

《数字图像处理》

Chapter 9

形态学图像处理 (morphology)

中山大学计算机学院

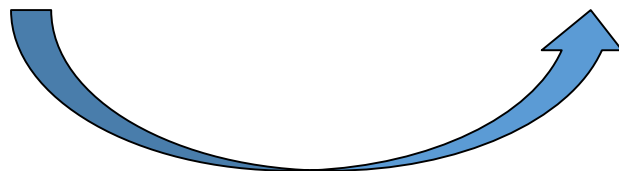
谢晓华

如何获得以下图像处理效果？

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

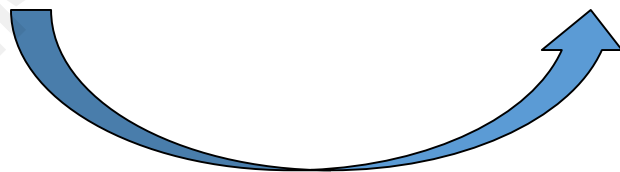


Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



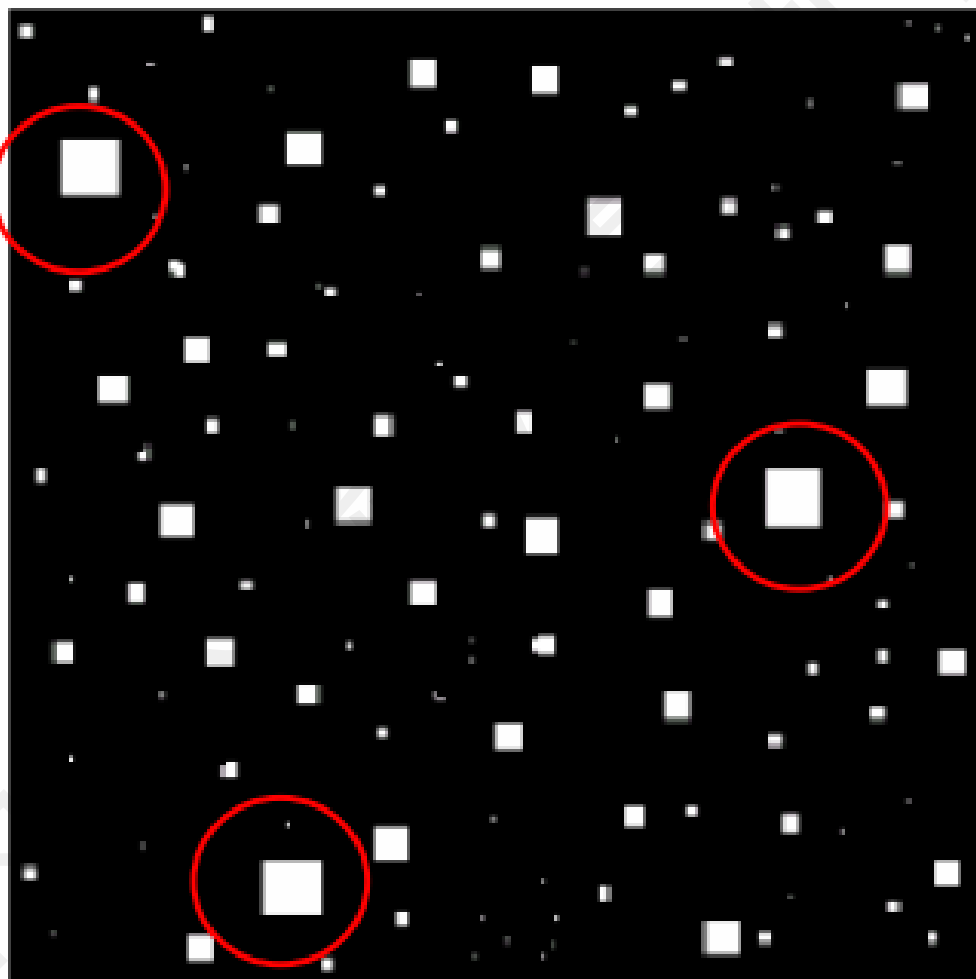
修补断裂

如何获得以下图像处理效果？



去噪但保护纹理

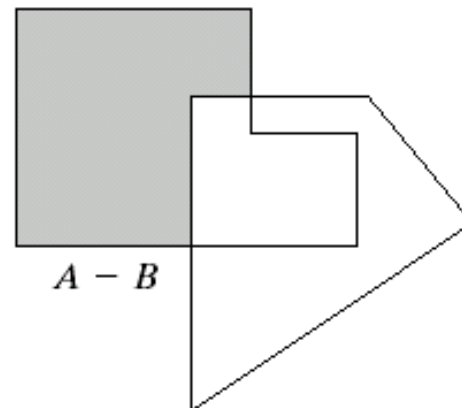
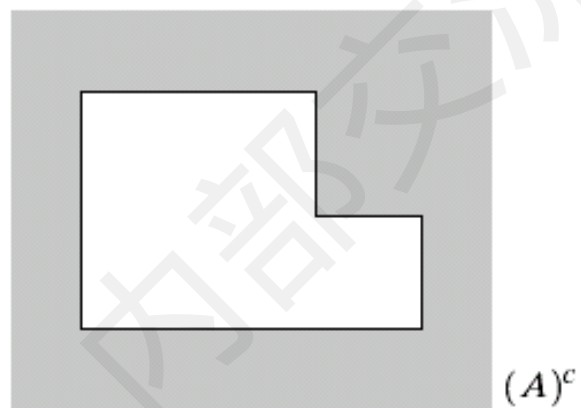
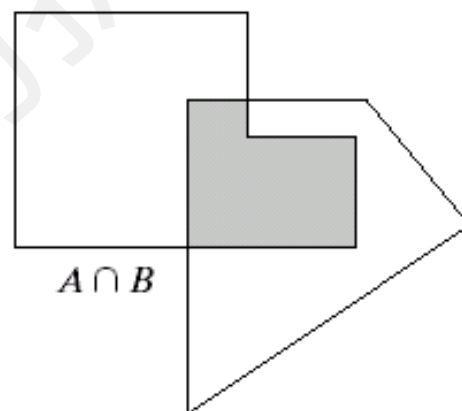
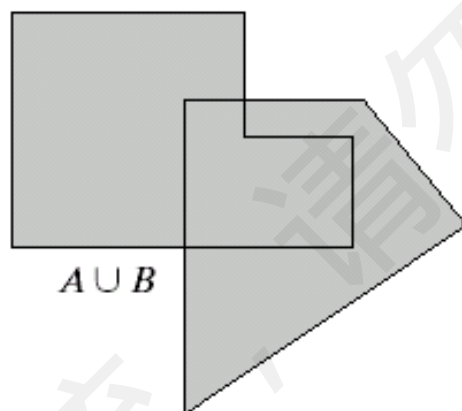
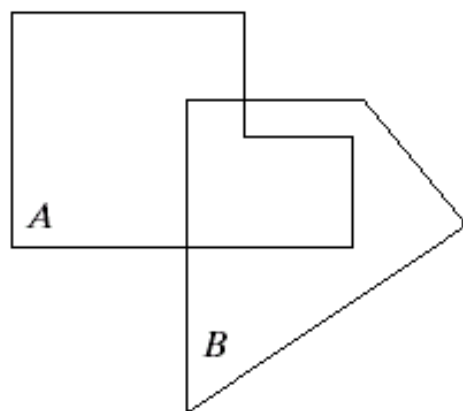
如何从图像中提取感兴趣的内容？



Mathematical morphology

- 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。(注：原本的形态学为生物学分支，本书用之为数学形态学的简称)。
- 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目的。
- 形态学图像处理的数学基础和所用语言是集合论。
- 一种简单的非线性代数算子。
- 主要用于二值图像，可扩展到灰度图像。

数学背景：集合



二值图像的逻辑运算

对于一幅二值图像，黑色表示1，白色表示0，有三个基本的逻辑运算。

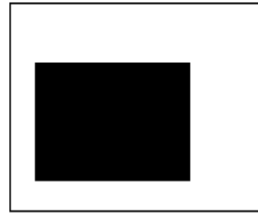
p	q	$p \text{ AND } q$ (also $p \cdot q$)	$p \text{ OR } q$ (also $p + q$)	NOT (p) (also \bar{p})
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

由这三种基本的逻辑运算可以构造其他的逻辑运算,如:

- A和B**异或**运算: 当两个像素的值不同时为1,否则为0.
- A和B**非与**运算: 当像素在B中而不在A中则为1,其他为0.

集合运算与逻辑运算有一一对应的关系.

A

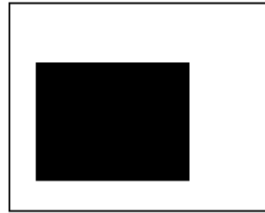


$\text{NOT}(A)$

NOT
→



A

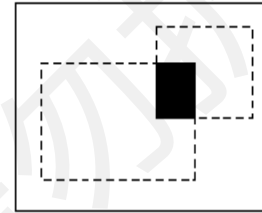


B



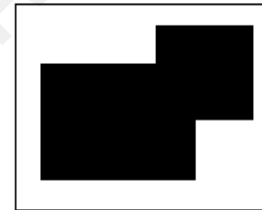
$(A) \text{ AND } (B)$

AND
→



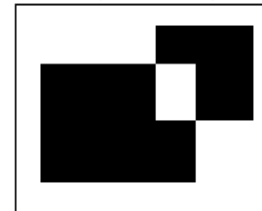
$(A) \text{ OR } (B)$

OR
→



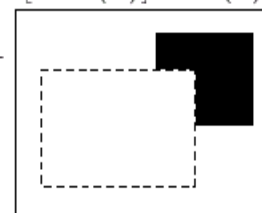
$(A) \text{ XOR } (B)$

XOR
→



$[\text{NOT}(A)] \text{ AND } (B)$

NOT-AND
→



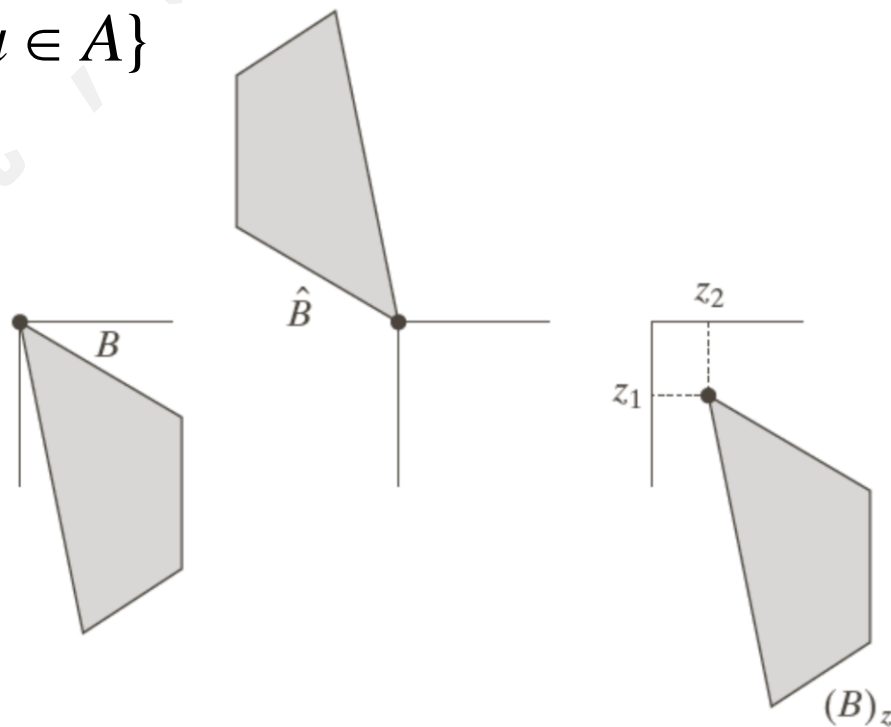
附加的运算定义：

集合的**反射**，表示为 \hat{B} 定义为：

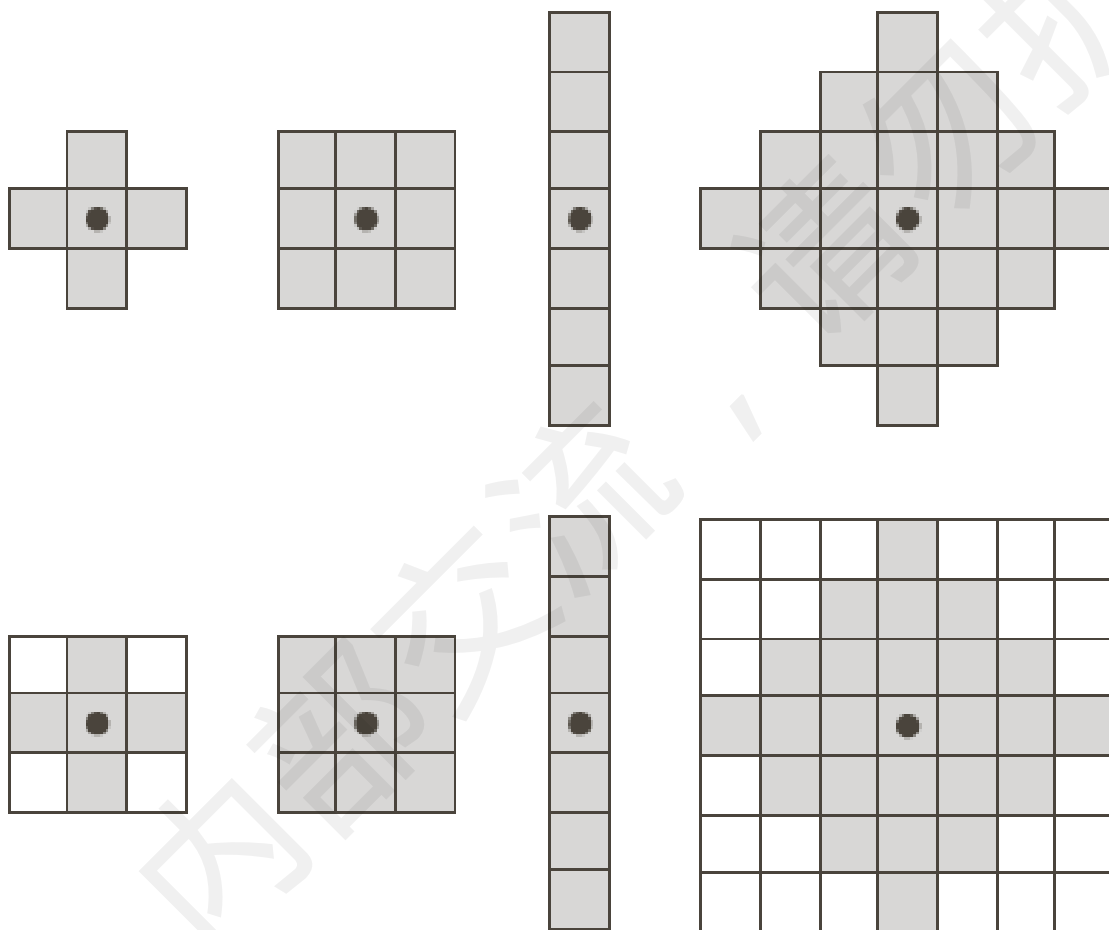
$$\hat{B} = \{w \mid w = -b, b \in B\}$$

集合A**平移**到点 $z = (z_1, z_2)$,表示为 $(A)_z$, 定义为：

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



一个重要的概念：结构元

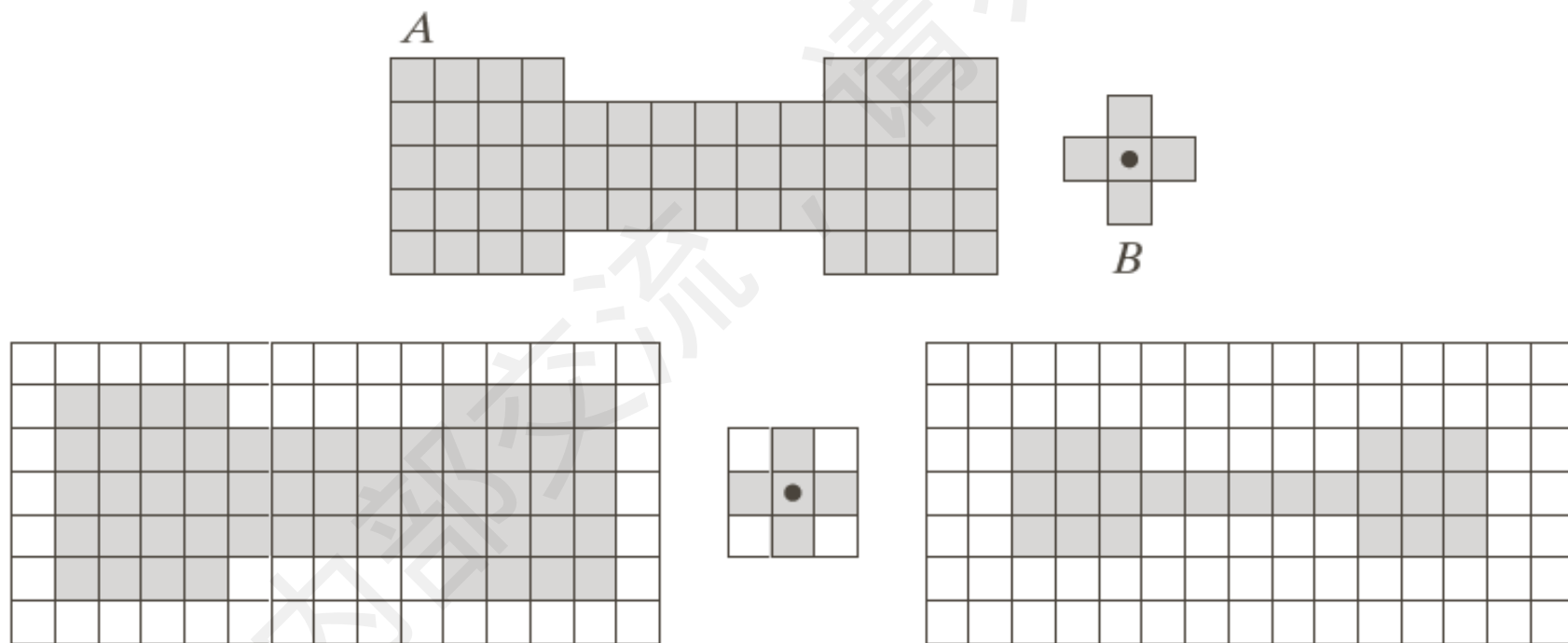


四种结构元
(黑点为其中心)

四种结构元对应的
矩阵形式

基于结构元的图像处理例子

定义操作： 在结构元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成员。



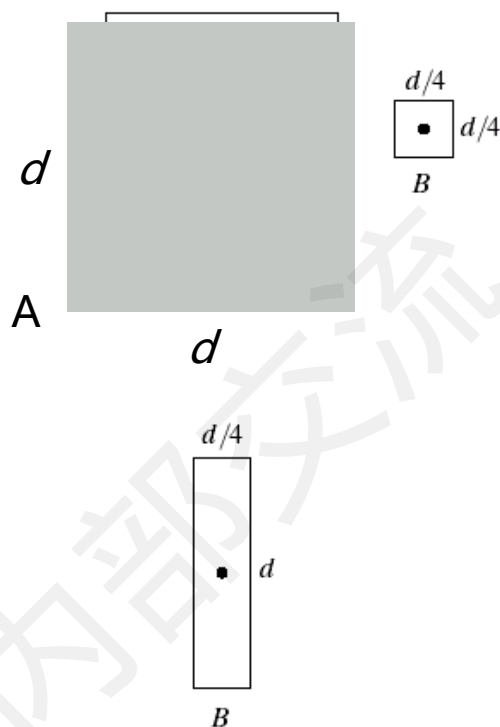
最终处理结果

如何用数学语言进行描述？

腐蚀: 假设B为结构元素.

A被B腐蚀定义为: $A \ominus B = \{z \mid (B)_z \subseteq A\}$
(定义不唯一) $= \{z \mid (B)_z \cap A^c = \phi\}$

注意变量为 z , 实际
为B移动的中心集合。



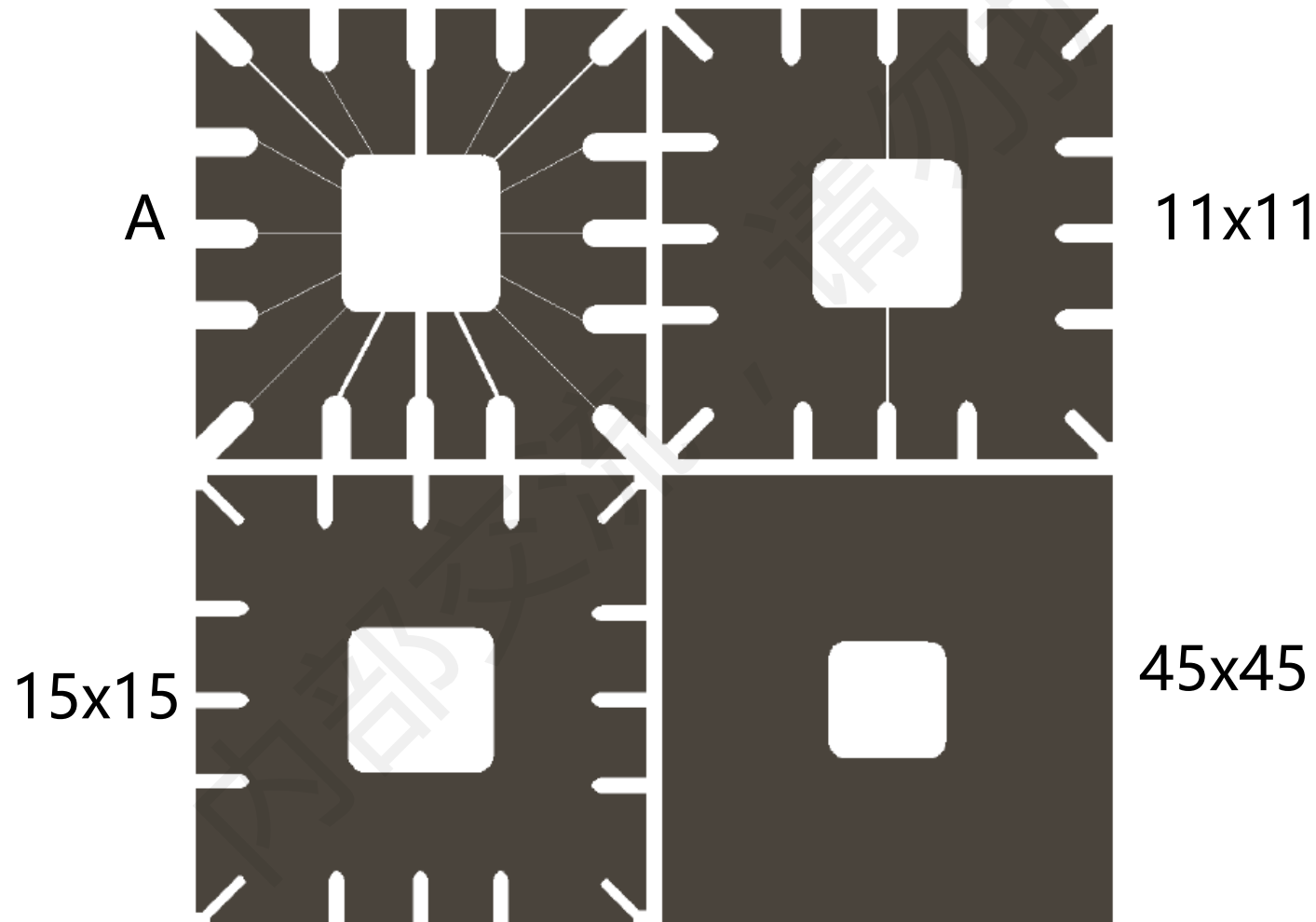
采用两种结构元B对A进行腐蚀的结果

提醒：

书上例子（包括本PPT）对集合元素的表示颜色并不一致，灰、白、黑均有可能，要仔细看。

腐蚀处理例子（白色表示元素集合）

B分别为边长为11、15、45
的方形图像，元素值全为1.

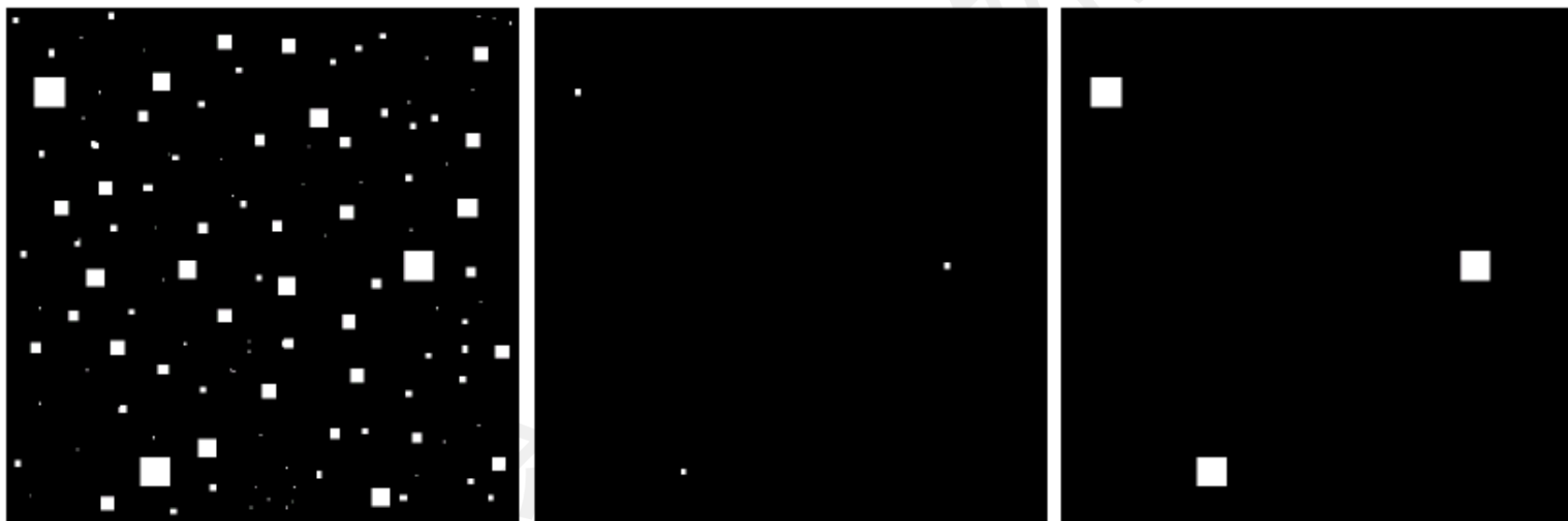


“腐蚀+膨胀”实现感兴趣结构提取（白色表示元素集合）：

原图

腐蚀

再膨胀



a b c

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.

膨胀如何定义？

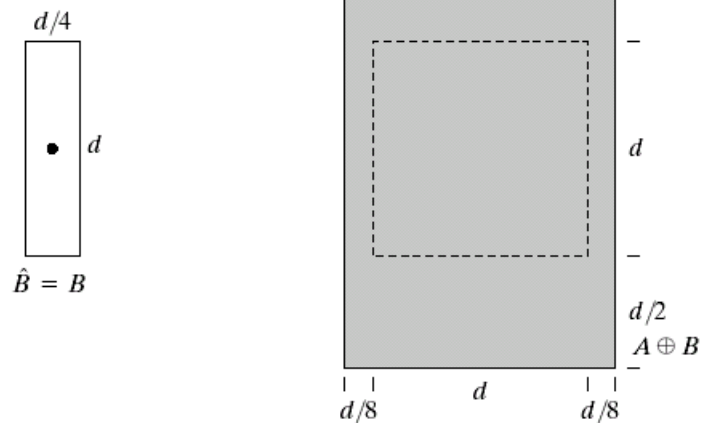
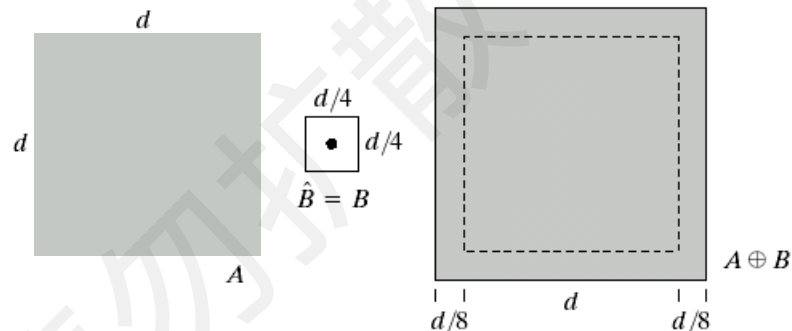
膨胀

假设B为结构元素.

A被B膨胀定义（不唯一）为:

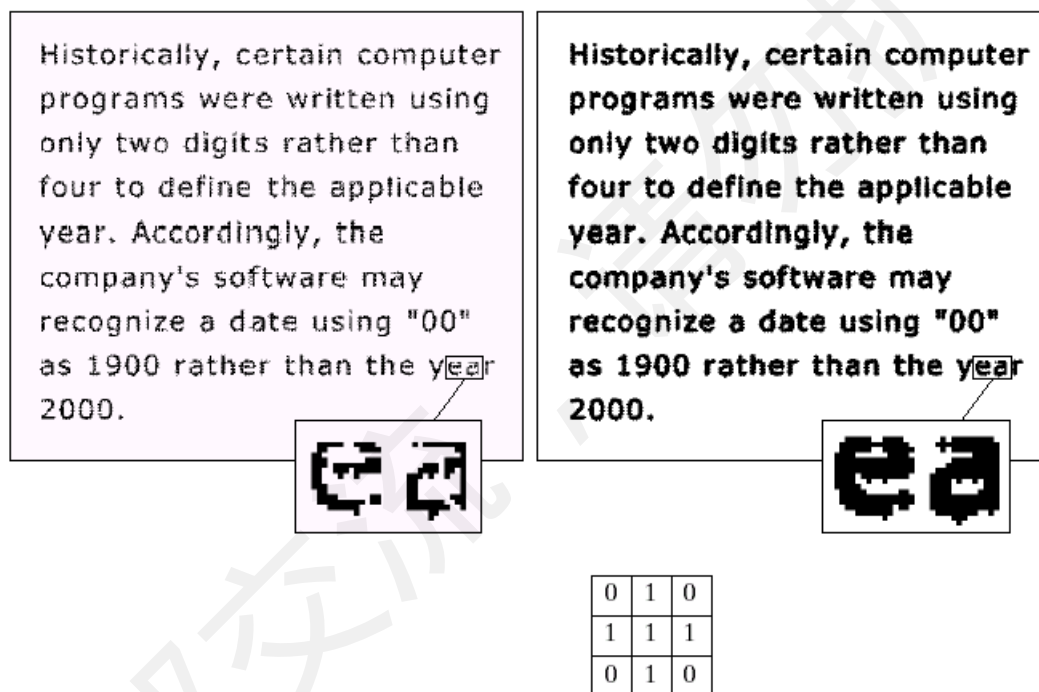
$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\}$$

$$= \{z \mid [(\hat{B})_z \cap A] \subseteq A\}$$



采用两种结构元B对
A进行膨胀的结果

例： 将裂缝桥接起来的形态学膨胀的应用



- 该结构元素能连接间断的最大长度为几个像素？ Re: 2
- 膨胀与均值滤波的异同。 Re: 变粗、边缘模糊

膨胀与腐蚀的关系：对偶性

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

证明：

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

如果集合 $(B)_z$ 包含于 A , 则等价于 $(B)_z \cap A^c = \phi$

于是： $(A \ominus B)^c = \{z \mid (B)_z \cap A^c = \phi\}^c = \{z \mid (B)_z \cap A^c \neq \phi\}$

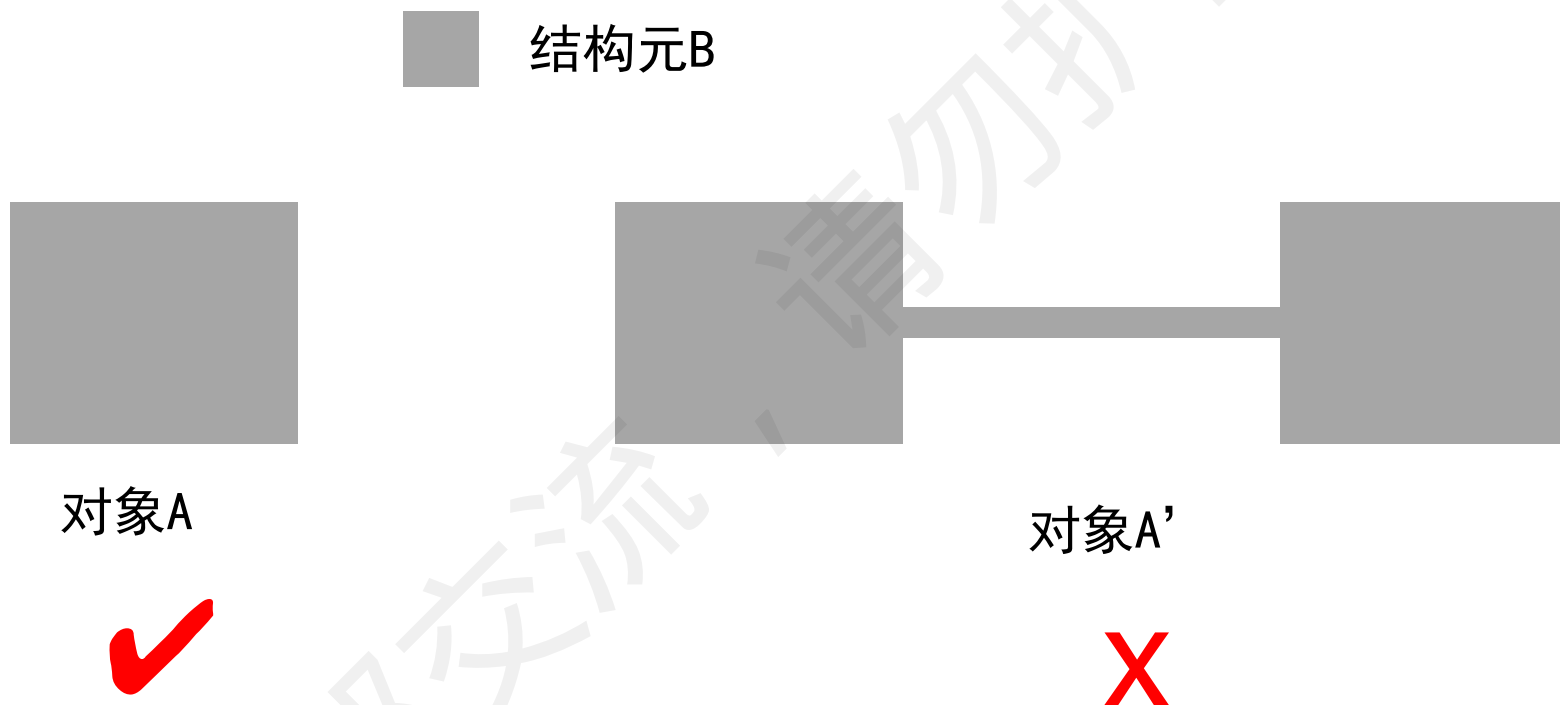
依定义： $A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \phi\}$

类似可定义： $A^c \oplus \hat{B} = \{z \mid (B)_z \cap A^c \neq \phi\}$

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

类似结论： $(A \oplus B)^c = A^c \ominus \hat{B}$

问题1： 对相同的结构元，膨胀与腐蚀是否为一对互逆操作？



问题2： “先膨胀再腐蚀” 是否等价于 “先腐蚀再膨胀” ？

三. 开操作与闭操作

使用结构元素**B**对集合**A**进行开操作的定义为：

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

即，使用结构元素**B**对集合**A**进行先腐蚀，然后膨胀。

使用结构元素**B**对集合**A**进行闭操作的定义为：

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

即，使用结构元素**B**对集合**A**进行先膨胀，然后腐蚀。

开操作结果的边界: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。(注意: 不是中心)

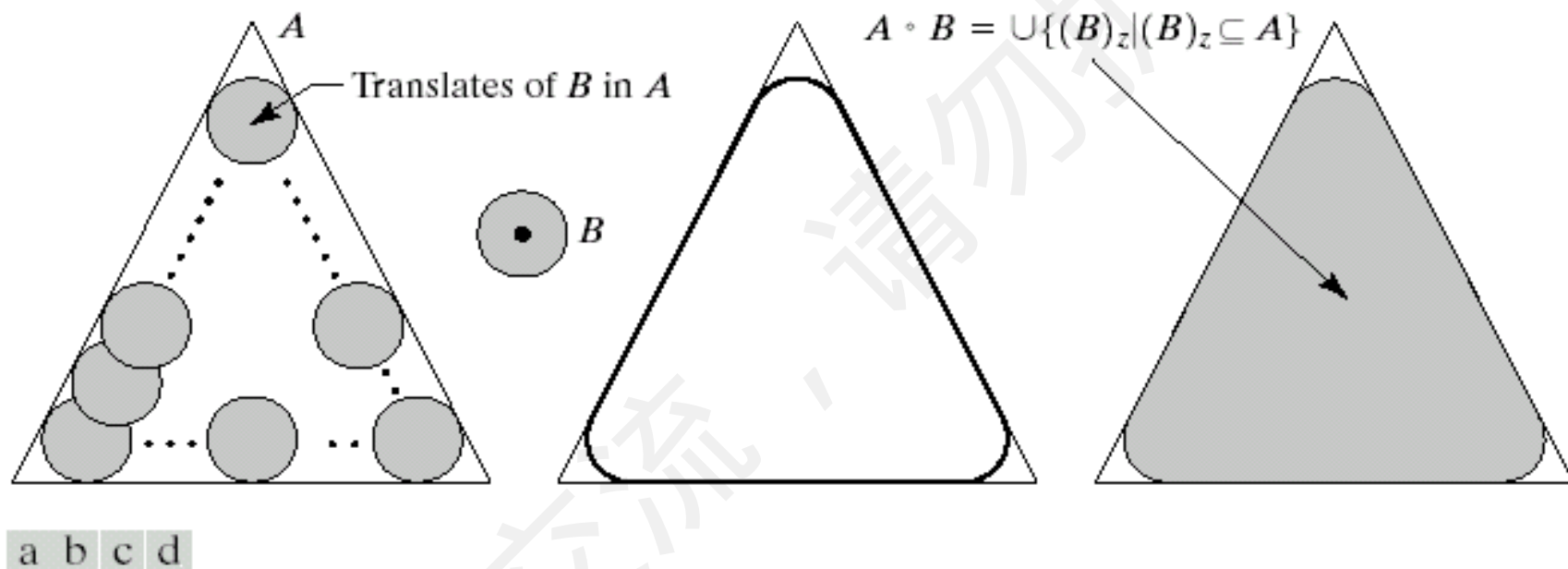


FIGURE 9.8 (a) Structuring element B “rolling” along the inner boundary of A (the dot indicates the origin of B). (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded).

- 开操作一般使对象的轮廓变得光滑, 断开狭窄的间断和消除细的突出物。

闭操作结果的边界:当B在A的边界外侧滚动时, **B**所能到达的**A的边界**的最远点。(注意: 不是中心)

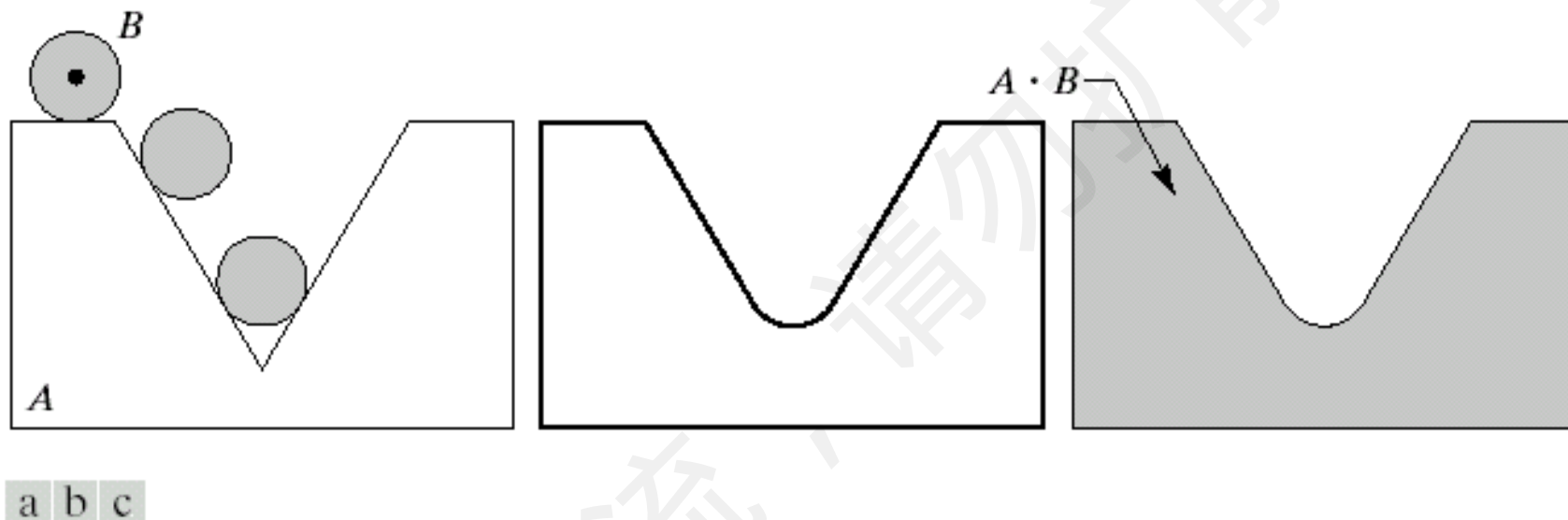
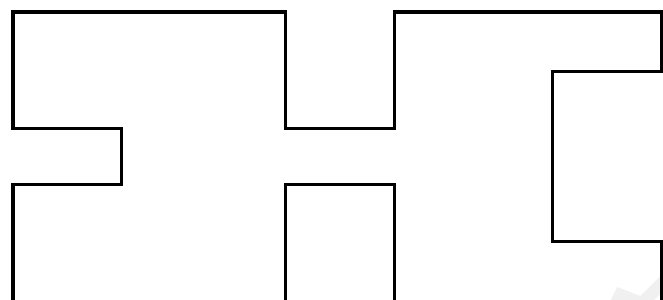
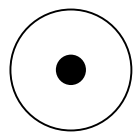


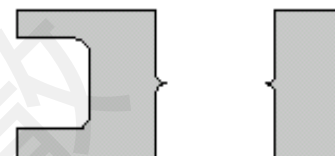
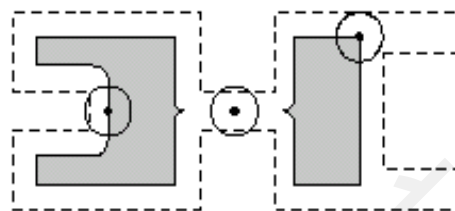
FIGURE 9.9 (a) Structuring element B “rolling” on the outer boundary of set A . (b) Heavy line is the outer boundary of the closing. (c) Complete closing (shaded).

- 闭操作使对象的更为连通, 它能消除小的孔洞, 并填补轮廓线中的断裂。

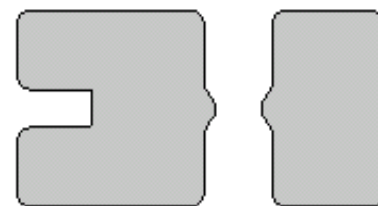
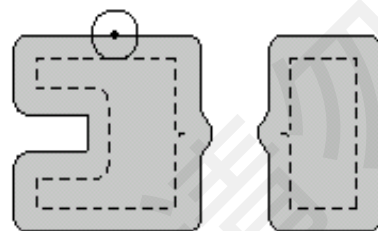
开操作与闭操作例子： 结构元为小圆



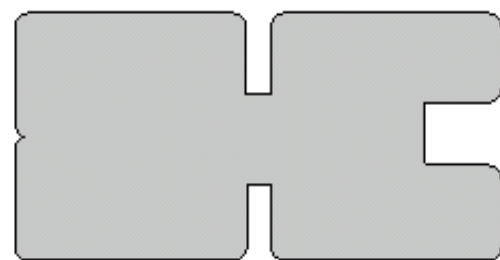
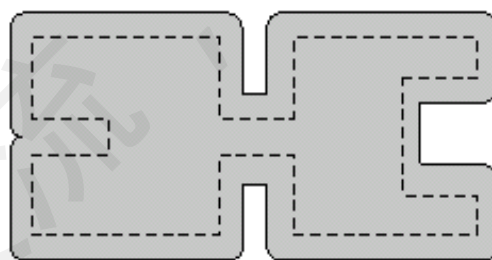
A



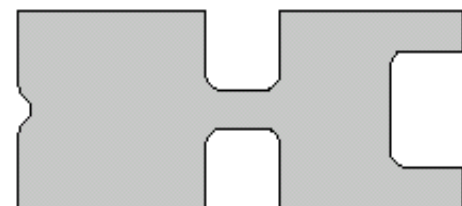
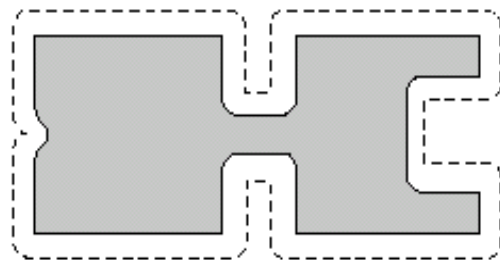
$A \ominus B$



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



$A \oplus B$



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

开操作与闭操作的性质：

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

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

开操作满足下列性质：

- (i) $A \circ B$ 是 A 的子集合（开操作后变小）。
- (ii) 如果 C 是 D 的子集合，则 $C \circ B$ 是 $D \circ B$ 的子集。
- (iii) $(A \circ B) \circ B = A \circ B$ （反复开等于一次开）

同样，闭操作也满足下列性质：

- (i) A 是 $A \bullet B$ 的子集（闭操作后变大）。
- (ii) 如果 C 是 D 的子集，则 $C \bullet B$ 是 $D \bullet B$ 的子集。
- (iii) $(A \bullet B) \bullet B = A \bullet B$ （反复闭等于一次闭）

问题：

开操作与闭操作是否为一对互逆操作？

开操作+闭操作实现指纹图像预处理

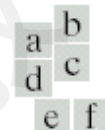


FIGURE 9.11

(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.)

四. 击中或击不中变换

➤ 击中或击不中变换是形状检测的基本工具.

➤ 令每种形状的重心为它的原点.

设形状D包含在一个小窗口W中, D的背景定义为W-D.
令B是由D和W-D组成的集合, 使用结构元素B对集合A进行匹配操作的定义为:

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

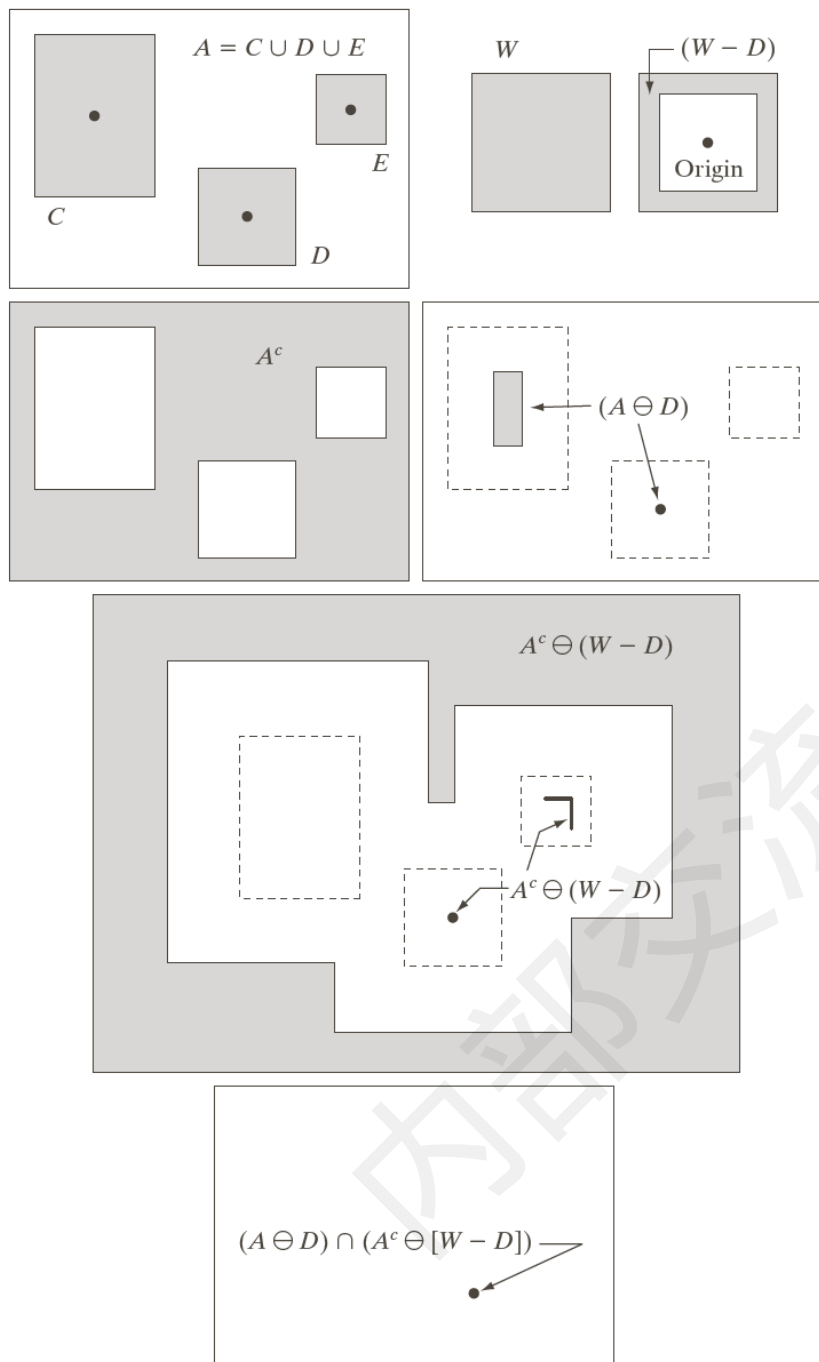
更一般地, 令 $B = (B_1, B_2)$, 使用结构元素B对集合A进行匹配操作的定义为:

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

B_1 、 B_2 分别与前景、背景对应, 联想腐蚀的定义。

由于 $A^c \oplus \hat{B} = (A \ominus B)^c$ 和 $A \cap B^c = A - B$

因此, $A \circledast B = (A \ominus B_1) - (A \oplus \hat{B}_2)$



击中或击中不中变换

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

找到了**D**在**A**中位置！

背景相当于以后学到的“上下文信息（**context**）”。

五. 一些基本的形态学算法

1. 介绍提取图像的边界、连通分量、凸壳和区域骨架的形态学算法；
2. 介绍区域填充、细化、粗化和修剪等预处理算法和后处理算法。

1. 边界提取

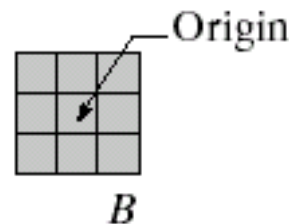
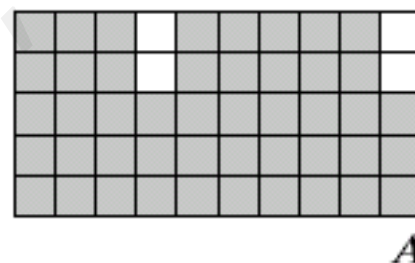
设集合A的边界表示为 $\beta(A)$ ，定义：

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

其中B为结构元素。

为什么不采用如下定义？

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



$A \ominus B$

$\beta(A)$

边界提取例子



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).

2. 洞孔填充

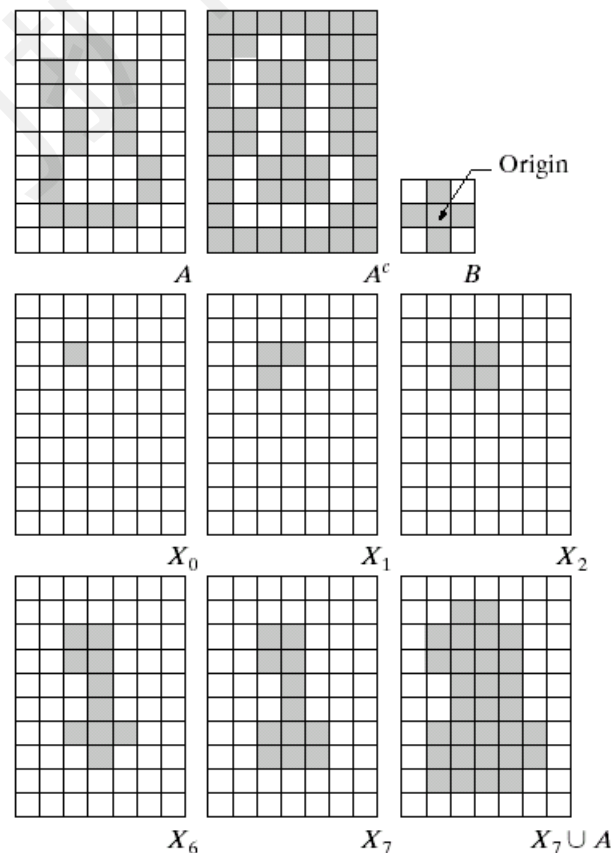
设**A**为集合，元素为八连通边界，区域填充算法是一个迭代算法：

1) 取一个洞内的初始点**p**，将**1**赋给**p**，
令 $X_0 = p$

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

3) 如果 $X_k = X_{k-1}$ ，结束，否则转到2)。
最后 $X_k \cup A$ 为所求，

第2) 步限制了 X_k
不会超出A的界



3 连通分量的提取

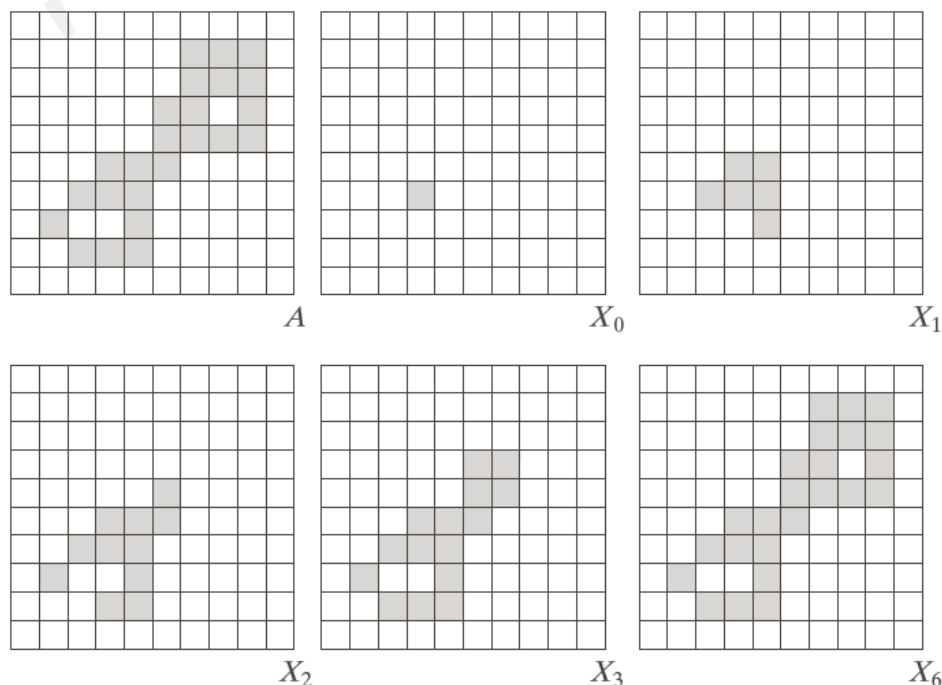
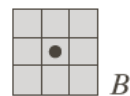
算法是一个迭代算法： $X_0 = p$

1) 取子集A中的点p为初始点（每个连通分量取一个），令

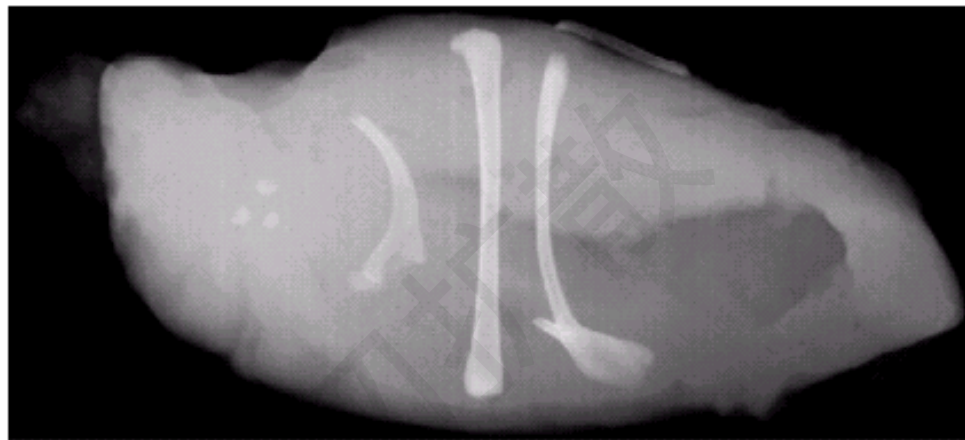
2) $X_k = (X_{k-1} \oplus B) \cap A, k = 1, 2, 3, \dots$

3) 如果 $X_k = X_{k-1}$ ，则

$$Y = X_k$$



3 连通分量的提取例子 (鸡胸肉中的碎骨检测)



二值化



连通区域
检测 (面
积小的为
碎骨)



4. 凸壳

凸集：连接集合A内任意两点的直线段都在A的内部.

凸壳：集合S的凸壳H是包含S的最小凸集.

集合差H-S称为S的**凸缺**.

集合A凸壳C(A)求取算法：

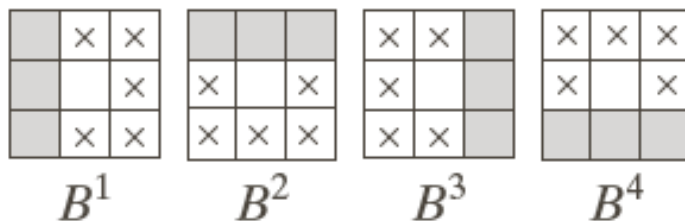
令 $B^i, i=1,2,3,4$, 表示下面4个结构元素。求取过程由计算公示组成：

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

令 $X_0^i = A$, 当收敛, 即 $X_k^i = X_{k-1}^i$, 令 $D^i = X_k^i$

则A的凸壳为：

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

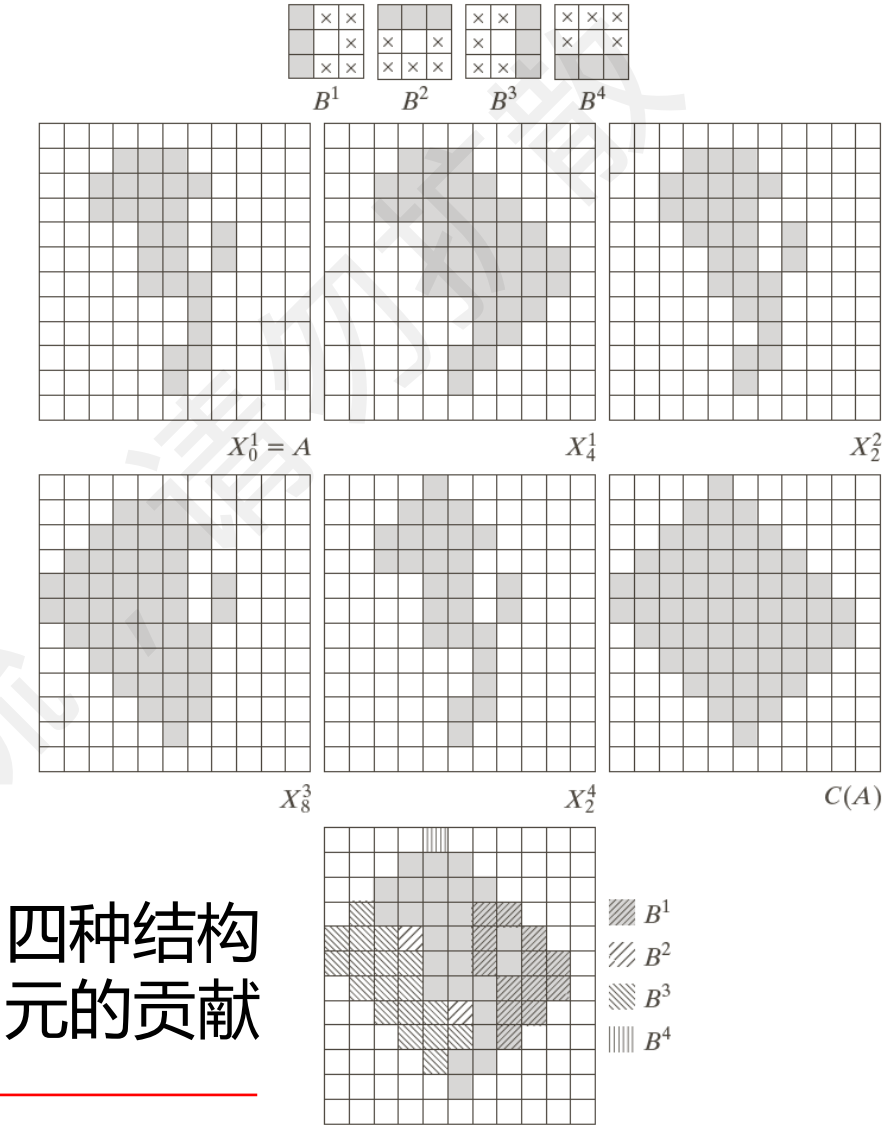
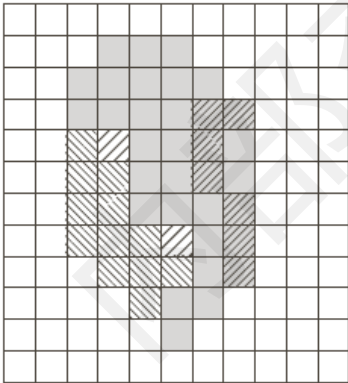


**仅关注灰色
和白色部分**

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

上述计算的凸壳不能确保凸性所需的最小尺寸.

因此要求:
限制生长以便凸壳不会超过初始集合在水平和垂直方向上的尺寸大小.



5. 细化

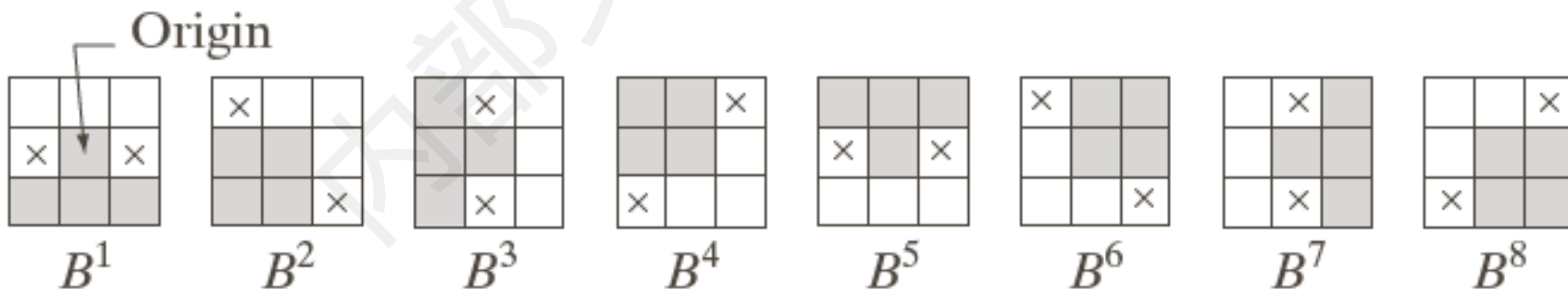
◆ 用结构元素B对集合A的细化定义为:

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

◆ 用结构元素序列 $\{B\} = \{B^1, B^2, B^3, \dots, B^n\}$ 对集合A的细化定义为:

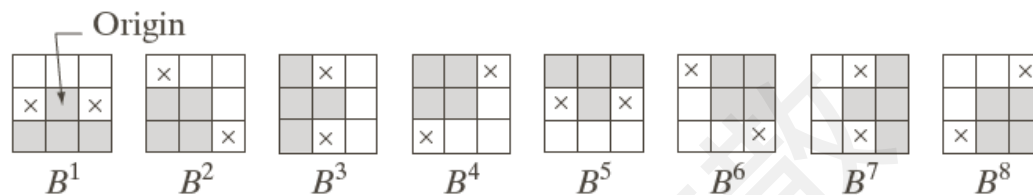
$$A \otimes \{B\} = (((\dots((A \otimes B^1) \otimes B^2) \dots) \otimes B^n)$$

B^i 是 B^{i-1} 的一种特殊变形 (旋转)

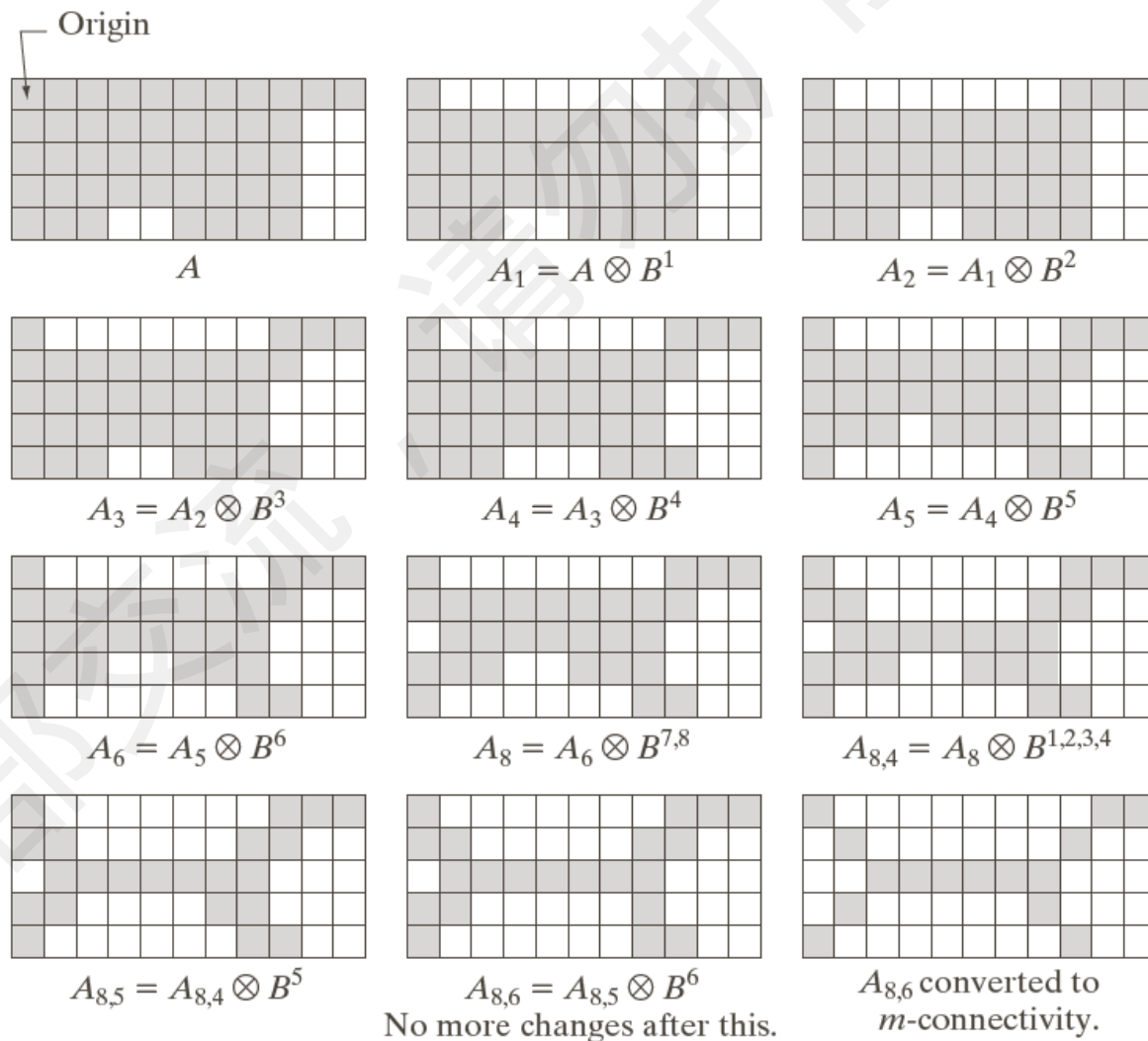


细化实例

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



注：A的外围
还是白色



最后转化为
m连通以消
除多重路径。

6. 粗化

用结构元素B对集合A的粗化定义为:

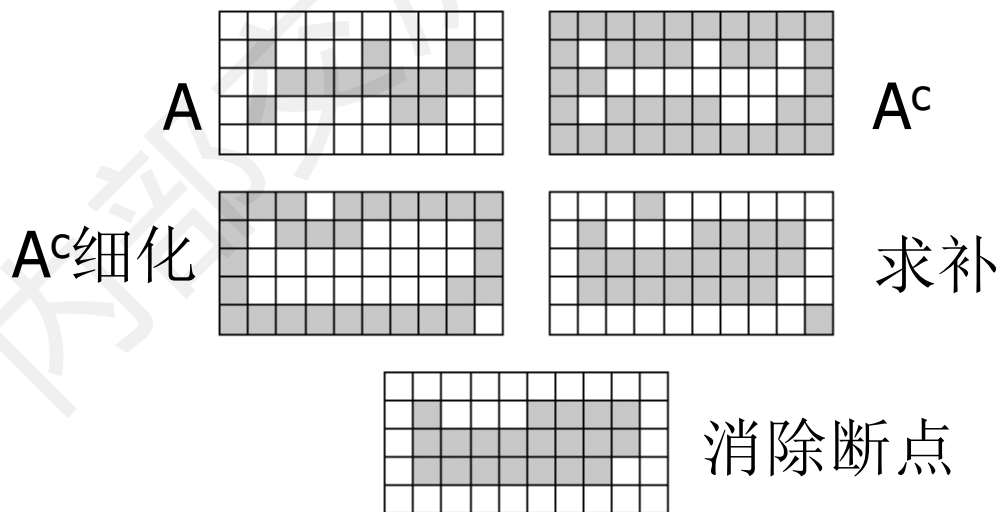
$$A \odot B = A \cup (A * B)$$

用结构元素序列 $\{B\} = \{B^1, B^2, B^3, \dots, B^n\}$ 对集合A的粗化定义:

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

注意:

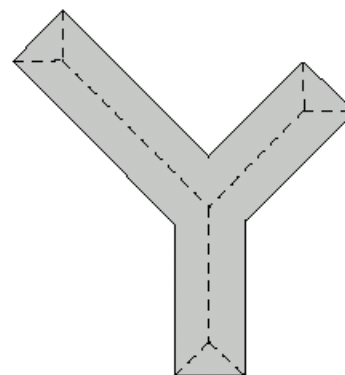
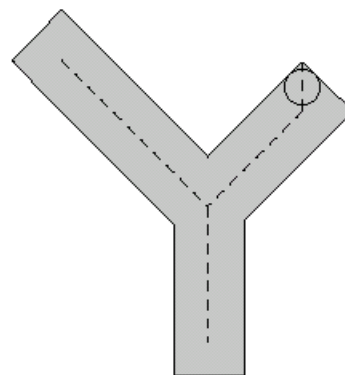
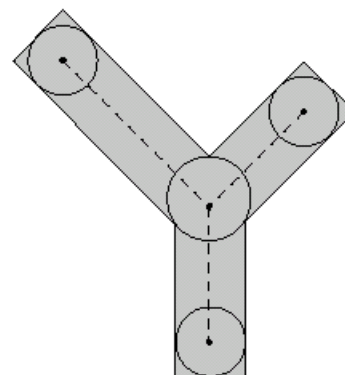
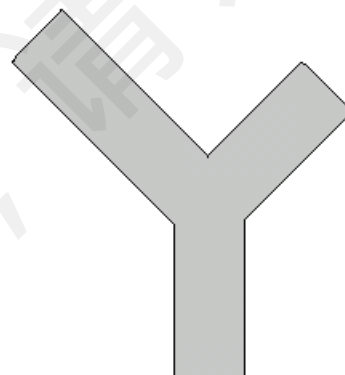
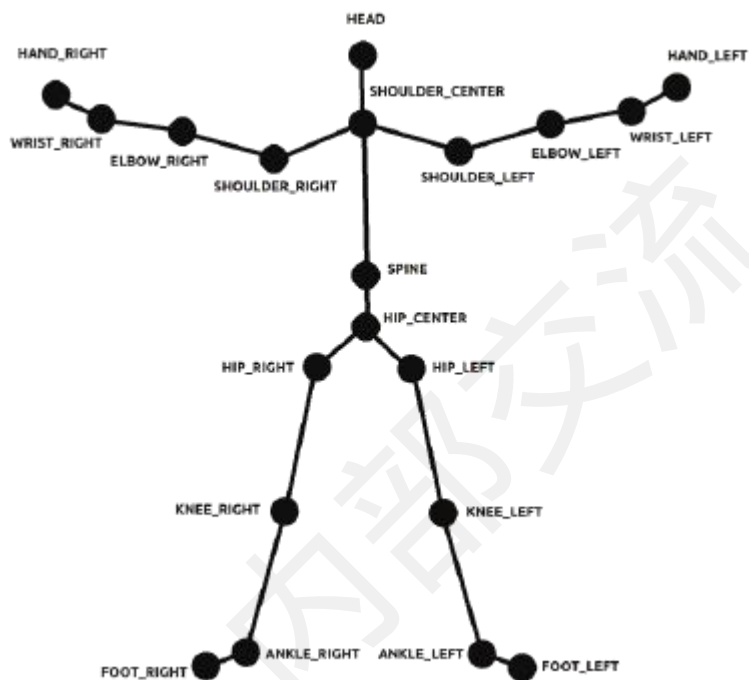
- 1) 粗化和细化的结构元素类似,但1和0要互换.
- 2) 通常用对背景的细化代替粗化,但需要消除断点.



7. 骨架

集合A的骨架 $S(A)$ 定义为:

$$S(A) = \{z \mid A \text{ 内最大圆盘 } (D)_z \text{ 的圆心}\}$$



A 的骨架可用腐蚀和开操作表达。即，骨架可以表达为如下式：

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

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

这里 B 是一个结构元素， $(A \ominus k B)$ 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。就是说：

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

可以证明（重构）： $A = \bigcup_{k=0}^K (S_k(A) \oplus k B)$

这里 $(S_k(A) \oplus k B)$ 表示对 $S_k(A)$ 的 k 次连续的膨胀，即，

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

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

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

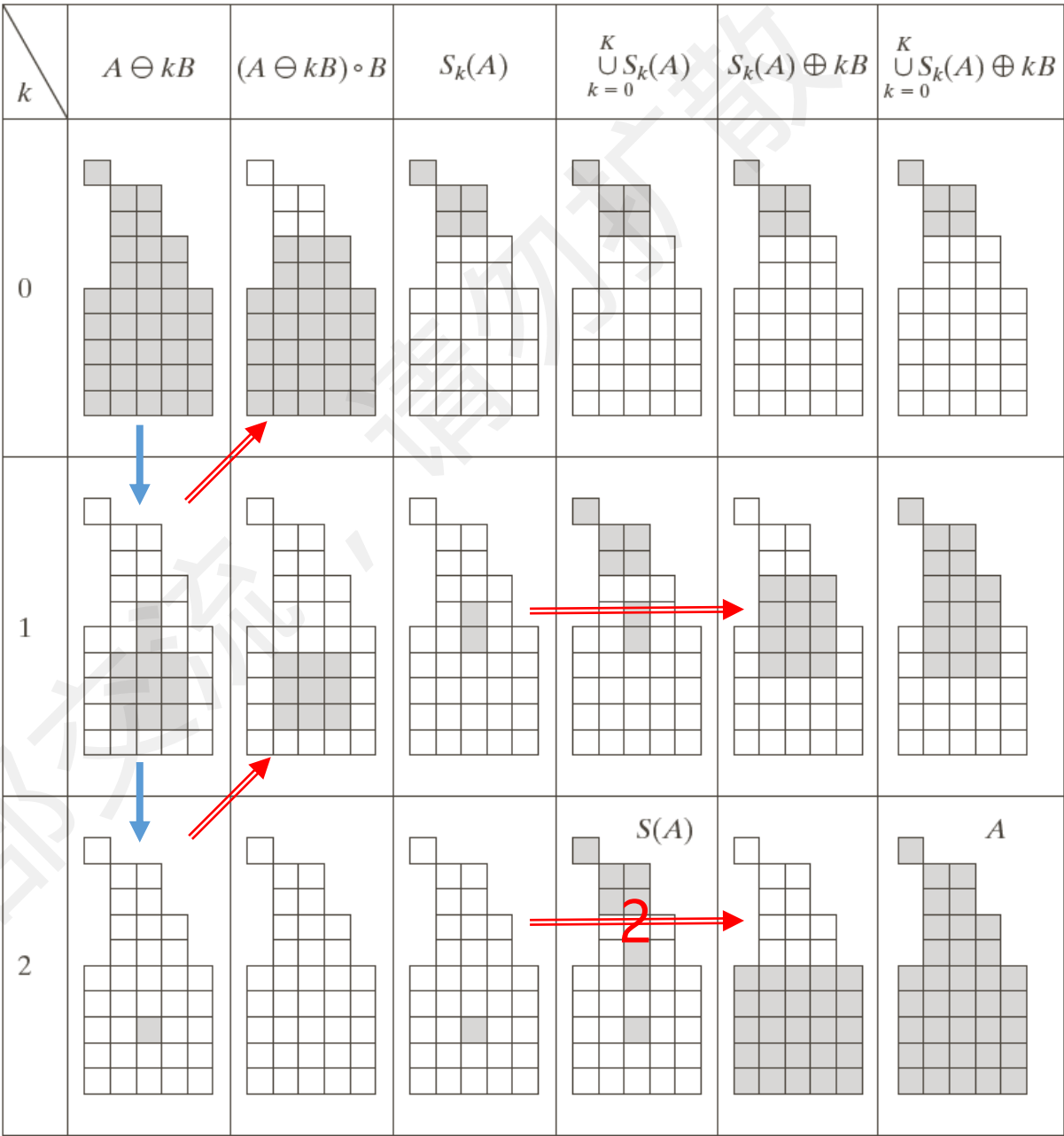
↓

腐蚀

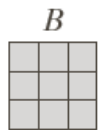
↓

膨胀

骨架提取



重建



8. 剪裁

剪裁通常用于细化和骨架绘制算法的后处理.

用于消除“毛刺”——**比较短的像素端点**，比如说小于等于3个像素长度.

剪裁算法（**例子**）：

1) **三次**细化 $X_1 = A \otimes \{B\}$

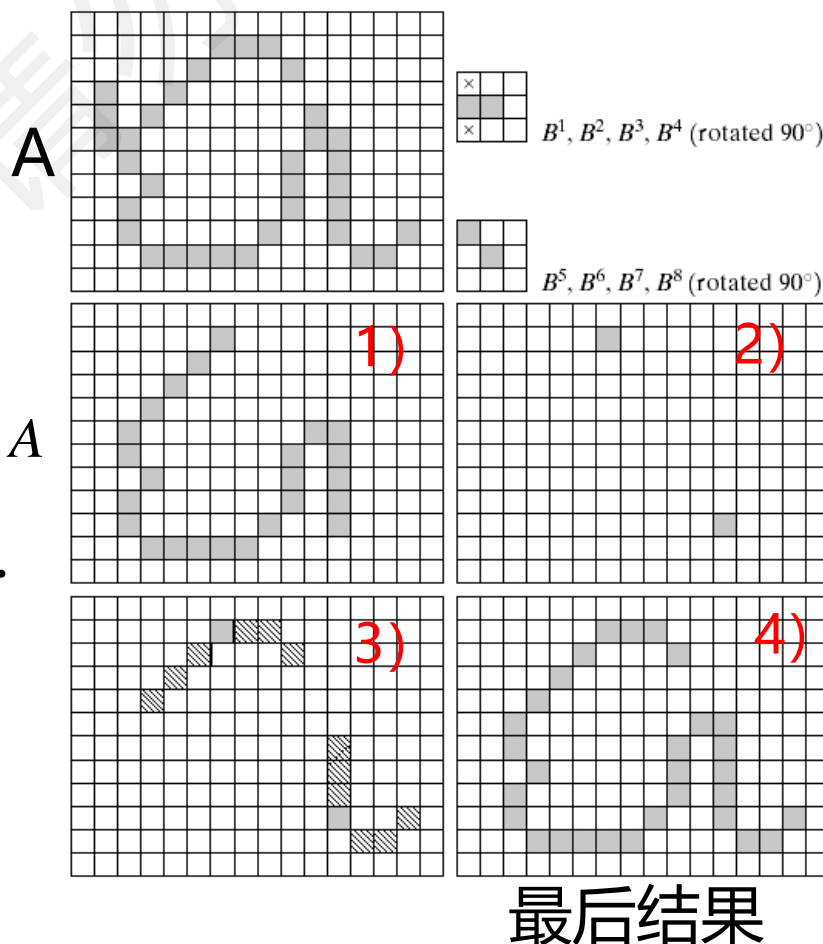
2) 提取端点 $X_2 = \bigcup_{k=1}^8 (X_1 \otimes B^k)$

3) **三次**限制膨胀 $X_3 = (X_2 \oplus H) \cap A$

其中H是元素值为1的3x3的结构元素.

4) 剪裁的结果: $X_4 = X_1 \cup X_3$

非常依赖于经验



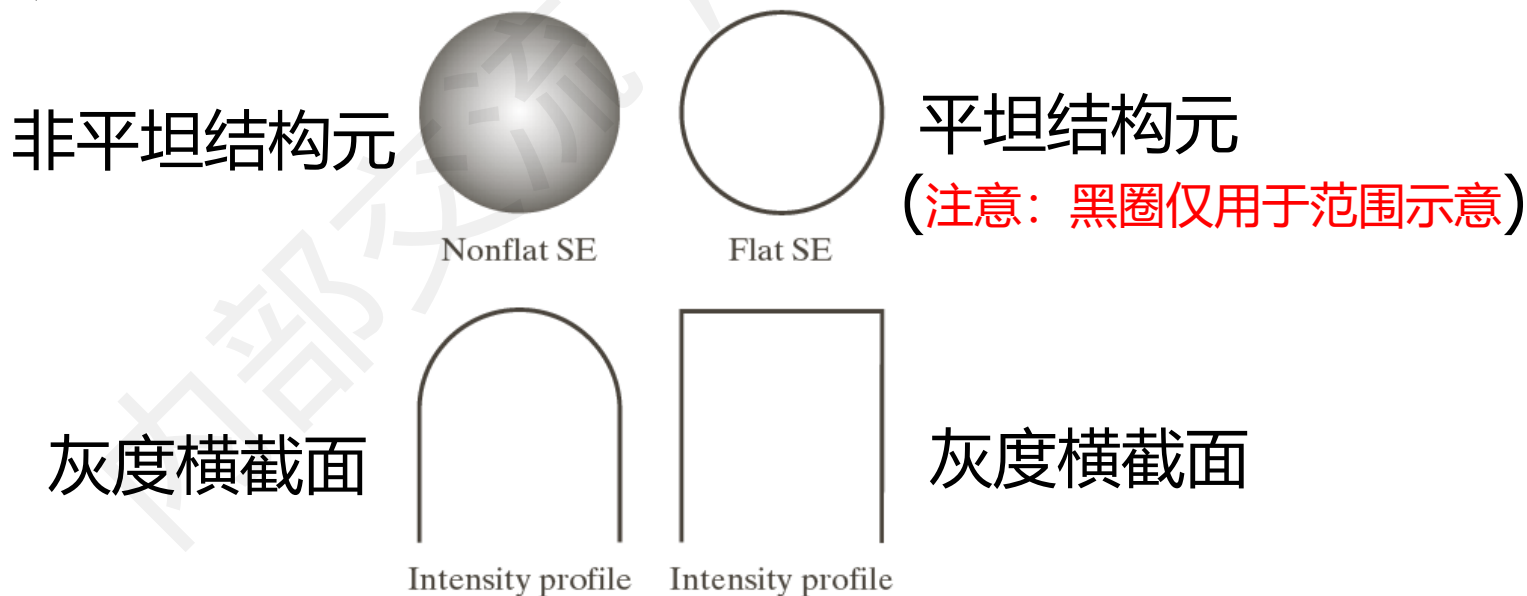
9. 形态学重建（略）

内部交流，请勿扩散

六. 灰度级图像形态学

1. 将形态学处理扩展到灰度图像的基本操作,即膨胀、腐蚀、开操作和闭操作;
2. 介绍基于形态学梯度运算的边界提取算法、基于纹理内容的区域分割算法、形态学图像平滑处理、Top-hat变换等。

处理形如 $f(x,y)$ 和 $b(x,y)$ 的图像, $f(x,y)$ 是输入图像和 $b(x,y)$ 是结构元素。



1. 平坦结构元下的膨胀和腐蚀

用平坦元 b 在 (x,y) 对图像 f 腐蚀定义：

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

用平坦元 b 在 (x,y) 对图像 f 膨胀定义 (f 与 \hat{b} 重合区域的最大值)：

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

2. 非平坦结构元下的膨胀和腐蚀

现实多用平坦结构元

用非平坦元 b_N 对图像 f 进行的灰度腐蚀定义为：

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

1) 图像将变暗？

如果所有结构元素值为正，则是。

2) 如果在输入图像中亮的细节比结构元素的面积小，则亮度效果将被消弱。

用非平坦元 b_N 在 (x, y) 对图像 f 膨胀定义：

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

2. 非平坦结构元下的膨胀和腐蚀

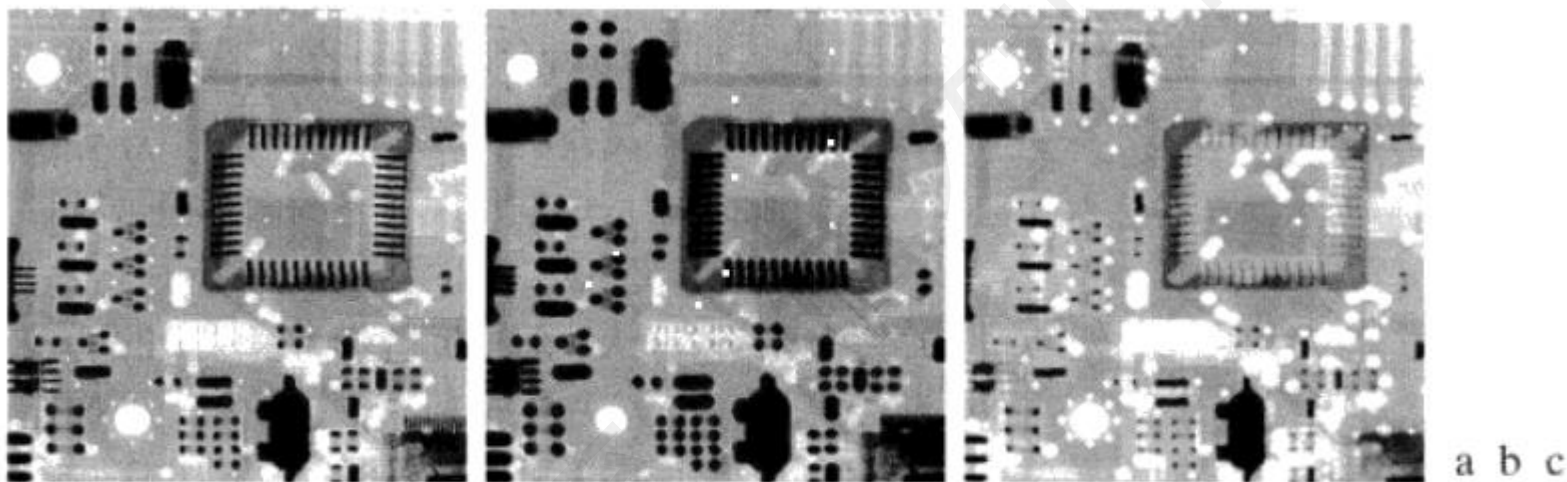


图 9.35 (a)大小为 448×425 像素的灰度级 X 射线图像；(b)使用半径为 2 个像素的圆盘形结构元对图像的腐蚀结果；(c)用相同结构元对图像的膨胀结果(原图像由 Lixi 公司提供)

灰度膨胀与腐蚀的关系：

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

其中 $f^c = -f(x, y)$, $\hat{b} = b(-x, -y)$

3. Opening and Closing

使用结构元素 b 对图像 f 进行开操作的定义为:

$$f \circ b = (f \ominus b) \oplus b$$

即, 使用结构元素 b 对图像 f 进行先腐蚀, 然后膨胀。

使用结构元素 b 对图像 f 进行闭操作的定义为:

$$f \bullet b = (f \oplus b) \ominus b$$

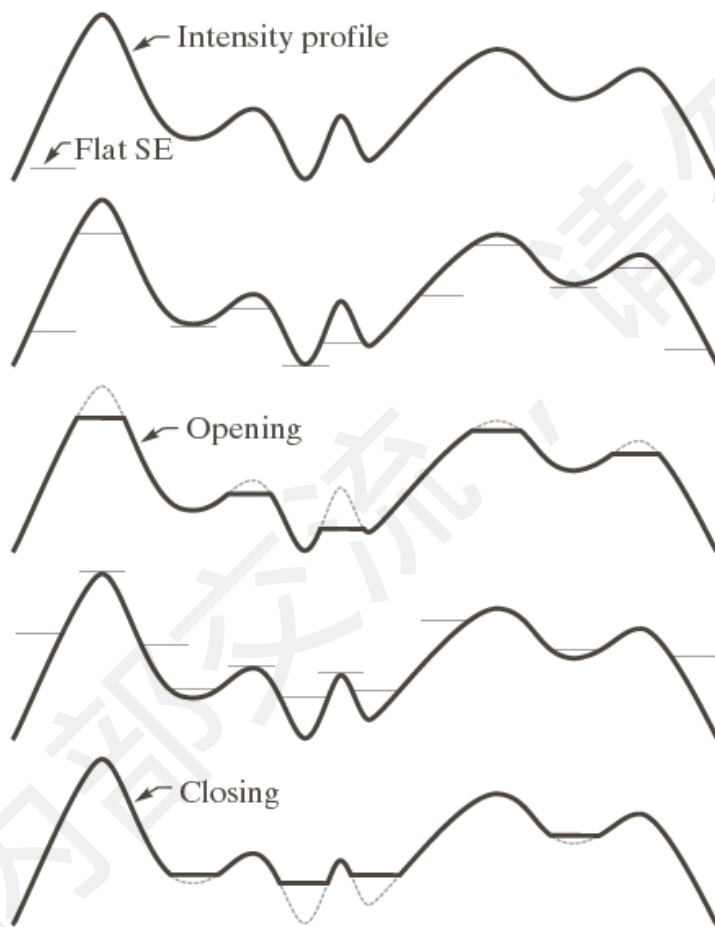
即, 使用结构元素 b 对图像 f 进行先膨胀, 然后腐蚀。

灰度开操作与闭操作的关系:

$$(f \bullet b)^c = (f^c \circ \hat{b})$$

其中 $f^c = -f(x, y)$ 即上式 $-(f \bullet b) = (-f \circ \hat{b})$

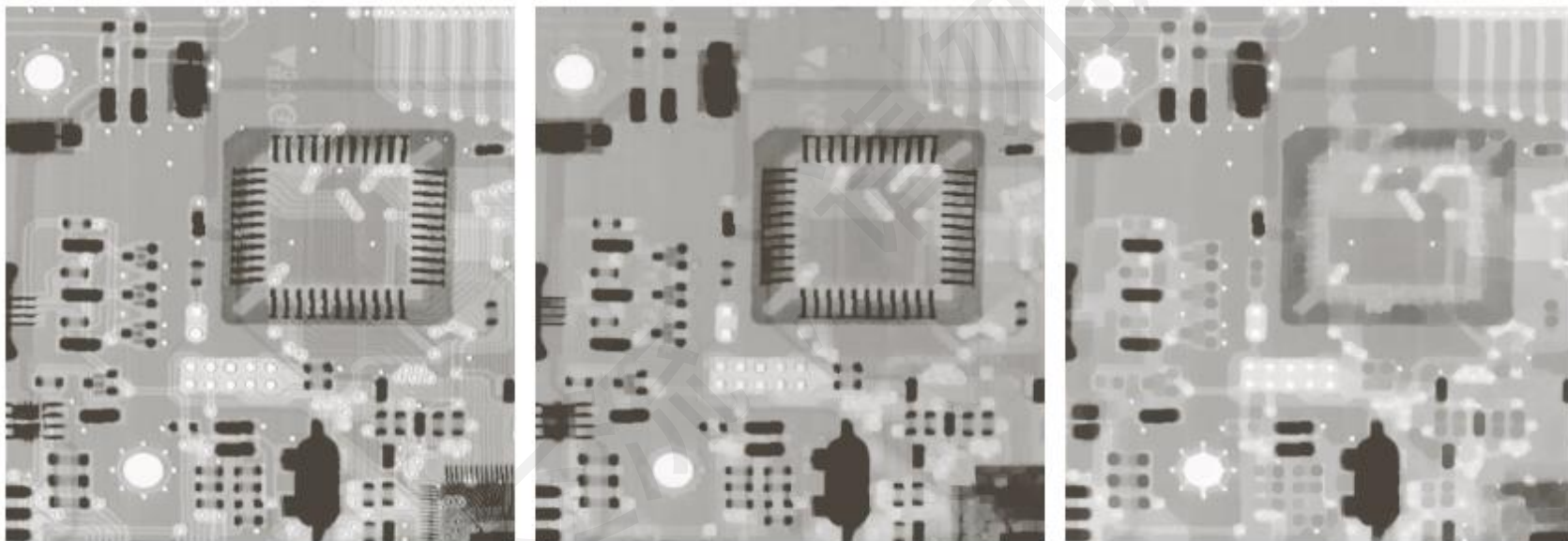
- 开操作经常用于去除小的明亮的细节，保持整体灰度级和较大的明亮特征；
- 闭操作经常用于去除小的黑暗的细节；



开：下表面往上推动
结构元所达到最高处

闭：上表面往下推动
结构元所达到最高处

开操作（半径**3**操作）和闭操作（半径**5**像素）的例子：



灰度级开操作满足下列性质：

- (i) $f \circ b \sqsubseteq f$
- (ii) 如果 $f_1 \sqsubseteq f_2$, 则 $f_1 \circ b \sqsubseteq f_2 \circ b$
- (iii) $(f \circ b) \circ b = f \circ b$

符号 $e \sqsubseteq r$ 表示 e 的域是 r 的域的子集, 且对 e 的域内的任何 (x,y) 有 $e(x,y) \leq r(x,y)$ 。

灰度级闭操作满足下列性质：

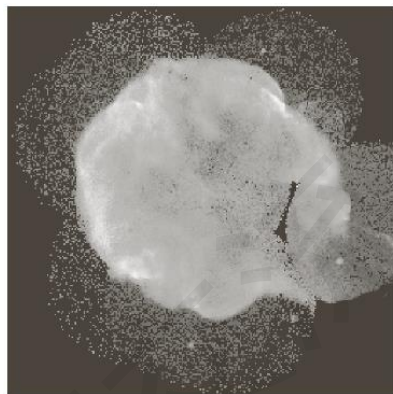
- (i) $f \sqsubseteq f \bullet b$
- (ii) 如果 $f_1 \sqsubseteq f_2$, 则 $f_1 \bullet b \sqsubseteq f_2 \bullet b$
- (iii) $(f \bullet b) \bullet b = f \bullet b$

4. 灰度级形态学的一些应用

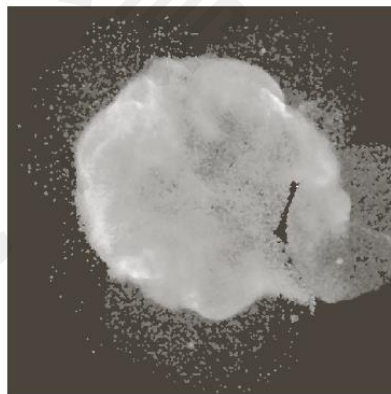
(1) 形态学图像平滑

**先采用开操作(抑制比结构元小的亮点),
然后采用闭操作(抑制暗细节)
以去除亮和暗的噪声。**

原图



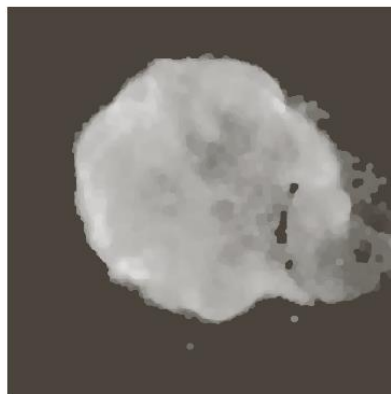
半径2平滑



半径3平滑



半径5平滑



4. 灰度级形态学的一些应用

(2) 形态学图像梯度

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

a b c d

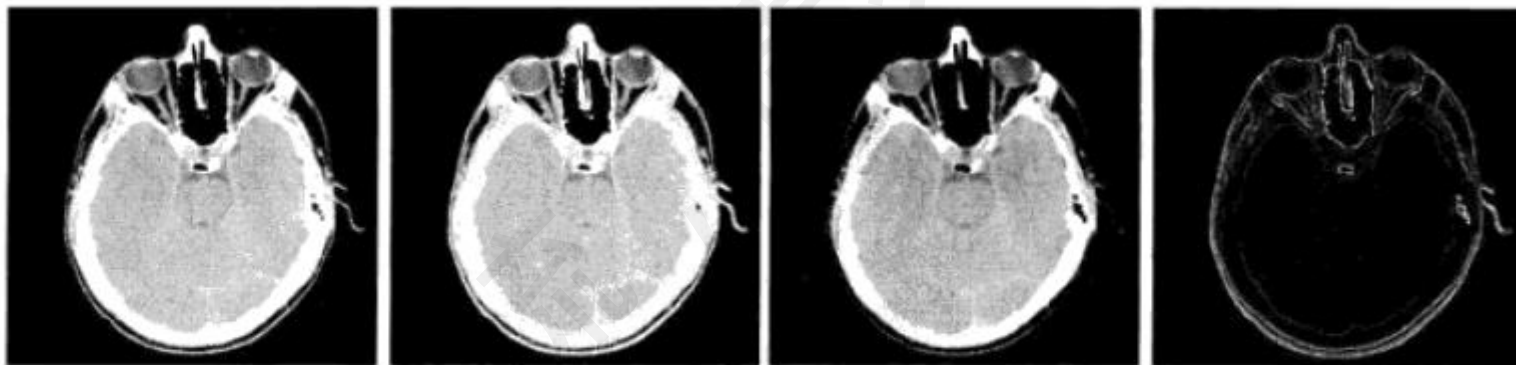


图 9.39 (a) 512×512 的头部 CT 扫描图像; (b) 膨胀的结果; (c) 腐蚀的结果; (d) 形态学梯度, 计算图 (b) 和图 (c) 间的差 (原图像由 Vanderbilt 大学的 David R. Pickens 博士提供)

使用全1的3*3结构元

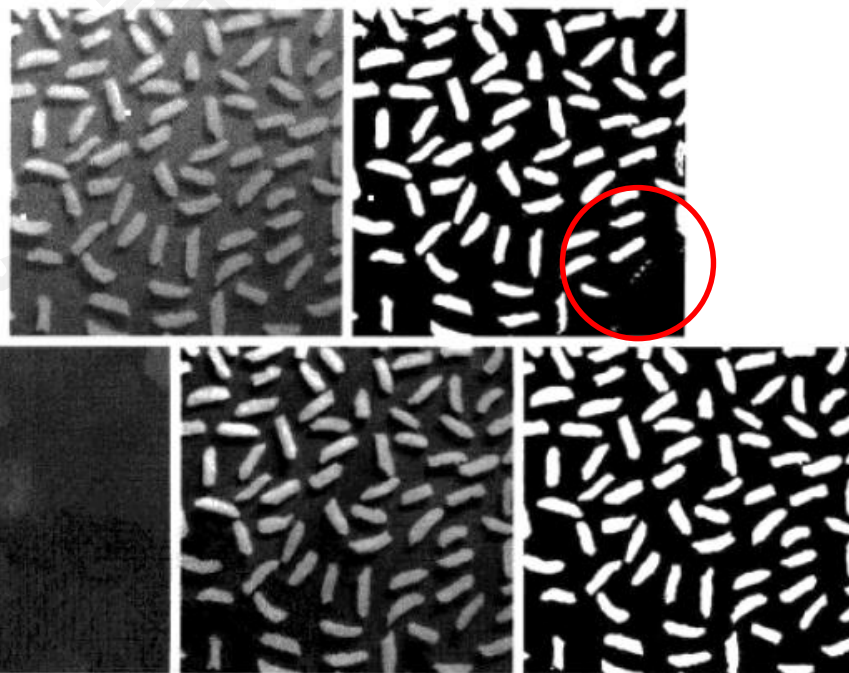
(3)形态学top-hat (顶帽)、bottom-hat (底帽) 变换

作用：矫正不均匀的光照影响。

$T_{hat} = f - f \circ b$ 亦称“白顶帽”变换，用于暗背景上亮物体

$B_{hat} = f \bullet b - f$ 亦称“黑底帽”变换，用于亮背景上暗物体

a b
c d e



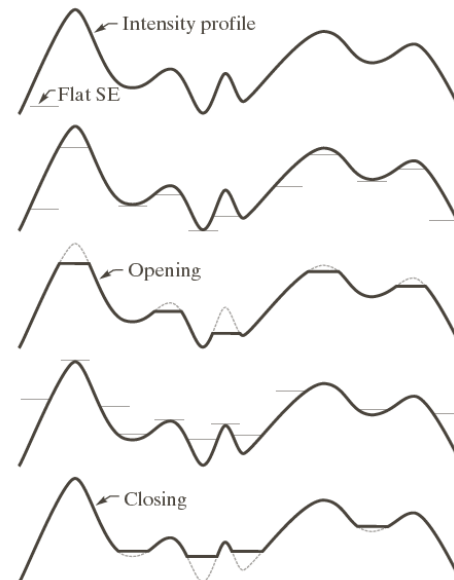
Why top-hat /
bottom-hat can
normalize the
illumination?

图 9.40 使用顶帽变换校正阴影：(a)大小为 600×600 的原图像；(b)阈值处理后的图像；(c)用半径为 40 的圆盘形结构元进行开操作的图像；(d)顶帽变换的图像(图像减去其开操作)；(e)顶帽变换的图像经阈值处理的结果

Why top-hat/bottom-hat can normalize illumination?

Illumination (shadow) variations mainly lie in the low-frequency band of images.

The opening / closing operation can obtain the low-frequency.



(4) Granulometry (粒度测定, 用于估计颗粒的尺寸分布)

使用逐渐增大的结构元对图像进行开操作。**某个特殊尺寸的开操作对包含类似尺寸的颗粒的输入图像的区域产生最大的效果。**通过计算输入图像和输出图像之间的差异可以对相近尺寸颗粒的相对数量进行测算。

a b c
d e f

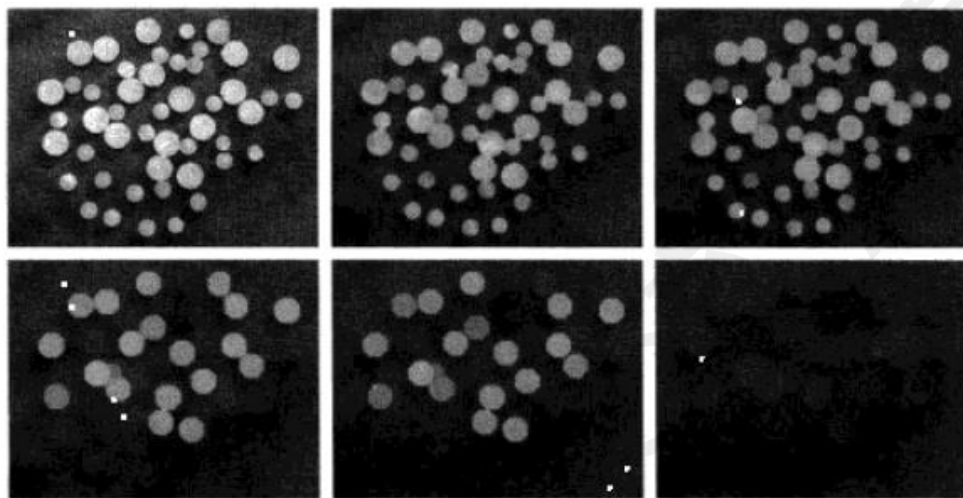


图 9.41 (a) 大小为 531×675 的木钉图像; (b) 平滑后的图像; (c)~(f) 分别用半径为 10, 20, 25 和 30 像素的圆盘形结构元对图像进行开操作后的结果 (原图像由 MathWorks 公司的 Steve Eddins 博士提供)

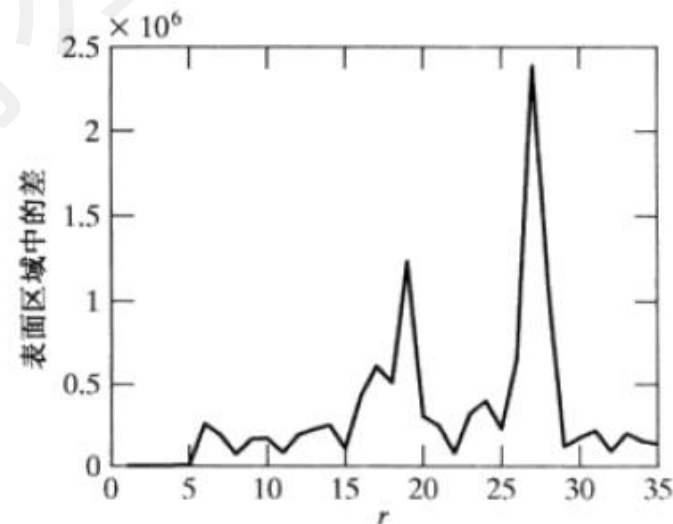


图 9.42 表面区域中的差, 它是圆盘形结构元半径 r 的函数。两个峰值表明图像中存在两个主要的颗粒尺寸

表面区域: 开操作后像素值的和。

(5) Textural Segmentation (没统一公式, 举例说明)

右边区域的圆点直径比左边大。目的是以纹理为基础找到区域的边界。算法如下:

- (i) 取尺寸比小斑点大的结构元做闭运算; (半径=30)
- (ii) 取比大斑点间隙大的结构元做开操作; (半径=60)
- (iii) 做二值化. (使用全1的3*3结构元执行梯度画界)

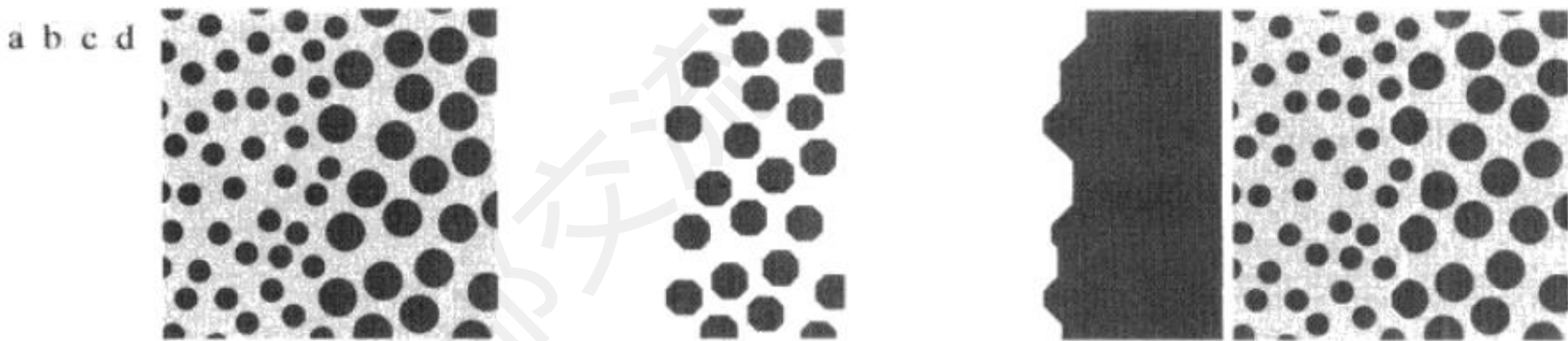


图 9.43 纹理分割: (a) 由两种斑点组成的大小为 600×600 的图像; (b) 对图 (a) 执行闭操作后删除了小斑点的图像; (c) 对图 (b) 执行开操作后删除了大斑点间的亮间隔的图像; (d) 将图 (c) 中两个区域间的边界叠加到原图像上后的结果。边界是使用形态学梯度操作得到的