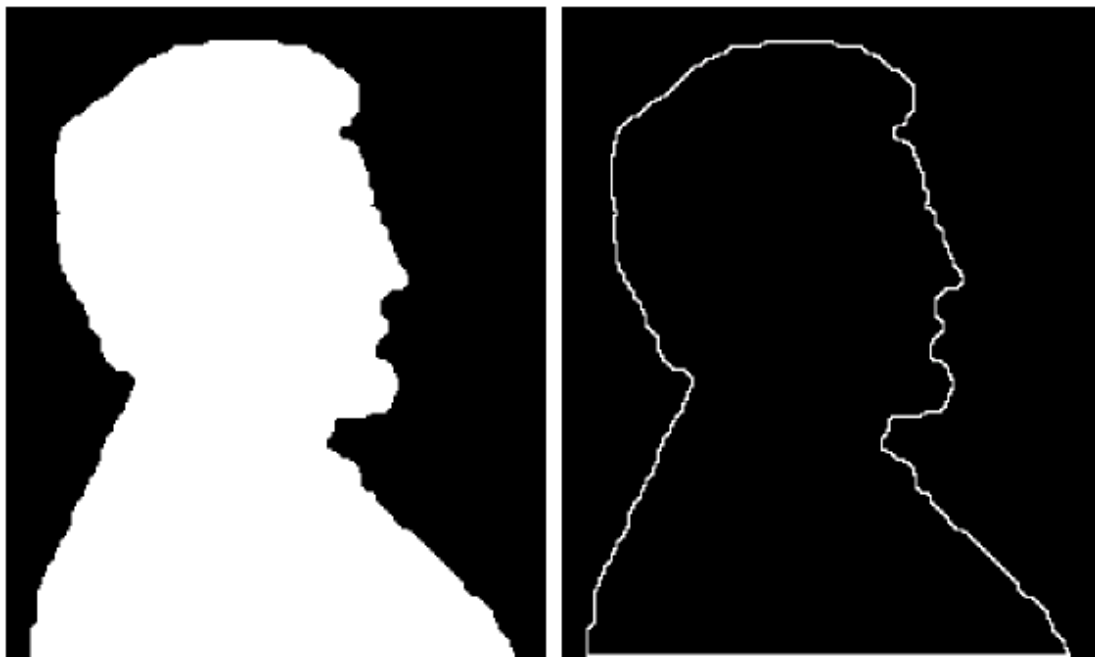
先用B对A腐蚀，然后用A减去腐蚀得到，
B是结构元素

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.



形态学的主要应用

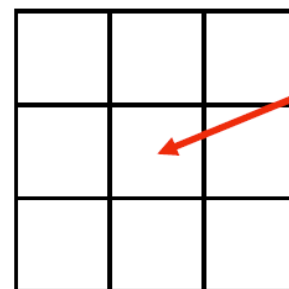


1表示为白色，0表示为黑色

a b

FIGURE 9.14

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



原点

结构元素B

形态学的主要应用

□ 区域填充

- 定义:设所有非边界(背景)点标记为0, 则将1赋给 p 点开始,

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

- 实现目的: 从边界内的一个点开始, 用1填充整个区域
- $X_0=p$, 如果 $X_k=X_{k-1}$, 则算法在迭代的第 k 步结束。 X_k 和 A 的并集包含被填充的集合和它的边界
- 条件膨胀: 如果对上述公式的左部不加限制, 则上述公式的膨胀将填充整个区域。利用 A^c 的交集将结果限制在感兴趣区域内, 实现条件膨胀

形态学的主要应用

a	b	c
d	e	f
g	h	i

FIGURE 9.15

Region filling.

(a) Set A .

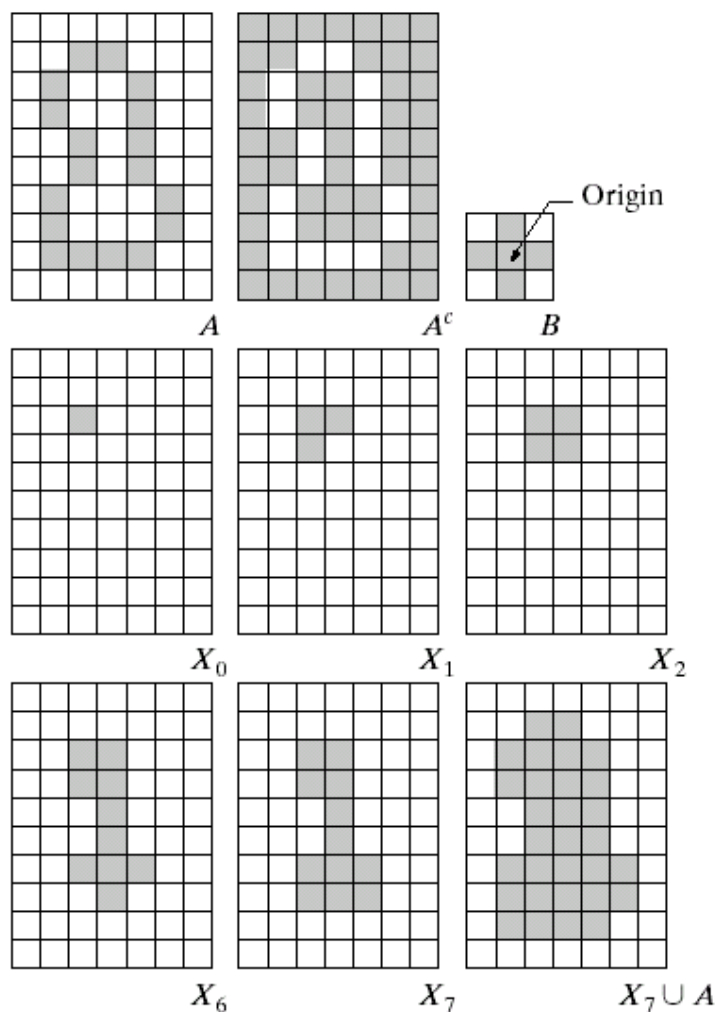
(b) Complement of A .

(c) Structuring element B .

(d) Initial point inside the boundary.

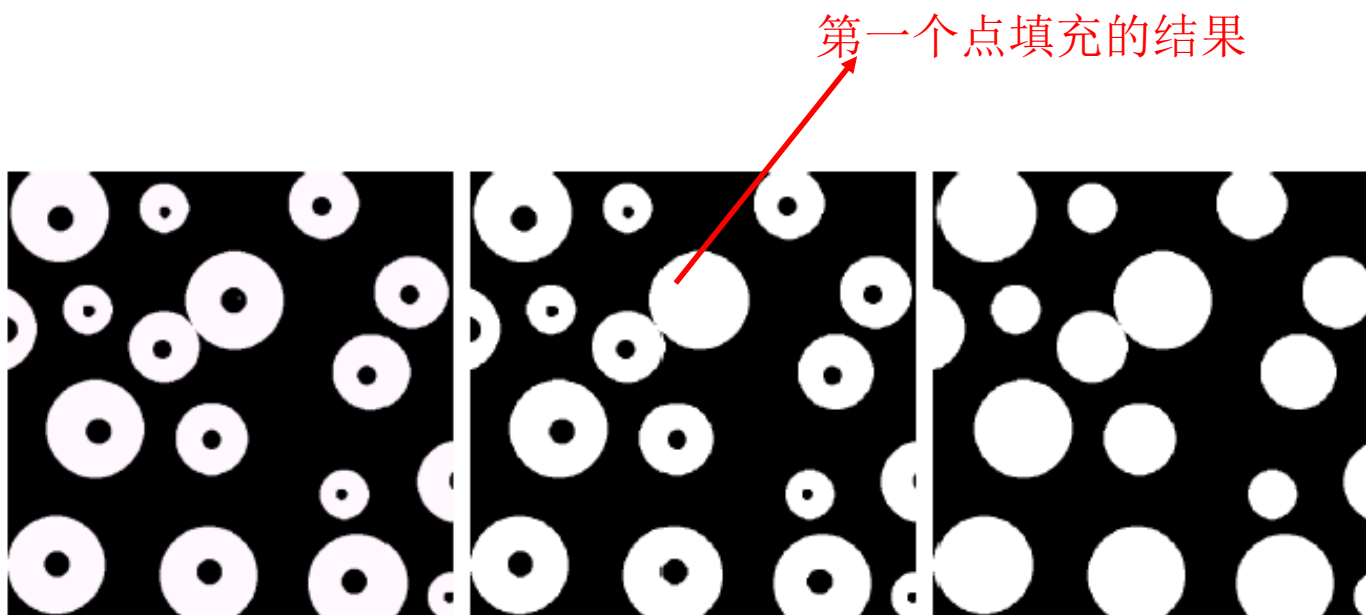
(e)–(h) Various steps of Eq. (9.5-2).

(i) Final result [union of (a) and (h)].



形态学的主要应用

- 通过区域填充消除白色圆圈内的黑点



a b c

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

形态学的主要应用

□ 连通分量的提取

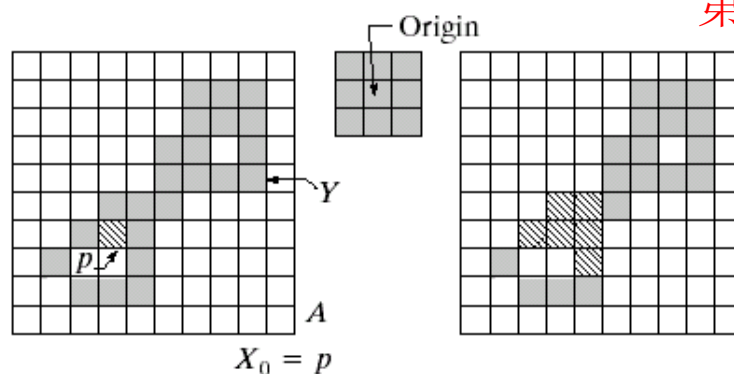
实现目的：在二值图像中提取连通分量

- 令 Y 表示一个包含于集合 A 中的连通分量，并假设 Y 中的一个点 p 是已知的。用下列迭代式生成 Y 的所有元素：
- $X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \dots$
- $x_0 = p$, 如果 $X_k = X_{k-1}$ ，算法收敛，令 $Y = X_k$

形态学的主要应用

8连通的结构元素

第一次迭代的结果



第二次迭代的结果

最终结果

a b c
d e

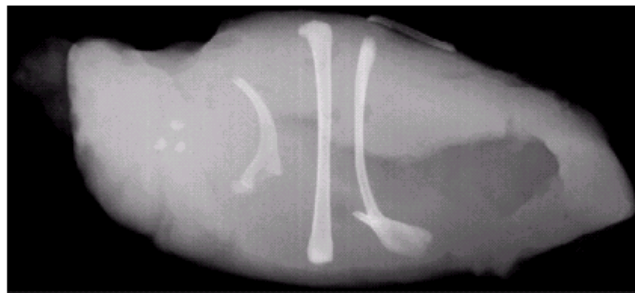
FIGURE 9.17 (a) Set A showing initial point p (all shaded points are valued 1, but are shown different from p to indicate that they have not yet been found by the algorithm). (b) Structuring element. (c) Result of first iterative step. (d) Result of second step. (e) Final result.

形态学的主要应用

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 1's.
(d) Number of pixels in the connected components of (c). (Image courtesy of NTB Elektronische Geraete GmbH, Diepholz, Germany, www.ntbxray.com.)



含有碎骨的鸡胸X光图像



使用阈值将骨头从背景中提取出来



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

提取连通分量，识别大尺寸对象，其中4个具有最大尺寸

消除细节，对阈值处理后的图像进行腐蚀，保留大尺寸物体

形态学的主要应用

□ 凸壳(are useful for object description)

- 如果连接集合A内任意两个点的直线段都在A的内部，则A是凸形的
- 集合S的凸壳H是包含S的最小凸集合。H-S称为S的convex deficiency
- 求取集合A的凸壳C(A)的简单形态学算法：

令 B_i 表示4个结构元素, $i=1,2,3,4$,

$$X_k^i = (X_{k-1} \oplus B^i) \cup A \quad i=1,2,3,4 \quad k=1,2,3,4,\dots$$

$X_0^i = A$ 令 $D^i = X_{conv}^i$, conv表示在 $X_k^i = X_{k-1}^i$ 收敛, 则

形态学的主要应用

□ 凸壳

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

- 先对A用 B_1 运用击中或击不中变换，反复使用，当不再发生变化时，执行与A的并集运算，用 D_1 表示结果
- 上述过程用 B_2 重复，直到不发生变化。
- 最后得到的4个 D 的并集组成了A的凸壳

形态学的主要应用

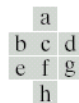
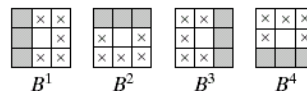
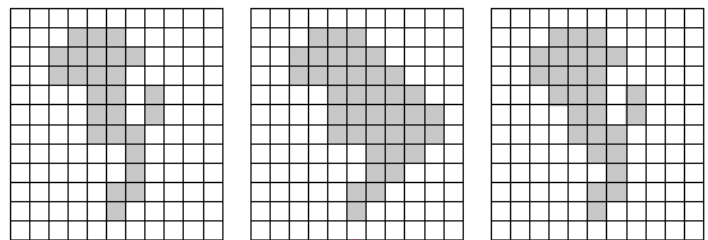


FIGURE 9.19
(a) Structuring elements. (b) Set A . (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull. (h) Convex hull showing the contribution of each structuring element.



图a



图b

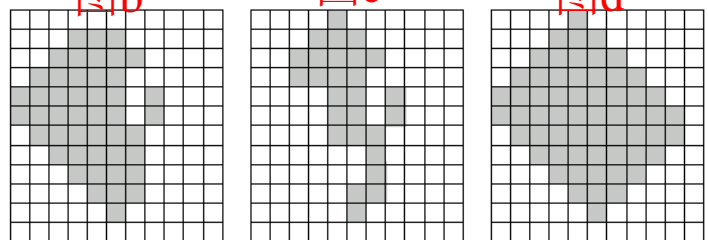
$X_0^1 = A$

图c

X_4^1

图d

X_2^2



图e

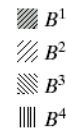
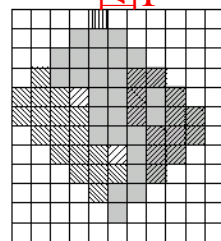
X_8^3

图f

X_2^4

图g

$C(A)$



图c-f是用图a中的结构元素

图h显示4个结构元素的属性

形态学的主要应用

- 上述过程的一个明显缺点是：凸壳可能超出确保凸性所需的最小尺寸
- 解决办法：限制水平和垂直方向上的尺寸大小，如下图所示。
- 也可限制水平、垂直和对角线方向上的最大尺寸。缺点是增加了算法的复杂性

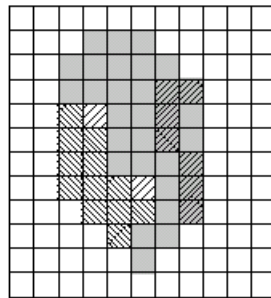


FIGURE 9.20 Result of limiting growth of convex hull algorithm to the maximum dimensions of the original set of points along the vertical and horizontal directions.

形态学的主要应用

□ 细化

- 细化过程根据击中或击不中变换定义

$$A \otimes B = A - (A \oplus B) = A \cap (A \oplus B)^c$$

定义结构元素序列为

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

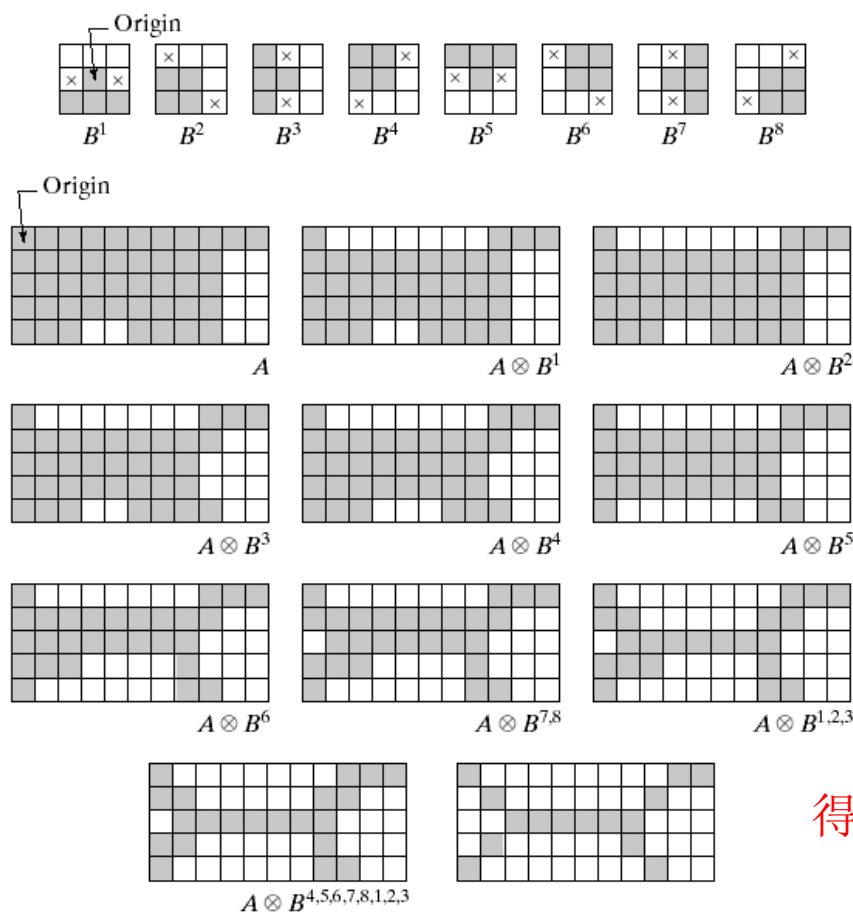
B^i 是 B^{i-1} 旋转后的形式，如在 B^4 中旋转 90°

- 用结构元素序列定义细化为

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

- 即连续使用 B^1, \dots, B^n 对A细化

形态学的主要应用



得到转化为 m 连通的细化

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

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 element again (there were no changes for the next two elements). (k) Result after convergence. (l) Conversion to m -connectivity.

形态学的主要应用

□ 粗化

- 粗化和细化在形态学上是对偶过程，定义为

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

- 用结构元素序列定义粗化为

$$A \odot \{B\} = \left(\left(\dots \left(\left(A \odot B^1 \right) \odot B^2 \right) \dots \right) \odot B^n \right)$$

- 即连续使用 B^1, B^2, \dots, B^n 对 A 粗化

形态学的主要应用

- 粗化可以通过细化算法求补集实现：先对所讨论集合的背景进行细化，然后对结果求补集，为了对集合A进行粗化，先令 $C=A^c$ ，然后对C进行细化，最后形成 C^c

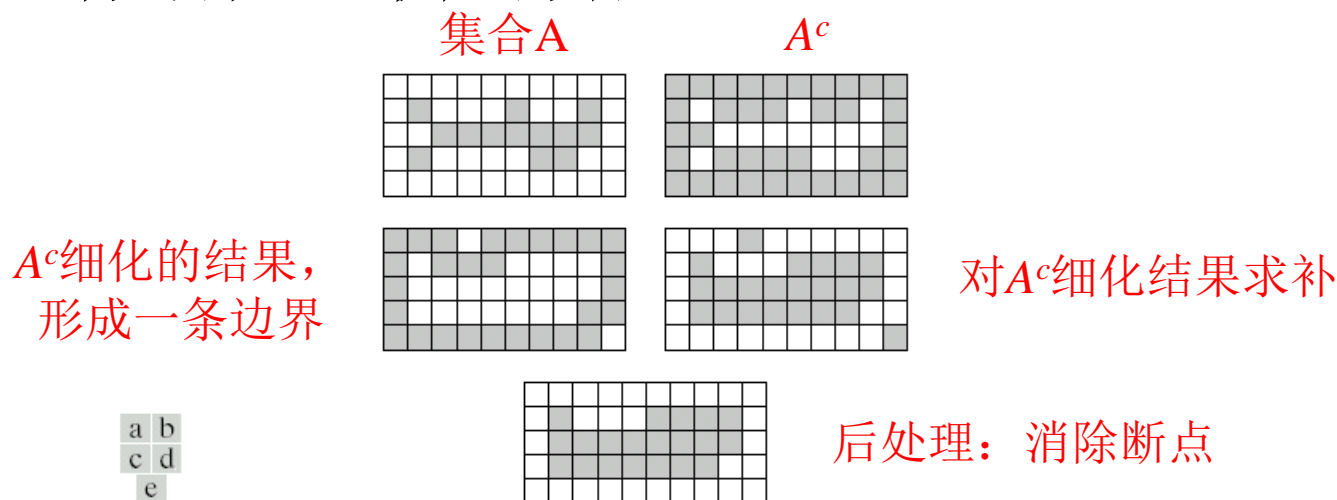


FIGURE 9.22 (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

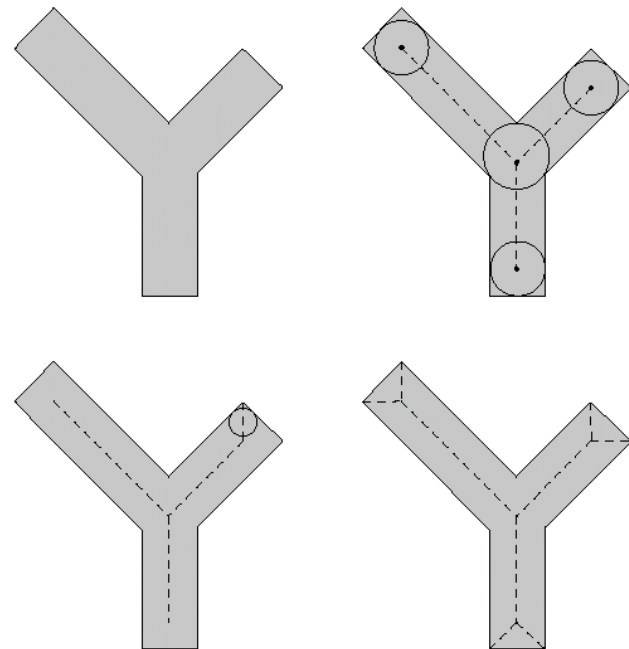
形态学的主要应用

□ 骨架

- 概念：骨架 $S(A)$ 可以从下图中直观看出，推断结果：

- 如果 z 是 $S(A)$ 的点并且 $(D)_z$ 是在 A 内以 z 为中心的最大的圆盘，则不存在位于 A 内的能包含 $(D)_z$ 的更大圆盘（不一定以 z 为中心）。圆盘 $(D)_z$ 就叫做最大盘。
- 圆盘 $(D)_z$ 在两个或更多的不同位置上与 A 的边界接触。

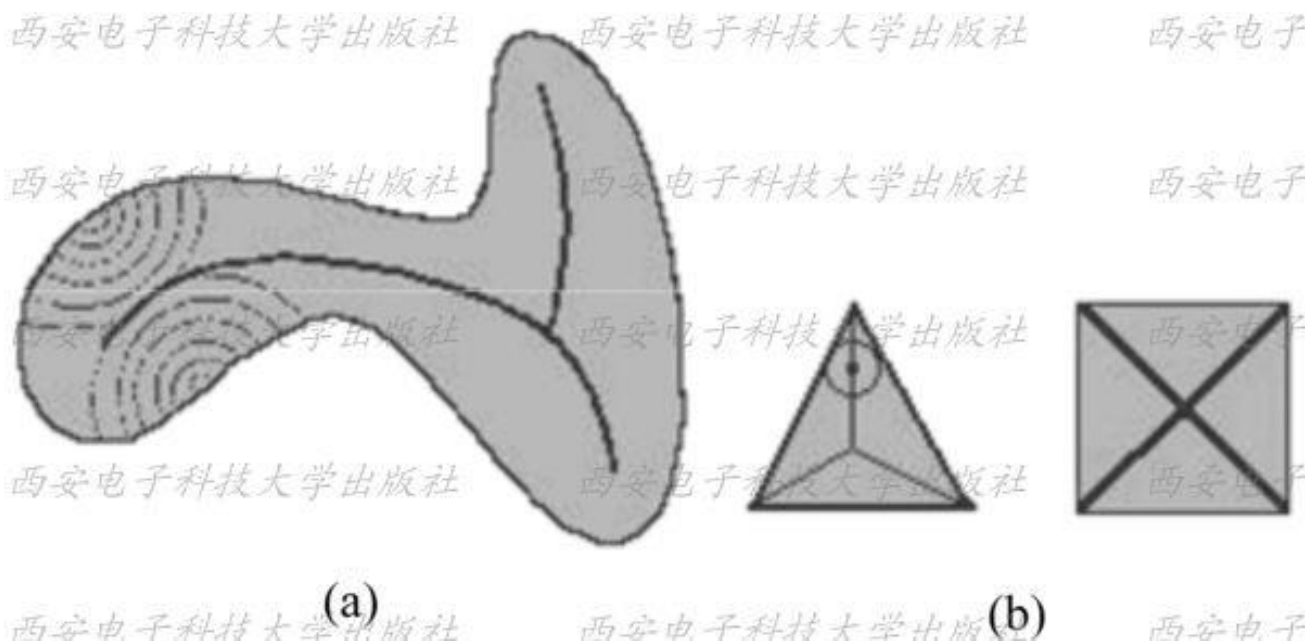
a b
c d
FIGURE 9.23
(a) Set A .
(b) Various positions of maximum disks with centers on the skeleton of A .
(c) Another maximum disk on a different segment of the skeleton of A .
(d) Complete skeleton.



形态学的主要应用

□ 骨架

骨架还可以用中轴表示。设想在 $t = 0$ 时刻，将目标边界各处同时点燃，火的前沿以匀速向目标内部蔓延，当前沿相交时火焰熄灭，火焰熄灭点的集合就构成了中轴。图8-26 (a)是这个过程的图示。



形态学的主要应用

□ 骨架

- A 的骨架可用腐蚀和开操作表达 $S(A) = \bigcup_{k=0}^K S_k(A)$

$$S_k(A) = (A \ominus k B) - (A \ominus k B) \circ B$$

- 这里 B 是一个结构元素，表示对 A 的连续 k 次腐蚀：

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

- 第 k 次是 A 被腐蚀为空集合前进行的最后一次迭代：

$$K = \max \{ k \mid (A \ominus k B) \neq \emptyset \}$$

形态学的主要应用

□ 裁剪

- 实现目的：裁剪方法实际上是对细化和骨架绘制算法的补充，因为要清除这些算法产生的一些不必要的附加成分。
- 应用实例：自动手写字符识别，分析每种字符的骨架形状。

形态学的主要应用

- 由于在字符骨架中经常有各种“毛刺”存在，毛刺是在腐蚀过程中由于构成字符的笔画不均匀造成的。图示如下：我们要消除字符“a”左边的寄生部分。假定寄生成分的长度都小于三个像素。
- 过程：
 - (1) 用一系列被设计用来检测终点的结构元素对A进行细化；{B}为结构元素序列，每种结构元素都全部8个像素进行90度的旋转。

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

形态学的主要应用

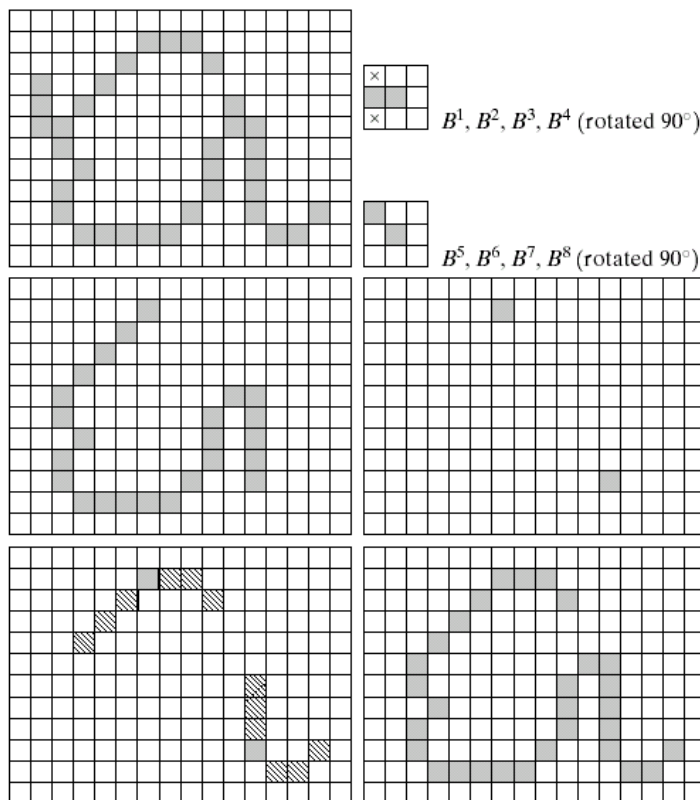
- (2) 得到建立在 X_1 中,包含的所有终点的集合 X_2 。
图中(e)所示

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

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.



形态学的主要应用

- (3) 对端点进行三次膨胀处理，用集合A作为消减因子；

$$X_3 = (X_2 \oplus H) \cap A$$

H 是元素值为1的 3×3 的结构元素; 图 (f) 为显示的结果

- (4) X_3 和 X_1 的并集

$$X_4 = X_3 \cup X_1$$

形态学图像处理

本章要点

- 概述
- 集合论基础知识
- 膨胀和腐蚀
- 开操作和闭操作
- 击中或击不中变换
- 形态学的主要应用
- 灰度级图像扩展

灰度级图像扩展

- 前面所讲的形态学方法都是基于二值图像的，下面我们要把形态学处理扩展到灰度图像的基本操作。
- 设 $f(x,y)$ 为输入图像，而 $b(x,y)$ 为结构元素。

灰度级图像扩展

□ 膨胀

■ 定义：用 b 对函数 f 进行的灰度膨胀表示为 $f \oplus b$

$$(f \oplus b)(s,t) = \max\{f(s-x, t-y) + b(x,y) \mid (s-x, t-y) \in D_f; (x,y) \in D_b\}$$

其中， D_f 和 D_b 分别是 f 和 b 的定义域。注意， f 和 b 是函数而不是二值形态学情况中的集合。

一维说明：用一维函数说明上式的表示法和运算原理。

$$(f \oplus b)(s) = \max\{f(s-x) + b(x) \mid (s-x) \in D_f; (x,y) \in D_b\}$$

□ 在此式中， $f(-x)$ 是 $f(x)$ 关于 x 轴的镜像。当 s 为正时函数 $f(s-x)$ 向右移动，为负则向左移动。而且 f 和 b 是彼此交叠的。我们也可以把 b 看作滑过 f 的函数。

灰度级图像扩展

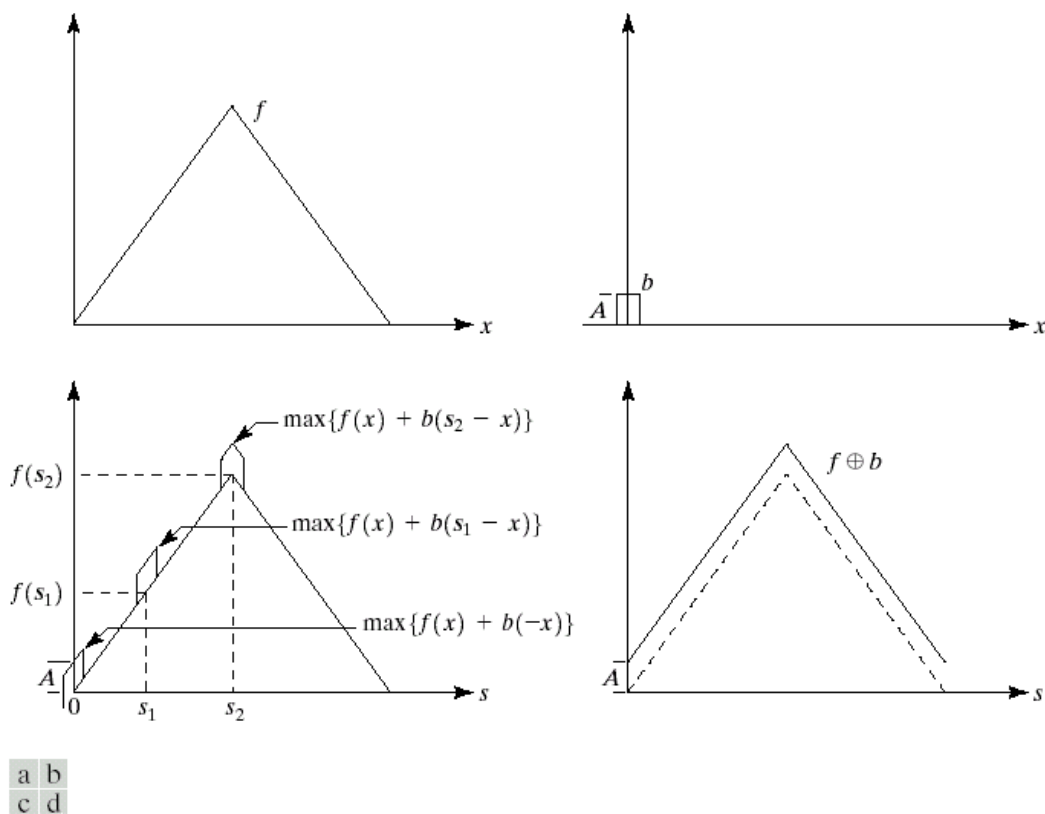


FIGURE 9.27 (a) A simple function. (b) Structuring element of height A . (c) Result of dilation for various positions of sliding b past f . (d) Complete result of dilation (shown solid).

(a)一个简单的函数，(b)高度A的结构元素，(c)b滑过f的不同位置进行膨胀的结果，(d)膨胀得到的完整结果（以实线表示）

灰度级图像扩展

- 从图中我们可以看出，在每个结构元素的位置上，这一点的膨胀值是在跨度为 b 的区间内 f 与 b 之和的最大值。
- 结果：通常对灰度图像进行膨胀处理的结果是双重的。
 - 若所有结构元素的值为正，则输出图像会趋向于比输入图像更亮；
 - 暗的细节部分全部减少了还是被消除了,取决于膨胀所用的结构元素的值和形状。

灰度级图像扩展

- 腐蚀：灰度腐蚀，表示为 $f \ominus b$ ，定义为：

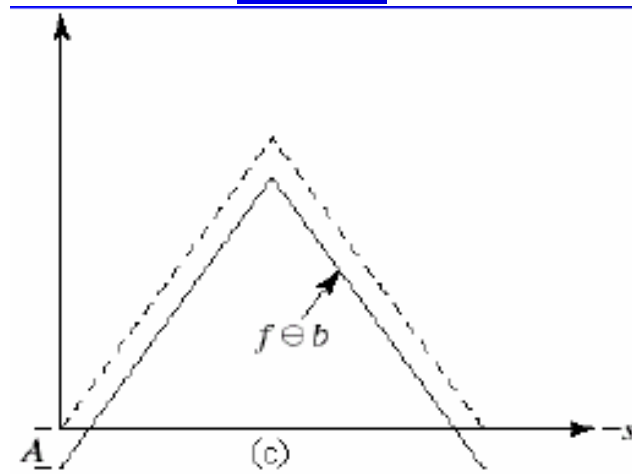
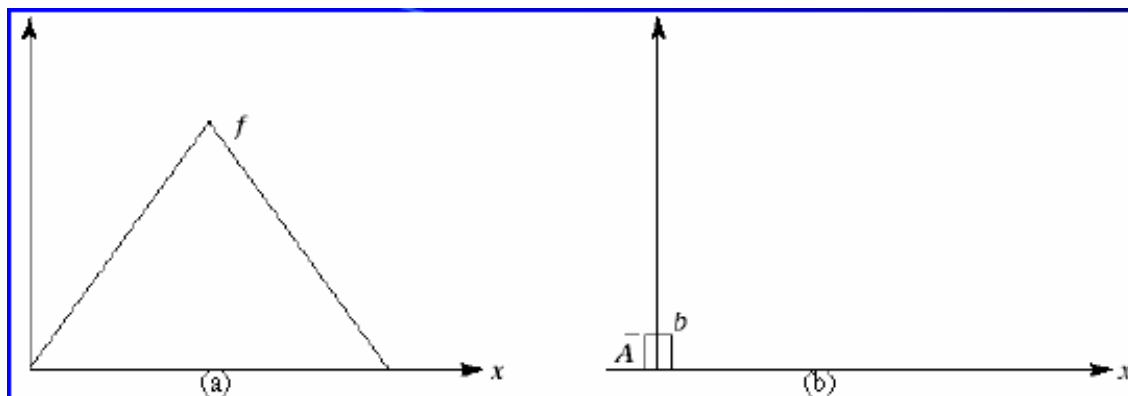
$$(f \ominus b)(s, t) = \min \{ f(s + x, t + y) - b(x, y) \mid (s + x, t + y) \in D_f; (x, y) \in D_b \}$$

其中， D_f 和 D_b 分别是 f 和 b 的定义域

- 一维说明：用一维函数说明上式的表示法和运算原理

$$(f \ominus b)(s) = \min \{ f(s + x) - b(x) \mid (s + x) \in D_f, x \in D_b \}$$

灰度级图像扩展



使用(b)中显示的结构元素对(a)中显示的函数进行腐蚀得到(c)

灰度级图像扩展

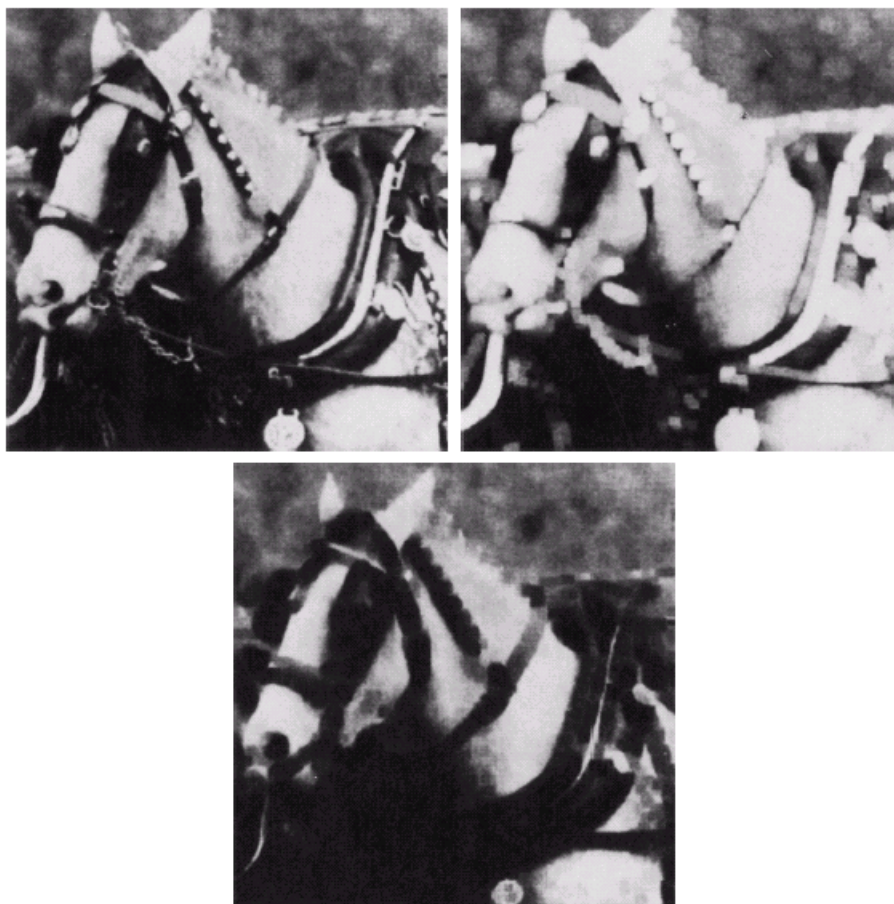
- 腐蚀操作是以结构元素形状定义的区间中选取 $(f-b)$ 最小值为基础的。结果：通常对灰度图像进行腐蚀处理的结果是双重的。
 - 若所有结构元素的值为正，则输出图像会趋向于比输入图像更暗；
 - 在输入图像中亮的细节的面积如果比结构元素的面积小，则亮的效果将被削弱。削弱的程度取决于环绕于亮细节周围的灰度值和结构元素自身的形状和幅值。

灰度级图像扩展

- 膨胀和腐蚀的比较：对偶关系

$$(f \ominus b)^c(s, t) = (f^c \oplus \hat{b})(s, t)$$

灰度级图像扩展



a b
c

FIGURE 9.29

(a) Original image. (b) Result of dilation.

(c) Result of erosion.

(Courtesy of Mr. A. Morris, Leica Cambridge, Ltd.)

图解：灰度图像的膨胀和腐蚀的说明，其中：(a)表示一幅 512×512 大小的灰度图像；(b)表示用“平顶”元素对图像进行膨胀的结果；(c)表示对原图像进行腐蚀的



THANKS!

Any Questions??